



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ &
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

«ΣΥΣΤΗΜΑΤΙΚΗ ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ ΓΙΑ
ΕΝΟΡΧΗΣΤΡΩΣΗ ΠΑΚΕΤΑΡΙΣΜΕΝΩΝ ΕΦΑΡΜΟΓΩΝ ΣΤΟ ΥΠΟΛΟΙΠΟ
ΣΥΝΕΧΕΣ»

Διπλωματική Εργασία

Θεοδώρου Αναστασία

Επιβλέπων: Βαρβαρίγου Θεοδώρα

Καθηγήτρια, ΕΜΠ, Ηλεκτρολογων Μηχανικων & Μηχανικών Υπολογιστών

Αθήνα, Σεπτέμβριος, 2025



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ &
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ**

**«ΣΥΣΤΗΜΑΤΙΚΗ ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ ΓΙΑ
ΕΝΟΡΧΗΣΤΡΩΣΗ ΠΑΚΕΤΑΡΙΣΜΕΝΩΝ ΕΦΑΡΜΟΓΩΝ ΣΤΟ
ΥΠΟΛΟΙΠΟ ΣΥΝΕΧΕΣ»**

Διπλωματική Εργασία

Θεοδώρου Αναστασία

Επιβλέπων: Βαρβαρίγου Θεοδώρα

Καθηγήτρια, ΕΜΠ, Ηλεκτρολογικών Μηχανικών & Μηχανικών Υπολογιστών

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30η Σεπτεμβρίου 2025

Βαρβαρίγου Θεοδώρα
Καθηγήτρια, ΕΜΠ

Εμμανουήλ Βαρβαρίγος
Καθηγητής, ΕΜΠ

Κωνσταντίνος Τσερπές
Επίκουρος Καθηγητής, ΕΜΠ

Αθήνα, Σεπτέμβριος, 2025

Θεοδώρου Αναστασία

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Copyright © Θεοδώρου Αναστασία, 2025

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τη σύγχρονη πρόκληση της ενορχήστρωσης containerized εφαρμογών στο πολυεπίπεδο και ετερογενές υπολογιστικό περιβάλλον του Cloud-to-Thing Continuum, το οποίο εκτείνεται από το κεντρικό cloud μέχρι το επίπεδο των edge συσκευών. Σε αυτό το δυναμικό και καταναεμημένο πλαίσιο, όπου συνυπάρχουν υποδομές cloud, fog και edge computing, η containerization αναδεικνύεται ως βασική τεχνολογική προσέγγιση για φορητή, συνεπή και αποδοτική εκτέλεση εφαρμογών. Ταυτόχρονα, τα εργαλεία ενορχήστρωσης —με κυριότερο το Kubernetes— καθίστανται κρίσιμα για την αυτοματοποιημένη διαχείριση του κύκλου ζωής των containers, περιλαμβάνοντας λειτουργίες όπως η δυναμική κλιμάκωση, η εξισορρόπηση φορτίου, η παρακολούθηση και η ανθεκτικότητα σε σφάλματα.

Η εργασία εφαρμόζει τη μεθοδολογία της Συστηματικής Ανασκόπησης Βιβλιογραφίας (Systematic Literature Review – SLR), ακολουθώντας ένα αυστηρά καθορισμένο πρωτόκολλο βάσει των κατευθυντήριων γραμμών PRISMA. Μετά από συστηματική αναζήτηση, φιλτράρισμα και ανάλυση, επιλέχθηκαν 38 επιστημονικές δημοσιεύσεις της τελευταίας δεκαετίας που πληρούσαν τα κριτήρια ένταξης. Η ανάλυση επικεντρώθηκε στην απάντηση τριών ερευνητικών ερωτημάτων: (α) ποιες πλατφόρμες και τεχνολογίες ενορχήστρωσης χρησιμοποιούνται στο Continuum, (β) ποια κριτήρια και μετρικές απόδοσης αξιοποιούνται για benchmarking, και (γ) ποιες είναι οι υφιστάμενες προκλήσεις και τα ερευνητικά κενά, με έμφαση στη χρήση τεχνητής νοημοσύνης και αποκεντρωμένων αρχιτεκτονικών.

Η συγκριτική αξιολόγηση των εργαλείων Kubernetes, Docker Swarm, Nomad και K3s ανέδειξε τα πλεονεκτήματα και μειονεκτήματα κάθε λύσης σε όρους επεκτασιμότητας, επιδόσεων, ευχρηστίας και υποστήριξης για περιβάλλοντα edge/fog. Επιπλέον, εντοπίστηκαν σημαντικά ερευνητικά κενά, ιδίως στην κατεύθυνση της αποκεντρωμένης ενορχήστρωσης, της τεχνητής νοημοσύνης στη λήψη αποφάσεων και της ενιαίας διαχείρισης εφαρμογών και δεδομένων σε καταναεμημένα περιβάλλοντα.

Η εργασία καταλήγει σε προτάσεις για μελλοντική έρευνα και παρέχει ένα ενοποιημένο θεωρητικό και πρακτικό υπόβαθρο για την ανάπτυξη ανθεκτικών, ευέλικτων και αποδοτικών συστημάτων container orchestration στο Cloud-to-Thing Continuum.

Λέξεις Κλειδιά

Εικονικοποίηση με Containers, Ενορχήστρωση, Υπολογιστικό Νέφος έως το Άκρο (Cloud-to-Thing Continuum), Διαχείριση Πόρων, Συστηματική Ανασκόπηση Βιβλιογραφίας

Abstract

This thesis explores the contemporary challenge of orchestrating containerized applications within the multi-layered and heterogeneous computing landscape of the Cloud-to-Thing Continuum, which spans from centralized cloud infrastructures to edge-level devices. In this dynamic and distributed environment — where cloud, fog, and edge computing infrastructures coexist — containerization emerges as a key technological approach for portable, consistent, and efficient application execution. Simultaneously, orchestration tools, with Kubernetes being the most prominent, become essential for the automated management of the container lifecycle, encompassing tasks such as dynamic scaling, load balancing, monitoring, and fault tolerance.

The study adopts the methodology of Systematic Literature Review (SLR), following a rigorously defined protocol based on the PRISMA guidelines. After a structured search, filtering, and qualitative analysis, 38 scientific publications from the past decade were selected based on predefined inclusion criteria. The review addresses three core research questions: (a) which orchestration platforms and technologies are used across the Continuum, (b) which performance metrics and benchmarking criteria are applied, and (c) what are the prevailing challenges and research gaps, with emphasis on the integration of artificial intelligence and decentralized architectures.

The comparative assessment of orchestration tools — including Kubernetes, Docker Swarm, Nomad, and K3s — highlights their respective advantages and limitations in terms of scalability, performance, usability, and support for edge/fog environments. Furthermore, significant research gaps were identified, particularly regarding decentralized orchestration models, AI-driven decision-making mechanisms, and unified application and data lifecycle management in distributed computing settings.

The thesis concludes with concrete directions for future research and offers a consolidated theoretical and practical foundation for the development of resilient, flexible, and efficient container orchestration systems within the Cloud-to-Thing Continuum.

KeyWords

Container Virtualization, Orchestration, Cloud-to-Thing Continuum, Resource Management, Systematic Literature Review

Πίνακας περιεχομένων

Contents

Περίληψη.....	5
Λέξεις Κλειδιά.....	5
Abstract.....	7
KeyWords.....	7
Πίνακας περιεχομένων.....	9
Πίνακας Εικόνων.....	12
Πίνακας Πινάκων.....	14
1. Εισαγωγή.....	16
2. Θεωρητικό Υπόβαθρο & Σχετική Έρευνα.....	18
2.1 Θεωρητικό Υπόβαθρο.....	18
2.1.1 Container Orchestration.....	18
2.1.2 Edge Computing.....	21
2.1.3 Fog Computing	23
2.1.4 Cloud-to-Thing Continuum.....	26
2.1.5 Τεχνητή Νοημοσύνη και Μηχανική Μάθηση.....	29
2.2 Σχετικές Συστηματικές Ανασκοπήσεις και Χαρτογραφήσεις Λογοτεχνίας.....	32
2.2.1 Μεθοδολογική προσέγγιση των SLR/SMS.....	32
2.2.2 Εργαλεία και τεχνολογίες που εξετάστηκαν.....	33
2.2.3 Κύρια συμπεράσματα και ερευνητικά κενά.....	33
3. Μεθοδολογία Συστηματικής Ανασκόπησης Λογοτεχνίας (SLR).....	35
3.1 Εισαγωγή.....	35
3.2 Ερευνητικά Ερωτήματα.....	36
3.3 Στρατηγική Ανασκόπησης.....	37
3.4 Λεπτομερής Διαδικασία Φιλτραρίσματος Μελετών.....	42
3.5 Σύνθεση και Ανάλυση Δεδομένων.....	43
4. Ανάλυση Αποτελεσμάτων.....	46
4.1 RQ1.....	46
4.1.1 Εισαγωγή.....	46
4.1.2 Ενορχήστρωση Containers με Εστίαση στο Edge.....	49
4.1.3 AI-based Scheduling και Προδραστική Ενορχήστρωση.....	52
4.1.4 Σύγκριση Kubernetes και Docker Swarm για Lightweight Υλοποιήσεις... 54	
4.1.5 Υβριδική και Πολυνεφική Διαλειτουργικότητα στην Ενορχήστρωση... 57	
4.1.6 Συνοπτική Αξιολόγηση και Συμπεράσματα RQ1.....	59
4.2 RQ2.....	63
4.2.1 Εισαγωγή.....	63

4.2.2	Ελαφριά Ενорχήστρωση και Προγραμματισμός για Edge Υποδομές....	65
4.2.3	Τεχνητή Νοημοσύνη και Προγνωστικές Τεχνικές στον Προγραμματισμό.	66
4.2.4	Υβριδική Τοποθέτηση Unikernels και Containers.....	68
4.2.5	Αποκεντρωμένος Προγραμματισμός και CRDTs στο Kubernetes.....	71
4.2.6	Προσαρμοστική Ενорχήστρωση σε SBCs με Επίγνωση Πόρων.....	74
4.2.7	Συνοπτική Αξιολόγηση.....	77
4.3	RQ3.....	83
4.3.1	Εισαγωγή.....	83
4.3.2	Ευφυείς Μηχανισμοί Ενорχήστρωσης με AI και ML.....	84
4.3.3	Εφαρμογή της Ενισχυτικής Μάθησης (RL) στην Κλιμάκωση Containers.	89
4.3.4	Συγκριτική Αξιολόγηση Τεχνικών και Αποτελεσμάτων.....	90
4.3.5	Ρόλος της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης στην Ενорχήστρωση.....	92
4.3.6	Προηγμένες Τεχνικές και Αλγόριθμοι για Αυτοπροσαρμοζόμενους Ενорχηστρωτές.....	93
4.3.7	Θεμελιώδεις Προϋποθέσεις για την Υλοποίηση AI στα Edge/Fog Περιβάλλοντα.....	95
4.3.8	Αρχιτεκτονικά Μοντέλα για Ευφυή Ενорχήστρωση στο Cloud-to-Thing Continuum.....	96
4.3.9	Συνοπτική Αξιολόγηση.....	98
5.	Συζήτηση & Συγκριτική Αξιολόγηση.....	101
5.1	Συγκριτική Ανάλυση Πλατφορμών και Τεχνικών Ενорχήστρωσης.....	101
5.2	Ερμηνεία των Ευρημάτων.....	104
5.3	Επιπτώσεις στην Πράξη.....	106
5.4	Περιορισμοί της Ανασκόπησης.....	108
5.5	Προτάσεις για Μελλοντική Έρευνα.....	110
6.	Συμπεράσματα.....	114
6.1	Επισκόπηση της Μελέτης.....	114
6.2	Κύρια Συμπεράσματα.....	115
6.3	Συμβολή της Εργασίας.....	117
6.4	Περιορισμοί της Μελέτης.....	118
6.5	Μελλοντική Ενέργεια.....	119
ΠΑΡΑΡΤΗΜΑ Α.....		122
DevOps, MLOps & CI/CD.....		125
Βιβλιογραφία.....		131

Πίνακας Εικόνων

Εικ. 1.1: Αρχιτεκτονική Cloud-to-Thing Continuum με ενδιάμεσα επίπεδα Fog Computing.....	15
Εικόνα 2.1: Αρχιτεκτονική Ενорχήστρωσης Containers σε Κατανεμημένες Πλατφόρμες Υποδοχής.....	17
Εικόνα 2.2 : Ιεραρχική Αρχιτεκτονική Υπολογισμού μεταξύ Edge Devices, Edge Nodes και Cloud.....	20
Εικόνα. 2.3: Ιεραρχικό Υπολογιστικό Μοντέλο Cloud–Fog–Edge και Κλίμακα Ανάπτυξης.....	22
Εικόνα 4.1.2 Ραβδόγραμμα Κατανομής Τεχνολογιών ανά Επίπεδο Υλοποίησης.....	50
Εικόνα 4.1.3 Πίτα Ποσοστών Χρήσης Τεχνολογιών.....	58
Εικόνα 4.1.4 Γραφική απεικόνιση της κατανομής των τεχνολογιών ενорχήστρωσης containerized εφαρμογών βάσει του επιπέδου υλοποίησης (Cloud, Fog, Edge, Hybrid/Multi-cloud) όπως καταγράφηκε στη συστηματική βιβλιογραφική ανασκόπηση.....	59
Εικόνα 4.1.5 Γραφική αποτύπωση της συχνότητας εμφάνισης κύριων τεχνικών χαρακτηριστικών στις λύσεις ενорχήστρωσης που εξετάστηκαν.....	60
Εικόνα 4.1.6: Γραφική απεικόνιση της κατανομής των λύσεων ενорχήστρωσης containerized εφαρμογών ανά τύπο intelligence.....	61
Εικόνα 4.3.1 Κατανομή των ερευνητικών εργασιών του RQ3 ανά κύρια πρόκληση που στοχεύουν να αντιμετωπίσουν οι προτεινόμενες τεχνικές ενорχήστρωσης.....	97
Εικόνα 1 – Κύκλος ζωής ανάπτυξης εφαρμογών με έμφαση στις φάσεις απαιτήσεων, μοντελοποίησης, υλοποίησης, δοκιμών, λειτουργίας, παρακολούθησης και ανατροφοδότησης.....	123

Εικόνα 2 – Μετάβαση από DevOps σε MLOps: ενσωμάτωση των διαδικασιών ανάπτυξης και λειτουργίας με τεχνικές μηχανικής μάθησης για αυτοματοποίηση και συνεχή μάθηση.....	125
Εικόνα 3 – Αντιστοίχιση φάσεων του DevOps κύκλου με τα αντίστοιχα αρχιτεκτονικά στοιχεία της πλατφόρμας AIDoArt που αξιοποιούν τεχνητή νοημοσύνη.....	126

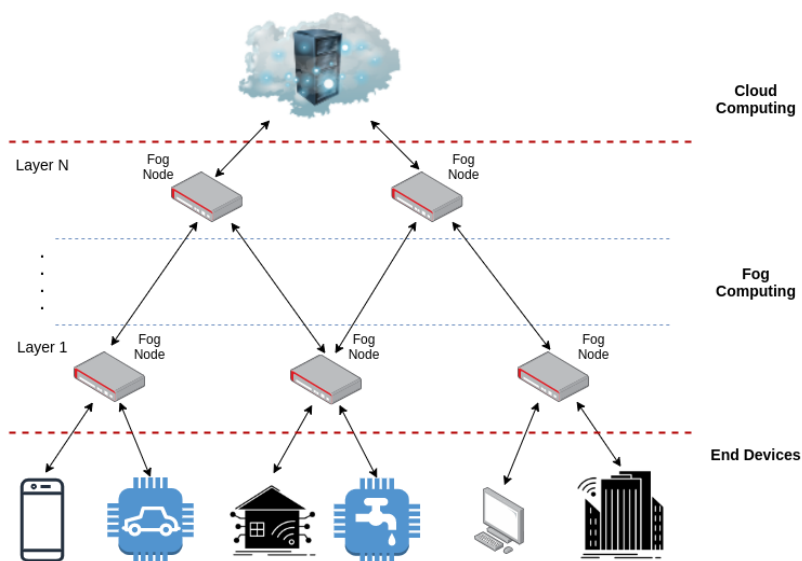
Πίνακας Πινάκων

Πίνακας 3.1 Διάγραμμα ροής επιλογής άρθρων σύμφωνα με τη μεθοδολογία PRISMA.....	37
Πίνακας 4.1.1 Πίνακας Συγκρίσεων μοντέλων.....	48
Πίνακας 4.1.2 Συγκριτική αξιολόγηση σύγχρονων πλατφορμών ενορχήστρωσης containers με χρήση σε περιβάλλοντα IoT/Fog.....	51
Πίνακας 4.1.2.2 Πίνακας κατανομής τεχνολογιών ανά επίπεδο υλοποίησης..	52
Πίνακας 4.1.3 Συνοπτική παρουσίαση AI/Optimization μοντέλων για έξυπνη ενορχήστρωση σε περιβάλλοντα Kubernetes και Edge/Cloud.....	54
Πίνακας 4.1.4 Συγκριτική αξιολόγηση Kubernetes και Docker Swarm με βάση απαιτήσεις, δυνατότητες και καταλληλότητα για IoT/Fog περιβάλλοντα.....	57
Πίνακας 4.1.5 Συγκριτική αποτύπωση σύγχρονων προσεγγίσεων δικτύωσης, federation και scaling για κατανεμημένες container-based υποδομές.....	59
Πίνακας 4.1.6 Πίνακας κατανομής τεχνολογιών ανά επίπεδο υλοποίησης.	61
Πίνακας 4.2.1 Συστηματική ταξινόμηση μοντέλων και στρατηγικών ενορχήστρωσης με έμφαση σε latency, AI, αποκέντρωση και lightweight υλοποιήσεις.....	65
Πίνακας 4.2.2 Συγκριτική αξιολόγηση AI-based μοντέλων scheduling και autoscaling σε edge/cloud περιβάλλοντα.....	67
Πίνακας 4.3.1 Κύριες προκλήσεις ενορχήστρωσης σε περιβάλλοντα Edge/Fog/Cloud.....	84
Πίνακας 5.1.1 – Συγκριτική αξιολόγηση πλατφορμών ενορχήστρωσης containerized.....	102
Εικόνα 5.1.1 Αξιολόγηση πλατφορμών ανά Ερευνητικό ερώτημα (RQ)..	104
Πίνακας Μεταδεδομένων Ερευνητικών Ερωτημάτων.....	122

1. Εισαγωγή

Η ραγδαία εξάπλωση των τεχνολογιών υπολογιστικού νέφους (cloud computing), σε συνδυασμό με την εξέλιξη του fog και edge computing, έχει οδηγήσει στην ανάδυση ενός νέου υπολογιστικού υποδείγματος γνωστού ως Cloud-to-Thing Continuum. Το εν λόγω παράδειγμα εκφράζει την αδιάλειπτη και ευέλικτη μετάβαση των δεδομένων και των εφαρμογών μεταξύ του κεντρικού cloud, των ενδιάμεσων fog κόμβων και του φυσικού επιπέδου των edge συσκευών. Σε αυτό το περιβάλλον, η ανάγκη για δυναμική κατανομή φορτίου, ελαχιστοποίηση της καθυστέρησης, αποτελεσματική αξιοποίηση των υπολογιστικών πόρων και ενίσχυση της αυτονομίας των συστημάτων καθιστά κρίσιμη την εφαρμογή μηχανισμών αυτοματοποιημένης ενορχήστρωσης. Η τεχνολογία της **containerization**, μέσω εργαλείων όπως το Docker, το Podman και το LXD, έχει καθιερωθεί ως το πρότυπο για την απομόνωση και τη φορητότητα εφαρμογών, ιδιαίτερα σε κατανεμημένα και ετερογενή περιβάλλοντα. Ωστόσο, η διαχείριση μεγάλου αριθμού containers, με διαφοροποιημένες απαιτήσεις και τοποθετήσεις σε πολλαπλά επίπεδα του Continuum, προϋποθέτει τη χρήση εξειδικευμένων **πλατφορμών ενορχήστρωσης (orchestration platforms)**. Το Kubernetes, το OpenShift, το Docker Swarm, το Nomad και άλλα εργαλεία έχουν διαδραματίσει καθοριστικό ρόλο στην αυτοματοποίηση του κύκλου ζωής των containerized εφαρμογών, ωστόσο η πλειονότητα αυτών έχει σχεδιαστεί πρωτίστως για cloud-centric υποδομές.

Η **πρόκληση της ενορχήστρωσης στο Continuum** αφορά τη μεταφορά αυτών των τεχνολογιών σε κατανεμημένα, ασύμμετρα και περιορισμένων πόρων περιβάλλοντα. Σε edge/fog σενάρια, η υπολογιστική ισχύς, η ενεργειακή κατανάλωση, η συνδεσιμότητα και η ανάγκη για τοπική λήψη αποφάσεων καθιστούν αναποτελεσματικές πολλές από τις υπάρχουσες cloud-native προσεγγίσεις. Παράλληλα, η ενσωμάτωση τεχνητής νοημοσύνης (AI/ML) στην ενορχήστρωση, αν και θεωρείται πολλά υποσχόμενη, παραμένει σε πρώιμο στάδιο, με ελλιπή υποστήριξη επεξηγησιμότητας, αυτονομίας και ενεργειακής αποδοτικότητας.



Εικ. 1.1: Αρχιτεκτονική Cloud-to-Thing Continuum με ενδιάμεσα επίπεδα Fog Computing

Μέσα σε αυτό το πλαίσιο, αναδεικνύεται η ανάγκη για **συστηματική και τεκμηριωμένη επισκόπηση της υπάρχουσας επιστημονικής γνώσης** αναφορικά με τις τεχνικές, τις πλατφόρμες, τα πρότυπα αξιολόγησης και τις ερευνητικές προκλήσεις που αφορούν την ενορχήστρωση containerized εφαρμογών στο Cloud-to-Thing Continuum. Η παρούσα εργασία υιοθετεί τη μεθοδολογία της **Systematic Literature Review (SLR)** με σκοπό την απάντηση των ακόλουθων ερευνητικών ερωτημάτων:

- **RQ1:** Ποιες τεχνικές και πλατφόρμες έχουν αξιοποιηθεί για την ενορχήστρωση containerized εφαρμογών σε cloud, fog και edge υποδομές;
- **RQ2:** Ποια κριτήρια απόδοσης και benchmarking μετρικές χρησιμοποιούνται για την αξιολόγηση των λύσεων αυτών;
- **RQ3:** Ποιες είναι οι βασικές ερευνητικές προκλήσεις και οι προοπτικές καινοτομίας, ιδιαίτερα σε σχέση με την εφαρμογή τεχνητής νοημοσύνης στην ενορχήστρωση;

Η εργασία είναι δομημένη ως εξής: το Κεφάλαιο 2 παρουσιάζει το θεωρητικό υπόβαθρο και τις σχετικές έρευνες, με έμφαση στις τεχνολογίες containerization, τις πλατφόρμες ενορχήστρωσης και τις εφαρμογές στο Continuum. Το Κεφάλαιο 3 αναπτύσσει τη μεθοδολογία της SLR, τα κριτήρια επιλογής και τη στρατηγική αναζήτησης. Στο Κεφάλαιο 4 παρουσιάζεται η ανάλυση των αποτελεσμάτων, ενώ το Κεφάλαιο 5 εστιάζει στη συγκριτική αξιολόγηση, τη συζήτηση και τα ερευνητικά συμπεράσματα. Η εργασία ολοκληρώνεται με το Κεφάλαιο 6, το οποίο περιλαμβάνει τα τελικά συμπεράσματα και προτάσεις για μελλοντική έρευνα.

2. Θεωρητικό Υπόβαθρο & Σχετική Έρευνα

2.1 Θεωρητικό Υπόβαθρο

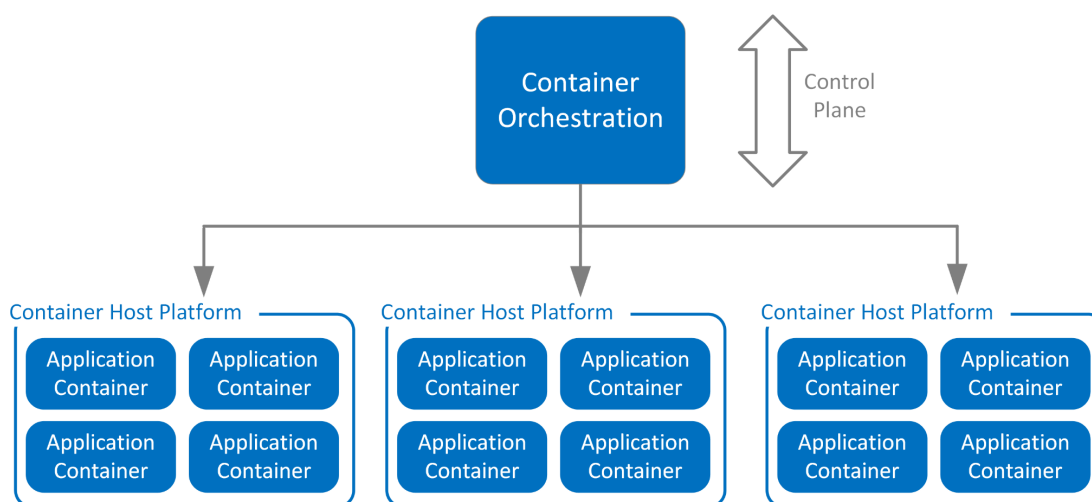
2.1.1 Container Orchestration

Αδιαμφισβήτητα, η ραγδαία εξάπλωση της χρήσης τεχνολογιών για containerization έχει δημιουργήσει σημαντικές προκλήσεις ως προς την διαχείριση τους σε μεγάλης κλίμακας περιβάλλοντα. Οι μέχρι τώρα τεχνολογίες, έχουν αποδειχθεί ανεπαρκείς στη σωστή και ενιαία εγκατάσταση ενημερώσεων του λογισμικού στα ταυτόχρονα πολλά και διαφορετικά περιβάλλοντα. Έτσι, λοιπόν, δημιουργήθηκε η ανάγκη για την ευέλικτη προσαρμογή της υπολογιστικής ισχύος, ώστε να καλύπτονται οι μεταβαλλόμενες απαιτήσεις των εφαρμογών (π.χ. αύξηση σε περιόδους υψηλής ζήτησης, μείωση όταν η χρήση είναι χαμηλή). Επιπλέον, η ταχεία, αυτοματοποιημένη και αξιόπιστη ανάπτυξη νέων εκδόσεων λογισμικού είναι κρίσιμη για την αποδοτικότητα. Ωστόσο, χωρίς εξειδικευμένα εργαλεία, ο στόχος για την ελαχιστοποίηση της μη διαθεσιμότητας των υπηρεσιών κατά τις αναβαθμίσεις, φαίνεται να είναι μη επιτεύξιμος.

Η ενορχήστρωση containers αποτελεί ένα θεμελιώδες τεχνολογικό βήμα για την αποτελεσματική διαχείριση σύγχρονων, κατανεμημένων εφαρμογών. Πρόκειται για ένα σύνθετο και προηγμένο σύστημα αυτοματοποιημένων διαδικασιών και εξειδικευμένων εργαλείων, που συνεργάζονται για την έξυπνη εκτέλεση των ακόλουθων λειτουργιών: τον προγραμματισμό, τη συνεχή ανάπτυξη και παρακολούθηση, την δυναμική κλιμάκωση, και την αποτελεσματική διαχείριση εφαρμογών, οι οποίες υλοποιούνται σε ευέλικτα και φορητά containers. Οι παραπάνω λειτουργίες εκτελούνται σε πολύπλοκα, κατανεμημένα υπολογιστικά περιβάλλοντα, που μπορεί να περιλαμβάνουν δημόσια και ιδιωτικά clouds, υποδομές edge computing πλησίον των πηγών δεδομένων, καθώς και σύνθετες υβριδικές αρχιτεκτονικές που συνδυάζουν ποικίλες υποδομές.[1]

Η ενορχήστρωση των container δεν είναι απλώς μια τεχνική λεπτομέρεια, αλλά αποτελεί βασικό δομικό στοιχείο του cloud-native λογισμικού. Αναδεικνύεται απαραίτητη για την αποτελεσματική διαχείριση των σύγχρονων αρχιτεκτονικών μικροϋπηρεσιών, οι οποίες αποτελούνται από μικρές, ανεξάρτητα αναπτύξιμες και διαχειρίσιμες υπηρεσίες. Προσφέρουν ευελιξία και ανθεκτικότητα, δημιουργώντας όμως σημαντικές προκλήσεις στη διαχείρισή τους. Η ενορχήστρωση τους βοηθά ώστε να ξεπεραστεί αυτή η δυσκολία, αξιοποιώντας τη μεταφερσιμότητα και ελαφρότητα των containers, όπως αυτά που δημιουργούνται με τεχνολογίες όπως το Docker και το CRI (Container Runtime Interface), επεκτείνοντας τις βασικές τους δυνατότητες σε ένα ευρύτερο επίπεδο, αυτό του cluster. Ένα cluster, στην προκειμένη περίπτωση, αναφέρεται σε μια ομάδα συνδεδεμένων υπολογιστικών κόμβων (nodes) που λειτουργούν σαν ένα ενιαίο σύστημα.[2] Το βασικό πλαίσιο για την ενορχήστρωση των container αποτελεί το Kubernetes το οποίο προσφέρει ένα πλαίσιο εξαιρετικά ισχυρό, ευέλικτο και επεκτάσιμο. Η ευρεία υιοθέτηση του Kubernetes έχει

αποδείξει δραστική μείωση εξόδων και κόστους συντήρησης της υποδομής που στηρίζει τα containers. Αυτό επιτυγχάνεται κυρίως μέσω της αυτοματοποίησης μεγάλου μέρους των χειροκίνητων και χρονοβόρων εργασιών των container. Για τους σύγχρονους προγραμματιστές λογισμικού, τους μηχανικούς DevOps και τους επαγγελματίες του IT γενικότερα, η ουσιαστική κατανόηση τόσο των θεμελιωδών τεχνολογιών containerisation, με κύριο εκπρόσωπο το Docker, όσο και των προηγμένων πλαισίων ενορχήστρωσης containers, όπως το Kubernetes είναι πλέον όχι απλώς επιθυμητή, αλλά επιτακτική. Η κατανόηση και αξιοποίηση αυτών των εργαλείων είναι απαραίτητη για την επιτυχημένη ανάπτυξη, την αξιόπιστη διαχείριση και την κλιμάκωση σύγχρονων κατανεμημένων εφαρμογών, οι οποίες αποτελούν τη βάση των σημερινών ψηφιακών υπηρεσιών. [1]



Εικόνα 2.1: Αρχιτεκτονική Ενορχήστρωσης Containers σε Κατανεμημένες Πλατφόρμες Υποδοχής

Στο σημερινό τεχνολογικό οικοσύστημα, το Docker κατέχει μια αδιαμφισβήτητη ηγετική θέση ως ο πιο διαδεδομένος και ευρέως χρησιμοποιούμενος μηχανισμός containerisation ανοιχτού κώδικα, αποτελώντας ένα εμβληματικό παράδειγμα της επαναστατικής αυτής τεχνολογίας που έχει αλλάξει τον τρόπο ανάπτυξης και διαχείρισης λογισμικού. Η ευρεία και παγκόσμια υιοθέτησή του έχει οδηγήσει σε μια εκρηκτική αύξηση του συνολικού αριθμού των διαθέσιμων και ενεργά χρησιμοποιούμενων containers σε παγκόσμιο επίπεδο. Μέσα σε αυτό το πλαίσιο, η προσπάθεια χειροκίνητης διαχείρισης ενός τόσο τεράστιου όγκου containers έχει καταστεί όχι μόνο εξαιρετικά δύσκολη και περίπλοκη, αλλά και πρακτικά ανέφικτη, αυξάνοντας σημαντικά τον κίνδυνο ανθρώπινων λαθών και μειώνοντας την αποτελεσματικότητα των λειτουργιών. Εδώ ακριβώς έρχονται να καλύψουν ένα κρίσιμο και υπαρκτό κενό τα εξειδικευμένα συστήματα ενορχήστρωσης containers, με το Kubernetes να πρωτοστατεί και να αποτελεί τον ηγέτη στον συγκεκριμένο χώρο. Αυτά τα προηγμένα συστήματα είναι απολύτως απαραίτητα για την αυτοματοποίηση ενός ευρέος φάσματος ζωτικών διαδικασιών που περιλαμβάνουν την αρχική ανάπτυξη νέων εφαρμογών σε μορφή containers, την αυτόματη προσαρμογή της κλίμακας των εφαρμογών (scale up ή scale down) ανάλογα με τις τρέχουσες απαιτήσεις και τη μεταβαλλόμενη ζήτηση, καθώς και τη γενικότερη διαχείριση του σύνθετου και πολυδιάστατου κύκλου ζωής αυτών των containerized εφαρμογών, από τη δημιουργία έως την κατάργησή τους. Το Kubernetes (K8S), αναγνωρισμένο ως το πιο δημοφιλές, ώριμο και ευρέως υιοθετημένο εργαλείο ενορχήστρωσης επιπέδου παραγωγής, απλοποιεί σημαντικά την πολυπλοκότητα της διαχείρισης μέσω της παροχής ενός πλούσιου συνόλου βασικών λειτουργιών που είναι

κρίσιμες για την ομαλή και αποδοτική λειτουργία των σύγχρονων, κατανεμημένων εφαρμογών. Αυτές οι λειτουργίες περιλαμβάνουν την εξασφάλιση υψηλής διαθεσιμότητας ώστε οι εφαρμογές να παραμένουν προσβάσιμες ακόμη και σε περίπτωση αστοχίας, την πλήρως αυτοματοποιημένη διαδικασία ανάπτυξης νέων εκδόσεων, την δυναμική και αυτόματη κλιμάκωση των υπολογιστικών πόρων για την αποτελεσματική αντιμετώπιση των μεταβολών στην ζήτηση, και την ομαλή και χωρίς διακοπές αναβάθμιση των εφαρμογών, διασφαλίζοντας τη συνεχή λειτουργία των υπηρεσιών. Ουσιαστικά, το K8S παρέχει ένα ολοκληρωμένο, ευέλικτο και φιλικό προς το χρήστη σύστημα για την αποτελεσματική και απρόσκοπτη διαχείριση ενός μεγάλου και διαρκώς μεταβαλλόμενου αριθμού containers σε πολύπλοκα περιβάλλοντα παραγωγής.

Ένα τυπικό Kubernetes cluster αποτελεί ένα περίπλοκο αλλά ταυτόχρονα εξαιρετικά ευέλικτο κατανεμημένο σύστημα, η θεμελιώδης αρχιτεκτονική του οποίου βασίζεται σε έναν αριθμό ανεξάρτητων υπολογιστικών κόμβων, οι οποίοι είναι ευρέως γνωστοί ως Nodes. Κάθε Node μέσα στο cluster αντιπροσωπεύει μια αυτόνομη υπολογιστική μονάδα, η οποία είναι εξοπλισμένη με το δικό της σύνολο υπολογιστικών πόρων, όπως η κεντρική μονάδα επεξεργασίας (CPU) για την εκτέλεση των εντολών, η μνήμη τυχαίας προσπέλασης (RAM) για την αποθήκευση δεδομένων που χρησιμοποιούνται ενεργά, και ο τοπικός ή δικτυακός αποθηκευτικός χώρος για την αποθήκευση των δεδομένων των εφαρμογών και του λειτουργικού συστήματος. Αυτή η θεμελιώδης αρχιτεκτονική δομή επιτρέπει την εύκολη και δυναμική οριζόντια κλιμάκωση των εφαρμογών, προσθέτοντας περισσότερους κόμβους στο cluster για την αντιμετώπιση αυξημένου φόρτου εργασίας, καθώς και την επίτευξη υψηλού επιπέδου διαθεσιμότητας και ανθεκτικότητας στις αστοχίες. Η ύπαρξη πολλαπλών ανεξάρτητων κόμβων διασφαλίζει ότι οι εφαρμογές παραμένουν προσβάσιμες και λειτουργικές ακόμη και σε περίπτωση απρόβλεπτης αστοχίας ενός ή περισσότερων μεμονωμένων κόμβων.

Η εμβέλεια της ενορχήστρωσης καλύπτει **ολόκληρο τον κύκλο ζωής** μιας containerized εφαρμογής, από τη στιγμή της αρχικής της ανάπτυξης μέχρι την τελική της απόσυρση. Οι βασικές της λειτουργίες περιλαμβάνουν:

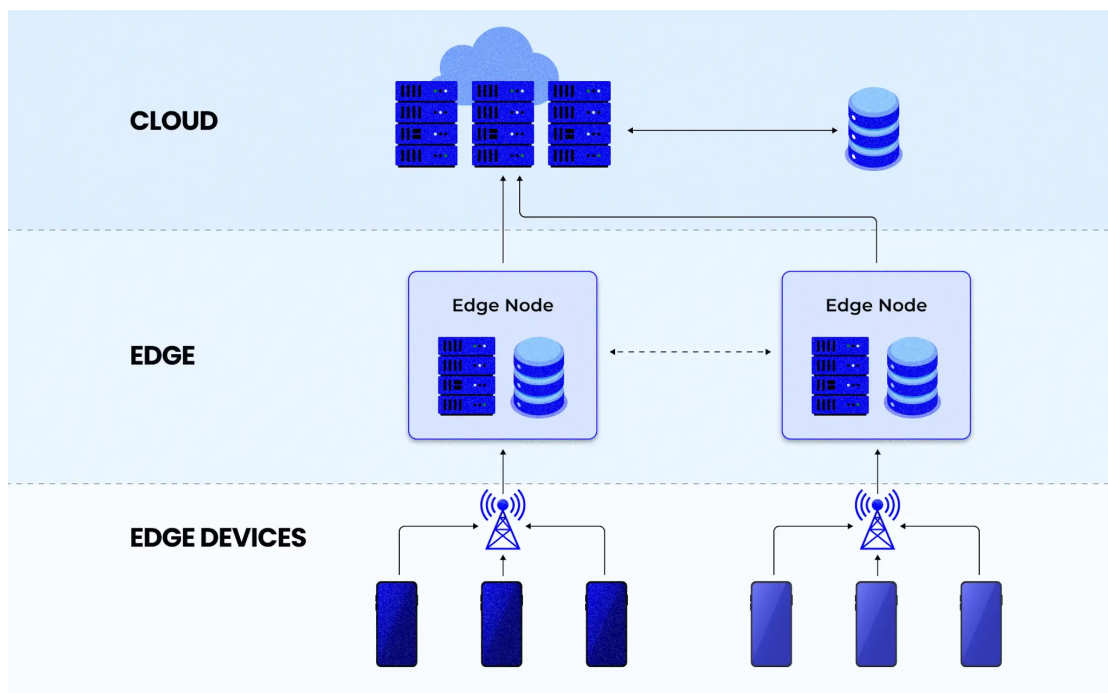
- **Προγραμματισμός και τοποθέτηση:** Αυτή η λειτουργία είναι υπεύθυνη για την ευφυή κατανομή των containers στους διαθέσιμους υπολογιστικούς κόμβους του cluster. Η απόφαση για το πού θα εκτελεστεί κάθε container λαμβάνεται βάσει ενός συνδυασμού παραγόντων, όπως η διαθεσιμότητα των απαραίτητων **πόρων**, η συμμόρφωση με προκαθορισμένες **πολιτικές** και διάφοροι άλλοι **περιορισμοί** που μπορεί να έχουν τεθεί. Ένας αποτελεσματικός μηχανισμός scheduling εξασφαλίζει την ομοιόμορφη κατανομή φορτίου και την αποδοτική χρήση των υπολογιστικών πόρων.
- **Ανάπτυξη:** Η ενορχήστρωση αυτοματοποιεί την διαδικασία δημιουργίας, ανανέωσης ή οριστικής κατάργησης containers μέσα στο cluster. Αυτό επιτυγχάνεται μέσω προηγμένων **abstractions**, όπως τα **Deployments**, τα **ReplicaSets** και τα **StatefulSets**. Τα Deployments παρέχουν έναν δηλωτικό τρόπο για την περιγραφή της επιθυμητής κατάστασης μιας εφαρμογής. Τα ReplicaSets εξασφαλίζουν ότι ένας συγκεκριμένος αριθμός αντιγράφων ενός pod είναι πάντα διαθέσιμος. Τα StatefulSets διαχειρίζονται εφαρμογές με κατάσταση, παρέχοντας μοναδικές ταυτότητες, σταθερό αποθηκευτικό χώρο και προβλέψιμη σειρά στην ανάπτυξη και την κλιμάκωση.[3]

- **Κλιμάκωση:** Η ικανότητα αυτόματης προσαρμογής του αριθμού των containers που εκτελούνται είναι ένα από τα βασικά πλεονεκτήματα της ενορχήστρωσης. Η **κλιμάκωση** μπορεί να είναι **οριζόντια** ή **κάθετη**. Η οριζόντια κλιμάκωση μπορεί να βασίζεται στο τρέχον **φορτίο** της εφαρμογής ή σε προκαθορισμένες **πολιτικές** που ορίζονται μέσω μηχανισμών όπως το **HPA**. Αυτό εξασφαλίζει ότι οι εφαρμογές μπορούν να ανταποκριθούν δυναμικά στις μεταβαλλόμενες απαιτήσεις χωρίς χειροκίνητη παρέμβαση.
- **Παρακολούθηση:** Η συνεχής **εποπτεία** της υγείας και της απόδοσης των **pods**, των υποκείμενων **κόμβων** και των παρεχόμενων **υπηρεσιών** είναι ζωτικής σημασίας. Αυτό επιτυγχάνεται μέσω **control loops**, μηχανισμών συνεχούς σύγκρισης της επιθυμητής με την πραγματική κατάσταση, και της συλλογής και ανάλυσης **metrics**. Η αποτελεσματική παρακολούθηση επιτρέπει την έγκαιρη ανίχνευση προβλημάτων και την αυτόματη ενεργοποίηση μηχανισμών αποκατάστασης.
- **Ανακάλυψη υπηρεσιών και δρομολόγηση:** Σε ένα περιβάλλον μικροϋπηρεσιών, οι διάφορες υπηρεσίες πρέπει να μπορούν να εντοπίζουν και να επικοινωνούν μεταξύ τους αξιόπιστα. Η ενορχήστρωση παρέχει μηχανισμούς **δυναμικής έκθεσης υπηρεσιών** μέσω **abstractions** όπως τα **Services**. Ένα Service παρέχει ένα σταθερό σημείο πρόσβασης σε ένα σύνολο pods που παρέχουν μια συγκεκριμένη λειτουργικότητα. Επιπλέον, μηχανισμοί όπως το **Ingress** επιτρέπουν την εξωτερική πρόσβαση στις υπηρεσίες του cluster, παρέχοντας δυνατότητες **δρομολόγησης** των εισερχόμενων αιτημάτων βάσει κανόνων.[4]
- **Ανθεκτικότητα:** Η ενορχήστρωση ενσωματώνει μηχανισμούς για την αυτόματη αντιμετώπιση αποτυχιών και τη διατήρηση της διαθεσιμότητας των εφαρμογών. Η **αυτόματη ανάκτηση** σημαίνει ότι εάν ένας container ή ένα pod αποτύχει, ο orchestrator μπορεί αυτόματα να προγραμματίσει την αντικατάστασή του. Τεχνικές όπως τα **rolling updates** και τα **canary deployments** επιτρέπουν την σταδιακή αναβάθμιση των εφαρμογών.[5], [6], [7]

2.1.2 Edge Computing

Η υπολογιστική στο άκρο συνιστά μια θεμελιώδη ανατροπή παραδείγματος στον τρόπο συλλογής, επεξεργασίας και ανάλυσης δεδομένων. Αντί να βασίζεται αποκλειστικά σε απομακρυσμένα κέντρα δεδομένων στο νέφος, η υπολογιστική στο άκρο μετατοπίζει την υπολογιστική ισχύ εγγύτερα στην πηγή δημιουργίας των δεδομένων, δηλαδή τις ακραίες συσκευές. Αυτή η μετάβαση έχει σημαντικές επιπτώσεις στην ταχύτητα λήψης αποφάσεων, την ασφάλεια των δεδομένων και την αποτελεσματικότητα ενός ευρέος φάσματος εφαρμογών, ιδίως στον ταχέως αναπτυσσόμενο τομέα του Διαδικτύου των Πραγμάτων.

Ένας από τους κύριους παράγοντες που προωθούν την υιοθέτηση της υπολογιστικής στο άκρο είναι η αυξανόμενη υπολογιστική ισχύς που διατίθεται στις τερματικές συσκευές. Οι σύγχρονες Ακραίες Συσκευές Άκρου, όπως αισθητήρες, κινητά τηλέφωνα, βιομηχανικά ρομπότ και αυτόνομα οχήματα, διαθέτουν αξιόλογες δυνατότητες επεξεργασίας που συχνά παραμένουν ανεκμετάλλευτες. Η αξιοποίηση αυτής της αδρανούς ισχύος μπορεί να οδηγήσει στην «αποκέντρωση» του άκρου, επιτρέποντας την τοπική ανάλυση δεδομένων και μειώνοντας την ανάγκη για συνεχή επικοινωνία με το νέφος.



Εικόνα 2.2 : Ιεραρχική Αρχιτεκτονική Υπολογισμού μεταξύ Edge Devices, Edge Nodes και Cloud

Η Υπολογιστική Νέφος, παρότι προσφέρει τεράστια υπολογιστική ισχύ και επεκτασιμότητα, ενδέχεται να αντιμετωπίσει δυσκολίες στην ανταπόκριση στις απαιτήσεις χαμηλής καθυστέρησης και υψηλής Ποιότητας Υπηρεσίας που προϋποθέτουν πολλές σύγχρονες εφαρμογές. Η μεταφορά μεγάλων όγκων δεδομένων σε απομακρυσμένα κέντρα δεδομένων συνεπάγεται αναπόφευκτα καθυστερήσεις, οι οποίες μπορεί να είναι κρίσιμες σε σενάρια πραγματικού χρόνου, όπως η αυτόνομη οδήγηση ή ο βιομηχανικός αυτοματισμός.[8]

Η Υπολογιστική Άκρου αναδεικνύεται ως μια ελκυστική εναλλακτική λύση, φέρνοντας την επεξεργασία των δεδομένων πλησιέστερα στους τελικούς χρήστες και τις συσκευές που τα παράγουν. Αυτή η προσέγγιση μειώνει σημαντικά την καθυστέρηση, βελτιώνει την ανταπόκριση των εφαρμογών και επιτρέπει τη λήψη αποφάσεων σε πραγματικό χρόνο με βάση τοπικά επεξεργασμένα δεδομένα.

Εντούτοις, οι σύγχρονες πλατφόρμες EC συχνά βασίζονται σε υποδομές που ελέγχονται από παρόχους νέφους και τηλεπικοινωνιακούς παρόχους. Παρότι αυτό παρέχει ορισμένα πλεονεκτήματα, μπορεί επίσης να οδηγήσει σε περιορισμούς όσον αφορά τον συγκεντρωτικό έλεγχο, τη μειωμένη ευελιξία και το υψηλό κόστος που συνδέεται με τη χρήση αυτών των υποδομών. Η αξιοποίηση των λανθανόντων υπολογιστικών πόρων των EEDs προσφέρει τη δυνατότητα δημιουργίας νέων, πιο αποκεντρωμένων και οικονομικά αποδοτικών λύσεων.

Η παράλληλη επεξεργασία δεδομένων μεταξύ πολλαπλών EEDs μπορεί να πολλαπλασιάσει τη συνολική υπολογιστική ισχύ που είναι διαθέσιμη στο άκρο, επιτρέποντας την εκτέλεση πιο σύνθετων εργασιών ανάλυσης δεδομένων τοπικά. Επιπλέον, η διανομή της επεξεργασίας μπορεί να βελτιώσει την ανθεκτικότητα και τη διαθεσιμότητα του συστήματος.

Παρά τα πολλά υποσχόμενα οφέλη, οι υπάρχουσες λύσεις EC αντιμετωπίζουν αρκετούς περιορισμούς. Η έλλειψη ευελιξίας όσον αφορά τη γεωγραφική τοποθέτηση των υποδομών και η περιορισμένη συμβατότητα με διαφορετικά λειτουργικά συστήματα μπορεί να εμποδίσουν την ευρεία υιοθέτηση και την επεκτασιμότητα των λύσεων άκρου. Υφίσταται

αυξανόμενη ανάγκη για οικονομικά αποδοτικές λύσεις που μπορούν να αξιοποιήσουν πλήρως τους πόρους των EEDs, να προσφέρουν μεγαλύτερη ευελιξία στην εγκατάσταση και να διασφαλίζουν υψηλό επίπεδο ιδιωτικότητας και ασφάλειας των δεδομένων.

Παραδείγματα υφιστάμενων τεχνολογιών περιλαμβάνουν το Azure IoT Edge της Microsoft, το οποίο επεκτείνει τις δυνατότητες ανάλυσης δεδομένων του cloud σε συσκευές άκρου, υποστηρίζοντας προηγμένες λειτουργίες όπως η μηχανική μάθηση και η αναγνώριση εικόνας απευθείας στις συσκευές.

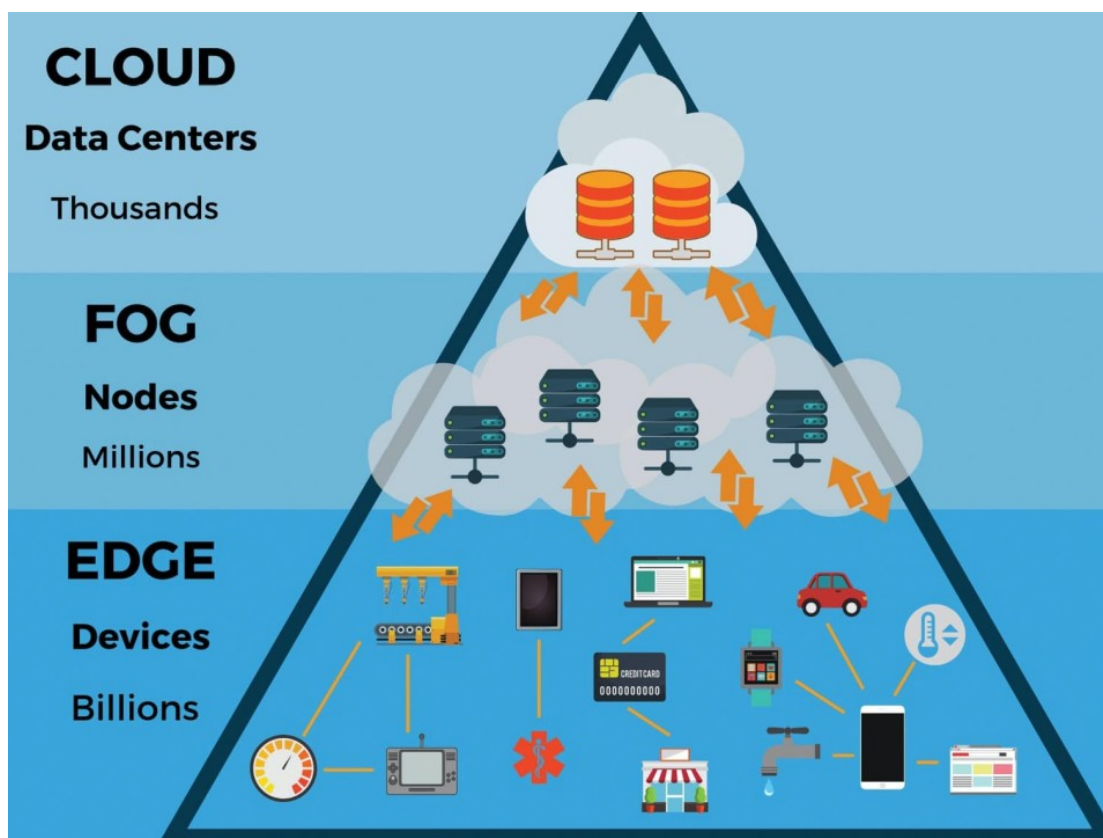
Η αρχιτεκτονική των μικροϋπηρεσιών προσφέρει σημαντικά πλεονεκτήματα για την ανάπτυξη και διαχείριση εφαρμογών στο άκρο. Η διάσπαση μιας μονολιθικής εφαρμογής σε μικρότερες, ανεξάρτητες υπηρεσίες επιτρέπει τη βελτίωση της απόδοσης, τη μείωση της καθυστέρησης και την ενίσχυση της αξιοπιστίας και της επεκτασιμότητας μέσω της αποκέντρωσης και της ευελιξίας στην ανάπτυξη και αναβάθμιση κάθε υπηρεσίας ξεχωριστά.

Ωστόσο, η υπολογιστική στο άκρο δημιουργεί και σημαντικές προκλήσεις. Η στενότητα του εύρους ζώνης για την αρχική συλλογή και την περιστασιακή μεταφορά δεδομένων στο νέφος, το κόστος και οι περιορισμοί αποθήκευσης στις ακραίες συσκευές, καθώς και οι αυξημένες ανησυχίες για την ασφάλεια σε ένα πιο κατακεντρωμένο περιβάλλον, είναι σημαντικοί παράγοντες που πρέπει να αντιμετωπιστούν.

Η μετάβαση από παραδοσιακές, μονολιθικές αρχιτεκτονικές σε αρχιτεκτονικές που βασίζονται σε μικροϋπηρεσίες αποτελεί μια εξελισσόμενη τάση που υπόσχεται να αντιμετωπίσει πολλούς από τους περιορισμούς των μονολιθικών λύσεων, προσφέροντας μεγαλύτερη ευελιξία, επεκτασιμότητα και ανθεκτικότητα για τις εφαρμογές υπολογιστικής στο άκρο. Η αποσύνθεση των εφαρμογών σε μικρότερες, διασυνδεδεμένες υπηρεσίες επιτρέπει την ανεξάρτητη ανάπτυξη, αναβάθμιση και κλιμάκωση κάθε τμήματος, οδηγώντας σε πιο ευέλικτα και αποτελεσματικά συστήματα.[9]

2.1.3 Fog Computing

Το υπολογιστικό νέφος (fog computing), σε άρρηκτη συνάφεια με την καθιερωμένη υπολογιστική στο νέφος (cloud computing), θεμελιώνει τις βάσεις μιας ευρέως διανεμημένης υπολογιστικής υποδομής, γνωστής ως το cloud-to-edge continuum. Αυτή η πρωτοποριακή αρχιτεκτονική επιτρέπει την αδιάλειπτη εκτέλεση ποικίλων υπηρεσιών του Διαδικτύου των Πραγμάτων (IoT), παρέχοντας κρίσιμα πλεονεκτήματα όπως η εξαιρετικά χαμηλή λανθάνουσα κατάσταση και η έμφυτη ευαισθησία στη γεωγραφική θέση των δεδομένων και των συσκευών. Εντούτοις, η διαχείριση ενός τέτοιου ετερογενούς και ευρέως διανεμημένου οικοσυστήματος εγείρει σημαντικές προκλήσεις πολυπλοκότητας, ιδίως όσον αφορά την ταυτόχρονη ικανοποίηση των αναγκών πολλαπλών χρηστών για πρόσβαση σε ένα ευρύ και ποικίλο φάσμα υπολογιστικών πόρων, που βρίσκονται σε διαφορετικές περιοχές της υποδομής. Επιπροσθέτως, καθίσταται απαραίτητη η διασφάλιση ενός αυστηρού επιπέδου απομόνωσης μεταξύ των χρηστών, αποτρέποντας πιθανές παρεμβολές ή παραβιάσεις ασφαλείας, καθώς και η εγγύηση σταθερής και προβλέψιμης απόδοσης για κάθε παρεχόμενη υπηρεσία, ανεξαρτήτως του φόρτου ή της θέσης των υπολοίπων χρηστών.[10], [11]



Εικόνα. 2.3: Ιεραρχικό Υπολογιστικό Μοντέλο Cloud–Fog–Edge και Κλίμακα Ανάπτυξης

Η ραγδαία και σχεδόν εκρηκτική αύξηση των ευρέως διαδεδομένων ψηφιακών υπηρεσιών, σε συνδυασμό με τον διαρκή πολλαπλασιασμό των συνδεδεμένων συσκευών IoT, έχει επιφέρει μια άνευ προηγουμένου πίεση στην χωρητικότητα και τις συνολικές δυνατότητες των σύγχρονων κινητών δικτύων. Η αποτελεσματική ανάλυση και αξιοποίηση αυτού του συνεχώς αυξανόμενου όγκου δεδομένων μπορεί να ανοίξει νέους και ανεξερεύνητους ορίζοντες, παρέχοντας στις επιχειρήσεις και τους οργανισμούς τη δυνατότητα να αναπτύξουν καινοτόμα επιχειρηματικά μοντέλα, να δημιουργήσουν νέες και διαφοροποιημένες πηγές εσόδων και να βελτιστοποιήσουν τις υφιστάμενες λειτουργίες τους σε μια πρωτοφανή κλίμακα. Αυτή η κυρίαρχη τάση στον ψηφιακό μετασχηματισμό της κοινωνίας και της οικονομίας καθιστά την αδιάλειπτη υποστήριξη της κινητικότητας των χρηστών και των συσκευών τους, την ευέλικτη γεωγραφική κατανομή των απαραίτητων υπολογιστικών πόρων ανάλογα με τη ζήτηση, την ενσωμάτωση της χωρικής αντίληψης στις διάφορες εφαρμογές και την εξασφάλιση εξαιρετικά χαμηλής καθυστέρησης στην επικοινωνία ως θεμελιώδεις και αδιαπραγμάτευτες απαιτήσεις για την υποκείμενη υπολογιστική υποδομή που θα στηρίζει αυτή τη νέα ψηφιακή εποχή.

Οι συσκευές του Διαδικτύου των Πραγμάτων (IoT), εγγενώς σχεδιασμένες για χαμηλή κατανάλωση ενέργειας και μικρό μέγεθος, συχνά αντιμετωπίζουν περιορισμούς στους διαθέσιμους υπολογιστικούς πόρους, στην επεξεργαστική ισχύ και στη χωρητικότητα της μπαταρίας τους. Αυτός ο θεμελιώδης περιορισμός υπαγορεύει την ανάγκη μεταφοράς της απαιτητικής υπολογιστικής επεξεργασίας των δεδομένων που συλλέγουν σε πιο ισχυρά υπολογιστικά συστήματα με μεγαλύτερες δυνατότητες. Το παραδοσιακό μοντέλο του cloud computing, που εστιάζει στην κεντρική συγκέντρωση των υπολογιστικών πόρων σε απομακρυσμένα κέντρα δεδομένων, συχνά χιλιάδες χιλιόμετρα μακριά από τις συσκευές και τους χρήστες, παρουσιάζει αυξανόμενες προκλήσεις στην ικανοποίηση της αυξανόμενης

ζήτησης για χαμηλή καθυστέρηση και τοπική επεξεργασία δεδομένων. Η συνεχής και μαζική μεταφορά μεγάλου όγκου δεδομένων προς τα κεντρικά data centers αναπόφευκτα συνεπάγεται σημαντικές καθυστερήσεις δικτύου, οι οποίες με τη σειρά τους επηρεάζουν αρνητικά την εμπειρία των τελικών χρηστών, ειδικά σε εφαρμογές που απαιτούν αλληλεπίδραση σε πραγματικό χρόνο, όπως η επαυξημένη και εικονική πραγματικότητα, η αυτόνομη οδήγηση, τα βιομηχανικά συστήματα αυτοματισμού και η τηλεϊατρική. Σε αυτό το εξελισσόμενο τεχνολογικό τοπίο, η αξιοποίηση των υπολογιστικών πόρων που βρίσκονται φυσικά πιο κοντά στις συσκευές IoT, δηλαδή στην άκρη του δικτύου, αναδεικνύεται ως μια πιο πρακτική, αποδοτική και βιώσιμη στρατηγική για την αποτελεσματική αντιμετώπιση αυτών των σύγχρονων τεχνολογικών προκλήσεων και την κάλυψη των απαιτήσεων για ταχύτερη επεξεργασία και λήψη αποφάσεων.[11]

Ενώ η εικονικοποίηση των υπολογιστικών πόρων στο άκρο του δικτύου συχνά αποτελεί μια ευέλικτη λύση, προσφέροντας σημαντικά πλεονεκτήματα στην δυναμική κατανομή των πόρων και βελτιώνοντας την αποδοτικότητα της χρήσης της υποδομής, παρατηρείται μια αυξανόμενη πληθώρα σύγχρονων υπηρεσιών που βασίζονται στην τεχνολογία του Διαδικτύου των Πραγμάτων και χαρακτηρίζονται από εξαιρετικά υψηλές απαιτήσεις σε υπολογιστική ισχύ. Αυτές οι υπηρεσίες συχνά περιλαμβάνουν την ανάλυση τεράστιων όγκων δεδομένων σε πραγματικό χρόνο για την εξαγωγή κρίσιμων πληροφοριών και την υποστήριξη άμεσων ενεργειών. Σε τέτοιες περιπτώσεις, τα εικονικοποιημένα περιβάλλοντα ενδέχεται να μην είναι σε θέση να παρέχουν την απαραίτητη απόδοση και αποτελεσματικότητα λόγω του πρόσθετου φόρτου που εισάγει το επίπεδο εικονικοποίησης. Αυτός ο φόρτος μπορεί να επιβραδύνει σημαντικά την επεξεργασία των δεδομένων, να αυξήσει την καθυστέρηση και να μειώσει τη συνολική απόδοση των εφαρμογών, καθιστώντας την εικονικοποίηση λιγότερο ελκυστική για φόρτους εργασίας με αυστηρές απαιτήσεις απόδοσης.

Αυτή η εγγενής καθυστέρηση στα εικονικοποιημένα περιβάλλοντα μπορεί να έχει σοβαρές επιπτώσεις, ειδικά σε εφαρμογές που απαιτούν άμεση και αδιάλειπτη ανταπόκριση για την ασφάλεια και την αποτελεσματική λειτουργία, όπως τα συστήματα αυτόνομης οδήγησης όπου η λήψη αποφάσεων σε χιλιοστά του δευτερολέπτου είναι κρίσιμη, τα βιομηχανικά συστήματα αυτοματισμού που ελέγχουν ευαίσθητες διαδικασίες παραγωγής και οι υπηρεσίες υγειονομικής περίθαλψης που βασίζονται σε δεδομένα πραγματικού χρόνου για την παρακολούθηση ασθενών και την παροχή έγκαιρης ιατρικής φροντίδας. Επιπλέον, η διαμοιραζόμενη φύση των υπολογιστικών πόρων σε ένα εικονικοποιημένο περιβάλλον ενέχει τον κίνδυνο παρενεργειών, όπου ένα ιδιαίτερα απαιτητικό υπολογιστικό φορτίο μιας εφαρμογής μπορεί να καταναλώσει δυσανάλογα μεγάλο μέρος των διαθέσιμων πόρων του φυσικού κόμβου, επηρεάζοντας αρνητικά την απόδοση άλλων υπηρεσιών ή εφαρμογών που φιλοξενούνται στον ίδιο φυσικό κόμβο. Αυτός ο ανταγωνισμός για πόρους μπορεί να οδηγήσει σε αστάθεια του συστήματος, απρόβλεπτες διακοπές λειτουργίας και γενικότερη υποβάθμιση της ποιότητας των παρεχόμενων υπηρεσιών, υπογραμμίζοντας την ανάγκη για προσεκτικό σχεδιασμό και διαχείριση των πόρων στα περιβάλλοντα edge computing.[12]

Τα bare-metal clouds, από την άλλη πλευρά, αντιπροσωπεύουν μια εναλλακτική αρχιτεκτονική που αντιμετωπίζει αποτελεσματικά αυτές τις προκλήσεις. Προσφέρουν την αποκλειστική χρήση ενός φυσικού διακομιστή σε έναν μόνο χρήστη, εξαλείφοντας τον ανταγωνισμό για πόρους και τον κίνδυνο παρενεργειών από γειτονικά εικονικά μηχανήματα. Παράλληλα, διατηρούν τα πλεονεκτήματα του cloud computing, όπως η δυνατότητα διάθεσης των πόρων κατ' απαίτηση και η χρέωση με βάση την πραγματική χρήση. Η

ισχυρότερη απομόνωση των υπολογιστικών πόρων που παρέχουν τα bare-metal clouds προσφέρει στους χρήστες μεγαλύτερη ελευθερία και ευελιξία για την ανάπτυξη εξατομικευμένων λύσεων διαχείρισης πόρων. Μπορούν να διαμορφώσουν το υλικό και το λογισμικό του διακομιστή ακριβώς σύμφωνα με τις ειδικές και μοναδικές ανάγκες της κάθε εφαρμογής τους, επιτυγχάνοντας βέλτιστη απόδοση. Αυτό επιτρέπει την ελαχιστοποίηση του πρόσθετου φόρτου που συνεπάγεται η εικονικοποίηση για ιδιαίτερα κρίσιμα και απαιτητικά υπολογιστικά φορτία. Είτε μέσω της χρήσης ελαφρών containers, τα οποία μειώνουν σημαντικά το overhead σε σύγκριση με τις παραδοσιακές εικονικές μηχανές, είτε μέσω της άμεσης και απρόσκοπτης πρόσβασης στους υποκείμενους φυσικούς πόρους, τα bare-metal clouds προσφέρουν ένα ιδανικό περιβάλλον για εφαρμογές IoT με υψηλές απαιτήσεις σε απόδοση και χαμηλή καθυστέρηση.[12]

2.1.4 Cloud-to-Thing Continuum

Το **Cloud-to-Thing Continuum** αναφέρεται σε ένα ενιαίο και δυναμικό υπολογιστικό οικοσύστημα που εκτείνεται από τα κεντρικά data centers του cloud έως τις συσκευές IoT ("things") στο άκρο του δικτύου. Περιλαμβάνει επίπεδα όπως το fog και το edge computing, επιτρέποντας την κατανομημένη επεξεργασία δεδομένων κοντά στην πηγή τους. Αυτό προσφέρει χαμηλή καθυστέρηση, αυξημένη ασφάλεια, αποδοτικότητα πόρων και real-time απόκριση — χαρακτηριστικά κρίσιμα για εφαρμογές όπως έξυπνες πόλεις, βιομηχανικούς αυτοματισμούς και την υγειονομική περίθαλψη. Το continuum αυτό απαιτεί προηγμένη οργάνωση για την αποτελεσματική αξιοποίηση των υπολογιστικών πόρων σε όλα τα επίπεδα.

Η έννοια του **Cloud-to-Thing Continuum** αποτελεί ένα άλμα εξέλιξης στην αρχιτεκτονική κατανομημένων συστημάτων, σηματοδοτώντας μια ριζική αναχώρηση από τα παραδοσιακά μοντέλα cloud-to-edge computing και αγκαλιάζοντας πλήρως το εκτεταμένο και ετερογενές οικοσύστημα του Διαδικτύου των Πραγμάτων (IoT). Αυτή η ολιστική και ενοποιημένη προσέγγιση αναδεικνύεται ως κρίσιμη για την αποτελεσματική διαχείριση και αξιοποίηση της πολυπλοκότητας που χαρακτηρίζει σύγχρονα σενάρια, ιδίως εκείνα που ενσωματώνουν serverless αρχιτεκτονικές. Η χρήση serverless υποδομών προσφέρει σημαντικά πλεονεκτήματα, όπως η αυξημένη ευελιξία και η δραστική μείωση της διαχείρισης υποδομής, επιτρέποντας στους developers να επικεντρωθούν στην ανάπτυξη καινοτόμων εφαρμογών χωρίς την επιβάρυνση της συντήρησης των διακομιστών. Επιπλέον, το Cloud-to-Thing Continuum διευκολύνει την ενσωμάτωση προηγμένων τεχνικών όπως το federated learning. Στο πλαίσιο αυτό, η γνώση και τα μοντέλα μηχανικής μάθησης αναπτύσσονται συλλογικά από έναν μεγάλο αριθμό αποκεντρωμένων συσκευών IoT, χωρίς την ανάγκη συγκέντρωσης όλων των δεδομένων σε ένα κεντρικό σημείο. Αυτή η προσέγγιση όχι μόνο βελτιώνει την αποτελεσματικότητα της εκπαίδευσης μοντέλων σε τεράστιους όγκους δεδομένων, αλλά παράλληλα διασφαλίζει την ιδιωτικότητα και την ασφάλεια των ευαίσθητων πληροφοριών που παράγονται από τις συσκευές.[13], [14]

Η εκρηκτική αύξηση του αριθμού των συνδεδεμένων συσκευών IoT έχει οδηγήσει στη δημιουργία ενός πρωτοφανούς όγκου δεδομένων. Αυτή η πλημμύρα πληροφοριών θέτει σημαντικές προκλήσεις για τις συμβατικές, συγκεντρωτικές υποδομές νέφους. Τα αναπόφευκτα σημεία συμφόρησης που προκύπτουν κατά την επεξεργασία και τη μεταφορά αυτών των τεράστιων ποσοτήτων δεδομένων, οι αυξημένες καθυστερήσεις (latency) που επηρεάζουν αρνητικά την ανταπόκριση σε εφαρμογές που απαιτούν επεξεργασία σε πραγματικό χρόνο (real-time), και η δυσκολία αποτελεσματικής κλιμάκωσης των υποδομών

για την αντιμετώπιση αυτής της διαρκώς αυξανόμενης ροής δεδομένων, καταδεικνύουν την επιτακτική ανάγκη για μια ριζική μετατόπιση της υπολογιστικής ισχύος πιο κοντά στην πηγή δημιουργίας των δεδομένων, δηλαδή στο άκρο (edge) του δικτύου.

Το Cloud-to-Thing Continuum επιτρέπει τη δημιουργία ενός συνεχούς και αδιάλειπτου φάσματος υπολογιστικών δυνατοτήτων, που εκτείνεται από το κεντρικό νέφος έως τις πιο απομακρυσμένες συσκευές IoT. Αυτή η αρχιτεκτονική προσφέρει μια ευέλικτη και δυναμική πλατφόρμα για την ανάπτυξη και την εκτέλεση κατανεμημένων εφαρμογών, επιτρέποντας την βελτιστοποίηση της επεξεργασίας δεδομένων, τη μείωση της καθυστέρησης, την ενίσχυση της ασφάλειας και την αποτελεσματικότερη διαχείριση του αυξανόμενου όγκου δεδομένων που παράγεται από το IoT. Με αυτόν τον τρόπο, το Cloud-to-Thing Continuum ανοίγει νέους ορίζοντες για την αξιοποίηση της δύναμης του IoT σε ένα ευρύ φάσμα βιομηχανιών και εφαρμογών, οδηγώντας σε πιο ευφυείς, αυτόνομες και αποδοτικές λύσεις.

Η **υπολογιστική στο άκρο (Edge Computing)** αναδεικνύεται ως μια καίρια λύση σε αυτές τις προκλήσεις, επιτρέποντας την ανάπτυξη έξυπνων συσκευών και την εγκατάσταση υπολογιστικών πόρων στρατηγικά τοποθετημένων πλησίον των πηγών δημιουργίας δεδομένων. Αυτή η γεωγραφική εγγύτητα προσφέρει μια σειρά από σημαντικά πλεονεκτήματα. Πρώτον και κυριότερο, μειώνει δραστικά την καθυστέρηση (latency) στην επεξεργασία δεδομένων και στην ανταπόκριση των συστημάτων, κάτι που είναι κρίσιμο για εφαρμογές που απαιτούν αλληλεπίδραση σε πραγματικό χρόνο, όπως η αυτόνομη οδήγηση, η βιομηχανική αυτοματοποίηση και η επαυξημένη/εικονική πραγματικότητα. Δεύτερον, περιορίζει σημαντικά την εξάρτηση από την κεντρική υποδομή νέφους, καθιστώντας τα συστήματα πιο αυτόνομα και ανθεκτικά σε διακοπές σύνδεσης. Τρίτον, βελτιώνει την ιδιωτικότητα και την ασφάλεια των δεδομένων, καθώς ευαίσθητες πληροφορίες μπορούν να υποστούν επεξεργασία και ανάλυση τοπικά, χωρίς να χρειάζεται να μεταφερθούν στο cloud. Τέταρτον, ενισχύει την ανθεκτικότητα των συστημάτων, καθώς οι τοπικές μονάδες μπορούν να συνεχίσουν τη λειτουργία τους ακόμα και σε περίπτωση απώλειας σύνδεσης με το κεντρικό cloud. Επιπλέον, η επεξεργασία δεδομένων στο άκρο οδηγεί σε μείωση του απαιτούμενου εύρους ζώνης για τη μεταφορά δεδομένων στο cloud, γεγονός που μεταφράζεται σε σημαντική μείωση του κόστους λειτουργίας, ειδικά για εφαρμογές που παράγουν τεράστιους όγκους δεδομένων.[15]

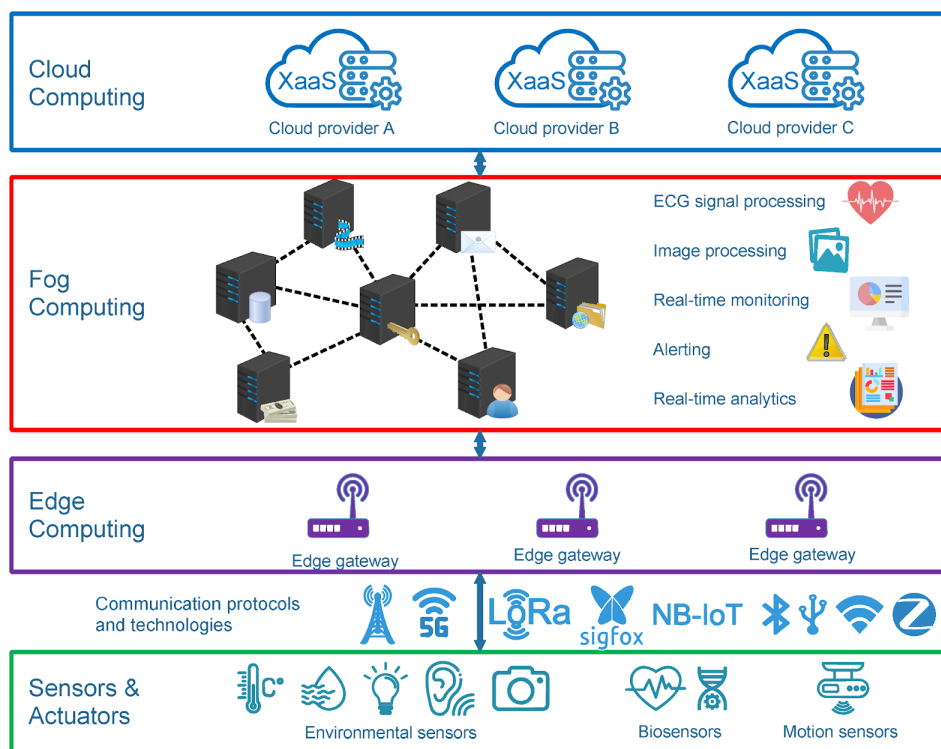
Παρά την αυξανόμενη σημασία του edge computing, το **cloud computing** διατηρεί έναν θεμελιώδη και αναντικατάστατο ρόλο στο ευρύτερο οικοσύστημα. Η εγγενής κλιμακωσιμότητα του cloud, η ικανότητα για κεντρική διαχείριση και αποθήκευση δεδομένων σε τεράστια κλίμακα, καθώς και η πληθώρα υπηρεσιών υψηλού επιπέδου (όπως analytics, machine learning, και data warehousing) που προσφέρει, το καθιστούν ιδανικό για εργασίες που απαιτούν σημαντικούς υπολογιστικούς πόρους, ανάλυση μεγάλων συνόλων δεδομένων και μακροπρόθεσμη αποθήκευση. Η αποτελεσματική ενοποίηση των υποδομών cloud και edge επιτρέπει τη δυναμική κατανομή του φόρτου εργασίας μεταξύ των δύο περιβάλλοντων, καθώς και τη δημιουργία ευέλικτων και προσαρμοστικών υποδομών που μπορούν να ανταποκριθούν στις μεταβαλλόμενες ανάγκες των εφαρμογών και στη διαθεσιμότητα των πόρων σε πραγματικό χρόνο.

Οι **εφαρμογές Machine Learning (ML) και Τεχνητής Νοημοσύνης (AI)** αξιοποιούν στο έπακρο αυτές τις υβριδικές αρχιτεκτονικές cloud-edge. Συγκεκριμένα, η εκπαίδευση πολύπλοκων μοντέλων ML και AI, η οποία απαιτεί σημαντικούς υπολογιστικούς πόρους και μεγάλα σύνολα δεδομένων, πραγματοποιείται συνήθως στο cloud. Στη συνέχεια, τα

εκπαιδευμένα μοντέλα αναπτύσσονται και εκτελούνται για την εξαγωγή συμπερασμάτων (inference) στο άκρο, πλησίον των πηγών δεδομένων, επιτρέποντας την λήψη γρήγορων και έξυπνων αποφάσεων σε πραγματικό χρόνο. Πρόσφατες εξελίξεις στον τομέα επιτρέπουν την επιτάχυνση των εργασιών ML στο edge μέσω της χρήσης εξειδικευμένων συνεπεξεργαστών χαμηλής κατανάλωσης ενέργειας, καθιστώντας δυνατή την εκτέλεση πιο σύνθετων μοντέλων τοπικά. Επιπλέον, η εφαρμογή του Federated Learning αντιπροσωπεύει μια πρωτοποριακή προσέγγιση, όπου πολλαπλοί edge κόμβοι συνεργάζονται για την εκπαίδευση ενός κοινού μοντέλου ML χωρίς να χρειάζεται να ανταλλάξουν τα τοπικά τους δεδομένα, διασφαλίζοντας έτσι την προστασία της ιδιωτικότητας των δεδομένων και ξεπερνώντας τους περιορισμούς συγκεντρωμένης εκπαίδευσης.

Το **fog computing** αναδύεται ως ένα κρίσιμο ενδιάμεσο στρώμα που γεφυρώνει το χάσμα μεταξύ του ισχυρού αλλά απομακρυσμένου cloud και των πολυάριθμων και συχνά περιορισμένων σε πόρους edge nodes. Το fog computing επεκτείνει τις δυνατότητες του cloud προς το άκρο του δικτύου μέσω της ανάπτυξης ευφών κόμβων που μπορούν να εγκατασταθούν σε διάφορες συσκευές με υπολογιστική ισχύ, όπως routers, gateways, βιομηχανικούς ελεγκτές και ακόμη και αυτοκίνητα. Η τυπική αρχιτεκτονική του fog computing είναι ιεραρχική και πολυεπίπεδη, επιτρέποντας την κατανομή λειτουργιών επεξεργασίας, αποθήκευσης και δικτύωσης σε διάφορα επίπεδα εγγύτητας προς τις συσκευές IoT. Αυτή η αρχιτεκτονική προσφέρει τη δυνατότητα συνεργατικής επεξεργασίας δεδομένων εντός του ίδιου επιπέδου fog, μειώνοντας περαιτέρω την καθυστέρηση και βελτιώνοντας την αποδοτικότητα.[15], [16]

Συνολικά, το **cloud-to-thing continuum** περιγράφει ένα βαθιά κατανεμημένο και εξαιρετικά δυναμικό υπολογιστικό περιβάλλον που εκτείνεται απρόσκοπτα από τις απλούστερες αισθητήριες συσκευές (things), οι οποίες συλλέγουν δεδομένα από το φυσικό κόσμο και συχνά διαθέτουν περιορισμένους πόρους επεξεργασίας και ενέργειας, έως τις πιο ισχυρές και κεντρικές υποδομές cloud, που προσφέρουν τεράστια υπολογιστική ισχύ, αποθηκευτικό χώρο και προηγμένες αναλυτικές δυνατότητες. Σε ένα τέτοιο περίπλοκο και διαρκώς εξελισσόμενο οικοσύστημα, η αλληλεπίδραση και η συνεργασία μεταξύ των άκρων του συνεχούς είναι ζωτικής σημασίας για την υλοποίηση πλήθους σύγχρονων εφαρμογών και υπηρεσιών, όπως η έξυπνη βιομηχανία, η αυτόνομη οδήγηση, η ακριβής γεωργία και οι έξυπνες πόλεις. Η έννοια αυτή υπογραμμίζει την ανάγκη για ευέλικτες αρχιτεκτονικές που μπορούν να διαχειριστούν την ετερογένεια των συσκευών, των δικτύων και των απαιτήσεων επεξεργασίας, καθώς και να εξασφαλίσουν την ασφάλεια και την ιδιωτικότητα των δεδομένων σε όλο το εύρος του συνεχούς. Επιπλέον, αναδεικνύει την σημασία της ορχήστρωσης και της διαχείρισης αυτών των κατανεμημένων πόρων για την επίτευξη βέλτιστης απόδοσης, χαμηλής καθυστέρησης και υψηλής διαθεσιμότητας.[17]



Εικόνα 2.6 – Αρχιτεκτονική κατακεντρωμένου υπολογιστικού συστήματος στο Cloud-to-Thing Continuum με διαστρωμάτωση σε επίπεδα Cloud, Fog, Edge και αισθητήρων.

2.1.5 Τεχνητή Νοημοσύνη και Μηχανική Μάθηση

Η Τεχνητή Νοημοσύνη (Artificial Intelligence - AI) ορίζεται ως το διεπιστημονικό πεδίο της επιστήμης των υπολογιστών που αποσκοπεί στη δημιουργία υπολογιστικών συστημάτων ικανών να προσομοιώνουν, να αναπαράγουν και, σε ορισμένες περιπτώσεις, να επεκτείνουν πτυχές της ανθρώπινης νοημοσύνης. Αυτό το ευρύ πεδίο περιλαμβάνει την ανάπτυξη και εφαρμογή προηγμένων αλγορίθμων και τεχνολογιών που επιτρέπουν στις μηχανές να εκτελούν ένα ευρύ φάσμα γνωστικών λειτουργιών, οι οποίες παραδοσιακά απαιτούν ανθρώπινη διάνοια.

Οι βασικές λειτουργίες που επιδιώκει να αναπαράγει και να βελτιώσει η AI περιλαμβάνουν τη συλλογιστική και την επίλυση προβλημάτων, την αναγνώριση προτύπων σε πολύπλοκα δεδομένα, τη λήψη αποφάσεων υπό συνθήκες αβεβαιότητας, την αντίληψη του περιβάλλοντος μέσω αισθητήρων (όπως η όραση και η ακοή), και την επεξεργασία φυσικής γλώσσας για την κατανόηση και παραγωγή ανθρώπινου λόγου. Όπως περιγράφεται εκτενώς στη σχετική βιβλιογραφία και τις σύγχρονες ερευνητικές τάσεις, η AI επιδιώκει όχι μόνο να μιμηθεί τις ικανότητες του ανθρώπινου νου, αλλά και να τις επεκτείνει, προσφέροντας δυνατότητες που υπερβαίνουν τα βιολογικά όρια.

Αυτό επιτυγχάνεται μέσω της ενσωμάτωσης πολλαπλών μεθοδολογιών και υπο-πεδίων, όπως:

- Μηχανική Μάθηση (Machine Learning - ML): Ένα υπο-πεδίο της AI που επικεντρώνεται στην ανάπτυξη αλγορίθμων που επιτρέπουν στους υπολογιστές να

"μαθαίνουν" από δεδομένα, χωρίς να είναι ρητά προγραμματισμένοι. Περιλαμβάνει την επιβλεπόμενη, μη επιβλεπόμενη, και ενισχυτική μάθηση.

- Βαθιά Μάθηση (Deep Learning - DL): Ένα εξειδικευμένο κομμάτι της ML που χρησιμοποιεί τεχνητά νευρωνικά δίκτυα με πολλές κρυφές στρώσεις (deep neural networks) για την αναγνώριση πολύπλοκων προτύπων σε μεγάλα σύνολα δεδομένων, όπως εικόνες, ήχο και κείμενο.
- Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP): Αφορά την αλληλεπίδραση μεταξύ υπολογιστών και ανθρώπινης (φυσικής) γλώσσας, συμπεριλαμβανομένης της κατανόησης, ερμηνείας και παραγωγής ανθρώπινου λόγου.
- Όραση Υπολογιστών (Computer Vision - CV): Επιδιώκει να προσδώσει στους υπολογιστές την ικανότητα να "βλέπουν" και να ερμηνεύουν οπτικές πληροφορίες από τον κόσμο, όπως εικόνες και βίντεο.
- Ρομποτική: Η ενσωμάτωση της ΑΙ σε φυσικά συστήματα (ρομπότ) για την εκτέλεση αυτόνομων λειτουργιών σε πραγματικά περιβάλλοντα.

Η Τεχνητή Νοημοσύνη (ΑΙ) έχει ξεπεράσει τα όρια της θεωρίας και έχει ενσωματωθεί βαθιά σε πρακτικές εφαρμογές σε ένα ευρύ φάσμα τομέων, επηρεάζοντας την υγεία, την εκπαίδευση, τις μεταφορές και τη βιομηχανία. Αυτή η εξέλιξη διαμορφώνει δυναμικά το μέλλον της τεχνολογίας και της κοινωνίας, εισάγοντας νέες δυνατότητες και προκλήσεις.

Ένα κεντρικό υποπεδίο της ΑΙ είναι η Μηχανική Μάθηση (Machine Learning - ML), η οποία επικεντρώνεται στην αυτόματη απόκτηση γνώσης από δεδομένα. Η ML λειτουργεί αναπτύσσοντας αλγόριθμους που επιτρέπουν στα υπολογιστικά συστήματα να βελτιώνουν την απόδοσή τους σε συγκεκριμένα καθήκοντα μέσω της εμπειρίας, χωρίς την ανάγκη ρητού προγραμματισμού. Η βιβλιογραφία αναδεικνύει τη μηχανική μάθηση ως το θεμέλιο πολλών σύγχρονων εφαρμογών ΑΙ, από την ακριβή διάγνωση ασθενειών και την αυτόνομη πλοήγηση οχημάτων έως την ανάλυση συναισθημάτων και την εξατομίκευση υπηρεσιών. Η ικανότητα της ML να αναγνωρίζει μοτίβα και να κάνει προβλέψεις από μεγάλα σύνολα δεδομένων έχει φέρει επανάσταση σε τομείς όπως η χρηματοοικονομική ανάλυση, η διαχείριση εφοδιαστικής αλυσίδας και η προγνωστική συντήρηση.[18]

Η έννοια της νοημοσύνης, είτε φυσικής είτε τεχνητής, περιλαμβάνει μια σειρά από πολύπλοκες ικανότητες, όπως η μάθηση, η μνήμη, η συλλογιστική, η αντίληψη και η γλώσσα. Ενώ η τεχνητή νοημοσύνη έχει σημειώσει τεράστια πρόοδο στην επεξεργασία αντικειμενικών δεδομένων (fact-based computation), η ανθρώπινη νοημοσύνη διαπρέπει στην εκτίμηση υποκειμενικών αξιών, στην κατανόηση περίπλοκων ηθικών διλημμάτων και κοινωνικών πλαισίων, καθώς και στην προσαρμογή της γνώσης βάσει του συνεχώς μεταβαλλόμενου κοινωνικού και φυσικού περιβάλλοντος. Αυτή η δυναμική προσαρμογή και η ικανότητα κατανόησης των αποχρώσεων του ανθρώπινου βίου παραμένουν σημαντικές προκλήσεις για τις σημερινές μηχανές. Η συνεχής έρευνα στην ΑΙ στοχεύει στη γεφύρωση αυτού του χάσματος, επιδιώκοντας την ανάπτυξη συστημάτων που μπορούν να αλληλεπιδρούν με τον κόσμο με πιο ανθρώπινο και ευέλικτο τρόπο, ενσωματώνοντας στοιχεία κοινής λογικής, συναισθηματικής νοημοσύνης και κριτικής σκέψης.[19]

Σύμφωνα με πρόσφατες μελέτες, η εξέλιξη της ΑΙ δεν περιορίζεται πλέον σε βελτιώσεις αλγοριθμικής απόδοσης, αλλά απαιτεί βαθύτερη ενσωμάτωση του ανθρώπου στο σύστημα (human-computer integration) ώστε να επιτευχθεί αληθινή τεχνητή νοημοσύνη. Αυτή η προσέγγιση εστιάζει στη συνεργασία ανθρώπου-μηχανής-περιβάλλοντος με στόχο την ενίσχυση της προσαρμοστικότητας, της ερμηνευσιμότητας και της αξιοπιστίας των συστημάτων. Παράλληλα, επισημαίνεται η ανάγκη για συστήματα που συνδυάζουν συλλογιστική βασισμένη σε γεγονότα και αξίες (fact-based και value-based reasoning), ώστε να επιτυγχάνεται ισορροπία μεταξύ ακρίβειας και ηθικής κατανόησης.

Η μηχανική μάθηση (Machine Learning) αποτελεί έναν κλάδο της τεχνητής νοημοσύνης που εστιάζει στην ανάπτυξη αλγορίθμων και μοντέλων που επιτρέπουν στους υπολογιστές να «μαθαίνουν» από δεδομένα, χωρίς να είναι ρητά προγραμματισμένοι για κάθε πιθανή περίπτωση. Αυτή η ικανότητα μάθησης επιτρέπει στα συστήματα να αναγνωρίζουν πρότυπα, να λαμβάνουν αποφάσεις και να κάνουν προβλέψεις, βελτιώνοντας την απόδοσή τους με την πάροδο του χρόνου και την έκθεση σε νέα δεδομένα.

Η μηχανική μάθηση διακρίνεται σε επιμέρους, θεμελιώδεις κατηγορίες, καθεμία με τη δική της προσέγγιση και εφαρμογές:

- **Επιβλεπόμενη μάθηση (Supervised Learning):** Σε αυτή την κατηγορία, τα μοντέλα εκπαιδεύονται σε δεδομένα που έχουν προκαθορισμένες ετικέτες ή "σωστές" απαντήσεις. Ουσιαστικά, το σύστημα "επιβλέπεται" κατά τη διαδικασία μάθησης από τα γνωστά αποτελέσματα. Στόχος είναι το μοντέλο να μάθει τη σχέση μεταξύ των δεδομένων εισόδου και των αντίστοιχων ετικετών εξόδου, ώστε να μπορεί να προβλέπει τις ετικέτες για νέα, άγνωστα δεδομένα. Χαρακτηριστικά παραδείγματα περιλαμβάνουν την ταξινόμηση (π.χ. αναγνώριση spam emails, διάγνωση ασθενειών) και την παλινδρόμηση (π.χ. πρόβλεψη τιμών κατοικιών, πρόβλεψη χρηματιστηριακών δεικτών).
- **Μη επιβλεπόμενη μάθηση (Unsupervised Learning):** Αντίθετα με την επιβλεπόμενη μάθηση, εδώ τα μοντέλα εκπαιδεύονται σε δεδομένα χωρίς προϋπάρχουσες ετικέτες. Ο κύριος στόχος είναι η ανίχνευση κρυμμένων προτύπων, δομών ή σχέσεων εντός των δεδομένων, χωρίς κάποια εξωτερική καθοδήγηση. Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη για την εξερεύνηση δεδομένων και την κατανόηση της εσωτερικής τους οργάνωσης. Βασικές εφαρμογές περιλαμβάνουν τη συσταδοποίηση (clustering), όπου δεδομένα ομαδοποιούνται με βάση την ομοιότητά τους (π.χ. τμηματοποίηση αγοράς, ομαδοποίηση γενετικών δεδομένων), και τη μείωση διαστάσεων, η οποία απλοποιεί σύνθετα δεδομένα διατηρώντας τις κρίσιμες πληροφορίες.
- **Ενισχυτική μάθηση (Reinforcement Learning):** Αυτή η προσέγγιση είναι εμπνευσμένη από τον τρόπο που οι άνθρωποι και τα ζώα μαθαίνουν μέσω της αλληλεπίδρασης με το περιβάλλον τους. Ένας "πράκτορας" (το μοντέλο) εκτελεί ενέργειες σε ένα περιβάλλον και λαμβάνει "ανταμοιβές" (θετικές ή αρνητικές) ως ανατροφοδότηση για τις ενέργειές του. Ο στόχος του πράκτορα είναι να μάθει μια στρατηγική ή "πολιτική" που θα μεγιστοποιεί τη συνολική ανταμοιβή του μακροπρόθεσμα. Η ενισχυτική μάθηση βρίσκει εφαρμογές σε τομείς όπως τα αυτόνομα οχήματα, τα ρομποτικά συστήματα, τα παιχνίδια (π.χ. AlphaGo) και η βελτιστοποίηση βιομηχανικών διεργασιών.

Συνοψίζοντας, η Τεχνητή Νοημοσύνη (AI) και η Μηχανική Μάθηση (ML) συνιστούν κρίσιμες τεχνολογίες για την ανάπτυξη ευφυών συστημάτων. Αυτά τα συστήματα φιλοδοξούν να γεφυρώσουν το χάσμα ανάμεσα στις υπολογιστικές δυνατότητες των μηχανών και τις γνωστικές και ηθικές διαστάσεις της ανθρώπινης νοημοσύνης. Μέσω της ικανότητας των μηχανών να μαθαίνουν, να προσαρμόζονται και να βελτιώνονται, η AI και η ML αναδιαμορφώνουν θεμελιωδώς πολλούς τομείς της ανθρώπινης δραστηριότητας, προσφέροντας πρωτοφανείς δυνατότητες για καινοτομία και επίλυση σύνθετων προβλημάτων.[18], [19]

2.2 Σχετικές Συστηματικές Ανασκοπήσεις και Χαρτογραφήσεις Λογοτεχνίας

Η παρούσα ενότητα παρέχει μια αναλυτική και κριτική επισκόπηση των ευρημάτων προηγούμενων Συστηματικών Ανασκοπήσεων Λογοτεχνίας (Systematic Literature Reviews - SLR) και Συστηματικών Χαρτογραφήσεων (Systematic Mapping Studies - SMS), οι οποίες εστιάζουν στην οργάνωση containerized εφαρμογών. Το ερευνητικό αυτό πεδίο αναδεικνύεται ως ιδιαίτερα κρίσιμο στο πλαίσιο της σύγχρονης πληροφορικής, καθώς η ορθολογική διαχείριση και ανάπτυξη τέτοιων εφαρμογών καλείται να ανταποκριθεί στις προκλήσεις ενός διαρκώς εξελισσόμενου και πολυεπίπεδου υπολογιστικού οικοσυστήματος.

Οι εν λόγω μελέτες εξετάζουν την ενορχήστρωση εφαρμογών που βασίζονται σε τεχνολογίες containers — όπως το Docker και το Kubernetes — σε περιβάλλοντα που καλύπτουν όλο το εύρος του **Cloud-to-Thing Continuum**, δηλαδή:

- **Περιβάλλοντα Cloud**, όπου οι υπολογιστικοί πόροι είναι άφθονοι, κλιμακούμενοι και προσφέρονται μέσω κέντρων δεδομένων και δημόσιων/ιδιωτικών cloud υποδομών.
- **Περιβάλλοντα Fog**, τα οποία λειτουργούν ως ενδιάμεσο στρώμα μεταξύ cloud και edge, προσφέροντας τοπική επεξεργασία δεδομένων και μειωμένη καθυστέρηση.
- **Περιβάλλοντα Edge**, που περιλαμβάνουν πόρους και συσκευές κοντά στις πηγές δεδομένων (όπως αισθητήρες, IoT συσκευές), με περιορισμένη υπολογιστική ισχύ αλλά αυξημένες απαιτήσεις σε χαμηλή καθυστέρηση και τοπική λήψη αποφάσεων.

Η ανάλυση επικεντρώνεται σε τρεις βασικούς άξονες: τη μεθοδολογική προσέγγιση, τα εργαλεία και τις πλατφόρμες που μελετήθηκαν, και τα κύρια συμπεράσματα που προκύπτουν από τη βιβλιογραφία.

2.2.1 Μεθοδολογική προσέγγιση των SLR/SMS

Οι ανασκοπήσεις που εξετάστηκαν, όπως αυτές των **Naydenov et al. (2023)** και **Ullah et al. (2023)**, εφάρμοσαν αυστηρά καθορισμένα και επαναλήψιμα ερευνητικά πρωτόκολλα, τα οποία βασίστηκαν σε καθιερωμένα πρότυπα όπως PRISMA, Kitchenham και Petersen. Η αναζήτηση πραγματοποιήθηκε σε έγκριτες επιστημονικές βάσεις (Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect) με στοχευμένες

συνθήκες αναζήτησης, ώστε να διασφαλιστεί η συνάφεια και η πληρότητα του δείγματος. Τα κριτήρια ένταξης και αποκλεισμού σχεδιάστηκαν με σκοπό να αποτρέψουν τα false positives και να επικεντρωθούν σε μελέτες με άμεση συνάφεια με την ορχήστρωση containerized εφαρμογών στο Cloud-to-Thing Continuum.[20], [21]

Η ανάλυση και ταξινόμηση των πρωτογενών μελετών βασίστηκε σε θεματικές κατηγορίες και ερευνητικά ερωτήματα, όπως:

- Ποια container orchestrators και runtimes χρησιμοποιούνται;
- Ποιες συνοδευτικές τεχνολογίες ενσωματώνονται στις πλατφόρμες ορχήστρωσης;
- Ποιες αρχιτεκτονικές και υποδείγματα cloud υιοθετούνται;
- Ποιες μεθοδολογίες αξιολόγησης και σύγκρισης εφαρμόζονται;

2.2.2 Εργαλεία και τεχνολογίες που εξετάστηκαν

Οι προηγούμενες SLR/SMS χαρτογράφησαν με συστηματικό τρόπο το τοπίο των εργαλείων και τεχνολογιών που αξιολογήθηκαν ή προτάθηκαν:

- **Container Orchestrators:** Το Kubernetes αποτέλεσε το κυρίαρχο εργαλείο (>80% των μελετών), συμπεριλαμβανομένων ελαφριών παραλλαγών του όπως K3s, KubeEdge και MicroK8s, για edge και fog περιβάλλοντα. Άλλα εργαλεία, όπως τα Docker Swarm, Apache Mesos, Nomad και OpenShift, χρησιμοποιήθηκαν σε μικρότερη έκταση, κυρίως για συγκριτικούς σκοπούς. Ορισμένες μελέτες πρότειναν custom orchestrators (π.χ. MiCADO-Edge, OpenIoTfog) ειδικά σχεδιασμένα για καταναεμημένα περιβάλλοντα.
- **Container Runtimes:** Ο Docker engine διατηρεί την κυριαρχία του ως runtime, αν και η εξάρτηση από αυτό τίθεται υπό αμφισβήτηση λόγω της κατάργησης του Dockershim. Εναλλακτικές όπως containerd, CRI-O, runC, Singularity, Podman και LXC/LXD εξετάστηκαν για λόγους απόδοσης, ασφάλειας και συμβατότητας με edge/fog περιβάλλοντα.
- **Edge/Fog εργαλεία:** Πλατφόρμες όπως KubeEdge, MicroK8s, K3s, StarlingX και OpenIoTfog αποτέλεσαν τη βάση για την επέκταση της ορχήστρωσης στο edge. Λύσεις όπως MiCADO-Edge και PrEstoCloud προτάθηκαν για cloud-to-edge orchestration με υποστήριξη advanced policy-based reconfiguration.
- **Συνοδευτικές τεχνολογίες:** Εργαλεία όπως Prometheus, Grafana, Istio, Kafka και Redis αξιοποιήθηκαν για παρακολούθηση, ανάλυση και service mesh λειτουργίες. Παράλληλα, πρότυπα όπως το TOSCA και τεχνολογίες όπως Ansible και Terraform χρησιμοποιήθηκαν για specification και υποστήριξη Infrastructure as Code.[20], [21], [22]

2.2.3 Κύρια συμπεράσματα και ερευνητικά κενά

Η σύγκριση και σύνθεση των ευρημάτων ανέδειξε σημαντικά κοινά σημεία και προκλήσεις:

- Η κυριαρχία του Kubernetes ως de facto πλατφόρμα ορχήστρωσης, συνοδεύεται από περιορισμένη αξιοποίηση εναλλακτικών container runtimes με εξειδικευμένες δυνατότητες.
- Υπάρχει έλλειψη υποστήριξης για δυναμικό resource discovery και context-aware placement, γεγονός που περιορίζει τη δυνατότητα των πλατφορμών να προσαρμόζονται σε πραγματικό χρόνο στις δυναμικές απαιτήσεις των edge/fog περιβαλλόντων.
- Οι λύσεις προληπτικής επαναδιάταξης (proactive reconfiguration) με χρήση AI/ML παραμένουν σε εμβρυακό στάδιο, παρά τη σαφή ανάγκη για έξυπνη αυτονομία σε κατανεμημένα περιβάλλοντα.
- Η υποστήριξη τυποποιημένων μοντέλων (π.χ. TOSCA) για edge-related λειτουργίες είναι περιορισμένη, με αποτέλεσμα την ανάγκη για ad-hoc προσαρμογές που υπονομεύουν τη διαλειτουργικότητα.
- Οι αρχιτεκτονικές που επικρατούν είναι κυρίως κεντριοποιημένες ή υβριδικές, γεγονός που δημιουργεί single points of failure και περιορίζει τη δυνατότητα αποκεντρωμένης διαχείρισης.
- Στο πεδίο της ασφάλειας, οι περισσότερες λύσεις παρέχουν βασικό access control, χωρίς ενσωμάτωση εξελιγμένων τεχνικών authentication και trust management για multi-domain και heterogeneous περιβάλλοντα.

Συνολικά, οι προηγούμενες ανασκοπήσεις επισημαίνουν την ανάγκη για περαιτέρω έρευνα και ανάπτυξη ολοκληρωμένων λύσεων που θα επιτρέψουν την αποτελεσματική και ασφαλή ορχήστρωση containerized εφαρμογών σε ολόκληρο το Cloud-to-Thing Continuum. [20], [21], [22]

3. Μεθοδολογία Συστηματικής Ανασκόπησης

Λογοτεχνίας (SLR)

3.1 Εισαγωγή

Η Συστηματική Ανασκόπηση Λογοτεχνίας (Systematic Literature Review – SLR) αποτελεί μια θεμελιώδη και αυστηρά καθορισμένη επιστημονική μεθοδολογία, η οποία χρησιμοποιείται για τη συστηματική συλλογή, ενδελεχή αξιολόγηση και εμπεριστατωμένη ανάλυση της συνολικής υφιστάμενης βιβλιογραφίας που σχετίζεται άμεσα με ένα συγκεκριμένο ερευνητικό ερώτημα ή ένα σαφώς ορισμένο θεματικό πεδίο. Διακρίνεται σαφώς από τις παραδοσιακές, αφηγηματικές ανασκοπήσεις, οι οποίες συχνά χαρακτηρίζονται από μια πιο ελεύθερη και λιγότερο δομημένη προσέγγιση. Αντίθετα, η SLR υιοθετεί και εφαρμόζει αυστηρά προκαθορισμένα και σαφή κριτήρια εισαγωγής και αποκλεισμού των μελετών, εξασφαλίζοντας ότι μόνο οι πλέον συναφείς και ποιοτικές πηγές θα συμπεριληφθούν στην ανάλυση. Επιπλέον, ακολουθεί μια πλήρως τεκμηριωμένη και διαφανή διαδικασία αναζήτησης και φιλτραρίσματος των ερευνητικών πηγών, η οποία στοχεύει στην ελαχιστοποίηση της υποκειμενικότητας του ερευνητή και στην αποφυγή συστηματικών ή μη συστηματικών προκαταλήψεων (biases) κατά τη διαδικασία επιλογής και αξιολόγησης των μελετών. Η διαφάνεια και η επαναληψιμότητα αποτελούν κεντρικά στοιχεία της SLR, επιτρέποντας σε άλλους ερευνητές να κατανοήσουν πλήρως τη μεθοδολογία που ακολουθήθηκε και ενδεχομένως να επαναλάβουν τη διαδικασία για να επαληθεύσουν τα ευρήματα.

Η επιλογή της μεθοδολογίας της Συστηματικής Ανασκόπησης Λογοτεχνίας στην παρούσα εργασία δεν είναι τυχαία, αλλά υπαγορεύεται από μια σειρά συγκεκριμένων αναγκών και στόχων που επιδιώκει η έρευνα. Συγκεκριμένα, η SLR κρίνεται απαραίτητη για τους ακόλουθους λόγους:

- Συστηματική χαρτογράφηση των τρεχουσών τάσεων και σύγχρονων τεχνικών στην ενорχήστρωση containerized εφαρμογών σε cloud-to-edge περιβάλλοντα. Η εργασία στοχεύει στην αναγνώριση και κατηγοριοποίηση των τάσεων, τεχνικών και αρχιτεκτονικών μέσω ενδελεχούς ανάλυσης της βιβλιογραφίας, παρέχοντας ολοκληρωμένη εικόνα του τρέχοντος τοπίου και αναδεικνύοντας βέλτιστες πρακτικές.
- Απαιτείται αντικειμενική και τεκμηριωμένη συγκριτική αποτίμηση των διαθέσιμων πλατφορμών ενорχήστρωσης containerized εφαρμογών, βασιζόμενη σε αξιόπιστες επιστημονικές μελέτες. Η SLR παρέχει αυστηρή, συστηματική ανάλυση, όχι υποκειμενική, αλλά βασισμένη σε ευρύ φάσμα τεκμηριωμένων ερευνών. Αυτή η αποτίμηση θα εντοπίσει τις καταλληλότερες λύσεις για διάφορα σενάρια και θα προσφέρει πολύτιμες πληροφορίες.

- Στοχευμένη ανάλυση για την ανίχνευση τεχνολογικών κενών και ερευνητικών ευκαιριών στην ενσωμάτωση τεχνητής νοημοσύνης (AI/ML) σε διαδικασίες και πλατφόρμες ενορχήστρωσης containerized εφαρμογών. Η συστηματική ανασκόπηση της βιβλιογραφίας αποσκοπεί στην ανάδειξη προκλήσεων, ερωτημάτων και περιοχών όπου η AI/ML μπορεί να προσφέρει σημαντικά οφέλη, όπως αυτοματοποίηση διαχείρισης πόρων, πρόβλεψη προβλημάτων απόδοσης, βελτιστοποίηση τοποθέτησης containers και ενίσχυση ασφάλειας. Στόχος είναι ο εντοπισμός τεχνολογικών κενών και η πρόταση κατευθύνσεων για μελλοντική έρευνα.

Η μεθοδολογική προσέγγιση που θα ακολουθηθεί στην παρούσα εργασία θα βασιστεί στις αυστηρές και ευρέως αναγνωρισμένες οδηγίες που έχουν προταθεί από τους Kitchenham et al. για τη διεξαγωγή Συστηματικών Ανασκοπήσεων Λογοτεχνίας στον τομέα της Πληροφορικής. Επιπλέον, θα ληφθούν υπόψη οι κατευθυντήριες αρχές της πρωτοβουλίας PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses), οι οποίες παρέχουν ένα πλαίσιο για τη διαφανή και πλήρη αναφορά των αποτελεσμάτων των συστηματικών ανασκοπήσεων. Η συνδυαστική εφαρμογή αυτών των οδηγιών και αρχών θα διασφαλίσει την επιστημονική *rigor*, την αξιοπιστία και την εγκυρότητα των ευρημάτων της παρούσας εργασίας, προσαρμοσμένη πάντα στις ιδιαιτερότητες και τις απαιτήσεις του συγκεκριμένου ερευνητικού πεδίου της Πληροφορικής και των Συστημάτων Υπολογιστικού Νέφους.[23]

3.2 Ερευνητικά Ερωτήματα

Η παρούσα μελέτη εμβαθύνει στην κρίσιμη περιοχή της ενορχήστρωσης containerized εφαρμογών στο εξελισσόμενο και πολύπλοκο συνεχές Cloud-Thing. Στον πυρήνα της βρίσκονται τρία θεμελιώδη ερευνητικά ερωτήματα (Research Questions – RQs), σχολαστικά σχεδιασμένα για να διασφαλίσουν μια ενδελεχή διερεύνηση και μια σφαιρική κατανόηση των πολλαπλών διαστάσεων αυτού του δυναμικού πεδίου. Κάθε ερώτημα επικεντρώνεται σε μια ξεχωριστή, αλλά άρρηκτα συνδεδεμένη πτυχή του προβλήματος, διευκολύνοντας έτσι μια συστηματική και ολοκληρωμένη προσέγγιση για την αντιμετώπιση των προκλήσεων και των ευκαιριών που αναδύονται.

Το **πρώτο ερευνητικό ερώτημα (RQ1)** αναζητά να αναλύσει το ευρύ φάσμα των τεχνικών, των μεθοδολογιών και των πλατφορμών που είτε χρησιμοποιούνται ήδη στην πράξη, είτε προτείνονται από την ερευνητική κοινότητα ως βέλτιστες πρακτικές για την αποτελεσματική ενορχήστρωση containerized εφαρμογών στο σύνθετο και διαρκώς μεταβαλλόμενο περιβάλλον του cloud-to-thing continuum. Αυτό το ερώτημα εμβαθύνει στην καταγραφή και την ενδελεχή αξιολόγηση των υφιστάμενων λύσεων, καθώς και των αναδυόμενων τάσεων στον συγκεκριμένο τομέα. Ιδιαίτερη έμφαση δίνεται στην ικανότητά τους να διαχειρίζονται την εγγενή πολυπλοκότητα, την ετερογένεια των υποκείμενων υποδομών και τις αυστηρές απαιτήσεις απόδοσης που χαρακτηρίζουν αυτό το υβριδικό περιβάλλον, το οποίο γεφυρώνει το κεντρικό νέφος με τις διάσπαρτες συσκευές του Internet of Things (IoT). Η ανάλυση αυτή στοχεύει στην ανάδειξη των δυνατών και αδύνατων σημείων κάθε προσέγγισης, καθώς και στην κατανόηση του βαθμού στον οποίο ανταποκρίνονται στις μοναδικές προκλήσεις της ενορχήστρωσης στο συνεχές Cloud-Thing.

Το **δεύτερο ερευνητικό ερώτημα (RQ2)** εστιάζει στην αξιολόγηση της απόδοσης, διερευνώντας ποια συγκεκριμένα κριτήρια απόδοσης (performance metrics) εφαρμόζονται στην καθημερινή πρακτική ή προτείνονται σε θεωρητικό επίπεδο για την ποσοτική και

ποιοτική αποτίμηση της αποτελεσματικότητας και της αποδοτικότητας των διαφόρων τεχνικών ενορχήστρωσης containerized εφαρμογών στο cloud-to-thing continuum. Το ερώτημα αυτό επιδιώκει την συστηματική αναγνώριση και την προσεκτική κατηγοριοποίηση των σημαντικότερων μετρικών που χρησιμοποιούνται για την αντικειμενική μέτρηση κρίσιμων παραμέτρων των συστημάτων ενορχήστρωσης. Μεταξύ αυτών περιλαμβάνονται η καθυστέρηση κατά την απόκριση (latency), η αξιοπιστία και η ανθεκτικότητα στις αστοχίες, η αποδοτική κατανάλωση πόρων (υπολογιστικών, δικτυακών, αποθηκευτικών), η ικανότητα για οριζόντια και κάθετη επεκτασιμότητα (scalability), καθώς και άλλες ζωτικές πτυχές που επηρεάζουν τη συνολική λειτουργία και την εμπειρία του χρήστη. Η σαφής οριοθέτηση και η εμπεριστατωμένη ανάλυση αυτών των μετρικών είναι απαραίτητη για την αντικειμενική σύγκριση και την τεκμηριωμένη επιλογή των καταλληλότερων τεχνικών ενορχήστρωσης για συγκεκριμένα σενάρια χρήσης και απαιτήσεις εφαρμογών.

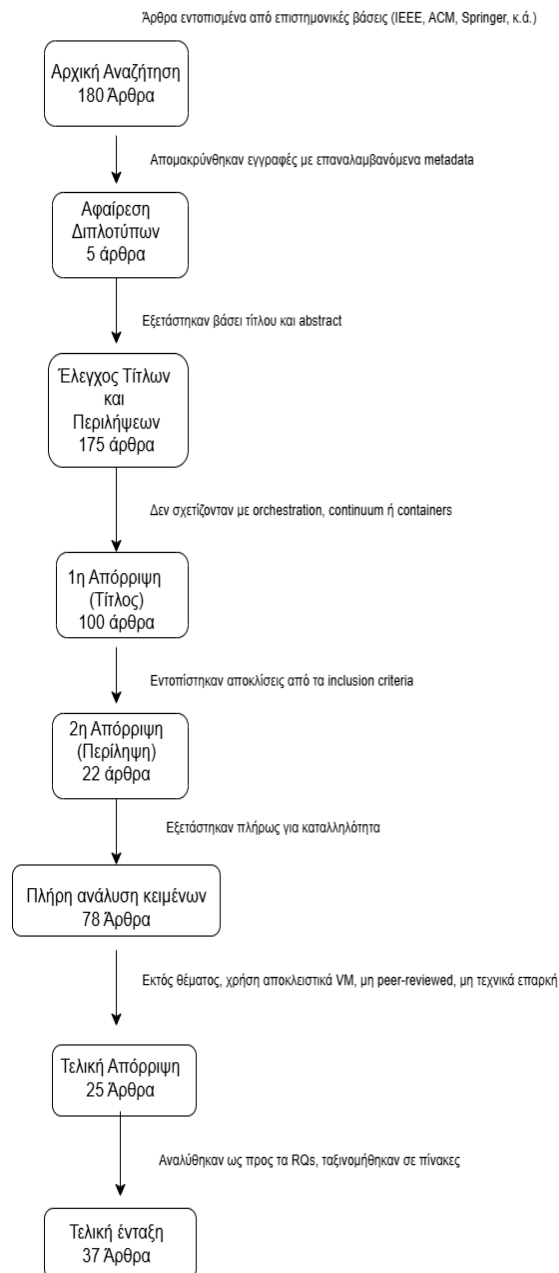
Τέλος, το **τρίτο ερευνητικό ερώτημα (RQ3)** στρέφει την προσοχή στις μελλοντικές προοπτικές και στις άλυτες προκλήσεις, εστιάζοντας στις σημαντικότερες ανοιχτές προκλήσεις που εξακολουθούν να υφίστανται στον τομέα και στις νέες ερευνητικές ευκαιρίες που αναδύονται, ιδίως στο πλαίσιο της αυξανόμενης ενσωμάτωσης τεχνολογιών τεχνητής νοημοσύνης (AI) και μηχανικής μάθησης (ML) στη διαδικασία ενορχήστρωσης containerized εφαρμογών στο cloud-to-thing continuum. Αυτό το ερώτημα επιχειρεί να εντοπίσει τα σημεία όπου η τρέχουσα ερευνητική δραστηριότητα υστερεί και να διερευνήσει τις δυνητικές δυνατότητες αξιοποίησης της AI και της ML για την αποτελεσματική αντιμετώπιση σύνθετων προβλημάτων ενορχήστρωσης. Τέτοια προβλήματα περιλαμβάνουν την αυτοματοποιημένη βελτιστοποίηση της τοποθέτησης και της διαχείρισης των containers, την πρόβλεψη και την πρόληψη πιθανών αστοχιών ή σημείων συμφόρησης, την δυναμική και προσαρμοστική διαχείριση των διαθέσιμων πόρων ανάλογα με τις τρέχουσες ανάγκες των εφαρμογών, καθώς και την ενίσχυση της συνολικής ευφυΐας και αυτονομίας των συστημάτων ενορχήστρωσης. Η διερεύνηση αυτών των ευκαιριών μπορεί να οδηγήσει στην ανάπτυξη νέων, πιο ευέλικτων, αποδοτικών και ανθεκτικών λύσεων ενορχήστρωσης για το απαιτητικό περιβάλλον του συνεχούς Cloud-Thing.

Τα τρία αυτά ερευνητικά ερωτήματα αποτελούν τον ακρογωνιαίο λίθο πάνω στον οποίο θεμελιώνεται η παρούσα ανασκόπηση της υπάρχουσας βιβλιογραφίας. Καθορίζουν με σαφήνεια το εύρος και τα όρια της έρευνας, λειτουργώντας ως ένα καλά καθορισμένο οργανωτικό πλαίσιο για τη συστηματική σύνθεση και την εμπεριστατωμένη ανάλυση των ευρημάτων που θα προκύψουν από την κριτική μελέτη των σχετικών επιστημονικών πηγών. Κατά τη διάρκεια της φάσης της ανάλυσης, τα ερωτήματα αυτά θα χρησιμοποιηθούν ως βασικός οδηγός για την ομαδοποίηση, τη συγκριτική αξιολόγηση και την ερμηνεία των διαφορετικών προσεγγίσεων, των τεχνικών λύσεων και των εμπειρικών αποτελεσμάτων που παρουσιάζονται στη βιβλιογραφία. Ο απώτερος στόχος αυτής της διαδικασίας είναι η εξαγωγή χρήσιμων και αξιόπιστων συμπερασμάτων σχετικά με την τρέχουσα κατάσταση της έρευνας στον τομέα, η ανάδειξη των κυριότερων προκλήσεων και των αναδυόμενων ευκαιριών, καθώς και η σκιαγράφηση πιθανών κατευθύνσεων για μελλοντική ερευνητική δραστηριότητα που θα συμβάλει στην περαιτέρω πρόοδο του πεδίου της ενορχήστρωσης containerized εφαρμογών στο συνεχές Cloud-Thing.

3.3 Στρατηγική Ανασκόπησης

Αρχικά, πραγματοποιήθηκε ένας λεπτομερής καθορισμός των κριτηρίων ένταξης και αποκλεισμού των δημοσιεύσεων. Τα κριτήρια αυτά σχεδιάστηκαν με σκοπό να οριοθετήσουν

με σαφήνεια το πεδίο της ανασκόπησης και να εξασφαλίσουν ότι θα συμπεριληφθούν μόνο οι πλέον σχετικές και ποιοτικές ερευνητικές εργασίες. Στη συνέχεια, έγινε η επιλογή των πλέον κατάλληλων επιστημονικών βάσεων δεδομένων, οι οποίες θα αποτελούσαν τις κύριες πηγές άντλησης βιβλιογραφικών αναφορών. Ένα κρίσιμο βήμα στην στρατηγική ανασκόπησης ήταν ο σχεδιασμός σύνθετων και στοχευμένων εκφράσεων αναζήτησης. Οι εκφράσεις αυτές βασίστηκαν στις τεχνολογικές λέξεις-κλειδιά που χαρακτηρίζουν το ερευνητικό πεδίο της ενορχήστρωσης containerized εφαρμογών σε διάφορα υπολογιστικά περιβάλλοντα. Η προσεκτική επιλογή και ο συνδυασμός των λέξεων-κλειδιών ήταν απαραίτητοι για τη μεγιστοποίηση της αποτελεσματικότητας των αναζητήσεων και την ανάκτηση του συνόλου των σχετικών δημοσιεύσεων.



Πίνακας 3.1 Διάγραμμα ροής επιλογής άρθρων σύμφωνα με τη μεθοδολογία PRISMA

Κριτήρια Ένταξης (Inclusion Criteria)

Τα κριτήρια ένταξης διαμορφώθηκαν με γνώμονα την εξασφάλιση της συνάφειας και της ποιότητας των επιλεγμένων δημοσιεύσεων. Αναλυτικότερα:

- **Δημοσιεύσεις στην αγγλική γλώσσα:** Επιλέχθηκαν δημοσιεύσεις στην αγγλική γλώσσα, καθώς αυτή αποτελεί την κυρίαρχη γλώσσα στην επιστημονική επικοινωνία στον τομέα της τεχνολογίας και διασφαλίζει την ευρύτερη δυνατή προσβασιμότητα και κατανόηση των ευρημάτων.
- **Άρθρα δημοσιευμένα σε επιστημονικά περιοδικά (journals), πρακτικά συνεδρίων (conference proceedings) ή ως κεφάλαια βιβλίων (book chapters):** Η επιλογή αυτών των τύπων δημοσιεύσεων έγινε λόγω της διαδικασίας αξιολόγησης (peer-review) που συνήθως διέπει τη δημοσίευσή τους, γεγονός που υποδηλώνει ένα υψηλότερο επίπεδο επιστημονικής αυστηρότητας και αξιοπιστίας. Τα επιστημονικά περιοδικά προσφέρουν εμπειριστατωμένες αναλύσεις και ερευνητικά αποτελέσματα, τα πρακτικά συνεδρίων παρουσιάζουν τις πιο πρόσφατες εξελίξεις και ιδέες, ενώ τα κεφάλαια βιβλίων παρέχουν συνήθως μια πιο συνολική και συνθετική θεώρηση των θεμάτων.
- **Μελέτες που επικεντρώνονται σε τεχνικές και εργαλεία ενορχήστρωσης containerized εφαρμογών (όπως Kubernetes, Docker Swarm) σε περιβάλλοντα cloud, edge ή fog:** Αυτό το κριτήριο αποτελεί τον πυρήνα της ανασκόπησης, καθώς εστιάζει αποκλειστικά σε εργασίες που πραγματεύονται την οργάνωση, διαχείριση και κλιμάκωση containerized εφαρμογών χρησιμοποιώντας συγκεκριμένα εργαλεία και σε σύγχρονα καταναμεμένα υπολογιστικά περιβάλλοντα όπως το cloud, το edge και το fog computing. Η αναφορά συγκεκριμένων τεχνολογιών όπως το Kubernetes και το Docker Swarm, καθώς και των περιβαλλόντων cloud, edge και fog, εξασφαλίζει την εστίαση στις πλέον επίκαιρες και σχετικές εξελίξεις του πεδίου.
- **Δημοσιεύσεις που περιλαμβάνουν μεθόδους αξιολόγησης απόδοσης ή συγκριτικές μελέτες εργαλείων ενορχήστρωσης:** Η συμπερίληψη μελετών που εστιάζουν στην αξιολόγηση της απόδοσης διαφόρων τεχνικών και εργαλείων ενορχήστρωσης, καθώς και συγκριτικών αναλύσεων μεταξύ τους, είναι ζωτικής σημασίας για την εξαγωγή ουσιαστικών συμπερασμάτων σχετικά με τα πλεονεκτήματα και τα μειονεκτήματα των διαφόρων προσεγγίσεων.

Κριτήρια Αποκλεισμού (Exclusion Criteria)

Παράλληλα με τα κριτήρια ένταξης, καθορίστηκαν σαφή κριτήρια αποκλεισμού για τον περιορισμό του όγκου της ανασκοπούμενης βιβλιογραφίας σε εργασίες άμεσα σχετικές με το ερευνητικό ερώτημα. Συγκεκριμένα:

- **Δημοσιεύσεις σε γλώσσα διαφορετική από την αγγλική:** Δημοσιεύσεις σε άλλες γλώσσες αποκλείστηκαν προκειμένου να διατηρηθεί η συνέπεια και η προσβασιμότητα της ανασκόπησης, λαμβάνοντας υπόψη την κυριαρχία της αγγλικής στην επιστημονική βιβλιογραφία του συγκεκριμένου τομέα.
- **Άρθρα που αποτελούν editorials, posters ή abstracts χωρίς διαθέσιμο πλήρες κείμενο:** Αυτοί οι τύποι δημοσιεύσεων αποκλείστηκαν καθώς συνήθως δεν περιέχουν επαρκείς λεπτομέρειες σχετικά με τη μεθοδολογία, τα αποτελέσματα και την

ανάλυση, καθιστώντας δύσκολη την ουσιαστική αξιολόγησή τους. Η διαθεσιμότητα του πλήρους κειμένου είναι απαραίτητη για την εμπεριστατωμένη ανάλυση και σύνθεση των πληροφοριών.

- **Εργασίες που δεν εστιάζουν στην ενορχήστρωση containerized εφαρμογών:** Εργασίες που πραγματεύονται γενικότερα θέματα cloud, edge ή fog computing, ή άλλες μορφές ενορχήστρωσης που δεν αφορούν containerized εφαρμογές, αποκλείστηκαν προκειμένου να διατηρηθεί η εστίαση της ανασκόπησης στο συγκεκριμένο ερευνητικό πεδίο.
- **Δημοσιεύσεις προγενέστερες του 2015:** Επιλέχθηκε ένα χρονικό όριο, αποκλείοντας δημοσιεύσεις πριν το 2015, με στόχο την επικέντρωση στις πιο σύγχρονες εξελίξεις και τάσεις στον δυναμικά εξελισσόμενο τομέα της ενορχήστρωσης containerized εφαρμογών. Η επιλογή αυτή βασίστηκε στην εκτίμηση ότι οι τεχνολογίες και οι προσεγγίσεις στον τομέα αυτό έχουν υποστεί σημαντικές αλλαγές και εξελίξεις τα τελευταία χρόνια.

Πηγές Αναζήτησης και Κριτήρια Επιλογής

Για την ολοκληρωμένη και εις βάθος διερεύνηση του ερευνητικού πεδίου, πραγματοποιήθηκε συστηματική αναζήτηση σε αναγνωρισμένες και έγκριτες επιστημονικές βάσεις δεδομένων. Η επιλογή των βάσεων δεδομένων έγινε με γνώμονα την ευρεία κάλυψη της σύγχρονης τεχνολογικής έρευνας, ιδίως στους κρίσιμους τομείς του cloud computing, του fog computing και του edge computing, οι οποίοι αποτελούν το επίκεντρο της παρούσας μελέτης. Επιπρόσθετα, καθοριστικό ρόλο στην επιλογή έπαιξε η αξιοπιστία των πηγών ως δεκτών φορέων δημοσίευσης επιστημονικών εργασιών που έχουν υποβληθεί σε αυστηρή διαδικασία αξιολόγησης από ομοτίμους (peer-review). Συγκεκριμένα, αξιοποιήθηκαν οι ακόλουθες βάσεις δεδομένων:

- **IEEE Xplore:** Θεωρείται μία από τις κορυφαίες ψηφιακές βιβλιοθήκες για επιστημονική και τεχνική βιβλιογραφία στους τομείς της ηλεκτρολογίας, της ηλεκτρονικής, της πληροφορικής και των συναφών κλάδων. Η πλατφόρμα παρέχει πρόσβαση σε ένα ευρύ φάσμα δημοσιεύσεων, συμπεριλαμβανομένων άρθρων από περιοδικά, πρακτικά συνεδρίων, τεχνικές αναφορές και πρότυπα.
- **ACM Digital Library:** Αποτελεί την εκτενή ψηφιακή βιβλιοθήκη του Association for Computing Machinery (ACM), του μεγαλύτερου παγκόσμιου επιστημονικού και εκπαιδευτικού οργανισμού για την πληροφορική. Η βιβλιοθήκη περιλαμβάνει ένα τεράστιο όγκο άρθρων από περιοδικά, πρακτικά συνεδρίων, ενημερωτικά δελτία και εκπαιδευτικό υλικό, καλύπτοντας όλους τους τομείς της πληροφορικής.
- **ScienceDirect (Elsevier):** Πρόκειται για μια ευρεία πλατφόρμα που παρέχει πρόσβαση σε έναν μεγάλο αριθμό επιστημονικών περιοδικών και ηλεκτρονικών βιβλίων που εκδίδονται από την Elsevier, έναν από τους μεγαλύτερους εκδοτικούς οίκους παγκοσμίως. Η ScienceDirect καλύπτει ένα ευρύ φάσμα επιστημονικών κλάδων, συμπεριλαμβανομένων των επιστημών της πληροφορικής και της μηχανικής.
- **SpringerLink:** Αποτελεί την ηλεκτρονική πύλη της Springer Nature, ενός μεγάλου εκδοτικού οίκου που δραστηριοποιείται στην έκδοση επιστημονικών βιβλίων,

περιοδικών και πρακτικών συνεδρίων. Η SpringerLink προσφέρει πρόσβαση σε εκατομμύρια επιστημονικών δημοσιεύσεων σε πολλούς επιστημονικούς τομείς.

- **Wiley Online Library:** Είναι μια πλατφόρμα που παρέχει πρόσβαση σε ένα ευρύ φάσμα επιστημονικών περιοδικών, ηλεκτρονικών βιβλίων και άλλων ακαδημαϊκών πόρων που εκδίδονται από τον εκδοτικό οίκο Wiley. Η βιβλιοθήκη καλύπτει ποικίλους επιστημονικούς, τεχνικούς, ιατρικούς και ανθρωπιστικούς τομείς.

Η επιλογή αυτών των συγκεκριμένων βάσεων δεδομένων διασφαλίζει την πρόσβαση σε μια πληθώρα ερευνητικών εργασιών υψηλής ποιότητας που σχετίζονται άμεσα με τους τομείς ενδιαφέροντος της παρούσας μελέτης.

Λέξεις-Κλειδιά, Εκφράσεις Αναζήτησης και Στρατηγική Αναζήτησης

Για την αποτελεσματική αναζήτηση σχετικής βιβλιογραφίας, αναπτύχθηκε μια στοχευμένη στρατηγική αναζήτησης που βασίστηκε στον συνδυασμό συγκεκριμένων λέξεων-κλειδιών και φράσεων αναζήτησης. Οι όροι αναζήτησης επιλέχθηκαν με γνώμονα την κάλυψη τόσο των τεχνολογικών πτυχών της ενορχήστρωσης container όσο και των παραμέτρων απόδοσης και των προκλήσεων που σχετίζονται με τα συστήματα cloud, fog και edge computing.

Η σύνθεση των όρων αναζήτησης πραγματοποιήθηκε με τη χρήση των λογικών τελεστών Boolean (AND, OR), επιτρέποντας τον ακριβέστερο καθορισμό των κριτηρίων αναζήτησης και τη βελτιστοποίηση των αποτελεσμάτων. Οι βασικοί συνδυασμοί όρων αναζήτησης που χρησιμοποιήθηκαν περιλάμβαναν:

- Συνδυασμοί όρων που σχετίζονται με την ενορχήστρωση container: ("container orchestration" OR "containerized applications orchestration" OR "Docker orchestration" OR Kubernetes OR "Docker Swarm"). Αυτή η ομάδα όρων στοχεύει στην ανάκτηση εργασιών που ασχολούνται με τις τεχνικές και τα εργαλεία για την αυτοματοποιημένη διαχείριση και ανάπτυξη εφαρμογών σε περιβάλλοντα container.
- Συνδυασμοί όρων που σχετίζονται με τα υπολογιστικά περιβάλλοντα: ("cloud computing" OR "edge computing" OR "fog computing" OR "cloud-to-thing continuum"). Αυτή η ομάδα όρων επικεντρώνεται στα διαφορετικά παραδείγματα καταναμημένων υπολογιστικών συστημάτων που αποτελούν το πλαίσιο της παρούσας έρευνας, συμπεριλαμβανομένων των κεντρικών υπολογιστικών νέφους, των καταναμημένων υπολογιστικών στην άκρη του δικτύου και των ενδιάμεσων αρχιτεκτονικών fog computing. Ο όρος "cloud-to-thing continuum" συμπεριλήφθηκε για να καλύψει εργασίες που εξετάζουν την ολιστική διαχείριση υπολογιστικών πόρων σε όλο το φάσμα από το νέφος έως τις τερματικές συσκευές.
- Συνδυασμοί όρων που σχετίζονται με την αξιολόγηση και τις προκλήσεις: ("performance evaluation" OR "resource management" OR scalability OR challenges OR benchmarking OR comparison"). Αυτή η ομάδα όρων στοχεύει στην ανάκτηση εργασιών που εστιάζουν στην αξιολόγηση της απόδοσης, στη διαχείριση των διαθέσιμων πόρων, στην επεκτασιμότητα των συστημάτων, στις προκλήσεις που ανακύπτουν, στην συγκριτική αξιολόγηση διαφορετικών προσεγγίσεων και στην δημιουργία σημείων αναφοράς (benchmarks).

Η συνδυαστική χρήση αυτών των ομάδων όρων, με τη διαμεσολάβηση του τελεστή AND, επέτρεψε την αναζήτηση άρθρων και ερευνητικών εργασιών που εστιάζουν ταυτόχρονα στην

τεχνική διάσταση της ενορχήστρωσης container και στην μεθοδολογική αξιολόγηση των διαφόρων εργαλείων, τεχνικών και αρχιτεκτονικών στα πλαίσια των κατανεμημένων υπολογιστικών περιβαλλόντων. Αυτή η προσέγγιση διασφάλισε την ανάκτηση βιβλιογραφίας άμεσα σχετικής με τους ερευνητικούς στόχους της παρούσας μελέτης.

3.4 Λεπτομερής Διαδικασία Φιλτραρίσματος Μελετών

Η επιλογή των επιστημονικών εργασιών που τελικά συμπεριλήφθηκαν στην παρούσα ανασκόπηση βιβλιογραφίας ακολούθησε μια αυστηρή και συστηματική διφασική διαδικασία φιλτραρίσματος, σχεδιασμένη σύμφωνα με τις κατευθυντήριες αρχές της μεθοδολογίας PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses). Η υιοθέτηση αυτής της μεθοδολογίας διασφαλίζει τη διαφάνεια, την αναπαραγωγιμότητα και την αντικειμενικότητα της διαδικασίας επιλογής, ενισχύοντας την αξιοπιστία και την εγκυρότητα των συμπερασμάτων της ανασκόπησης.

Φάση 1: Πρώτο Στάδιο Φιλτραρίσματος - Αξιολόγηση Τίτλων και Περίληψεων

Κατά το πρώτο στάδιο της διαδικασίας, πραγματοποιήθηκε μια εκτενής αξιολόγηση των τίτλων και των περιλήψεων όλων των ερευνητικών εργασιών που εντοπίστηκαν μέσω των προκαθορισμένων πηγών και στρατηγικών αναζήτησης. Ο κύριος στόχος αυτού του αρχικού φιλτραρίσματος ήταν ο άμεσος αποκλεισμός εκείνων των άρθρων που, βάσει των πληροφοριών που παρείχαν ο τίτλος και η περίληψη, δεν σχετίζονταν άμεσα με το ευρύτερο θεματικό πεδίο της ενορχήστρωσης containerized εφαρμογών. Επιπλέον, απορρίφθηκαν άρθρα που δεν φαίνονταν να συμβάλλουν στην απάντηση των σαφώς καθορισμένων ερευνητικών ερωτημάτων (Research Questions - RQs) που τέθηκαν στην αρχή της μελέτης. Η διαδικασία αυτή πραγματοποιήθηκε από τουλάχιστον δύο ανεξάρτητους ερευνητές, οι οποίοι αξιολόγησαν κάθε τίτλο και περίληψη με βάση προκαθορισμένα κριτήρια αποκλεισμού.

Φάση 2: Δεύτερο Στάδιο Φιλτραρίσματος - Εξέταση του Πλήρους Κειμένου για Επιλεξιμότητα

Οι εργασίες που πέρασαν επιτυχώς το πρώτο στάδιο του φιλτραρίσματος, δηλαδή εκείνες που φάνηκαν να σχετίζονται με το θέμα και τα ερευνητικά ερωτήματα βάσει τίτλου και περιλήψης, προχώρησαν στο δεύτερο στάδιο. Σε αυτή τη φάση, έγινε πλήρης ανάγνωση και λεπτομερής ανάλυση του πλήρους κειμένου κάθε άρθρου. Η τελική ένταξη μιας μελέτης στην ανασκόπηση εξαρτήθηκε από την ικανοποίηση συγκεκριμένων κριτηρίων ένταξης (inclusion criteria) που είχαν καθοριστεί εκ των προτέρων. Τα κριτήρια αυτά διασφαλίζουν ότι οι επιλεγμένες μελέτες παρείχαν ουσιαστική και τεκμηριωμένη συνεισφορά, σχετική με τουλάχιστον ένα από τα ερευνητικά ερωτήματα που αποτελούσαν τον πυρήνα της παρούσας μελέτης. Η αξιολόγηση της επιλογής του πλήρους κειμένου πραγματοποιήθηκε επίσης από τουλάχιστον δύο ανεξάρτητους ερευνητές, ακολουθώντας την ίδια διαδικασία διαχείρισης των διαφωνιών που περιγράφηκε για το πρώτο στάδιο.

Λεπτομερής Καταγραφή και Συστηματική Οργάνωση Δεδομένων

Μετά την ολοκλήρωση της διαδικασίας φιλτραρίσματος και τον καθορισμό της τελικής λίστας των μελετών που ανταποκρίνονταν στα κριτήρια ένταξης, ακολούθησε ένα κρίσιμο στάδιο καταγραφής και ταξινόμησης των δεδομένων. Κάθε μία από τις μελέτες που επιλέχθηκαν καταγράφηκε με μεγάλη προσοχή σε έναν ειδικά διαμορφωμένο πίνακα, ο οποίος περιείχε ένα σύνολο προκαθορισμένων μεταδεδομένων. Σκοπός της συλλογής αυτών

των μεταδεδομένων ήταν η συστηματική κατηγοριοποίηση και ανάλυση των χαρακτηριστικών κάθε μελέτης, ώστε να διευκολυνθεί η μετέπειτα σύνθεση και ερμηνεία των αποτελεσμάτων.

Τα μεταδεδομένα που καταγράφηκαν για κάθε μελέτη ήταν τα εξής:

- **Μοναδικός Αναγνωριστικός Κωδικός (DOI) και Ολοκληρωμένος Τίτλος της Δημοσίευσης:** Η καταγραφή του DOI διασφαλίζει την μοναδική αναγνώριση και την εύκολη αναζήτηση κάθε μελέτης, ενώ ο πλήρης τίτλος δίνει άμεση πληροφόρηση για το περιεχόμενό της.
- **Έτος Δημοσίευσης:** Η ημερομηνία δημοσίευσης είναι σημαντική για την κατανόηση της εξέλιξης της έρευνας στον συγκεκριμένο τομέα και για την αναγνώριση πιθανών χρονικών τάσεων.
- **Όνομα της Πλατφόρμας ή του Εργαλείου Ενορχήστρωσης που Εξετάζεται:** Καταγράφηκε το όνομα της βασικής πλατφόρμας ή του εργαλείου ενορχήστρωσης containerized εφαρμογών που ήταν το αντικείμενο της μελέτης (π.χ., Kubernetes, Nomad, OpenShift). Αυτή η πληροφορία είναι απαραίτητη για την ταξινόμηση των ευρημάτων με βάση την τεχνολογία.
- **Σύνδεση με τα Ερευνητικά Ερωτήματα (RQs):** Για κάθε μελέτη, σημειώθηκε σε ποιο ή ποια από τα προκαθορισμένα ερευνητικά ερωτήματα (1, 2 ή 3) παρείχε σχετικές πληροφορίες ή συμβολή. Αυτή η αντιστοίχιση επέτρεψε την ομαδοποίηση των αποτελεσμάτων ανάλογα με τους ερευνητικούς στόχους.
- **Ύπαρξη ή Απουσία Αξιολόγησης Απόδοσης (Performance Evaluation):** Καταγράφηκε αν η μελέτη περιλάμβανε ή όχι μια αξιολόγηση της απόδοσης των συστημάτων ή πλατφορμών που μελετήθηκαν. Η ύπαρξη αξιολογήσεων απόδοσης θεωρήθηκε κρίσιμο στοιχείο για την εξαγωγή πρακτικών συμπερασμάτων.
- **Πεδίο Εφαρμογής:** Καθορίστηκε το κύριο πεδίο εφαρμογής που κάλυπτε η μελέτη, όπως cloud computing, edge computing, fog computing ή υβριδικά περιβάλλοντα, με σκοπό να αναδειχθούν οι διαφορετικές οπτικές και προκλήσεις ανάλογα με το περιβάλλον ανάπτυξης.

Η συστηματική και δομημένη καταγραφή αυτών των δεδομένων αποτέλεσε θεμελιώδη λίθο για την επόμενη φάση της εργασίας, επιτρέποντας την στοχευμένη και συγκριτική σύνθεση των αποτελεσμάτων από τις επιλεγμένες μελέτες. Μέσω αυτής της διαδικασίας, κατέστη δυνατή η εξαγωγή ουσιαστικών συμπερασμάτων και η παροχή μιας ολοκληρωμένης εικόνας του τρέχοντος ερευνητικού τοπίου στον τομέα της ενορχήστρωσης containerized εφαρμογών.

3.5 Σύνθεση και Ανάλυση Δεδομένων

Η σύνθεση και ανάλυση των δεδομένων που προέκυψαν από τις επιλεγμένες μελέτες αποτελεί ένα κρίσιμο και θεμελιώδες στάδιο στο πλαίσιο μιας Συστηματικής Ανασκόπησης Λογοτεχνίας (Systematic Literature Review - SLR). Αυτό το στάδιο είναι καθοριστικής σημασίας για την εξαγωγή έγκυρων συμπερασμάτων και την απάντηση στα ερευνητικά ερωτήματα που τέθηκαν αρχικά. Η επιλογή της κατάλληλης μεθόδου σύνθεσης δεδομένων είναι άρρηκτα συνδεδεμένη με τη φύση των δεδομένων που συλλέγονται και τους στόχους της ανασκόπησης.

Για την παρούσα εργασία, επιλέχθηκε η **ποιοτική και θεματική σύνθεση** (qualitative and thematic synthesis). Η επιλογή αυτή αιτιολογείται από το γεγονός ότι η πλειονότητα των εντοπισμένων εργασιών, οι οποίες αποτέλεσαν το σώμα της ανασκόπησης, χαρακτηρίζεται κυρίως από περιγραφικές αναλύσεις. Ειδικότερα, οι μελέτες αυτές επικεντρώνονται στην παρουσίαση και εξήγηση τεχνικών λύσεων, την περιγραφή αρχιτεκτονικών συστημάτων και την ανάλυση ερευνητικών πλαισίων. Αντίθετα, παρατηρήθηκε έλλειψη ή περιορισμένος αριθμός μελετών που να παρέχουν ποσοτικά συγκρίσιμα αποτελέσματα, τα οποία θα επέτρεπαν τη χρήση στατιστικών ή μετα-αναλυτικών μεθόδων σύνθεσης, όπως η μετα-ανάλυση. Η ποιοτική και θεματική σύνθεση είναι ιδανική για τέτοιου είδους δεδομένα, καθώς επιτρέπει την αναγνώριση, ανάλυση και ερμηνεία κοινών θεμάτων, μοτίβων και εννοιών που αναδύονται από το ερευνητικό υλικό, ακόμη και αν οι αρχικές μελέτες χρησιμοποιούν διαφορετικές μεθοδολογίες ή προσεγγίσεις.

Η μεθοδολογική προσέγγιση που ακολουθήθηκε για τη σύνθεση των δεδομένων περιλαμβάνει τα εξής στάδια, τα οποία διασφαλίζουν τη συστηματικότητα, τη διαφάνεια και την αξιοπιστία της διαδικασίας:

Θεματική Ομαδοποίηση και Κωδικοποίηση Περιεχομένου

Κάθε επιλεγμένη εργασία καταγράφηκε αρχικά σε μια ειδικά διαμορφωμένη βάση δεδομένων με καθορισμένα μεταδεδομένα, όπως: τίτλος, συγγραφείς, DOI, έτος δημοσίευσης, πλατφόρμα ή εργαλείο ενορχήστρωσης που εξετάζεται, πεδίο εφαρμογής (cloud, edge, fog), καθώς και σύνδεση με τα τρία ερευνητικά ερωτήματα (RQ1–RQ3).

Ακολούθησε η **θεματική κωδικοποίηση** του περιεχομένου κάθε μελέτης, με σκοπό την επισήμανση και συστηματική καταγραφή βασικών εννοιών και τεχνικών στοιχείων. Η κωδικοποίηση έγινε **χειροκίνητα**, προκειμένου να διασφαλιστεί η ακρίβεια και η συνέπεια ως προς την ερμηνεία των εννοιών. Οι βασικοί άξονες κωδικοποίησης περιλάμβαναν:

- Τεχνικές, αρχιτεκτονικά σχήματα και πλατφόρμες που αφορούν την ενορχήστρωση containers.
- Χρησιμοποιούμενα κριτήρια και μετρικές αξιολόγησης απόδοσης.
- Εφαρμογές τεχνητής νοημοσύνης και μηχανικής μάθησης σε περιβάλλοντα ενορχήστρωσης.
- Αναφερόμενες τεχνικές προκλήσεις, περιορισμοί και τεκμηριωμένα ερευνητικά κενά.
- Προτάσεις και ευκαιρίες για μελλοντική έρευνα.

Η οργάνωση της βάσης δεδομένων επέτρεψε το **φιλτράρισμα και την κατηγοριοποίηση των μελετών ανά Ερευνητικό Ερώτημα (RQ)**, διευκολύνοντας τη στοχευμένη ανάλυση ανά θεματικό άξονα.

Οργάνωση της Σύνθεσης

Η τελική σύνθεση της βιβλιογραφίας οργανώθηκε θεματικά, βάσει των τριών Ερευνητικών Ερωτημάτων που τέθηκαν στην ενότητα 3.2. Για κάθε RQ δημιουργήθηκαν **θεματικοί πίνακες ανάλυσης**, οι οποίοι χρησιμοποίησαν τα μεταδεδομένα και την κωδικοποιημένη πληροφορία για να ομαδοποιήσουν τις σχετικές μελέτες σύμφωνα με:

- Την τεχνολογία/εργαλείο που εξετάζουν.

- Το περιβάλλον εφαρμογής (cloud/edge/fog/hybrid).
- Τις μετρικές αξιολόγησης που χρησιμοποιούν.
- Τη μεθοδολογία που ακολουθούν.
- Την παρουσία ή απουσία AI/ML εφαρμογών και τον τύπο αυτών (π.χ. reinforcement learning, supervised learning).
- Τη συμβολή τους σε προτάσεις για μελλοντική έρευνα.

Η παρουσίαση των **αποτελεσμάτων** της θεματικής σύνθεσης αυτής, καθώς και η **κριτική ερμηνεία των ευρημάτων**, θα παρουσιαστούν αναλυτικά στο επόμενο κεφάλαιο (Κεφάλαιο 4), το οποίο επικεντρώνεται στην απάντηση των ερευνητικών ερωτημάτων και στη συγκριτική αποτίμηση των προσεγγίσεων.

4. *Ανάλυση Αποτελεσμάτων*

4.1 RQ1

4.1.1 Εισαγωγή

Το πρώτο ερευνητικό ερώτημα (RQ1), αποτελεί το επίκεντρο αυτής της εργασίας, καθώς εξετάζει τις καινούριες αρχιτεκτονικές προσεγγίσεις σχετικά με την αποτελεσματική ενορχήστρωση containerized εφαρμογών σε όλο το Cloud-to-Thing Continuum. Παράλληλα εστιάζει στην πολυπλοκότητα της, με σκοπό να εξηγήσει τον τρόπο διαχείρισης και συγχρονισμού των εφαρμογών σε containers, από το κεντρικό Cloud έως τις απομακρυσμένες συσκευές και αισθητήρες (Thing).

Στην παρούσα ανάλυση, εξετάζονται τα χαρακτηριστικά, τα πλεονεκτήματα αλλά και οι προκλήσεις των περισσότερων προσεγγίσεων ενορχήστρωσης containerized εφαρμογών στο συνεχές Cloud-to-Thing. Με τη σύγκριση και την μελέτη των σημαντικότερων προσεγγίσεων που υπάρχουν την τρέχουσα περίοδο, αναδεικνύονται οι βέλτιστες πρακτικές αλλά και παρουσιάζονται οι μελλοντικές κατευθύνσεις που χρήζουν διερεύνηση. Η τεχνολογία του σήμερα, υποστηρίζει την υλοποίηση των μικροϋπηρεσιών σε cloud υποδομές (δημόσιες ή ιδιωτικές) για ευελιξία και επεκτασιμότητα μέσω της χρήσης ενδιάμεσων επιπέδων (fog) για την επίτευξη μειωμένης καθυστέρησης και τοπικής αυτονομίας. Τα τερματικά - edge συσκευές- αποτελούν τις πηγές αλλά και τους καταναλωτές των δεδομένων. Με τη βοήθεια της ενορχήστρωσης, επιτυγχάνεται υψηλή απόδοση και αποτελεσματικότητα με οριζόντια και κάθετη κλιμάκωση αλλά και η ευελιξία για προσαρμοστικότητα σε διάφορες υποδομές. Ωστόσο, η καθυστέρηση αποτελεί σημαντική πρόκληση για εφαρμογές που απαιτούν άμεση ανταπόκριση.

Στο σύνθετο και διαρκώς εξελισσόμενο χώρο των κατανεμημένων υπολογιστικών συστημάτων, η αποτελεσματική ενορχήστρωση των containers αναδεικνύεται ως ένας κρίσιμος παράγοντας για την πλήρη αξιοποίηση των δυνατοτήτων που προσφέρουν τα ασύμμετρα περιβάλλοντα Cloud-to-Thing. Η ικανότητα διαχείρισης και συντονισμού εφαρμογών που εκτελούνται σε ετερογενείς υπολογιστικούς πόρους, από τις συσκευές στην άκρη του δικτύου έως τις ισχυρές υποδομές του νέφους, αποτελεί πλέον απαίτηση για την ανάπτυξη σύγχρονων ψηφιακών λύσεων. Η πρόκληση έγκειται στην επιλογή και την εφαρμογή κατάλληλων αρχιτεκτονικών σχεδίων και εργαλείων για την αντιμετώπιση της πολυπλοκότητας των υβριδικών περιβαλλόντων. Η διαφορετικότητα των υπολογιστικών πόρων και των συνδέσεων του κάθε δικτύου και εφαρμογής χρειάζεται βαθιά κατανόηση της τεχνολογίας αλλά και των ιδιοτήτων κάθε διαφορετικού επιπέδου και εργαλείου. Επιπλέον, σημαντικό τέλμα για την εύκολη μεταφορά μεταξύ των επιπέδων, αποτελεί το μειωμένο απόθεμα πόρων των IoT συσκευών, η χαμηλή κατανάλωση ενέργειας αλλά και οι διακοπτόμενες συνδέσεις με το δίκτυο.

Σε αντίθεση με τις IoT συσκευές, το νέφος προσφέρει εύκολη επεκτασιμότητα και ισχυρές υπολογιστικές δυνατότητες, οι οποίες ωστόσο, έχουν αντίβαρο τον κίνδυνο της καθυστέρησης αλλά και του υψηλού κόστους για την μεταφορά δεδομένων. Όπως είναι αντιληπτό, η κατάλληλη επιλογή της αρχιτεκτονικής ενορχήστρωσης, πρέπει να λαμβάνει υπόψη παράλληλα και τους περιορισμούς και τις δυνατότητες. Ο κύριος στόχος, φυσικά είναι η ομαλή ανάπτυξη, διαχείριση και παρακολούθηση εφαρμογών σε όλο το συνεχές. Γι' αυτό και συνεχώς αυξάνεται η ανάγκη για ευέλικτα εργαλεία προσαρμόσιμα στις απαιτήσεις κάθε επιπέδου, τα οποία θα μπορούν να υποστηρίξουν τα περισσότερα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται σήμερα αλλά και τα μοντέλα προγραμματισμού στα οποία βασίζεται η σημερινή τεχνολογία. Όπως κάθε εργαλείο, βέβαια, έτσι και σε αυτά, είναι απαραίτητοι οι μηχανισμοί ασφαλείας τόσο για τα λάθη απροσεξίας όσο και τα θεμιτά προβλήματα που μπορεί να προκληθούν. [24]

Η ανάλυση των βέλτιστων πρακτικών που έχουν ήδη αναπτυχθεί και δοκιμαστεί σε πραγματικά σενάρια χρήσης μπορεί να προσφέρει πολύτιμες γνώσεις και να καθοδηγήσει μελλοντικές προσπάθειες. Συγκεκριμένα, η απάντηση στο RQ1 αποκτά ιδιαίτερη σημασία καθώς δεν περιορίζεται στην απλή καταγραφή της τρέχουσας τεχνολογικής κατάστασης, αλλά στοχεύει στην ανάδειξη των κρίσιμων σημείων που καθορίζουν την αποτελεσματική ενορχήστρωση των containers σε περιβάλλοντα Cloud-to-Thing. Η αναγνώριση και η λεπτομερής εξέταση των ανοιχτών προκλήσεων που απειλούν την αποτελεσματική υιοθέτηση αυτού του μοντέλου είναι απαραίτητη για την προώθηση της καινοτομίας. Αυτές οι προκλήσεις μπορεί να αφορούν την ασφάλεια των δεδομένων και των επικοινωνιών σε κατανεμημένα περιβάλλοντα, τη διαχείριση της πολυπλοκότητας των υβριδικών αρχιτεκτονικών, την εξασφάλιση της αξιοπιστίας και της διαθεσιμότητας των εφαρμογών, καθώς και την αντιμετώπιση των ζητημάτων διαλειτουργικότητας μεταξύ διαφορετικών τεχνολογιών και πλατφορμών.

Η ανάλυση των μελλοντικών κατευθύνσεων για έρευνα και περαιτέρω ανάπτυξη είναι υψίστης σημασίας τόσο για την αντιμετώπιση αυτών των προκλήσεων, αλλά και την περαιτέρω εξέλιξη του Cloud-to-Thing Continuum. Η έρευνα νέων μοντέλων ενορχήστρωσης, η ανάπτυξη πιο ευφυών και αυτοματοποιημένων εργαλείων διαχείρισης, αλλά και η ενσωμάτωση των πλέον απαραίτητων τεχνολογιών όπως η τεχνητή νοημοσύνη και η μηχανική μάθηση για την πρόβλεψη και την αντιμετώπιση προβλημάτων, αποτελούν μερικές από τις πιθανές κατευθύνσεις. Η κατανόηση αυτών των πλευρών είναι απαραίτητη για την προώθηση της καινοτομίας, την αξιοποίηση των δυνατοτήτων των σύγχρονων ψηφιακών εφαρμογών και την δημιουργία νέων ευκαιριών στον τομέα της τεχνολογίας. Η επιτυχής αντιμετώπιση των προκλήσεων της ενορχήστρωσης containers στο συνεχές Cloud-to-Thing θα επιτρέψει την ανάπτυξη πιο ευφυών, αυτόνομων και διασυνδεδεμένων συστημάτων που θα έχουν σημαντικό αντίκτυπο στην κοινωνία και την οικονομία.

Πλαίσιο / Προσέγγιση	Τύπος Υποδομής	Υποστήριξη Edge	Latency- awareness	Autoscaling
-------------------------	----------------	--------------------	-----------------------	-------------

Kubernetes	Cloud, Hybrid	Περιορισμένη	Όχι εγγενώς	HPA, VPA
Docker Swarm	Cloud, Fog, Edge	Ναι	Περιορισμένη	Manual
k3s / KubeEdge	Edge/Fog	Ναι	Ναι	Ναι (με extensions)
GAIKube (AI-based)	Edge/Heterogeneous	Ναι	Ναι	Ναι (AI-driven)
Load-Aware K8s	Cloud/Edge	Ναι	Ναι	Ναι
Blockchain-Orchestrated Edge	Edge	Ναι	Όχι	Όχι
CRDT-based K8s (Rearchitecting)	Edge-Decentralized	Ναι (κατά κόμβο)	Ναι	Ναι (με lazy sync)
Cost-aware Autoscaling	Cloud	Όχι εγγενώς	Όχι	Ναι

Πίνακας 4.1.1 Πίνακας Συγκρίσεων μοντέλων

Συνολικά, η δυνατότητα λειτουργίας σε περιβάλλοντα Edge είναι πολύ σημαντική για σύγχρονες εφαρμογές, ειδικά στο Βιομηχανικό Διαδίκτυο των Πραγμάτων (IIoT). Εκεί, οι υπολογιστικοί πόροι είναι συχνά περιορισμένοι και διασκορπισμένοι. Για αυτό τον λόγο, έχουν αναπτυχθεί διάφορες παραλλαγές του Kubernetes που είναι πιο κατάλληλες για τέτοιες συνθήκες:

- k3s: μια ελαφριά έκδοση του Kubernetes, κατάλληλη για μικρές και απλές συσκευές.[25]
- GAIKube: χρησιμοποιεί τεχνητή νοημοσύνη για να διαχειρίζεται τον φόρτο εργασίας έξυπνα στο Edge.[26]
- KubeEdge: επεκτείνει το Kubernetes ώστε να μπορεί να λειτουργεί και σε συσκευές στο άκρο του δικτύου (όχι μόνο στο cloud).[27]
- Conflict-free Replicated Data-based Kubernetes: βοηθά στη σταθερή διαχείριση δεδομένων, ακόμα και όταν υπάρχει διακοπή στη σύνδεση.[28]

Όλα αυτά επιτρέπουν την επεξεργασία δεδομένων πιο κοντά στη συσκευή ή στον αισθητήρα, μειώνοντας την καθυστέρηση και αυξάνοντας την αυτονομία του συστήματος.

Στο Internet of Things, η καθυστέρηση είναι πολύ σημαντική – πολλές εφαρμογές πρέπει να αντιδρούν σχεδόν άμεσα. Ωστόσο, λίγες πλατφόρμες Kubernetes λαμβάνουν υπόψη την καθυστέρηση όταν αποφασίζουν πού να τρέξουν κάθε εφαρμογή. Υπάρχουν όμως μερικές εξαιρέσεις:

- GAIKube: προβλέπει πού είναι καλύτερο να τρέξει η εφαρμογή, για να είναι πιο κοντά στον χρήστη ή την πηγή δεδομένων.[26]
- Load-Aware προσεγγίσεις: ελέγχουν την κίνηση στο δίκτυο και προσαρμόζονται ανάλογα.[29]
- Conflict-free Replicated Data Type (Τύπος Αναπαραγόμενων Δεδομένων χωρίς Συγκρούσεις): βοηθούν στην ταχύτερη ανταπόκριση όταν τα δεδομένα είναι κατανεμημένα. [28]

Γενικά, η υποστήριξη για καθυστέρηση είναι περιορισμένη και αποτελεί πρόβλημα για κρίσιμες εφαρμογές στο βιομηχανικό περιβάλλον.

Ο αυτοματισμός και η αυτόματη κλιμάκωση είναι απαραίτητα για δυναμικούς φόρτους εργασίας. Πέρα από το HPA/VPA του Kubernetes, προηγμένες τεχνικές αξιοποιούν telemetry σε πραγματικό χρόνο και AI για πρόβλεψη αναγκών και αυτόματη προσαρμογή. Η χρήση AI για λήψη εξυπνων αποφάσεων κλιμάκωσης βελτώνει την αποδοτικότητα, την ανθεκτικότητα και την απόδοση των συστημάτων Kubernetes, ιδίως σε περιβάλλοντα με μεγάλο και απρόβλεπτο φόρτο εργασίας.[25], [30]

Ειδικά Χαρακτηριστικά: Τα μοντέλα ορχήστρωσης διαφέρουν σημαντικά. Κάποια ενσωματώνουν blockchain για διαφάνεια, το GAIKube προβλεπτική νοημοσύνη για βελτιστοποίηση, ενώ η Green Orchestration εστιάζει στη βιωσιμότητα μέσω περιβαλλοντικών παραμέτρων. Αυτά τα χαρακτηριστικά υπογραμμίζουν την ποικιλία των λύσεων και την αντιμετώπιση ειδικών αναγκών.[31]

4.1.2 Ενορχήστρωση Containers με Εστίαση στο Edge

Η διάδοση εφαρμογών σε περιβάλλοντα που εκτείνονται από το cloud έως το edge απαιτεί από τις πλατφόρμες ενορχήστρωσης να προσαρμοστούν ουσιαστικά σε ένα νέο, σύνθετο υπολογιστικό τοπίο. Σε αυτό το συνεχές, όπου το cloud προσφέρει αφθονία πόρων και ευελιξία, ενώ το edge χαρακτηρίζεται από περιορισμένη υποδομή και συχνά ασταθείς συνθήκες, οι απαιτήσεις για ελαφριές, αποκεντρωμένες και αυτόνομες λύσεις είναι κρίσιμες. Οι παραδοσιακές παραδοχές για σταθερότητα, ισχυρή συνδεσιμότητα και συνεχή διαθεσιμότητα δεν ισχύουν πλέον στο edge. Αντιθέτως, η μεταβλητότητα και οι περιορισμοί δημιουργούν σημαντικές προκλήσεις για την αξιόπιστη λειτουργία κατανεμημένων συστημάτων, επιβάλλοντας νέες αρχές σχεδίασης και ενορχήστρωσης.

Στο σύνθετο αυτό πλαίσιο, η ερευνητική κοινότητα και η βιομηχανία έχουν επιδείξει έντονη δραστηριότητα, προτείνοντας και αξιολογώντας ένα ευρύ φάσμα αρχιτεκτονικών προσεγγίσεων. Μέσα σε αυτό το σύνθετο πλαίσιο, τόσο η επιστημονική κοινότητα όσο και η βιομηχανία έχουν δείξει έντονο ενδιαφέρον, προτείνοντας πολλές λύσεις για την ενορχήστρωση σε edge περιβάλλοντα. Οι προσεγγίσεις αυτές περιλαμβάνουν είτε την τροποποίηση υπαρχόντων συστημάτων — όπως το γνωστό Kubernetes — ώστε να καλύπτουν τις ανάγκες του edge, είτε τον σχεδιασμό εντελώς νέων, ελαφριών και αποκλειστικά για το edge πλατφορμών. Οι βασικές απαιτήσεις αυτών των λύσεων περιλαμβάνουν τη δυνατότητα να παίρνονται αποφάσεις τοπικά, χωρίς να χρειάζεται συνεχής επικοινωνία με κεντρικά σημεία, ένα αποκεντρωμένο μοντέλο ελέγχου για μεγαλύτερη αυτονομία και ανθεκτικότητα, υψηλή ενεργειακή αποδοτικότητα λόγω περιορισμένων πόρων,

καθώς και υποστήριξη για ελαχιστοποίηση της καθυστέρησης, ώστε να εξυπηρετούνται εφαρμογές με ανάγκες για άμεση απόκριση.[32]

Αναλύοντας τις σχετικές εργασίες, μπορούμε να διακρίνουμε τις ακόλουθες αρχιτεκτονικές προσεγγίσεις:

Το k3s και το KubeEdge είναι δύο δημοφιλείς και ελαφριές εκδόσεις του Kubernetes, σχεδιασμένες για χρήση σε περιβάλλοντα edge computing. Τέτοιες λύσεις είναι απαραίτητες λόγω των ιδιαίτερων δυσκολιών στο edge, όπως οι περιορισμένοι υπολογιστικοί πόροι, η αστάθεια στη σύνδεση και η ανάγκη για αυτόνομη λειτουργία όταν δεν υπάρχει σύνδεση με το cloud.

Το k3s, που αναπτύχθηκε από τη Rancher Labs, απλοποιεί σημαντικά το Kubernetes. Αφαιρεί μη απαραίτητα κομμάτια και συνδυάζει υπηρεσίες ώστε να έχει μικρότερο αποτύπωμα. Είναι κατάλληλο για περιβάλλοντα IoT και edge, υποστηρίζει τα ίδια APIs με το Kubernetes και είναι εύκολο στην εγκατάσταση και τη διαχείριση, με δυνατότητα αυτόματης αναβάθμισης. Το KubeEdge, που υποστηρίζεται από το CNCF, επεκτείνει το Kubernetes στο edge, διατηρώντας το βασικό control plane στο cloud και χρησιμοποιώντας έναν ελαφρύ agent (EdgeCore) σε κάθε edge κόμβο. Υποστηρίζει συγχρονισμό δεδομένων και μπορεί να λειτουργεί αυτόνομα όταν δεν υπάρχει σύνδεση, ενώ επιτρέπει και τη διαχείριση συσκευών στο edge. Παρόλο που τόσο το k3s όσο και το KubeEdge έχουν κοινό στόχο — τη διαχείριση containerized εφαρμογών στο edge — έχουν διαφορετική σχεδίαση και διαφορετικές προτεραιότητες. Το k3s είναι κατάλληλο όταν θέλουμε ένα πολύ ελαφρύ Kubernetes σε κάθε edge συσκευή, ενώ το KubeEdge είναι καλύτερο για περιπτώσεις όπου υπάρχει μεγάλος αριθμός edge κόμβων και απαιτείται κεντρική διαχείριση, μεγαλύτερη αυτονομία και ανθεκτικότητα. Παρά τις διαφορές τους, και οι δύο τεχνολογίες βασίζονται σε ένα κεντρικό control plane. Ωστόσο, προσφέρουν κρίσιμες δυνατότητες για το edge, όπως απομακρυσμένη διαχείριση κόμβων και εφαρμογών, έλεγχο της κατάστασης των containers και των συστημάτων, καθώς και αυτόματη επανεκκίνηση σε περίπτωση προβλήματος. Έτσι εξασφαλίζουν τη σταθερότητα και τη διαθεσιμότητα που απαιτούν οι edge εφαρμογές. Τελικά, η επιλογή ανάμεσα σε k3s και KubeEdge εξαρτάται από τις ανάγκες και τους περιορισμούς του εκάστοτε edge περιβάλλοντος.[33]

Πλατφόρμα Μοντέλο	Αρχιτεκτονική Εστίαση	Latency-awareness	Scheduling	Decentralisation	Κατανάλωση Πόρων	Χρήση σε IoT/Fog
k3s	Lightweight K8s	Όχι εγγενώς	Static	Όχι	Πολύ χαμηλή	Ναι

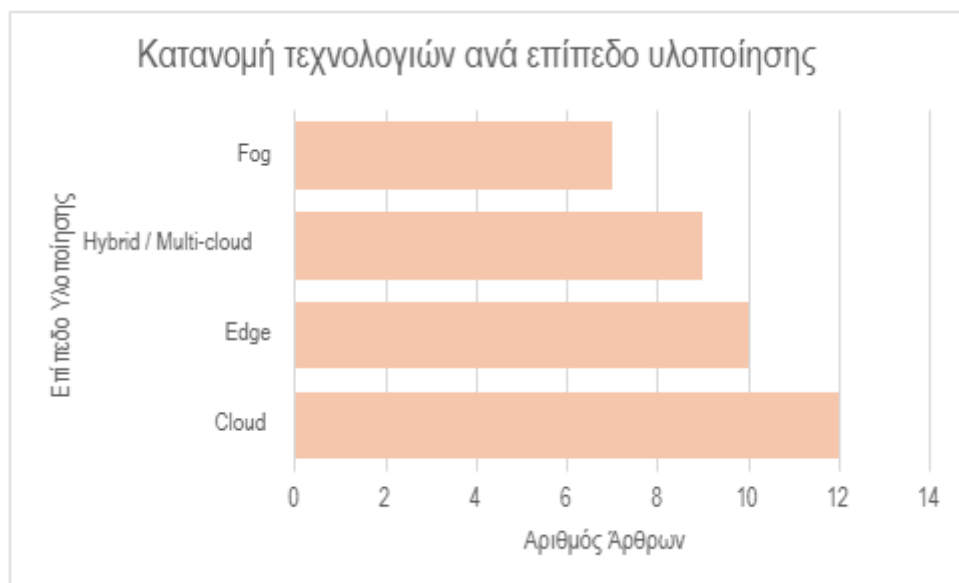
KubeEdge	Edge-native Kubernetes	Μέτρια	Centralized with edge sync	Όχι	Χαμηλή	Ναι
GAIKube	AI-based edge orchestration	Ναι	Proactive AI	Όχι	Μέτρια	Ναι
CRDT-based K8s	Decentralized K8s	Ναι	Local-only	Ναι	Χαμηλή	Ναι
MOSGD	Multi-objective optimization	Έμμεσα μέσω KPIs	AI-based Gradient Descent	Όχι	Optimized	Ναι
Blockchain-based orchestration	Secure edge orchestration	Όχι	Decentralized trust consensus	Ναι	Μέτρια	Ναι
Fog K8s with GBA (Latency-Aware)	Industry-focused fog	Ναι (GBA algorithm)	Greedy Border Allocation	Όχι	Χαμηλή	Ναι

Πίνακας 4.1.2 Συγκριτική αξιολόγηση σύγχρονων πλατφορμών ενορχήστρωσης containers με χρήση σε περιβάλλοντα IoT/Fog

Ο Πίνακας 4.1.2 παρουσιάζει τη συγκριτική αποτύπωση σύγχρονων πλατφορμών ενορχήστρωσης containerized εφαρμογών σε περιβάλλοντα IoT και fog computing. Οι λύσεις διαφέρουν σημαντικά ως προς τη latency-awareness, το επίπεδο αποκέντρωσης και την κατανάλωση πόρων, γεγονός που υπογραμμίζει την ανάγκη προσεκτικής επιλογής της κατάλληλης πλατφόρμας με βάση το προφίλ του εκάστοτε edge/fog deployment. Στον Πίνακα 4.1.5 παρουσιάζεται η ποσοτική ανάλυση της βιβλιογραφίας ως προς το επίπεδο υλοποίησης των προσεγγίσεων ενορχήστρωσης. Το cloud παραμένει το βασικό πεδίο μελέτης (**12 άρθρα**), ενώ το edge (**10 άρθρα**) και το fog (**7 άρθρα**) αναδεικνύονται ως κρίσιμα πεδία με αυξανόμενο ενδιαφέρον. Ιδιαίτερη σημασία έχει η παρουσία υβριδικών και πολυνεφικών αρχιτεκτονικών (**9 άρθρα**), γεγονός που υπογραμμίζει την ανάγκη για ενοποιημένες λύσεις στο Cloud-to-Thing Continuum.

Επίπεδο Υλοποίησης	Sum of Αριθμός Άρθρων
Cloud	12
Edge	10
Hybrid / Multi-cloud	9
Fog	7
Grand Total	38

Πίνακας 4.1.2.2 Πίνακας κατανομής τεχνολογιών ανά επίπεδο υλοποίησης



Εικόνα 4.1.2 Ραβδόγραμμα Κατανομής Τεχνολογιών ανά Επίπεδο Υλοποίησης

4.1.3 AI-based Scheduling και Προδραστική Ενορχήστρωση

Η παραδοσιακή αρχιτεκτονική του Kubernetes βασίζεται στο etcd για την αποθήκευση και τον συντονισμό, όμως αυτή η κεντρική προσέγγιση ενδέχεται να προκαλέσει συμφόρηση. Η ιδέα του CRDT-based Kubernetes αντικαθιστά το etcd με ένα κατακευματισμένο μοντέλο που βασίζεται σε CRDTs, επιτρέποντας ανεξάρτητες τροποποιήσεις και αυτόματη επίλυση συγκρούσεων με τελική συνέπεια. Η μετάβαση σε CRDTs επιτρέπει στους edge κόμβους να λειτουργούν πιο αυτόνομα, διατηρώντας τοπικές αναπαραγωγές δεδομένων και λαμβάνοντας αποφάσεις ανεξάρτητα, με ασύγχρονη αναπαραγωγή αλλαγών και σύγκλιση μέσω CRDTs, προσφέροντας πλεονεκτήματα για edge deployments.[34]

Βελτιωμένη επεκτασιμότητα (Scalability): Η απουσία ενός κεντρικού σημείου συντονισμού μειώνει τα bottlenecks και επιτρέπει την εύκολη προσθήκη και διαχείριση μεγάλου αριθμού edge κόμβων.

Μειωμένο latency: Οι edge κόμβοι μπορούν να λαμβάνουν αποφάσεις τοπικά, χωρίς την ανάγκη να επικοινωνούν με ένα κεντρικό σύστημα που μπορεί να βρίσκεται σε μεγάλη απόσταση, μειώνοντας σημαντικά τις καθυστερήσεις.

Αυξημένη ανθεκτικότητα (Resilience): Οι edge κόμβοι μπορούν να συνεχίσουν τη λειτουργία τους ακόμα και σε περιπτώσεις προσωρινής αποσύνδεσης από το κεντρικό δίκτυο, καθώς διαθέτουν την απαραίτητη κατάσταση για να λαμβάνουν αποφάσεις αυτόνομα. Αυτό είναι

ιδιαίτερα σημαντικό για disconnected ή delay-tolerant edge deployments, όπου η αξιοπιστία και η συνεχής λειτουργία είναι κρίσιμες.

Απλοποιημένη διαχείριση σε κατανεμημένα περιβάλλοντα: Η αυτονομία των edge κόμβων μπορεί να απλοποιήσει τη διαχείριση και την παρακολούθηση μεγάλων και γεωγραφικά διασκορπισμένων deployments.

Ωστόσο, η υιοθέτηση ενός CRDT-based Kubernetes εισάγει επίσης νέες προκλήσεις. Η διασφάλιση της τελικής συνέπειας απαιτεί προσεκτικό σχεδιασμό των CRDTs και των μηχανισμών αναπαραγωγής. Επιπλέον, η παρακολούθηση και η διαχείριση ενός πλήρως αποκεντρωμένου συστήματος μπορεί να είναι πιο σύνθετη σε σχέση με ένα κεντρικά ελεγχόμενο σύστημα. Παρόλα αυτά, το CRDT-based Kubernetes αντιπροσωπεύει μια πολλά υποσχόμενη κατεύθυνση για την εξέλιξη της ενορχήστρωσης container, ιδιαίτερα καθώς η υπολογιστική στο edge και τα κατανεμημένα συστήματα γίνονται όλο και πιο διαδεδομένα. Η δυνατότητα αυτόνομης λειτουργίας των edge κόμβων, η βελτιωμένη επεκτασιμότητα και η μειωμένη λανθάνουσα κατάσταση ανοίγουν νέους δρόμους για την ανάπτυξη και τη διαχείριση κατανεμημένων εφαρμογών σε ένα ευρύ φάσμα περιπτώσεων χρήσης. Η περαιτέρω έρευνα και ανάπτυξη σε αυτόν τον τομέα αναμένεται να οδηγήσει σε καινοτόμες λύσεις που θα αξιοποιούν πλήρως τις δυνατότητες της αποκεντρωμένης ενορχήστρωσης.

Το μοντέλο MOSGD, αρχικά για ενεργειακή αποδοτικότητα και χωροθέτηση, χρησιμοποιεί πολυκριτηριακή βελτιστοποίηση για ταυτόχρονη μείωση λανθάνοντος χρόνου και κατανάλωσης ενέργειας, κρίσιμο σε περιβάλλοντα με αυστηρές απαιτήσεις. Η επιτυχής λειτουργία του σε edge συσκευές όπως το Raspberry Pi αποδεικνύει ότι έξυπνοι αλγόριθμοι μπορούν να αξιοποιηθούν αποτελεσματικά σε πλατφόρμες περιορισμένων πόρων, προσφέροντας βελτιωμένη απόδοση και αυτονομία. Η σύγκριση προσεγγίσεων ενορχήστρωσης Cloud-to-Thing Continuum αποκαλύπτει διαφοροποιήσεις σε τρεις άξονες, ζωτικής σημασίας για την επιλογή της κατάλληλης αρχιτεκτονικής.[35]

Πλατφόρμα / Μοντέλο	Τύπος AI / Optimization	Σκοπός	Latency-awareness	Προδραστικότητα	Περιβάλλον Υλοποίησης
GAIKube	Generative AI (LLMs + RL)	Προδραστικό scheduling & scaling	Ναι	Ναι (predictive orchestration)	Edge
MOSGD	Multi-objective Gradient Descent	Latency-energy trade-off optimization	Εμμέσα	Όχι	Edge (Carinthian Computing Continuum)

RL-based Kubernetes Scheduler (Literature)	Deep Reinforcement Learning	Dynamic pod placement	Nαι	Nαι	Cloud/Hybrid
Auto-tuned Descheduler (Load-Aware K8s)	Telemetry + Custom rules (light ML)	Rebalancing workload in runtime	Nαι	Εν μέρει	Kubernetes clusters
AI + ILP Autoscaler (Joint Autoscaling)	Predictive + Optimization model	Scaling VMs & pods with cost-efficiency	Όχι	Nαι	Cloud (AWS EKS)

Πίνακας 4.1.3 Συνοπτική παρουσίαση AI/Optimization μοντέλων για έξυπνη εννοχρήστρωση σε περιβάλλοντα Kubernetes και Edge/Cloud

Οι αρχιτεκτονικές εννοχρήστωσης στο Cloud-to-Thing Continuum διαφέρουν κυρίως ως προς το επίπεδο αποκέντρωσης (από κεντρικό έλεγχο έως πλήρη αυτονομία), την ευφυΐα του συστήματος (από στατικούς αλγορίθμους έως AI-driven schedulers) και την αποδοτικότητα (latency, κατανάλωση ενέργειας, χρήση πόρων). Η επιλογή της κατάλληλης πλατφόρμας εννοχρήστωσης είναι κρίσιμη για την επιτυχία, καθώς κάθε εφαρμογή έχει μοναδικές ανάγκες που πρέπει να αξιολογηθούν προσεκτικά βάσει αυτών των τριών αξόνων.

4.1.4 Σύγκριση Kubernetes και Docker Swarm για Lightweight Υλοποιήσεις

Η επιλογή της κατάλληλης πλατφόρμας εννοχρήστωσης αναδεικνύεται σε ένα θεμελιώδες και κρίσιμο αρχιτεκτονικό ζήτημα που διαπερνά όλα τα στάδια του κύκλου ζωής των containerized εφαρμογών, από τον αρχικό σχεδιασμό και την ενδελεχή ανάπτυξη έως την αδιάλειπτη και αποτελεσματική διαχείρισή τους σε πραγματικό χρόνο. Η σημασία της επιλογής αυτής γίνεται ακόμη πιο έντονη και καθοριστική όταν οι εν λόγω εφαρμογές προορίζονται για λειτουργία σε υπολογιστικά περιβάλλοντα που χαρακτηρίζονται από αυστηρούς και συχνά ασφυκτικούς περιορισμούς στους διαθέσιμους υπολογιστικούς πόρους, όπως είναι εγγενώς τα κατανεμημένα και ετερογενή συστήματα fog και edge computing.

Στο πλαίσιο της εννοχρήστωσης containers, το Kubernetes και το Docker Swarm αξιολογούνται για την υποστήριξη ευέλικτων και αποκεντρωμένων εφαρμογών, κρίσιμων για το Cloud-to-Thing Continuum. Παρά την κυριαρχία του Kubernetes σε σύνθετα περιβάλλοντα, η πολυπλοκότητά του μπορεί να αποτελέσει πρόκληση για edge περιβάλλοντα χαμηλών πόρων. Αντίθετα, το Docker Swarm προσφέρει μια ελαφριά και απλούστερη εναλλακτική, ενσωματωμένη στον Docker engine, καθιστώντας το κατάλληλο για

συγκεκριμένες χρήσεις όπου η αποδοτικότητα πόρων και η ευκολία διαχείρισης είναι πρωταρχικές.

Ένα από τα θεμελιώδη πλεονεκτήματα του Docker Swarm έγκειται στην απρόσκοπτη και ταχύτατη διαδικασία εγκατάστασης και αρχικής διαμόρφωσης. Σε αντίθεση με την πολυπλοκότητα που συχνά συνοδεύει το Kubernetes κατά τα πρώτα του βήματα, το Docker Swarm προσφέρει μια πολύ πιο άμεση και κατανοητή εμπειρία, επιτρέποντας στις ομάδες να ξεκινήσουν την ενορχήστρωση των containers τους με ελάχιστη αρχική προσπάθεια και εξειδικευμένες γνώσεις. Αυτή η ευκολία εγκατάστασης μειώνει σημαντικά τον χρόνο και τους πόρους που απαιτούνται για την έναρξη ενός έργου ενορχήστρωσης.

Επιπρόσθετα, το Docker Swarm διακρίνεται για την εντυπωσιακή ταχύτητα με την οποία μπορεί να εκκινήσει νέα containers. Αυτή η αμεσότητα στην διάθεση νέων instances εφαρμογών είναι ιδιαίτερα πολύτιμη σε δυναμικά περιβάλλοντα όπου η ανάγκη για κλιμάκωση και απόκριση σε μεταβαλλόμενες απαιτήσεις είναι συνεχής. Παράλληλα, η αρχιτεκτονική του Docker Swarm είναι σχεδιασμένη για χαμηλή κατανάλωση υπολογιστικών πόρων. Αυτό το χαρακτηριστικό το καθιστά μια εξαιρετικά ελκυστική λύση για μικρότερα και μεσαία deployments, όπου η αποτελεσματική χρήση των διαθέσιμων πόρων είναι κρίσιμη για τον έλεγχο του κόστους και τη διατήρηση της απόδοσης.

Πέρα από τα παραπάνω, η χαμηλή κατανάλωση πόρων καθιστά το Docker Swarm μια ιδανική επιλογή για περιβάλλοντα edge computing. Σε αυτές τις περιπτώσεις, οι υπολογιστικοί πόροι είναι συχνά περιορισμένοι και η ικανότητα για ευέλικτη και ελαφριά ανάπτυξη εφαρμογών είναι υψίστης σημασίας. Το Docker Swarm επιτρέπει την αποτελεσματική διαχείριση containers σε τέτοια περιβάλλοντα, χωρίς να επιβαρύνει υπερβολικά τους διαθέσιμους πόρους.

Τέλος, ένα σημαντικό πλεονέκτημα του Docker Swarm είναι η στενή και άρρηκτη ενσωμάτωσή του στο ευρύτερο Docker ecosystem. Για ομάδες που ήδη χρησιμοποιούν το Docker για τη δημιουργία, διαχείριση και εκτέλεση containers, η μετάβαση στο Docker Swarm για την ενορχήστρωσή τους είναι ομαλή και φυσική. Η εξοικείωση με τις βασικές έννοιες και εργαλεία του Docker διευκολύνει την υιοθέτηση του Swarm, μειώνοντας την καμπύλη εκμάθησης και επιτρέποντας στις ομάδες να αξιοποιήσουν άμεσα τις υπάρχουσες γνώσεις και δεξιότητες. Αυτή η εγγενής ενσωμάτωση προσφέρει μια συνεκτική και αποτελεσματική εμπειρία διαχείρισης ολόκληρου του κύκλου ζωής των containerized εφαρμογών.

Αναλύοντας λεπτομερώς τα συγκριτικά ευρήματα, αναδεικνύονται σημαντικές διαφοροποιήσεις στις δυνατότητες και τις απαιτήσεις των διαφόρων διανομών Kubernetes. Το Kubernetes στην πλήρη του μορφή είναι ένα ισχυρό πλαίσιο ενορχήστρωσης container με προηγμένες λειτουργίες όπως RBAC, αυτόματη κλιμάκωση, CRDs και fault-tolerant control plane, ιδανικό για σύνθετα συστήματα. Ωστόσο, το αυξημένο λειτουργικό βάρος και οι απαιτήσεις πόρων το καθιστούν λιγότερο κατάλληλο για απλές εφαρμογές ή συσκευές περιορισμένων πόρων. Η πολυπλοκότητα στην εγκατάσταση και διαχείριση είναι επίσης ένας παράγοντας. Αντίθετα, παραλλαγές όπως το k3s έχουν σχεδιαστεί για μειωμένο αποτύπωμα και ευκολία εγκατάστασης, διατηρώντας τις βασικές λειτουργίες αλλά αφαιρώντας λιγότερο απαραίτητα components. Αυτό επιτρέπει την αποτελεσματική λειτουργία σε μικρότερες πλατφόρμες και απλοποιεί την εγκατάσταση και διαχείριση, καθιστώντας το ελκυστικό για

ελαφριά workloads. Η επιλογή εξαρτάται από τις ανάγκες της εφαρμογής, τους διαθέσιμους πόρους και τις απαιτήσεις διαχείρισης.[36], [37], [38]

Docker Swarm

Το Docker Swarm είναι μια αξιόπιστη και εύχρηστη λύση ενορχήστρωσης container, ιδανική για περιβάλλοντα με περιορισμένους πόρους ή για απλούστερη διαχείριση σε σχέση με το Kubernetes. Προσφέρει ευκολία εγκατάστασης, αυτόματη αναπαραγωγή container, service discovery, load balancing και rolling updates, εξασφαλίζοντας υψηλή διαθεσιμότητα και αποτελεσματική διαχείριση. Ωστόσο, υστερεί σε προηγμένες δυνατότητες scheduling και πολιτικές ασφαλείας. Συνολικά, είναι μια ισχυρή επιλογή για μικρές έως μεσαίες αναπτύξεις όπου η απλότητα και η ταχύτητα είναι σημαντικές, αλλά για πιο σύνθετες ανάγκες, άλλες λύσεις μπορεί να είναι προτιμότερες. Η επιλογή εξαρτάται από τις συγκεκριμένες απαιτήσεις του έργου.

Σε δοκιμές με SBCs και IoT, το Docker Swarm είχε χαμηλότερο latency και ταχύτερους χρόνους εκκίνησης containers από το Kubernetes, υποδεικνύοντας ευέλικτη διαχείριση σε περιβάλλοντα με περιορισμένους πόρους. Το Kubernetes, ενώ προσφέρει πολλές λειτουργίες ενορχήστρωσης, επιβάλλει αυξημένο overhead, επηρεάζοντας την απόδοση σε συσκευές χαμηλών προδιαγραφών. Ωστόσο, υπερτερεί σε υψηλή διαθεσιμότητα, ανθεκτικότητα και παρακολούθηση κρίσιμων εφαρμογών.

Η επιλογή εξαρτάται από τις ανάγκες του έργου και το υλικό. Το Docker Swarm είναι προτιμότερο για IoT και SBCs με έμφαση στην ταχύτητα και το χαμηλό latency. Το Kubernetes είναι καλύτερο για συστήματα που απαιτούν ανθεκτικότητα, παρακολούθηση και κλιμάκωση, παρά το μεγαλύτερο overhead.[39]

Κριτήριο	Kubernetes (πλήρες ή k3s)	Docker Swarm
Απαιτήσεις συστήματος	Υψηλές (RAM, CPU) – εκτός από k3s	Πολύ χαμηλές
Ευκολία εγκατάστασης	Πολύπλοκη (ειδικά το vanilla)	Πολύ απλή και άμεση
Ελαφρότητα	Βαρύ (εκτός από k3s), modular	Πολύ ελαφρύ, native στο Docker
Λειτουργικότητα	Πολύ πλήρης (autoscaling, RBAC, CRDs)	Περιορισμένη (μόνο βασικά orchestration features)
Υποστήριξη Edge/Fog	Ναι (μέσω K3s, KubeEdge)	Ναι (native, χωρίς πρόσθετα)
Κατανάλωση πόρων	Υψηλή (εκτός από K3s)	Χαμηλή
Scalability	Εξαιρετική, enterprise-ready	Μέτρια (πρακτικά έως ~100 nodes)

Δυνατότητες παρακολούθησης (observability)	Προηγμένες (Prometheus, Grafana, etc.)	Βασικές
Ανθεκτικότητα self-healing	& Ναι, εγγενώς	Περιορισμένη (leader-follower failover)
Κατάλληλο production IoT/Fog;	για Ναι, με παραμετροποίηση	Ναι, για μικρές/μεσαίες εφαρμογές

Πίνακας 4.1.4 Συγκριτική αξιολόγηση Kubernetes και Docker Swarm με βάση απαιτήσεις, δυνατότητες και καταλληλότητα για IoT/Fog περιβάλλοντα

Το Kubernetes ενδείκνυται για πολύπλοκα περιβάλλοντα με αυστηρές απαιτήσεις κλιμάκωσης και επιχειρησιακή κρισιμότητα, προσφέροντας προηγμένες λειτουργίες με αυξημένες απαιτήσεις σε πόρους και διαχείριση. Το Docker Swarm αποτελεί μια ελκυστική εναλλακτική για ελαφριές και edge υλοποιήσεις, με απλή χρήση και χαμηλή κατανάλωση πόρων, κατάλληλο για εφαρμογές χωρίς τις προηγμένες δυνατότητες του Kubernetes και για περιβάλλοντα με περιορισμένους πόρους.

Η αυξανόμενη ανάγκη για ελαφρές και ευέλικτες πλατφόρμες που μπορούν να λειτουργήσουν αποτελεσματικά σε edge και fog επίπεδα, όπου οι πόροι είναι συχνά περιορισμένοι και η συνδεσιμότητα μπορεί να είναι ασταθής, οδήγησε στην ανάπτυξη εναλλακτικών λύσεων όπως το k3s. Το k3s αποτελεί μια ελαφριά έκδοση του Kubernetes, πιστοποιημένη από το Cloud Native Computing Foundation (CNCF), η οποία έχει σχεδιαστεί για να γεφυρώσει το χάσμα μεταξύ της πληρότητας λειτουργιών του Kubernetes και της ελαφρότητας του Docker Swarm. Διατηρώντας τις βασικές λειτουργίες και τα APIs του Kubernetes, το k3s προσφέρει σημαντικά μειωμένες απαιτήσεις σε πόρους, ευκολότερη εγκατάσταση και διαχείριση, καθιστώντας το ιδανικό για edge computing, IoT και άλλα περιβάλλοντα με περιορισμούς. Με αυτόν τον τρόπο, οι χρήστες μπορούν να αξιοποιήσουν την ωριμότητα και το ευρύ οικοσύστημα του Kubernetes σε περιπτώσεις όπου το κανονικό Kubernetes θα ήταν υπερβολικά βαρύ ή πολύπλοκο. Η επιλογή της κατάλληλης πλατφόρμας ενορχήστρωσης απαιτεί προσεκτική ανάλυση των αναγκών της εφαρμογής, των χαρακτηριστικών του περιβάλλοντος ανάπτυξης και λειτουργίας, καθώς και των διαθέσιμων πόρων και της τεχνογνωσίας της ομάδας.[40], [41]

4.1.5 Υβριδική και Πολυνεφική Διαλειτουργικότητα στην Ενορχήστρωση

Η ενσωμάτωση του Cloud-to-Thing Continuum απαιτεί απρόσκοπτη λειτουργία εφαρμογών και υποδομών σε πολλαπλούς παρόχους και επίπεδα, από data centers έως edge συσκευές. Η υβριδική και πολυνεφική διαλειτουργικότητα είναι απαραίτητη για ευέλικτες αρχιτεκτονικές, επιτρέποντας αλληλεπίδραση, συνεργασία και μεταφορά φόρτων εργασίας/δεδομένων μεταξύ νέφων, για μέγιστη ευελιξία, επεκτασιμότητα και ανθεκτικότητα. Η απουσία διαλειτουργικότητας οδηγεί σε απομονωμένα συστήματα και μειωμένη αποδοτικότητα.

Κύριες προκλήσεις:

- Διασύνδεση και συντονισμός κατανεμημένων clusters:
 - Αποτελεσματική επικοινωνία και συνεργασία αυτόνομων κόμβων.

- Αντιμετώπιση τεχνικών δυσκολιών, NAT, firewalls και γεωγραφικής διασποράς.
- Ανεξαρτησία από παρόχους:
 - Αποφυγή εξάρτησης μέσω φορητών τεχνολογιών.
- Ενοποιημένη πολιτική ενορχήστρωσης και ασφαλούς διαχείρισης workloads:
 - Κεντρικός μηχανισμός για απλοποίηση, αξιοποίηση πόρων και ενίσχυση ασφάλειας.
- Έλλειψη ενοποίησης:
 - Οδηγεί σε ασυνέπεια και αυξημένο κίνδυνο.

Η σχετική βιβλιογραφία προτείνει πληθώρα προσεγγίσεων που απαντούν στις ανωτέρω προκλήσεις:

Το EdgeVPN.io αποτελεί μια καινοτόμα πλατφόρμα που διευκολύνει τη διασύνδεση ετερογενών υποδομών μέσω ενός ευέλικτου και αυτοδιαχειριζόμενου peer-to-peer VPN. Χάρη στη χρήση τεχνολογιών όπως το WebRTC, το XMPP και τεχνικές NAT traversal, επιτρέπει τη δημιουργία καταναμημένων Kubernetes clusters, ακόμα και όταν οι κόμβοι βρίσκονται σε διαφορετικά διαχειριστικά domains, γεωγραφικά απομακρυσμένες περιοχές ή ανήκουν σε διαφορετικούς οργανισμούς. Η πλατφόρμα υποστηρίζει επίσης αυτόματη ανακάλυψη κόμβων, λογική οργάνωση μέσω namespaces, καθώς και εγγενή ενσωμάτωση με το Kubernetes μέσω CNI plugins, διευκολύνοντας τη διαχείριση containerized εφαρμογών σε καταναμημένο περιβάλλον. Το CRDT-based Kubernetes εισάγει ένα αποκεντρωμένο μοντέλο συνοχής δεδομένων, βασισμένο στην έννοια της τελικής συνέπειας (eventual consistency), επιτρέποντας την αυτόνομη λειτουργία κόμβων χωρίς κεντρική συμφωνία. Αυτή η προσέγγιση είναι ιδιαίτερα κατάλληλη για edge περιβάλλοντα με ασταθείς συνδέσεις ή γεωγραφική διασπορά.[41]

Πλατφόρμα / Πλαίσιο	Τύπος Υλοποίησης	Networking /Federation	NAT Firewall Traversal	Federation Support	Χρήση με Kubernetes
EdgeVPN.io	Overlay VPN SDN	Πλήρες	Ναι	Ναι	Ναι (CNI plugin)
Blockchain-based Orchestration	Decentralized ledger	Όχι	Όχι	Όχι	Μερικώς
CRDT-based Kubernetes	Eventually consistent datastore	Όχι	Όχι	Όχι	Ναι (με τροποποιήσεις)

Cloud Container Architectures Mapping Study	Mapping / Taxonomy	Διαφορετικά μοντέλα	Όχι	Ποικίλλει	Ναι
Hybrid Scaling με Foreman + Puppet	Private cloud scaling	Ναι (vCenter alarms)	Όχι	Όχι	Ναι

Πίνακας 4.1.5 Συγκριτική αποτύπωση σύγχρονων προσεγγίσεων δικτύωσης, federation και scaling για κατανεμημένες container-based υποδομές

Η Blockchain-based ενορχήστρωση προσφέρει μια ασφαλή και αξιόπιστη λύση συντονισμού κατανεμημένων εφαρμογών, εξασφαλίζοντας την ακεραιότητα και την εμπιστοσύνη μέσω αποκεντρωμένων μηχανισμών συναίνεσης. Είναι ιδιαίτερα χρήσιμη σε περιβάλλοντα με πολλαπλούς οργανισμούς ή stakeholders. Παράλληλα, η έρευνα για federated orchestrators προτείνει αρχιτεκτονικές που επιτρέπουν τον συντονισμό μεταξύ πολλών Kubernetes clusters ή διαφορετικών ενορχηστρωτών, ενισχύοντας τη διαλειτουργικότητα και τη φορητότητα εφαρμογών. Τέλος, το Hybrid Elastic Scaling μέσω εργαλείων όπως το Foreman και το Puppet προσφέρει μια ανοικτού κώδικα λύση για δυναμική κλιμάκωση υποδομής σε ιδιωτικά σύννεφα όπως OpenStack ή vCenter. Βασίζεται σε γεγονότα (event-driven scaling) και υποστηρίζει σενάρια υβριδικού cloud.[31]

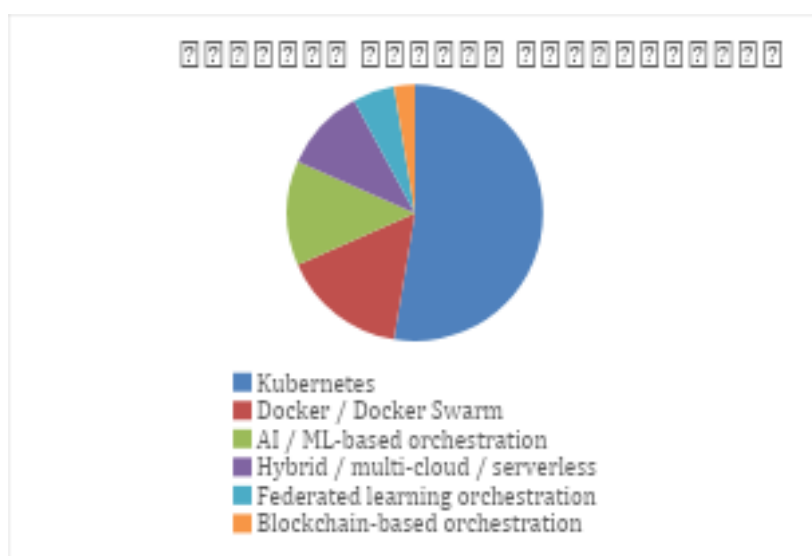
4.1.6 Συνοπτική Αξιολόγηση και Συμπεράσματα RQ1

Οι λύσεις ενορχήστρωσης για περιβάλλοντα πολλαπλών νέφων και υβριδικών υποδομών επιδεικνύουν ενοποιημένα και διαλειτουργικά συστήματα, αν και χαρακτηρίζονται από σημαντική τεχνική πολυπλοκότητα. Αρχιτεκτονικές που χρησιμοποιούν overlay δίκτυα (π.χ., EdgeVPN.io), μηχανισμούς αποκέντρωσης βασισμένους σε CRDTs, ή κατανεμημένα συστήματα εμπιστοσύνης όπως το blockchain, αποτελούν βιώσιμες προσεγγίσεις για ενορχήστρωση χωρίς κεντρικό έλεγχο και με μειωμένη εξάρτηση από έναν μόνο πάροχο cloud.

Επιπλέον, οι τεχνικές ομοσπονδοποίησης μεταξύ ενορχηστρωτών και τα εργαλεία για αποτελεσματική διαχείριση υβριδικών αναπτύξεων προσφέρουν κρίσιμη επεκτασιμότητα και αυτονομία. Αυτά είναι ζωτικής σημασίας για το Cloud-to-Thing Continuum, το οποίο εκτείνεται από το κεντρικό cloud έως τις IoT συσκευές. Σε αυτό το πλαίσιο, οι εφαρμογές πρέπει να λειτουργούν απρόσκοπτα και με συνέπεια σε ετερογενείς υποδομές και διάφορους παρόχους, διατηρώντας απόδοση και αξιοπιστία. Η ικανότητα επέκτασης και η αυτονομία των τμημάτων του συστήματος είναι καθοριστικές για την αντιμετώπιση της πολυπλοκότητας του Cloud-to-Thing Continuum.

Συμπερασματικά για το RQ1, δεν υπάρχει μία ενιαία αρχιτεκτονική ή τεχνολογία ενορχήστρωσης που να καλύπτει αποτελεσματικά όλο το Cloud-to-Thing Continuum. Η επιλογή της κατάλληλης λύσης εξαρτάται από μια στοχευμένη διαδικασία λήψης αποφάσεων που λαμβάνει υπόψη πολλούς παράγοντες και περιορισμούς σε κάθε επίπεδο του Continuum.

Η ποσοτική ανάλυση των μελετών έρχεται να ενισχύσει τη διαπίστωση αυτή, αποτυπώνοντας την επικράτηση συγκεκριμένων τεχνολογιών. Συγκεκριμένα, το Kubernetes εντοπίζεται στο **52,63%** των εξεταζόμενων μελετών, γεγονός που υπογραμμίζει τον κυρίαρχο ρόλο της ως πλατφόρμας ενορχήστρωσης στο Cloud-to-Thing Continuum. Το Docker / Docker Swarm καταγράφηκε στο **15,79%** των εργασιών, ενώ τεχνολογίες βασισμένες σε τεχνητή νοημοσύνη ή μηχανική μάθηση (AI/ML-based orchestration) εμφανίστηκαν σε ποσοστό **13,16%**. Λύσεις που αξιοποιούν federated learning ή blockchain παρουσιάζουν μικρότερη διείσδυση (με ποσοστά **5,26%** και **2,63%** αντίστοιχα), ενώ τα σενάρια υβριδικής/πολυνεφικής ενορχήστρωσης καλύπτουν περίπου **10,53%** του συνόλου. Η κατανομή αυτή απεικονίζει την εστίαση της έρευνας σε Kubernetes-centric λύσεις και την αυξανόμενη τάση ενσωμάτωσης ευφυών και αποκεντρωμένων προσεγγίσεων.

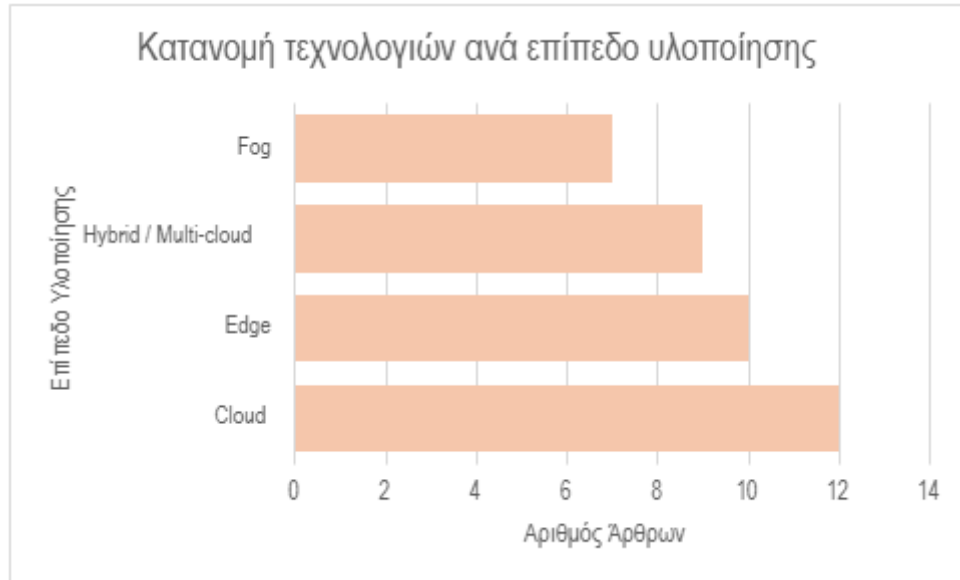


Εικόνα 4.1.3 Πίτα Ποσοστών Χρήσης Τεχνολογιών

Συγκεκριμένα, το περιβάλλον ανάπτυξης είναι καθοριστικός παράγοντας. Λύσεις ενορχήστρωσης αποδοτικές στο cloud, με πολλούς πόρους και χαμηλή λανθάνουσα κατάσταση, μπορεί να μην είναι κατάλληλες για fog ή edge, όπου οι πόροι είναι περιορισμένοι και οι απαιτήσεις για χαμηλή λανθάνουσα κατάσταση και αυτόνομη λειτουργία είναι αυξημένες. Στον Πίνακα 4.1.6 παρουσιάζεται η ποσοτική ανάλυση της βιβλιογραφίας ως προς το επίπεδο υλοποίησης των προσεγγίσεων ενορχήστρωσης. Το cloud παραμένει το βασικό πεδίο μελέτης (**12 άρθρα**), ενώ το edge (**10 άρθρα**) και το fog (**7 άρθρα**) αναδεικνύονται ως κρίσιμα πεδία με αυξανόμενο ενδιαφέρον. Ιδιαίτερη σημασία έχει η παρουσία υβριδικών και πολυνεφικών αρχιτεκτονικών (**9 άρθρα**), γεγονός που υπογραμμίζει την ανάγκη για ενοποιημένες λύσεις στο Cloud-to-Thing Continuum.

Επίπεδο Υλοποίησης	Sum of Αριθμός Άρθρων
Cloud	12
Edge	10
Hybrid / Multi-cloud	9
Fog	7

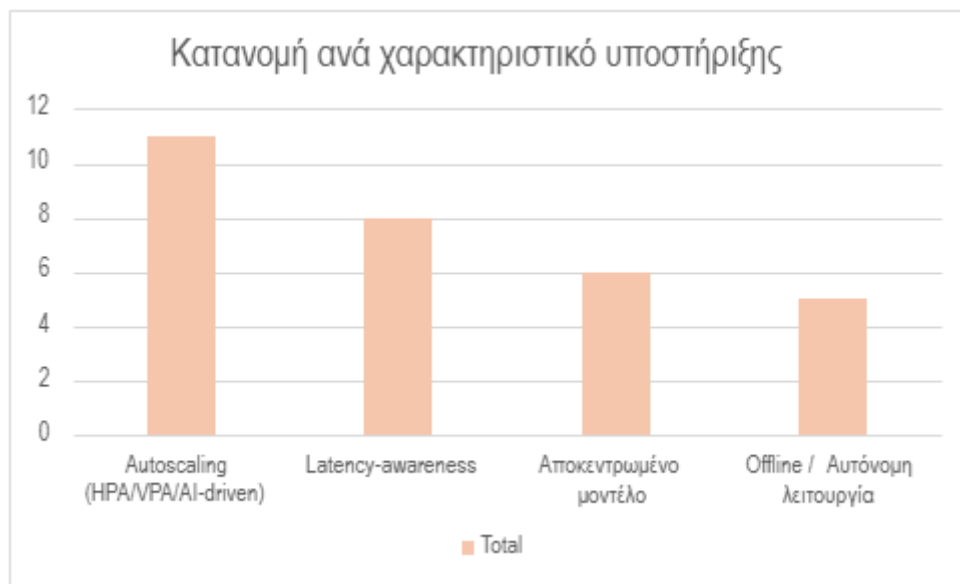
Πίνακας 4.1.6 Πίνακας κατανομής τεχνολογιών ανά επίπεδο υλοποίησης



Εικόνα 4.1.4 Γραφική απεικόνιση της κατανομής των τεχνολογιών ενορχήστρωσης containerized εφαρμογών βάσει του επιπέδου υλοποίησης (Cloud, Fog, Edge, Hybrid/Multi-cloud) όπως καταγράφηκε στη συστηματική βιβλιογραφική ανασκόπηση.

Αντίστοιχα, τεχνολογίες για IoT συσκευές με χαμηλή υπολογιστική ισχύ και ενεργειακούς περιορισμούς δεν μπορούν να χρησιμοποιηθούν αυτούσιες στο cloud. Επιπλέον, οι λειτουργικές απαιτήσεις των εφαρμογών είναι κεντρικές για την επιλογή της αρχιτεκτονικής ενορχήστρωσης. Εφαρμογές με εξαιρετικά χαμηλή λανθάνουσα κατάσταση (π.χ., αυτόνομη οδήγηση, βιομηχανικός αυτοματισμός) απαιτούν επεξεργασία και ενορχήστρωση δεδομένων κοντά στην πηγή (edge). Η ανάγκη για αυτονομία σε περιπτώσεις διακοπής σύνδεσης απαιτεί λύσεις ενορχήστρωσης αυτόνομης λειτουργίας. Παράλληλα, οι απαιτήσεις ασφάλειας διαφέρουν ανάλογα με το περιβάλλον και τα δεδομένα, επηρεάζοντας την επιλογή μηχανισμών ενορχήστρωσης και διαχείρισης ταυτότητας.

Η εικόνα 4.1.8 παρουσιάζει τη συνολική κατανομή των λύσεων ενορχήστρωσης βάσει των κύριων τεχνικών χαρακτηριστικών που υποστηρίζουν. Η δυνατότητα autoscaling (HPA, VPA ή AI-driven) εντοπίζεται στο μεγαλύτερο ποσοστό των περιπτώσεων (**11 άρθρα**), γεγονός που επιβεβαιώνει τη σημασία της αυτόματης προσαρμογής των πόρων σε δυναμικά περιβάλλοντα. Η latency-awareness καταγράφεται σε **8 άρθρα**, αναδεικνύοντας την αυξανόμενη ανάγκη για μειωμένη καθυστέρηση σε εφαρμογές edge και fog. Οι αποκεντρωμένες προσεγγίσεις εμφανίζονται σε **6 άρθρα**, ενώ η δυνατότητα offline ή αυτόνομης λειτουργίας σε **5 άρθρα**, υποδηλώνοντας περιορισμένη διείσδυση σε αυτό το κρίσιμο χαρακτηριστικό για edge περιβάλλοντα.

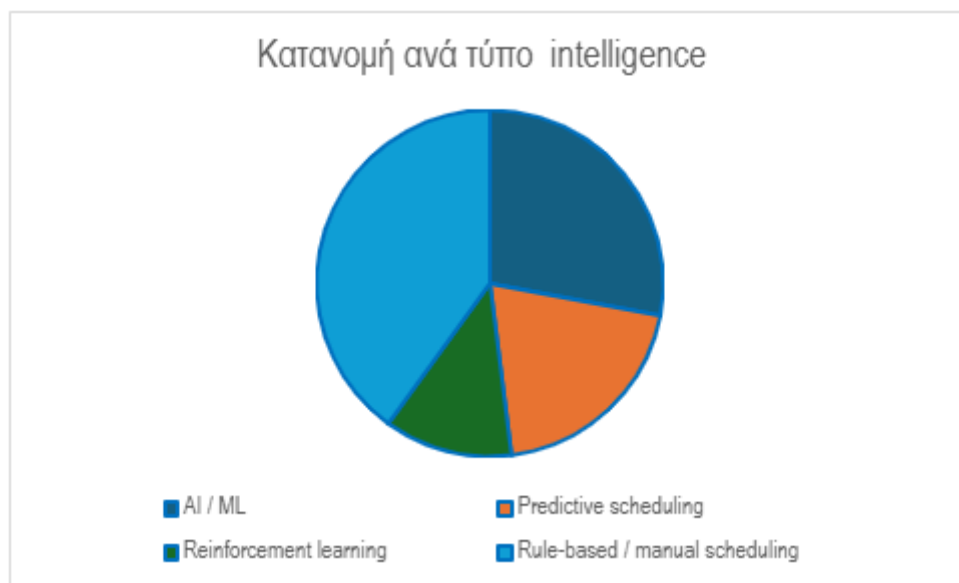


Εικόνα 4.1.5 Γραφική αποτύπωση της συχνότητας εμφάνισης κύριων τεχνικών χαρακτηριστικών στις λύσεις ενορχήστρωσης που εξετάστηκαν.

Οι επιχειρησιακοί περιορισμοί, όπως το κόστος και η ενεργειακή κατανάλωση, είναι επίσης σημαντικοί. Στο edge, όπου οι συσκευές μπορεί να είναι μπαταριοκίνητες ή με περιορισμένους πόρους ενέργειας, η επιλογή μιας ελαφριάς και ενεργειακά αποδοτικής λύσης ενορχήστρωσης είναι κρίσιμη. Το συνολικό κόστος ιδιοκτησίας, συμπεριλαμβανομένου του κόστους ανάπτυξης, διαχείρισης και συντήρησης, πρέπει να ευθυγραμμίζεται με τους επιχειρησιακούς στόχους. Οι πιο υποσχόμενες τάσεις στην ενορχήστρωση στο Cloud-to-Thing Continuum εστιάζουν στην ενσωμάτωση ευφυΐας, στην αποκέντρωση και στην ανάπτυξη ελαφρών λύσεων. Η χρήση CRDT-based orchestration επιτρέπει τη συνεπή διαχείριση καταναμεμημένων δεδομένων με ανθεκτικότητα σε σφάλματα και καθυστερήσεις. Οι AI-enhanced schedulers αξιοποιούν μηχανική μάθηση για βελτιστοποίηση της κατανομής και διαχείρισης πόρων, λαμβάνοντας υπόψη πολλαπλές παραμέτρους και προβλέποντας μελλοντικές ανάγκες. Τα overlay federated συστήματα προσφέρουν ευελιξία και διαλειτουργικότητα, επιτρέποντας τη διαχείριση ετερογενών υποδομών ως ένα ενιαίο λογικό σύστημα. Ωστόσο, η βιβλιογραφική ανασκόπηση αναδεικνύει σημαντικά ερευνητικά κενά. Η έλλειψη ομοιογενούς υποστήριξης για ενορχήστρωση σε όλο το Continuum αποτελεί σημαντικό εμπόδιο για την ανάπτυξη πλήρως καταναμεμημένων εφαρμογών. Οι υπάρχουσες λύσεις συχνά επικεντρώνονται σε συγκεκριμένα επίπεδα, δυσχεραίνοντας την απρόσκοπτη μετάβαση και συνεργασία. Η περιορισμένη εφαρμογή της τεχνητής νοημοσύνης σε πραγματικά edge production συστήματα υποδηλώνει ανεκμετάλλευτο δυναμικό για βελτιστοποίηση της ενορχήστρωσης σε περιβάλλοντα με περιορισμένους πόρους. Επιπλέον, η ανάγκη για τυποποιημένα υβριδικά orchestration APIs είναι επιτακτική για τη διευκόλυνση της διαλειτουργικότητας και της δημιουργίας φορητών εφαρμογών.

Στην εικόνα 4.1.9 παρουσιάζεται η κατανομή των λύσεων ενορχήστρωσης containerized εφαρμογών βάσει του τύπου ευφυΐας που αξιοποιούν. Η τεχνητή νοημοσύνη και η μηχανική μάθηση (AI/ML) εντοπίζονται σε **7 άρθρα**, ενώ οι προσεγγίσεις με **predictive scheduling** εμφανίζονται σε **5 περιπτώσεις**, αντανακλώντας τη σταδιακή ενσωμάτωση πιο προδραστικών μηχανισμών διαχείρισης. Οι λύσεις που βασίζονται σε **reinforcement**

learning είναι περιορισμένες (**3 άρθρα**), γεγονός που αναδεικνύει το περιθώριο περαιτέρω ανάπτυξης στο συγκεκριμένο πεδίο. Αντιθέτως, οι **rule-based ή manual scheduling** τεχνικές παραμένουν οι επικρατέστερες (**10 άρθρα**), γεγονός που υποδηλώνει ότι η ευρύτερη υιοθέτηση ευφύων αλγορίθμων εντοχικής βροχίας βρίσκεται ακόμη σε πρώιμο στάδιο.



Εικόνα 4.1.6: Γραφική απεικόνιση της κατανομής των λύσεων εντοχικής βροχίας containerized εφαρμογών ανά τύπο intelligence

Συμπερασματικά για το RQ1, δεν υπάρχει μία ενιαία αρχιτεκτονική ή τεχνολογία εντοχικής βροχίας που να καλύπτει αποτελεσματικά όλο το Cloud-to-Thing Continuum. Η επιλογή της κατάλληλης λύσης εξαρτάται από στοχευμένη διαδικασία λήψης αποφάσεων που λαμβάνει υπόψη πολλούς παράγοντες και περιορισμούς σε κάθε επίπεδο του Continuum. Η ποσοτική ανάλυση ενισχύει το συμπέρασμα αυτό, αποτυπώνοντας την κυριαρχία του Kubernetes (52,63% των μελετών) και την περιορισμένη διείσδυση νεότερων προσεγγίσεων όπως το blockchain ή το federated learning. Ο Πίνακας 4.1.6 και η Εικόνα 4.1.7 συνοψίζουν την κατανομή ανά επίπεδο υλοποίησης, ενώ οι Πίνακες 4.1.7 και 4.1.8 παρουσιάζουν αντίστοιχα τα τεχνικά χαρακτηριστικά και τον τύπο intelligence που ενσωματώνουν οι λύσεις. Η συνολική αποτύπωση επιβεβαιώνει την ανάγκη για διεπιστημονικές, διαλειτουργικές και προσαρμόσιμες προσεγγίσεις ώστε να καλυφθεί το πλήρες εύρος του Cloud-to-Thing Continuum.

4.2 RQ2

4.2.1 Εισαγωγή

Η αποτελεσματική διαχείριση containerized εφαρμογών στο συνεχές φάσμα Cloud-to-Thing στηρίζεται στην ενοποίηση εξελιγμένων τεχνικών τοποθέτησης και προγραμματισμού. Η τοποθέτηση καθορίζει τον κόμβο εκτέλεσης ενός container, ενώ ο προγραμματισμός ρυθμίζει τη διάρκεια εκτέλεσης. Αυτές οι αλληλένδετες πτυχές είναι ζωτικής σημασίας για την απόδοση και την αποδοτικότητα σε σύνθετα, καταναεμημένα περιβάλλοντα, όπως τονίζεται στο δεύτερο ερευνητικό ερώτημα (RQ2).

Θεματική Κατηγορία	Μοντέλο / Πλατφόρμα	Τύπος / Στρατηγική	Κύρια Χαρακτηριστικά και Οφέλη
Lightweight Orchestration	K3s, FLEDGE, Docker Swarm	Lightweight orchestrators	Χαμηλό overhead, λειτουργία σε SBCs, συμβατότητα με Kubernetes API
	Docker (Ismail et al.)	Fast deployment, caching, migration	Ευκίνησία σε edge nodes, χαμηλή κατανάλωση πόρων
Latency-aware Scheduling	Latency-Aware K8s (Bin et al.)	RTT-based scheduling	Μέτρηση latency για node επιλογή, έως 40% βελτίωση καθυστέρησης
	GAIKube	AI-based latency prediction	Προδραστική μετεγκατάσταση pods, σταθερή απόκριση
	Kaiser et al. (Unikernels)	Runtime-aware placement	Μείωση latency variance μέσω επιλογής unikernel ή container ανά task
	Docker (Gupta & Nahrstedt)	Host networking, benchmarking	Μικρότερο network latency, προβλέψιμο behavior
AI-based & Predictive Scheduling	GAIKube	Generative AI + RL	Προβλέπει bottlenecks, αυτόνομο scheduling
	MOSGD	Multi-objective SGD optimizer	Βελτίωση latency & ενεργειακής αποδοτικότητας
	Joint Autoscaling (Li et al.)	ILP-based forecasting	Συγχρονισμένο scaling containers & VMs
	Survey (Al-Doghman et al.)	Κατηγοριοποίηση AI schedulers	Επισκόπηση supervised, RL, DQN, fuzzy logic
Hybrid Unikernel/Container Placement	Hybrid System (Kaiser et al.)	Task-aware scheduling	Έως 36% RAM saving, 25% latency βελτίωση για ελαφριά workloads
Decentralized / CRDT-based Scheduling	Rearchitected Kubernetes (Jeffery et al.)	CRDT-based, Local-first	Autonomous nodes, quorum-free, μειωμένο latency
	EdgeVPN.io	P2P VPN overlay	Συνδεσιμότητα μεταξύ edge nodes, ενίσχυση τοπολογικής αποκέντρωσης
Resource-aware Orchestration σε SBCs	K3s / KubeEdge / k0s (Yakubov et al.)	Limits, profiling, minimal overhead	Εξαιρετικό scaling, πλήρης λειτουργία σε 4GB SBCs
	Docker (Gupta & Nahrstedt)	Manual tuning & container limits	Προβλέψιμη χρήση πόρων, βέλτιστη συμπεριφορά σε Pi 3

	FLEDGE	Containerd-based, no kube-proxy	Έως 60% savings σε resources σε ARM-based κόμβους
--	--------	---------------------------------	---

Πίνακας 4.2.1 Συστηματική ταξινόμηση μοντέλων και στρατηγικών ενορχήστρωσης με έμφαση σε latency, AI, αποκέντρωση και lightweight υλοποιήσεις

Η εμβάθυνση στο RQ2 είναι κρίσιμη λόγω της ετερογένειας πόρων, των μεταβαλλόμενων δικτυακών συνθηκών και των ποικίλων απαιτήσεων εφαρμογών στα καταναμεμημένα περιβάλλοντα Cloud-to-Thing. Λανθασμένες αποφάσεις τοποθέτησης, όπως η εκτέλεση σε υπερφορτωμένους ή απομακρυσμένους κόμβους, μπορούν να οδηγήσουν σε υπερφόρτωση κόμβων, μειωμένη απόδοση, αυξημένη καθυστέρηση (κρίσιμη για εφαρμογές πραγματικού χρόνου), αναποτελεσματική χρήση ενέργειας και υποβάθμιση της ποιότητας υπηρεσιών (QoS). Αντίθετα, η προσεκτική και δυναμική επιλογή του κατάλληλου κόμβου, λαμβάνοντας υπόψη τις τρέχουσες συνθήκες και τις απαιτήσεις της εφαρμογής, αποτελεί θεμελιώδη πυλώνα για την αποδοτική ενορχήστρωση containerized εφαρμογών, εξασφαλίζοντας βέλτιστη απόδοση, μέγιστη αξιοποίηση πόρων και ελάχιστη καθυστέρηση.

Στο πλαίσιο του RQ2 εξετάζονται δύο αλληλένδετες έννοιες:

Ο προγραμματισμός είναι κρίσιμος για καταναμεμημένα συστήματα όπως το Kubernetes και περιλαμβάνει πολύπλοκες αποφάσεις για τη βελτιστοποίηση μετρικών. Η τοποθέτηση καθορίζει τον κόμβο ανάπτυξης container, βασιζόμενη σε στατικά και δυναμικά κριτήρια. Υπάρχουν διάφορες τεχνικές προγραμματισμού, με πλατφόρμες που υποστηρίζουν προσαρμοσμένους schedulers, και η ανάλυση εστιάζει στην αξιολόγηση των τεχνικών και την επίδρασή τους. Μέσα από συστηματική ανασκόπηση της υπάρχουσας επιστημονικής βιβλιογραφίας και των σχετικών άρθρων, το ερευνητικό ερώτημα RQ2 στοχεύει στην εις βάθος καταγραφή και ανάλυση των βέλτιστων πρακτικών που έχουν αναπτυχθεί στον τομέα της scheduling-aware ενορχήστρωσης εμπορευματοκιβωτίων (containers) στο συνεχές Cloud-to-Thing. Επιπλέον, επιδιώκει την επισημάνση και την κριτική αξιολόγηση των μειονεκτημάτων και των περιορισμών που χαρακτηρίζουν τις υφιστάμενες λύσεις, αναδεικνύοντας τις αδυναμίες τους στην αντιμετώπιση των σύνθετων προκλήσεων του συγκεκριμένου πεδίου. Τέλος, ένα κρίσιμο σκέλος του RQ2 είναι η αναγνώριση και η σαφής οριοθέτηση των ερευνητικών κενών που εξακολουθούν να υφίστανται και παραμένουν αναξιοποίητα, προσδιορίζοντας δυνητικές κατευθύνσεις για μελλοντική έρευνα και καινοτομία στον δυναμικά εξελισσόμενο χώρο της ενορχήστρωσης containers σε ένα περιβάλλον που εκτείνεται από το Cloud έως τις άκρες του δικτύου (Things).

4.2.2 Ελαφριά Ενορχήστρωση και Προγραμματισμός για Edge Υποδομές

Η διαχείριση containerized εφαρμογών σε edge nodes με περιορισμένους πόρους απαιτεί lightweight πλατφόρμες ενορχήστρωσης και scheduling μηχανισμούς χαμηλού overhead, λόγω των διαφορών με τα παραδοσιακά data centers που διαθέτουν άφθονους πόρους και αξιόπιστα δίκτυα. Στα edge περιβάλλοντα, η μικρή κατανάλωση πόρων, η αποδοτικότητα και η ευελιξία είναι κρίσιμες. Η ανάγκη για μικρές, αποδοτικές και ευέλικτες λύσεις ενορχήστρωσης είναι επιτακτική, καθώς οι παραδοσιακές πλατφόρμες επιβαρύνουν τους edge κόμβους. Η έρευνα και ανάπτυξη lightweight εναλλακτικών είναι ζωτικής σημασίας για το

edge computing. Η βιβλιογραφία παρουσιάζει λύσεις όπως τα K3s, FLEDGE και Docker Swarm, οι οποίες αξιολογούνται ως προς latency, κατανάλωση πόρων και καταλληλότητα για edge περιβάλλοντα, βοηθώντας στην επιλογή της κατάλληλης πλατφόρμας.

Παραδείγματα και Αξιολόγηση

Η ανάγκη για αποτελεσματική ενορχήστρωση σε περιβάλλοντα περιορισμένων πόρων, όπως το edge computing και το IoT, οδήγησε στην ανάπτυξη ελαφριών orchestrators που μπορούν να λειτουργούν αξιόπιστα ακόμα και σε συσκευές με περιορισμένη υπολογιστική ισχύ. Το **K3s**, μια ελαφριά διανομή του Kubernetes, είναι ιδανικό για τέτοια περιβάλλοντα, χάρη στο μικρό του μέγεθος, τη χαμηλή κατανάλωση ενέργειας και τη συμβατότητα με τα Kubernetes APIs. Προσφέρει επίσης ταχύτερο χρόνο εκκίνησης και σταθερή απόδοση.

Το **FLEDGE**, με ενσωματωμένο **Virtual Kubelet** και ένα ελαφρύ container runtime, επιτυγχάνει σημαντική εξοικονόμηση πόρων – έως και 60% σε ορισμένα edge σενάρια. Από την άλλη, το **Docker Swarm**, παρά την απλότητά του και τον χαμηλό χρόνο εκκίνησης, υπολείπεται σε προηγμένες δυνατότητες παρακολούθησης, ασφάλειας και ανθεκτικότητας που παρέχουν πιο ώριμες λύσεις.

Γενικά, οι ελαφριές πλατφόρμες ενορχήστρωσης διευκολύνουν την ανάπτυξη μικροϋπηρεσιών με περιορισμένες απαιτήσεις, καθιστώντας δυνατή τη λειτουργία εφαρμογών με χαμηλή καθυστέρηση και δυνατότητα επεξεργασίας σε πραγματικό χρόνο. Ωστόσο, υστερούν σε advanced features όπως προχωρημένες πολιτικές scheduling, πολιτικές ασφάλειας και ενσωματωμένη παρατηρησιμότητα.[42], [43]

4.2.3 Τεχνητή Νοημοσύνη και Προγνωστικές Τεχνικές στον Προγραμματισμό

Η συνεχής αύξηση της πολυπλοκότητας και ετερογένειας των κατανεμημένων υποδομών, σε συνδυασμό με την ανάγκη για ταχεία προσαρμογή σε δυναμικές συνθήκες, καθιστά τις παραδοσιακές μεθόδους scheduling ανεπαρκείς. Σε αυτό το πλαίσιο, αναδύονται ευφυείς τεχνικές scheduling που βασίζονται σε Machine Learning (ML), Reinforcement Learning (RL) και προγνωστικά μοντέλα.

Αντί για στατικούς ή heuristic schedulers, οι AI-powered schedulers μαθαίνουν από ιστορικά δεδομένα, εντοπίζουν μοτίβα κατανάλωσης πόρων, και βελτιώνουν συνεχώς τις αποφάσεις τοποθέτησης containers. Επιπλέον, μπορούν να προβλέψουν με ακρίβεια μελλοντικά επίπεδα φόρτου, επιτρέποντας την έγκαιρη μετακίνηση workloads ή την ενεργοποίηση μηχανισμών scaling πριν προκύψουν προβλήματα. Αυτή η προληπτική προσέγγιση scheduling συμβάλλει στη βελτιστοποίηση της χρήσης πόρων, τη μείωση των καθυστερήσεων και την ενίσχυση της αξιοπιστίας των υπηρεσιών, ιδιαίτερα σε σύνθετα, πολυδιάστατα περιβάλλοντα όπως το cloud-to-edge continuum.

Αυτόνομη λήψη αποφάσεων scheduling και placement με πολύπλοκα και πολλαπλά κριτήρια: Οι AI-based schedulers είναι σε θέση να λαμβάνουν αποφάσεις σχετικά με την τοποθέτηση και τη διαχείριση των containers λαμβάνοντας υπόψη ταυτόχρονα μια πληθώρα παραμέτρων και αντικρουόμενων στόχων. Αυτά μπορεί να περιλαμβάνουν την ελαχιστοποίηση της χρήσης πόρων, τη μεγιστοποίηση της απόδοσης των εφαρμογών (π.χ., throughput, latency), τη

βελτίωση της ενεργειακής απόδοσης, τη συμμόρφωση με συγκεκριμένες πολιτικές ασφαλείας και την ικανοποίηση συμφωνιών επιπέδου υπηρεσιών (SLAs). Η ικανότητα αυτόνομης λήψης αποφάσεων μειώνει την ανάγκη για χειροκίνητη παρέμβαση και επιτρέπει στο σύστημα να αντιδρά γρήγορα και αποτελεσματικά στις δυναμικές αλλαγές του περιβάλλοντος.[44], [45], [46]

Μοντέλο / Πλατφόρμα	Τύπος AI	Στόχος	Κριτήρια Scheduling	Πλεονεκτήματα
GAIKube	Generative AI + Reinforcement Learning	Proactive pod migration, latency prediction	Latency, resource load, prediction-based triggers	Μείωση latency spikes, σταθερή απόδοση σε edge
MOSGD	Multi-objective optimization (SGD)	Minimize execution time & energy consumption	Delay, power usage	Έως 30% βελτίωση σε latency, ενεργειακή αποδοτικότητα
Joint Autoscaling (Li et al.)	Forecast-based optimization (ILP)	Optimize pods & VMs jointly	Predicted load, cost efficiency	Συνδυασμένη βελτιστοποίηση scaling και κόστους
Survey (Survey & Assessment Paper)	Καταγραφή: RL, Fuzzy Logic, DQN, Supervised	Επισκόπηση AI-based τεχνικών scheduling	SLA, latency, CPU, memory	Κατηγοριοποίηση τεχνικών και εφαρμογών

Πίνακας 4.2.2 Συγκριτική αξιολόγηση AI-based μοντέλων scheduling και autoscaling σε edge/cloud περιβάλλοντα

Το GAIKube είναι ένα πρωτοποριακό σύστημα ευφυούς scheduling που ενσωματώνει Generative AI και Reinforcement Learning για την ανάλυση patterns επιβάρυνσης και την πρόταση προδραστικών μετεγκαταστάσεων pods σε ετερογενείς edge κόμβους μέσω lightweight containers, βελτιώνοντας τη σταθερότητα και μειώνοντας τα latency spikes. Το MOSGD, ένας αλγόριθμος στοχαστικής βελτιστοποίησης, διαχειρίζεται πολλαπλούς στόχους, ελαχιστοποιώντας ταυτόχρονα το execution delay και την ενεργειακή κατανάλωση σε edge υποδομές μέσω real-time feedback και δυναμικής προσαρμογής παραμέτρων, επιτυγχάνοντας σημαντικές βελτιώσεις σε latency και εξοικονόμηση ενέργειας. Η εργασία "Container Scheduling Techniques: A Survey and Assessment" παρουσιάζει AI-based στρατηγικές (supervised learning, RL, fuzzy logic, deep Q-learning) για scheduling containers, αναδεικνύοντας τις δυνατότητες αυτονομίας, scaling prediction και βελτίωσης της SLA compliance, κυρίως σε πειραματικά περιβάλλοντα. Το μοντέλο των Li et al. για Joint Autoscaling συνδυάζει προγνωστικά μοντέλα χρήσης πόρων με Integer Linear Programming για την ταυτόχρονη κλιμάκωση pods και VMs, στοχεύοντας σε cost-optimal deployments και predictable load responses χωρίς άμεση χρήση AI. Οι schedulers Τεχνητής Νοημοσύνης (AI) φέρνουν αυτονομία και αυξημένη αποδοτικότητα στην διαχείριση σύνθετων συστημάτων. Η ικανότητά τους να μαθαίνουν από δεδομένα και να αντιδρούν σε πραγματικό χρόνο προσφέρει πλεονεκτήματα που υπερβαίνουν τις δυνατότητες των παραδοσιακών schedulers.

- **Πρόληψη Συμφόρησης:** Οι AI schedulers προβλέπουν πιθανά σημεία συμφόρησης μέσω ανάλυσης δεδομένων με machine learning, επιτρέποντας έγκαιρη λήψη μέτρων.
- **Αναγνώριση Προτύπων:** Οι AI schedulers αναγνωρίζουν πολύπλοκα πρότυπα στην κατανάλωση πόρων και τις καθυστερήσεις, οδηγώντας σε βελτιωμένες αποφάσεις.
- **Προσαρμοστικές και Εξατομικευμένες Στρατηγικές:** Οι AI schedulers προσαρμόζουν δυναμικά τις στρατηγικές τους και προσφέρουν εξατομικευμένο scheduling, βελτιώνοντας την αποδοτικότητα και την ανταπόκριση.

Η υιοθέτηση AI schedulers αντιμετωπίζει προκλήσεις όπως η πολυπλοκότητα εκπαίδευσης μοντέλων με ποιοτικά δεδομένα, η δυσκολία γενίκευσης σε ετερογενή συστήματα και η περιορισμένη συμβατότητα με orchestrators χωρίς επεκτάσεις. Λύσεις όπως GAIKube και MOSGD δείχνουν τις δυνατότητες ενσωμάτωσης AI σε περιβάλλοντα με περιορισμένους πόρους, αλλά απαιτούνται ελαφριές υλοποιήσεις και προσαρμογή στις ιδιαιτερότητες κάθε περιβάλλοντος για αποδοτική λειτουργία.[26]

4.2.4 Υβριδική Τοποθέτηση Unikernels και Containers

Σε ένα edge περιβάλλον, όπου οι υπολογιστικοί πόροι είναι περιορισμένοι και οι απαιτήσεις για υψηλή απόδοση και χαμηλή καθυστέρηση αυξημένες, η επιλογή του κατάλληλου τύπου virtualization είναι κρίσιμη και άρρηκτα συνδεδεμένη με τις ανάγκες του κάθε φόρτου εργασίας. Στο πλαίσιο αυτό, αναδεικνύεται μια καινοτόμος προοπτική: η υιοθέτηση μιας υβριδικής ενορχήστρωσης που συνδυάζει containers και unikernels. Η συνδυαστική αυτή μέθοδος επιδιώκει να αξιοποιήσει τα ξεχωριστά πλεονεκτήματα και των δύο μοντέλων virtualization, επιτρέποντας την ανάπτυξη και εκτέλεση εφαρμογών με βελτιστοποιημένη απόδοση, αυξημένη ασφάλεια και μειωμένες απαιτήσεις πόρων στο δυναμικό και συχνά απρόβλεπτο edge περιβάλλον. Η ενορχήστρωση μιας τέτοιας υβριδικής αρχιτεκτονικής απαιτεί προηγμένες τεχνικές διαχείρισης και κατανομής πόρων, λαμβάνοντας υπόψη τις ανάγκες κάθε τύπου φόρτου εργασίας και τις δυνατότητες της υποκείμενης υποδομής.

Τα Unikernels αποτελούν μια πρωτοποριακή προσέγγιση για την κατασκευή και εκτέλεση εφαρμογών στο cloud και σε ενσωματωμένα συστήματα. Ουσιαστικά, πρόκειται για εξαιρετικά ελαφριές, εξειδικευμένες εικονικές μηχανές (VMs) που περιλαμβάνουν μόνο τον απαραίτητο κώδικα για τη λειτουργία μιας συγκεκριμένης εφαρμογής, συμπεριλαμβανομένων των αναγκαιών βιβλιοθηκών και ενός ελαχιστοποιημένου πυρήνα λειτουργικού συστήματος. Σε αντίθεση με τις παραδοσιακές, σύνθετες εικονικές μηχανές που περιέχουν πολλές υπηρεσίες και διεργασίες συχνά περιττές για την εκάστοτε εφαρμογή, η προσέγγιση των unikernels οδηγεί σε σημαντικά πλεονεκτήματα. Ένα κύριο πλεονέκτημα είναι το μικρό αποτύπωμα σε πόρους όπως μνήμη RAM και CPU. Περιέχοντας μόνο τον απολύτως απαραίτητο κώδικα, οι απαιτήσεις τους σε υπολογιστικούς πόρους είναι σημαντικά μειωμένες σε σύγκριση με τις συμβατικές εικονικές μηχανές. Αυτό συνεπάγεται αυξημένη πυκνότητα εφαρμογών σε έναν φυσικό διακομιστή, μείωση του λειτουργικού κόστους και βελτίωση της ενεργειακής απόδοσης. Ένα επιπλέον σημαντικό πλεονέκτημα είναι ο πολύ γρήγορος χρόνος εκκίνησης, της τάξης των χιλιοστών του δευτερολέπτου. Λόγω του ελαχιστοποιημένου μεγέθους και της εξειδικευμένης φύσης τους, τα unikernels εκκινούνται σχεδόν άμεσα. Αυτό είναι ιδιαίτερα χρήσιμο σε cloud περιβάλλοντα όπου η γρήγορη δημιουργία και καταστροφή instances είναι ζωτικής σημασίας, καθώς και σε σενάρια που απαιτούν άμεση ανταπόκριση σε αιτήματα. Επιπλέον, τα unikernels παρέχουν υψηλό επίπεδο απομόνωσης. Κάθε unikernel

περιέχει μόνο μία εφαρμογή και τις απολύτως απαραίτητες βιβλιοθήκες και λειτουργίες του πυρήνα, μειώνοντας σημαντικά την επιφάνεια επίθεσης. Σε περίπτωση παραβίασης ενός unikernel, οι επιπτώσεις περιορίζονται στην συγκεκριμένη εικονική μηχανή, χωρίς να επηρεάζονται άλλες εφαρμογές που εκτελούνται στον ίδιο φυσικό διακομιστή, ενισχύοντας έτσι την ασφάλεια των εφαρμογών και των δεδομένων.

Συνολικά, τα unikernels αποτελούν μια ριζοσπαστική προσέγγιση στην εικονικοποίηση, προσφέροντας σημαντικά οφέλη όσον αφορά την αποδοτικότητα των πόρων, την ταχύτητα εκκίνησης και την ασφάλεια, καθιστώντας τα μια ελκυστική επιλογή για ένα ευρύ φάσμα εφαρμογών και περιβαλλόντων. Από την άλλη πλευρά, τα containers παρέχουν πληθώρα εργαλείων, καλύτερη εντοπιστικότητα (π.χ., μέσω Kubernetes) και υποστήριξη για πιο σύνθετους φόρτους εργασίας. Η υβριδική διαχείριση αυτών των δύο τύπων virtualization επιτρέπει την εφαρμογή adaptive scheduling policies, με βάση το προφίλ του κάθε task (π.χ., ευαισθησία στην καθυστέρηση, κατανάλωση πόρων).

Στην ερευνητική εργασία "Edge System Design Using Containers and Unikernels for IoT Applications" των Kaiser et al. (2023) αναλύεται και επιδεικνύεται μια ολοκληρωμένη αρχιτεκτονική για υβριδική εντοπιστικότητα σε συστήματα edge, προσαρμοσμένα για τις απαιτήσεις των IoT εφαρμογών. Η καινοτομία της προσέγγισης έγκειται στην ταυτόχρονη αξιοποίηση της ευελιξίας των containers και της αποδοτικότητας των unikernels, προσφέροντας μια λύση που μπορεί να αντιμετωπίσει το ευρύ φάσμα των υπολογιστικών αναγκών που προκύπτουν σε ένα περιβάλλον IoT.

Πιο συγκεκριμένα, η εργασία εμβαθύνει στην επιλογή της τεχνολογίας containerization για την εκτέλεση υπολογιστικά απαιτητικών διεργασιών, όπως η εκπαίδευση και η συμπερίληψη μοντέλων Τεχνητής Νοημοσύνης. Τα containers, με την ικανότητά τους να παρέχουν ένα απομονωμένο και συνεπές περιβάλλον εκτέλεσης, καθίστανται ιδανικά για workloads που χαρακτηρίζονται από υψηλές απαιτήσεις σε πόρους και σύνθετες εξαρτήσεις. Από την άλλη πλευρά, τα unikernels προτείνονται για την υλοποίηση ελαφριών και επαναλαμβανόμενων εργασιών, όπως η συνεχής συλλογή και επεξεργασία δεδομένων από αισθητήρες. Η μινιμαλιστική φύση των unikernels, που περιλαμβάνουν μόνο τα απαραίτητα στοιχεία του λειτουργικού συστήματος για την εκτέλεση μιας συγκεκριμένης εφαρμογής, οδηγεί σε μειωμένη κατανάλωση πόρων, αυξημένη ασφάλεια και ταχύτερους χρόνους εκκίνησης, χαρακτηριστικά ιδιαίτερα επιθυμητά για εφαρμογές IoT με περιορισμένους πόρους.

Ένα θεμελιώδες στοιχείο της παρούσας εργασίας εστιάζει στην ανάπτυξη ενός εξελιγμένου μηχανισμού αυτόματης επιλογής, ικανού να προσδιορίζει δυναμικά τον πλέον ενδεδειγμένο τύπο εικονικοποίησης – είτε πρόκειται για containers είτε για unikernels – για την εκτέλεση κάθε συγκεκριμένου φόρτου εργασίας (workload). Η κρίσιμη αυτή απόφαση δεν λαμβάνεται στατικά, αλλά βασίζεται σε μια ενδεδειγμένη και συνεχή διαδικασία ανάλυσης προφίλ των διαφόρων εργασιών που αναπτύσσονται στο σύστημα. Η εν λόγω διαδικασία profiling αποτελεί ένα ζωτικής σημασίας βήμα, καθώς επιτρέπει στο σύστημα να αποκτήσει μια βαθιά κατανόηση των εγγενών απαιτήσεων κάθε διεργασίας, όπως οι ανάγκες της σε υπολογιστική ισχύ (CPU), κατανάλωση μνήμης (RAM), καθώς και άλλους κρίσιμους πόρους του συστήματος. Με βάση τα λεπτομερή αυτά προφίλ, το σύστημα στη συνέχεια προχωρά στην ευφυή ανάπτυξη κάθε εργασίας στο πλέον κατάλληλο και αποδοτικό περιβάλλον εκτέλεσης. Αυτή η δυναμική και ευφυής κατανομή των φόρτων εργασίας οδηγεί σε μια σημαντική βελτιστοποίηση της συνολικής απόδοσης του edge συστήματος, καθώς και σε μια αισθητά

πιο αποτελεσματική χρήση των περιορισμένων διαθέσιμων πόρων, μεγιστοποιώντας έτσι την αποδοτικότητα και την αξιοπιστία του συστήματος στο σύνολό του. Για την επίτευξη της υβριδικής ενορχήστρωσης, η ερευνητική ομάδα αξιοποίησε μια σειρά από σύγχρονα εργαλεία διαχείρισης containers και εφαρμογών. Το K3s, μια ελαφριά έκδοση του Kubernetes, επιλέχθηκε για τη διαχείριση των containers σε περιβάλλοντα με περιορισμένους πόρους. Το KubeEdge, μια επέκταση του Kubernetes για edge computing, χρησιμοποιήθηκε για τη διαχείριση των κόμβων edge και την επικοινωνία τους με το κεντρικό cloud. Επιπλέον, το Nomad, ένας ευέλικτος orchestrator που υποστηρίζει πολλαπλούς τύπους workloads (συμπεριλαμβανομένων containers και unikernels), χρησιμοποιήθηκε για τη συνολική διαχείριση και τον προγραμματισμό των εφαρμογών στους διαθέσιμους κόμβους.

Η προτεινόμενη αρχιτεκτονική υλοποιήθηκε και αξιολογήθηκε πειραματικά χρησιμοποιώντας πραγματικό υλικό, συγκεκριμένα κόμβους Raspberry Pi και συσκευές που βασίζονται σε ARM. Η επιλογή αυτού του υλικού δεν ήταν τυχαία, καθώς στοχεύει στην ανάδειξη της άμεσης πρακτικής εφαρμογής της προτεινόμενης λύσης σε συνήθη σενάρια edge computing, όπου συσκευές με περιορισμένους πόρους είναι κοινές. Η εργασία παρουσιάζει αναλυτικά τα αποτελέσματα των ενδεδειγμένων πειραμάτων που διεξήχθησαν, τα οποία αναλύονται σε βάθος, αναδεικνύοντας τα σημαντικά πλεονεκτήματα της υβριδικής προσέγγισης ενορχήστρωσης που προτείνεται. Συγκεκριμένα, η απόδοση της προτεινόμενης μεθόδου συγκρίνεται με αυτήν παραδοσιακών προσεγγίσεων που βασίζονται αποκλειστικά στη χρήση containers ή εικονικών μηχανών, φανερώνοντας την υπεροχή της.

Η παρούσα εργασία πραγματεύεται την προοπτική δημιουργίας αποδοτικών, ευέλικτων και ασφαλών edge συστημάτων για IoT, μέσω της συνδυαστικής χρήσης containers και unikernels. Κεντρικό στοιχείο είναι η υποστήριξη από κατάλληλα εργαλεία ενορχήστρωσης και έναν έξυπνο μηχανισμό αυτόματης επιλογής του βέλτιστου τύπου εικονικοποίησης για κάθε φόρτο εργασίας. Η εργασία αναλύει σε βάθος την προτεινόμενη αρχιτεκτονική, εστιάζοντας στον ρόλο κάθε επιμέρους στοιχείου στην επίτευξη των συνολικών πλεονεκτημάτων. Τα αποτελέσματα της έρευνας κατέδειξαν μια αξιοσημείωτη διαφορά στην κατανάλωση μνήμης μεταξύ των unikernels και των containers κατά την εκτέλεση πανομοιότυπων φόρτων εργασίας. Συγκεκριμένα, παρατηρήθηκε ότι οι unikernels επέδειξαν μείωση στην κατανάλωση μνήμης που έφτανε έως και το 36% σε σύγκριση με τα containers. Αυτή η σημαντική εξοικονόμηση πόρων υπογραμμίζει την ενδεχόμενη βελτίωση της αποδοτικότητας των υπολογιστικών συστημάτων μέσω της χρήσης unikernels, ειδικά σε περιβάλλοντα με περιορισμένους πόρους ή υψηλές απαιτήσεις κλιμάκωσης.

Περαιτέρω, η ανάλυση της απόκρισης των lightweight tasks αποκάλυψε βελτιώσεις που άγγιξαν το 25%. Αυτή η βελτίωση αποδίδεται κυρίως στη μείωση του χρόνου εκκίνησης (startup time) και του χρόνου προγραμματισμού (scheduling time) που χαρακτηρίζουν την αρχιτεκτονική των unikernels. Η ταχύτερη εκκίνηση και ο πιο αποδοτικός προγραμματισμός έχουν άμεσο θετικό αντίκτυπο στην απόδοση εφαρμογών που αποτελούνται από πολλές μικρές και γρήγορα εκτελούμενες εργασίες. Επιπροσθέτως, η ενσωμάτωση των unikernels με το K3s, μια ελαφριά διανομή του Kubernetes, επέτρεψε την υλοποίηση ενός ολοκληρωμένου κύκλου ζωής ενορχήστρωσης. Αυτό περιλαμβάνει κρίσιμες λειτουργίες όπως η δυναμική κλιμάκωση (scaling), η συνεχής παρακολούθηση (monitoring) της υγείας και της απόδοσης των εφαρμογών, καθώς και η αυτόματη ανάκαμψη σε περίπτωση αστοχίας (failover). Παρά την προσθήκη αυτών των εξελιγμένων δυνατοτήτων ενορχήστρωσης, το σύστημα διατήρησε την ευελιξία επιλογής ενός task-specific runtime χωρίς την εισαγωγή σημαντικού

υπολογιστικού ή διαχειριστικού κόστους (overhead). Η ικανότητα προσαρμογής του περιβάλλοντος εκτέλεσης σε συγκεκριμένες ανάγκες κάθε εργασίας, ελαχιστοποιώντας παράλληλα την επιβάρυνση του συστήματος, αναδεικνύει ένα σημαντικό πλεονέκτημα της προτεινόμενης αρχιτεκτονικής.

Η υβριδική προσέγγιση ενορχήστρωσης που συνδυάζει containers και unikernels αναδύεται ως μια ιδιαίτερα ευέλικτη και προσαρμοστική λύση, ειδικά σχεδιασμένη για τις απαιτήσεις των IoT και edge περιβαλλόντων. Η ικανότητα της προσέγγισης αυτής να λαμβάνει υπόψη το συγκεκριμένο πλαίσιο (context-aware) κάθε φόρτου εργασίας την καθιστά ιδιαίτερα ελκυστική για περιπτώσεις χρήσης όπου οι ανάγκες σε απόδοση, πόρους και ετερογένεια υλικού ποικίλουν σημαντικά. Ένα από τα πρωταρχικά οφέλη της υβριδικής ενορχήστρωσης είναι η δυναμική αντιστοίχιση μεταξύ του φόρτου εργασίας και του καταλληλότερου τύπου εκτέλεσης. Αυτό σημαίνει ότι εφαρμογές με αυστηρές απαιτήσεις χαμηλής καθυστέρησης (latency-sensitive) ή μικρού μεγέθους και σύντομης διάρκειας μπορούν να εκτελεστούν αποδοτικότερα ως unikernels, αξιοποιώντας τον μειωμένο χρόνο εκκίνησης και το μικρότερο αποτύπωμα μνήμης. Αντίστοιχα, πιο σύνθετες και μακροχρόνιες εφαρμογές μπορούν να διαχειριστούν αποτελεσματικότερα μέσω της τεχνολογίας των containers, επωφελούμενες από την ευρεία υποστήριξη και τα ώριμα εργαλεία διαχείρισης. Η δυνατότητα αυτή λεπτομερούς ρύθμισης της εκτέλεσης ανάλογα με τις εκάστοτε ανάγκες οδηγεί σε μια πιο αποδοτική και ευέλικτη διαχείριση των εφαρμογών.

Η **υβριδική ενορχήστρωση**, που συνδυάζει την εκτέλεση εφαρμογών ως **unikernels** ή **containers** ανάλογα με τις απαιτήσεις, προσφέρει σημαντικά πλεονεκτήματα σε περιβάλλοντα με περιορισμένους πόρους, όπως το IoT και το edge computing. Οι **unikernels**, χάρη στο ελαφρύ τους αποτύπωμα και τη χαμηλή κατανάλωση ενέργειας, συμβάλλουν στη βέλτιστη αξιοποίηση πόρων όπως η μνήμη και η CPU. Παράλληλα, η δυνατότητα επιλογής μεταξύ διαφορετικών runtime περιβαλλόντων επιτρέπει την ευέλικτη υποστήριξη ετερογενούς φόρτου εργασίας και υλικού. Ωστόσο, η υλοποίηση υβριδικών αρχιτεκτονικών ενορχήστρωσης δεν είναι χωρίς προκλήσεις. Η περιορισμένη υποστήριξη των **unikernels** από τις κυρίαρχες πλατφόρμες ενορχήστρωσης, η ανάγκη για εξειδικευμένα εργαλεία και η αυξημένη πολυπλοκότητα στη διαχείριση του κύκλου ζωής των εφαρμογών αποτελούν σημαντικά εμπόδια. Επιπλέον, η συνύπαρξη διαφορετικών χρόνων εκτέλεσης δυσχεραίνει τη διατήρηση της συνέπειας και τον ομοιόμορφο έλεγχο του συστήματος.

Παρά τις προκλήσεις, η υβριδική ενορχήστρωση αναδεικνύεται ως πολλά υποσχόμενη επιλογή για την ανάπτυξη και διαχείριση αποδοτικών και ευέλικτων edge εφαρμογών. Η περαιτέρω ερευνητική δραστηριότητα κρίνεται αναγκαία για την υπέρβαση των τεχνικών περιορισμών και την αξιοποίηση του πλήρους δυναμικού της. [47], [48]

4.2.5 Αποκεντρωμένος Προγραμματισμός και CRDTs στο Kubernetes

Η ανάγκη για αποκεντρωμένη ενορχήστρωση αναδεικνύεται ως θεμελιώδης λίθος στο εξελισσόμενο παράδειγμα του Cloud-to-Thing Continuum. Οι υποδομές edge και fog, εγγενώς κατανεμημένες και ετερογενείς, παρουσιάζουν μοναδικές προκλήσεις που καθιστούν τις παραδοσιακές προσεγγίσεις ενορχήστρωσης ανεπαρκείς. Χαρακτηριστικά όπως η ασταθής συνδεσιμότητα δικτύου, το περιορισμένο εύρος ζώνης και η γεωγραφική διασπορά μεγάλης κλίμακας απαιτούν ευέλικτες και αυτόνομες λύσεις διαχείρισης.

Η παραδοσιακή αρχιτεκτονική του Kubernetes, ένα ευρέως υιοθετημένο σύστημα ενορχήστρωσης container, βασίζεται σε ένα κεντρικό control plane για τη λήψη αποφάσεων και τη διαχείριση της κατάστασης του συστήματος. Επιπλέον, η εξάρτηση από ένα strongly consistent etcd datastore για την αποθήκευση κρίσιμων δεδομένων εισάγει εγγενείς περιορισμούς σε περιβάλλοντα edge και fog. Η απώλεια σύνδεσης με το κεντρικό control plane μπορεί να οδηγήσει σε αδυναμία αυτόνομης λειτουργίας των κόμβων, επηρεάζοντας την ανθεκτικότητα και τη διαθεσιμότητα των εφαρμογών. Επιπλέον, η συγκέντρωση της λογικής ελέγχου σε ένα κεντρικό σημείο μπορεί να δημιουργήσει σημεία συμφόρησης (bottlenecks) σε περιβάλλοντα με εκατοντάδες ή χιλιάδες καταναμημένους κόμβους, υποβαθμίζοντας την απόδοση και την επεκτασιμότητα.

Για την ουσιαστική αντιμετώπιση αυτών των εγγενών περιορισμών και την πλήρη αξιοποίηση των δυνατοτήτων του Cloud-to-Thing Continuum, η ερευνητική κοινότητα έχει στραφεί στην ανάπτυξη αποκεντρωμένων μοντέλων scheduling και state management. Αυτές οι προσεγγίσεις επιδιώκουν να καταλείψουν τη λογική ελέγχου και τις ευθύνες λήψης αποφάσεων σε όλο το καταναμημένο σύστημα, επιτρέποντας στους κόμβους να λειτουργούν πιο αυτόνομα και ανθεκτικά στις διακοπές δικτύου. Βασίζονται σε μια ποικιλία καινοτόμων τεχνικών, όπως:

- Eventual consistency
- Conflict-Free Replicated Data Types (CRDTs)
- Peer-to-peer synchronization
- Virtual Kubelets and lightweight agents

Το "Rearchitecting Kubernetes for the Edge" άρθρο τονίζει τους περιορισμούς του Kubernetes για edge computing, κυρίως λόγω της εξάρτησης από το etcd και τη συνέπεια που βασίζεται σε quorum. Αυτό οδηγεί σε μειωμένη διαθεσιμότητα, αυξημένο latency και δυσκολία στην αυτονομία των κόμβων σε ασταθή δίκτυα. Ως λύση, προτείνεται η αντικατάσταση του etcd με ένα καταναμημένο datastore βασισμένο σε CRDTs για ανεξάρτητη επεξεργασία δεδομένων χωρίς συγκρούσεις.

Η υιοθέτηση ενός CRDT datastore στο Kubernetes για το edge προσφέρει αυτόνομη λειτουργία κόμβων με τοπικές αποφάσεις και αλλαγές που συγχρονίζονται αργότερα, βελτιώνοντας την αξιοπιστία σε ασταθή δίκτυα. Ο lazy συγχρονισμός των CRDTs διασφαλίζει συνέπεια χωρίς χειροκίνητη επίλυση. Η κατάργηση του quorum μειώνει σημαντικά το scheduling latency (έως 60%), καθιστώντας την αρχιτεκτονική πιο γρήγορη και αποδοτική. Συνολικά, η εργασία αναδεικνύει μια κρίσιμη πρόκληση για την επέκταση του Kubernetes στο edge και προτείνει μια καινοτόμο λύση βασισμένη στην αντικατάσταση του etcd με ένα distributed, CRDT-based datastore. Τα πλεονεκτήματα αυτής της προσέγγισης, όπως η αυτόνομη λειτουργία των κόμβων, ο lazy synchronization και η μειωμένη λανθάνουσα κατάσταση, καθιστούν την εργασία αυτή θεμελιώδη για την κατανόηση και την ανάπτυξη του Kubernetes σε περιβάλλοντα edge computing. Η ποσοτικοποίηση της μείωσης του scheduling latency κατά 60% παρέχει ισχυρή εμπειρική υποστήριξη για την αποτελεσματικότητα της προτεινόμενης λύσης. [30]

Κάθε κόμβος στο cluster διατηρεί την ικανότητα να λαμβάνει αυτόνομα αποφάσεις σχετικά με τον προγραμματισμό των pods. Αυτή η δυνατότητα βασίζεται στην αξιοποίηση των τοπικών πόρων και των τρεχουσών δυνατοτήτων του κόμβου, επιτρέποντας την άμεση

ανταπόκριση στις ανάγκες ανάπτυξης εφαρμογών χωρίς την απαίτηση για συνεχή επικοινωνία με ένα κεντρικό σημείο ελέγχου. Ο τοπικός προγραμματισμός ενισχύει την ανθεκτικότητα και τη διαθεσιμότητα του συστήματος, καθώς η ικανότητα εκτέλεσης εργασιών δεν εξαρτάται από την κατάσταση ή τη συνδεσιμότητα ενός κεντρικού διαχειριστή. Ωστόσο, για τη διατήρηση μιας συνεκτικής εικόνας του cluster σε όλους τους κόμβους, υιοθετείται η αρχή της eventual propagation. Αυτό σημαίνει ότι οι αποφάσεις τοπικού προγραμματισμού και οι αλλαγές στην κατάσταση του cluster δεν διαδίδονται άμεσα σε όλους τους κόμβους. Αντίθετα, η διάδοση πραγματοποιείται περιοδικά, μέσω μηχανισμών συγχρονισμού. Η συχνότητα και ο τρόπος αυτού του συγχρονισμού μπορούν να ρυθμιστούν ανάλογα με τις απαιτήσεις του περιβάλλοντος και τις ανάγκες για συνέπεια.

Για την επιτυχή διάδοση αλλαγών, το μοντέλο χρησιμοποιεί CRDTs για την αυτόματη συγχώνευση ταυτόχρονων αλλαγών χωρίς συγκρούσεις, εξασφαλίζοντας τελική συνέπεια κατά τον περιοδικό συγχρονισμό των κόμβων, απλοποιώντας τη διαχείριση του cluster και ενισχύοντας την αξιοπιστία. Εναλλακτικά, το EdgeVPN.io προσφέρει λογική αποκέντρωση μέσω ενός peer-to-peer overlay VPN, διασυνδέοντας κατανεμημένα Kubernetes nodes που βρίσκονται σε απομονωμένα δίκτυα, παρακάμπτοντας τους περιορισμούς των παραδοσιακών δικτύων και ενισχύοντας την ανθεκτικότητα χωρίς κεντρικό σημείο αποτυχίας. Το EdgeVPN.io παρέχει την υποδομή για αποκεντρωμένους orchestrators και virtual kubelets, επιτρέποντας την ανάπτυξη κατανεμημένων instances του control plane για αυξημένη ανθεκτικότητα και επεκτασιμότητα, καθώς και τη διαχείριση φόρτου εργασίας σε ετερογενή περιβάλλοντα μέσω ελαφρών αναπαραστάσεων κόμβων που συνδέονται με ένα κεντρικό control plane. Συνολικά, το μοντέλο συνδυάζει τοπικό προγραμματισμό με eventual consistency μέσω CRDTs για αυτόνομη διαχείριση, ενώ το EdgeVPN.io προσφέρει μια ισχυρή εναλλακτική για διασύνδεση απομονωμένων κόμβων και υποστήριξη αποκεντρωμένων αρχιτεκτονικών orchestration, αυξάνοντας την ευελιξία και την επεκτασιμότητα των Kubernetes σε πολύπλοκα περιβάλλοντα.

Η μετάβαση από το παραδοσιακό, συγκεντρωτικό μοντέλο Kubernetes σε αρχιτεκτονικές που βασίζονται σε αποκεντρωμένα συστήματα scheduling αντιπροσωπεύει ένα σημαντικό εξελικτικό άλμα στην ενορχήστρωση εφαρμογών, ειδικά σε περιβάλλοντα edge που χαρακτηρίζονται από διανομή και αυτονομία. Αυτή η αλλαγή παραδείγματος υπόσχεται να ξεκλειδώσει νέες δυνατότητες και να επιλύσει εγγενείς περιορισμούς που παρουσιάζουν οι κεντρικές λύσεις σε τέτοια σενάρια.

Τα αποκεντρωμένα μοντέλα ενορχήστρωσης προσφέρουν σημαντικά πλεονεκτήματα σε σχέση με τις κεντρικές υλοποιήσεις Kubernetes:

- **Αυξημένη Λειτουργική Αυτονομία:** Οι κόμβοι μπορούν να συνεχίσουν τη λειτουργία τους ακόμα και χωρίς συνεχή σύνδεση με τον κεντρικό control plane. Αυτό είναι ιδιαίτερα σημαντικό σε edge περιβάλλοντα με ασταθείς δικτυακές συνδέσεις, εξασφαλίζοντας αδιάλειπτη διαθεσιμότητα εφαρμογών και υπηρεσιών.
- **Μείωση Καθυστερήσεων Προγραμματισμού:** Η αποκέντρωση εξαλείφει την ανάγκη για quorum locks κατά τον προγραμματισμό νέων pods. Κάθε κόμβος αποφασίζει τοπικά και ασύγχρονα, μειώνοντας σημαντικά τον χρόνο αναμονής και βελτιώνοντας την ανταπόκριση του συστήματος.
- **Ανθεκτικότητα και Συνέπεια μέσω Ασύγχρονης Συγχώνευσης:** Χρησιμοποιώντας τεχνολογίες όπως τα CRDTs, οι κόμβοι λειτουργούν ανεξάρτητα και συγχωνεύουν τις καταστάσεις τους ασύγχρονα. Αυτός ο μηχανισμός εξασφαλίζει συνέπεια και

ανθεκτικότητα, καθώς οι αλλαγές ενσωματώνονται ακόμα και μετά από προσωρινή αποσύνδεση κόμβων, χωρίς απώλεια δεδομένων.

Παρόλα αυτά, η υλοποίηση αποκεντρωμένων συστημάτων ενορχήστρωσης παρουσιάζει ορισμένες τεχνικές προκλήσεις και απαιτεί προσεκτική εξέταση:

- **Διαχείριση και Κόστος CRDTs σε Μεγάλη Κλίμακα:** Η αποτελεσματική διαχείριση και αποθήκευση μεγάλων όγκων δεδομένων CRDTs μπορεί να είναι πολύπλοκη και απαιτητική σε πόρους, καθιστώντας την βελτιστοποίηση τους για μεγάλα datasets μια σημαντική πρόκληση.
- **Ενσωμάτωση με το Υπάρχον Οικοσύστημα Kubernetes:** Η μετάβαση απαιτεί τροποποιήσεις στα Kubernetes APIs, στον scheduler και στον controller manager. Η διασφάλιση συμβατότητας και ομαλής συνύπαρξης με υπάρχουσες εφαρμογές και εργαλεία είναι κρίσιμη για την ευρεία υιοθέτηση.
- **Εξάρτηση Απόδοσης από Workloads και Τοπολογία:** Η αποδοτικότητα δεν είναι δεδομένη και εξαρτάται από τη φύση των workloads (π.χ., ανάγκη για συχνή επικοινωνία και ισχυρή συνέπεια) και τη σταθερότητα του δικτύου. Συχνές αλλαγές στην τοπολογία μπορούν να επηρεάσουν την απόδοση της ασύγχρονης συγχώνευσης.

Συμπεράσματα και Προοπτικές:

Οι αρχιτεκτονικές πρωτοβουλίες για αποκεντρωμένο Kubernetes scheduling αντιπροσωπεύουν μια μετάβαση προς συστήματα ενορχήστρωσης επόμενης γενιάς για το edge computing. Για την υποστήριξη του Cloud-to-Thing Continuum, απαιτούνται αποκεντρωμένα, fault-tolerant και consistency-aware συστήματα ικανά να λειτουργούν αξιόπιστα σε περιβάλλοντα με περιορισμένη συνδεσιμότητα, να ανταποκρίνονται γρήγορα και να διατηρούν τη συνέπεια δεδομένων. Η αντιμετώπιση τεχνικών προκλήσεων και η καινοτομία στη διαχείριση CRDTs και την ενσωμάτωση με το Kubernetes είναι κρίσιμες για την επιτυχή ανάπτυξη αυτών των αρχιτεκτονικών.

4.2.6 Προσαρμοστική Ενορχήστρωση σε SBCs με Επίγνωση Πόρων

Η ευρεία χρήση οικονομικών SBCs όπως τα Raspberry Pi έχει οδηγήσει στην ανάπτυξη edge υποδομών για κατανομημένα συστήματα IoT με περιορισμένους πόρους. Ωστόσο, η ενορχήστρωση containerized workloads σε αυτά τα συστήματα παρουσιάζει τεχνικές προκλήσεις λόγω περιορισμένων πόρων (CPU, RAM, αποθηκευτικός χώρος), αρχιτεκτονικής ετερογένειας, περιορισμένης συνδεσιμότητας δικτύου, διαχείρισης κύκλου ζωής, θεμάτων ασφάλειας και ενεργειακών περιορισμών. Οι τεχνικές resource-aware orchestration στοχεύουν στην αντιμετώπιση αυτών των προκλήσεων μέσω της βελτιστοποίησης της χρήσης πόρων.[25]

Μοντέλο Πλατφόρμα	Ορχηστρωτής	Resource-aware Τεχνικές	Κύρια Ευρήματα / Οφέλη
Yakubov & Hästbacka (2025)	k3s, KubeEdge	k0s, CPU/Memory limits, light pods	Γραμμικό scaling έως 60+ pods, καλή απόδοση σε 4 GB RAM

Gupta & Nahrstedt (2021)	Docker	Host networking, tuning quotas	Predictable behavior, low latency για light workloads
Goethals et al. (FLEDGE)	Containerd + Virtual Kubelet	Minimal runtime, no CNI, low memory	Μείωση πόρων έως 60%, πλήρης συμβατότητα με K8s APIs
Performance Eval. of Orchestrators (Mohamed et al.)	K3s, Swarm	K8s, Monitoring CPU, memory profiles	K3s πιο αποτελεσματικό σε ARM-based devices από Kubernetes και Swarm

Η ερευνητική εργασία των Yakubov και Hästbacka (2025) εμβαθύνει σε μια συγκριτική αξιολόγηση διαφόρων διανομών Kubernetes, με ιδιαίτερη έμφαση στην απόδοσή τους σε Single-Board Computers (SBCs). Η μελέτη εξέτασε τις επιδόσεις των k3s, k8s, k0s, KubeEdge και OpenYurt σε περιβάλλοντα με περιορισμένους πόρους, αποκαλύπτοντας αξιοσημείωτες διαφορές μεταξύ τους. Το k3s αναδείχθηκε ως η πιο αξιόπιστη λύση, επιδεικνύοντας γραμμική επεκτασιμότητα, ένα κρίσιμο χαρακτηριστικό για δυναμικά περιβάλλοντα με μεταβαλλόμενο φόρτο εργασίας. Επιπλέον, η διανομή k3s παρουσίασε χαμηλή κατανάλωση μνήμης και CPU, καθιστώντας την ιδανική για συσκευές με περιορισμένους πόρους. Ένα σημαντικό εύρημα της έρευνας ήταν η ικανότητα του k3s να υποστηρίξει πάνω από 60 pods ακόμη και σε SBCs. Οι ερευνητές τονίζουν ότι, με την κατάλληλη διαμόρφωση των ορίων πόρων, είναι εφικτή η ενορχήστρωση σύνθετων αρχιτεκτονικών microservices ακόμη και σε συστήματα με μόλις 4 GB RAM. Αυτό αποτελεί ένα σημαντικό πλεονέκτημα για σενάρια edge computing και IoT συσκευές, όπου η αποδοτική χρήση των πόρων είναι πρωταρχικής σημασίας. Η σταθερότητα και η χαμηλή κατανάλωση πόρων του k3s το καθιστούν μια ελκυστική επιλογή για την ανάπτυξη καταναμεμένων εφαρμογών σε περιβάλλοντα με περιορισμούς.[49]

Σε μια άλλη ερευνητική εργασία, οι Gupta και Nahrstedt (αναφοράς απουσιάζει στο παρεχόμενο κείμενο) επικεντρώθηκαν στην ανάλυση των επιδόσεων των Docker containers σε ένα Raspberry Pi 3, ένα ευρέως χρησιμοποιούμενο SBC σε εφαρμογές edge computing. Η μελέτη διεξήγαγε benchmarking με φόρτους εργασίας πραγματικού χρόνου, όπως αναγνώριση προσώπου, κρυπτογράφηση και clustering, για την αξιολόγηση της συμπεριφοράς των containers σε ρεαλιστικά σενάρια.

Τα κυριότερα ευρήματα της έρευνας περιλαμβάνουν τα εξής:

- Χρήση host networking αντί bridge networking μείωσε σημαντικά καθυστέρηση και overhead, ωφελώντας ευαίσθητες στην καθυστέρηση edge εφαρμογές.
- Το Docker έδειξε γραμμική συμπεριφορά στον προγραμματισμό containers, με κάποιο overhead σε εργασίες υψηλών απαιτήσεων.
- Κατάλληλη ρύθμιση ορίων πόρων εξασφαλίζει σταθερή απόδοση χωρίς υπερφόρτωση, κρίσιμο για αξιόπιστη λειτουργία edge εφαρμογών.

Συνολικά, τα ευρήματα αυτών των ερευνών παρέχουν πολύτιμες γνώσεις για την αποτελεσματική ανάπτυξη και διαχείριση containerized εφαρμογών σε περιβάλλοντα edge computing με περιορισμένους πόρους.[50]

Η ερευνητική ομάδα των Goethals et al. πρότεινε το FLEDGE, έναν ελαφρύ orchestrator ειδικά σχεδιασμένο για συσκευές edge με περιορισμένους πόρους. Η καινοτομία του έγκειται στην επιλογή του Containerd ως runtime container αντί του πλήρους Kubernetes runtime (kubelet), γεγονός που επιφέρει μείωση του αποτυπώματος έως και 60% σε ARM-based SBCs, καθιστώντας το FLEDGE ιδανικό για περιβάλλοντα με αυστηρούς περιορισμούς πόρων. Σχεδιασμένο για να υποστηρίζει Kubernetes-compatible workloads, το FLEDGE ελαχιστοποιεί τις εξαρτήσεις που συνήθως απαιτούνται από ένα πλήρες cluster Kubernetes (όπως CNI, kube-proxy). Αυτή η προσέγγιση καθιστά το FLEDGE μια πλήρως βιώσιμη λύση για αποκεντρωμένη ενορχήστρωση σε constrained περιβάλλοντα, επιτρέποντας την αποδοτική ανάπτυξη και διαχείριση εφαρμογών σε μεγάλο αριθμό edge συσκευών. Η δυνατότητα εκτέλεσης Kubernetes-compatible workloads με μειωμένο overhead ανοίγει νέες προοπτικές για την αξιοποίηση της ενορχήστρωσης containers σε περιβάλλοντα όπου οι απαιτήσεις πόρων του Kubernetes ήταν μέχρι πρότινος απαγορευτικές.

Παράλληλα, οι τεχνικές resource-aware orchestration σε Single Board Computers (SBCs) είναι κρίσιμες για την αποτελεσματική διαχείριση εφαρμογών σε περιβάλλοντα περιορισμένων πόρων. Στόχος τους είναι η βελτιστοποίηση της χρήσης υπολογιστικών πόρων, μνήμης, δικτύου και ενέργειας, διασφαλίζοντας τη σταθερή και αξιόπιστη λειτουργία των εφαρμογών.

Οι βασικοί άξονες των τεχνικών resource-aware orchestration στα SBCs περιλαμβάνουν:

- **Ακριβής κατανομή πόρων:** Όρια (limits) για μέγιστη κατανάλωση και αιτήματα (requests) για ελάχιστη, βάσει αναγκών, για αποδοτική χρήση πόρων SBC. Δυναμική ανακατανομή σε πραγματικό χρόνο.
- **Επιλογή orchestrator:** Lightweight orchestrators (k3s, Containerd/FLEDGE) για μειωμένο overhead σε σχέση με τον παραδοσιακό Kubernetes, ανάλογα με ανάγκες, πολυπλοκότητα και διαθέσιμους πόρους, αφήνοντας περισσότερους πόρους για τις εφαρμογές.
- **Αποφυγή overcommit & bottlenecks:** Lightweight images για λιγότερη κατανάλωση, απλοποιημένη δικτύωση για αποφυγή bottlenecks και ομαλή επικοινωνία. Network policy enforcement για βελτιστοποίηση κυκλοφορίας.

Κύρια Πλεονεκτήματα - Επεκτάσεις

Η εφαρμογή τεχνικών resource-aware orchestration στα SBCs προσφέρει πολλαπλά και σημαντικά πλεονεκτήματα:

- Μείωση ενεργειακού αποτυπώματος και σταθερή λειτουργία σε edge nodes μέσω αποτελεσματικής διαχείρισης πόρων.
- Κρίσιμη για συσκευές με μπαταρία ή περιορισμένη ενέργεια, αποτρέποντας υπερφόρτωση και αυξάνοντας τη διάρκεια ζωής του hardware.
- Υποστήριξη lightweight real-time εφαρμογών σε τυπικό hardware SBCs, μειώνοντας το κόστος.
- Επαναχρησιμοποίηση K8s APIs με lightweight Kubernetes (k3s) για απλοποίηση διαχείρισης εφαρμογών σε SBCs και ενοποίηση με cloud υποδομές.
- Μείωση πολυπλοκότητας και κόστους.

Οι περιορισμοί στην απόδοση και αξιοπιστία της τρέχουσας αρχιτεκτονικής περιλαμβάνουν:

- Έλλειψη native υποστήριξης για advanced AI scheduling, οδηγώντας σε υποβέλτιστη χρήση πόρων.
- Ανάγκη για tuning ανά workload, αυξάνοντας την πολυπλοκότητα.
- Ευαισθησία σε περιβαλλοντικές μεταβολές, επηρεάζοντας τη σταθερότητα.[43]

4.2.7 Συνοπτική Αξιολόγηση

Η εμβάθυνση στο RQ2 κατέδειξε ένα ευρύ φάσμα τεχνικών και μηχανισμών scheduling και placement που έχουν διατυπωθεί και τεθεί σε εφαρμογή σε υπολογιστικά νέφη (cloud), ακριακές υπολογιστικές υποδομές (edge) και ενδιάμεσες αρχιτεκτονικές (fog). Ο πρωταρχικός σκοπός αυτών των μηχανισμών είναι η βελτιστοποίηση της κατανομής των containerized εφαρμογών και η αποδοτική αξιοποίηση των προσφερόμενων πόρων. Η ενορχήστρωση σε αυτά τα ετερογενή και συχνά απρόβλεπτα περιβάλλοντα επιβάλλει την ανάπτυξη λύσεων ικανών να επιτυγχάνουν μια λεπτή ισορροπία μεταξύ ποικίλων παραμέτρων:

- Απόδοση: Οι τεχνικές scheduling και placement οφείλουν να εξασφαλίζουν την ικανοποίηση των απαιτήσεων απόδοσης των εφαρμογών, όπως ο χρόνος απόκρισης, ο ρυθμός διεκπεραίωσης εργασιών και η διαθεσιμότητα.
- Αποδοτικότητα Πόρων: Η μέγιστη δυνατή αξιοποίηση των υπολογιστικών, αποθηκευτικών και δικτυακών πόρων είναι ζωτικής σημασίας για τη μείωση του κόστους και τη βελτίωση της ενεργειακής απόδοσης.
- Latency: Στο πλαίσιο του Cloud-to-Thing Continuum, η ελαχιστοποίηση της καθυστέρησης (latency) είναι κρίσιμη για εφαρμογές που απαιτούν αλληλεπίδραση σε πραγματικό χρόνο με φυσικές συσκευές και αισθητήρες.

Επιπλέον, οι μηχανισμοί scheduling και placement πρέπει να λαμβάνουν υπόψη και άλλες σημαντικές παραμέτρους, όπως:

- Διαθεσιμότητα και Ανθεκτικότητα: Η ικανότητα των εφαρμογών να παραμένουν διαθέσιμες και λειτουργικές ακόμη και σε περίπτωση αστοχίας κάποιου κόμβου ή μέρους του συστήματος.
- Κινητικότητα: Η δυνατότητα μετακίνησης containerized εφαρμογών μεταξύ διαφορετικών υπολογιστικών κόμβων ή επιπέδων (cloud, edge, fog) ανάλογα με τις ανάγκες και τις συνθήκες.
- Ασφάλεια: Η διασφάλιση της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας των εφαρμογών και των δεδομένων τους.
- Κόστος: Η ελαχιστοποίηση του κόστους λειτουργίας των εφαρμογών, λαμβάνοντας υπόψη τις τιμολογιακές πολιτικές των παρόχων cloud και τις δαπάνες για την υποδομή edge/fog.
- Ενεργειακή Απόδοση: Η μείωση της κατανάλωσης ενέργειας των υπολογιστικών υποδομών, συμβάλλοντας στη βιωσιμότητα και τη μείωση του περιβαλλοντικού αποτυπώματος.

Κύριες Τάσεις και Τεχνικές που Καταγράφηκαν

Πλατφόρμες όπως το k3s, μια ελαφριά διανομή Kubernetes, και το FLEDGE της OpenYurt, ένα πλαίσιο διαχείρισης edge εφαρμογών, ενσωματώνουν ελαφριούς ενορχηστρωτές. Αυτό επιτρέπει την εκτέλεση του Kubernetes σε περιβάλλοντα περιορισμένων πόρων, όπως συσκευές SBC (π.χ. Raspberry Pi). Οι λύσεις αυτές παρέχουν απλοποιημένη ενορχήστρωση

με σημαντικά χαμηλότερο overhead σε σύγκριση με τις παραδοσιακές, πιο απαιτητικές εγκαταστάσεις Kubernetes. Η επίτευξη γραμμικής απόδοσης και προβλέψιμης συμπεριφοράς σε αυτά τα περιβάλλοντα βασίζεται σε διάφορες τεχνικές. Η χρήση host network modes επιτρέπει στα containers να χρησιμοποιούν απευθείας το δίκτυο του host, απλοποιώντας την δικτύωση. Επιπλέον, η χρήση minimal runtimes, όπως containerd ή CRI-O αντί του Docker, μειώνει την κατανάλωση πόρων. Τέλος, η εφαρμογή βελτιστοποιημένων ορίων και αιτημάτων για τους πόρους των containers εξασφαλίζει την αποδοτική χρήση των διαθέσιμων πόρων, οδηγώντας σε σταθερή και προβλέψιμη απόδοση των εφαρμογών edge. Αυτές οι τεχνικές είναι ζωτικής σημασίας για την αξιόπιστη λειτουργία εφαρμογών σε edge υποδομές όπου οι πόροι είναι περιορισμένοι και η απόδοση κρίσιμη.

Συστήματα όπως το GAIKube, ένα πλαίσιο βελτιστοποίησης τοποθέτησης εφαρμογών σε fog και edge περιβάλλοντα, και πολιτικές ευαισθησίας στην καθυστέρηση ενσωματωμένες στο Kubernetes, επιτρέπουν τη δυναμική τοποθέτηση containers σε κόμβους fog/edge. Η τοποθέτηση βασίζεται σε μετρήσεις σχετικές με την καθυστέρηση, όπως το Round-Trip Time (RTT) μεταξύ κόμβων και συσκευών, ιστορικά μοτίβα χρήσης πόρων και προφίλ φόρτου εργασίας εφαρμογών. Η αξιοποίηση αυτών των παραμέτρων επιτρέπει στο σύστημα να λαμβάνει έξυπνες αποφάσεις για την τοποθέτηση των containers, με στόχο τη μείωση της καθυστέρησης έως και 40%. Αυτό είναι ιδιαίτερα σημαντικό για εφαρμογές IoT σε πραγματικό χρόνο, όπου η χαμηλή καθυστέρηση είναι κρίσιμη για τη σωστή λειτουργία (π.χ., αυτόνομη οδήγηση, βιομηχανική αυτοματοποίηση, τηλεϊατρική). Η δυνατότητα δυναμικής προσαρμογής της τοποθέτησης των containers ανάλογα με τις συνθήκες δικτύου και τις απαιτήσεις των εφαρμογών εξασφαλίζει βέλτιστη εμπειρία χρήσης και αξιόπιστη λειτουργία των συστημάτων.

Η χρήση τεχνικών τεχνητής νοημοσύνης, όπως η ενισχυτική μάθηση στο GAIKube, η βελτιστοποίηση πολλαπλών στόχων (με αλγορίθμους όπως ο MOSGD) και η προγνωστική μοντελοποίηση (μέσω τεχνικών όπως το Integer Linear Programming - ILP), ανοίγει νέους ορίζοντες για τον προδραστικό προσδιορισμό των ιδανικών κόμβων εκτέλεσης εφαρμογών. Αυτές οι προσεγγίσεις επιτρέπουν την έγκαιρη αναγνώριση και την αποφυγή πιθανών συμφόρησης (bottlenecks) στους κόμβους του edge, καθώς και τη βελτιστοποίηση της χρήσης των ενεργειακών πόρων, οδηγώντας σε σημαντική εξοικονόμηση ενέργειας. Παρά τα σημαντικά πλεονεκτήματα που προσφέρουν αυτές οι προηγμένες τεχνικές, όπως η βελτιωμένη απόδοση και η μειωμένη κατανάλωση ενέργειας, υπάρχουν και σημαντικοί περιορισμοί που εμποδίζουν την ευρεία υιοθέτησή τους σε παραγωγικά περιβάλλοντα. Η αυξημένη πολυπλοκότητα των αλγορίθμων και των συστημάτων που απαιτούνται για την εφαρμογή τους, καθώς και η έλλειψη ωριμότητας ορισμένων από αυτές τις τεχνολογίες, αποτελούν σημαντικές προκλήσεις που πρέπει να αντιμετωπιστούν για να καταστεί δυνατή η ευρεία διάδοσή τους. Η συνεχής έρευνα και ανάπτυξη στον τομέα αυτό αναμένεται να οδηγήσει στην ωρίμανση αυτών των τεχνικών και στην ευρύτερη εφαρμογή τους σε πραγματικά περιβάλλοντα edge computing.

Για βελτιστοποίηση της διαχείρισης πόρων, ιδιαίτερα σε περιπτώσεις μεταβαλλόμενου φόρτου εργασίας και απαιτήσεων χαμηλού λανθάνοντος χρόνου, προτείνεται μια ευέλικτη στρατηγική εναλλαγής μεταξύ unikernels και containers. Τα unikernels, ελαφριά και εξειδικευμένα, προσφέρουν αυξημένη απόδοση και μειωμένο αποτύπωμα μνήμης για συγκεκριμένες εργασίες. Αντίστοιχα, τα containers παρέχουν ευελιξία και ευκολία ανάπτυξης για ένα ευρύτερο φάσμα εφαρμογών. Η επιλογή του κατάλληλου μοντέλου εικονικοποίησης

βάσει των αναγκών του φόρτου εργασίας επιτρέπει τη βέλτιστη αξιοποίηση των διαθέσιμων πόρων και τη σημαντική μείωση της κατανάλωσης μνήμης, πλεονέκτημα κρίσιμο για εφαρμογές Internet of Things (IoT) με μεγάλο αριθμό συσκευών και περιορισμένους πόρους. Η πρακτική εφαρμογή αυτής της υβριδικής προσέγγισης έχει επιδείξει βελτίωση στη χρήση μνήμης έως και 36%.

Η παραδοσιακή αρχιτεκτονική του Kubernetes, με ένα κεντρικό etcd για την αποθήκευση και τον συντονισμό, μπορεί να δημιουργήσει σημεία συμφόρησης και καθυστέρησης, ιδίως σε καταναμεμένα περιβάλλοντα όπως το edge computing. Η αντικατάσταση του κεντρικού datastore με καταναμεμένα datastores που βασίζονται σε Conflict-free Replicated Data Types (CRDTs) αλλάζει ριζικά το μοντέλο διαχείρισης. Τα CRDTs επιτρέπουν σε κάθε κόμβο να διατηρεί μια τοπική, συνεπή κατάσταση, μειώνοντας την ανάγκη για συνεχή επικοινωνία με ένα κεντρικό σημείο και ενισχύοντας την τοπική αυτονομία. Αυτή η αλλαγή οδηγεί σε σημαντική μείωση του scheduling latency σε περιβάλλοντα edge, όπου η χαμηλή καθυστέρηση είναι ζωτικής σημασίας. Επιπλέον, η δυνατότητα λειτουργίας χωρίς σταθερό quorum ανοίγει νέες δυνατότητες για αποκεντρωμένη ενορχήστρωση, επιτρέποντας την αυτόνομη λειτουργία τμημάτων του συστήματος ακόμη και σε περιπτώσεις προσωρινής αποσύνδεσης.

Η αποτελεσματική εκτέλεση containerized υπηρεσιών σε συσκευές περιορισμένων πόρων, όπως τα Single Board Computers (SBCs) με 512MB έως 4GB RAM, αποτελεί σημαντική πρόκληση. Για την αντιμετώπιση αυτής της πρόκλησης, απαιτείται μια συνδυαστική προσέγγιση που περιλαμβάνει τη χρήση βελτιστοποιημένων ορίων πόρων (resource limits) για κάθε container, την ανάπτυξη custom orchestrators προσαρμοσμένων στις ιδιαιτερότητες των SBCs και την αξιοποίηση ελαφριών λειτουργικών συστημάτων. Μέσω αυτών των τεχνικών, είναι δυνατή η ομαλή εκτέλεση containerized υπηρεσιών ακόμη και σε περιβάλλοντα με αυστηρούς περιορισμούς hardware, χωρίς αστοχίες ή υποβάθμιση της απόδοσης. Αυτό επεκτείνει το φάσμα των συσκευών που μπορούν να συμμετέχουν σε καταναμεμένα συστήματα και επιτρέπει την ανάπτυξη σύνθετων εφαρμογών σε υπολογιστικά περιβάλλοντα χαμηλού κόστους.

Η συνολική εξέταση του υπάρχοντος ερευνητικού και πρακτικού τοπίου καταλήγει σε ένα κομβικό σημείο: η επιδίωξη μιας ενιαίας, καθολικής τεχνικής scheduling και placement για την αποτελεσματική διαχείριση φόρτων εργασίας σε διάφορα καταναμεμένα περιβάλλοντα δεν φέρνει αποτελέσματα. Αντίθετα, η πείρα και η ανάλυση των επιτυχημένων εφαρμογών υποδεικνύουν ότι η ιδανική λύση βρίσκεται στην ανάπτυξη και εφαρμογή ευέλικτων και προσαρμόσιμων λύσεων.

Συγκεκριμένα, οι αποδοτικότερες στρατηγικές scheduling και placement ενσωματώνουν τα εξής θεμελιώδη χαρακτηριστικά:

- Ευέλικτη και Ευαίσθητη Διαχείριση (Adaptive, Context-Aware Scheduling):

Η αποτελεσματικότητα μιας τεχνικής scheduling είναι άρρηκτα συνδεδεμένη με τις ειδικές απαιτήσεις του φόρτου εργασίας (π.χ., υπολογιστικά απαιτητικός, με αυξημένο όγκο εισόδου/εξόδου, ευαίσθητος στην καθυστέρηση) και τις δυνατότητες και τους περιορισμούς των διαθέσιμων κόμβων (π.χ., υπολογιστική ισχύς, διαθέσιμη μνήμη, δικτυακή χωρητικότητα, παρουσία εξειδικευμένου υλικού όπως GPUs). Οι επιτυχημένες προσεγγίσεις αναλύουν δυναμικά αυτά τα στοιχεία και τροποποιούν ανάλογα τις αποφάσεις scheduling και placement, αντί να στηρίζονται σε στατικές ή προκαθορισμένες

πολιτικές. Για παράδειγμα, ένας φόρτος εργασίας που απαιτεί χαμηλή καθυστέρηση μπορεί να τοποθετηθεί σε κόμβους με μικρότερη δικτυακή απόσταση από τους χρήστες ή τις πηγές δεδομένων του.

- Αποκεντρωμένες ή Συνεργατικές Στρατηγικές (Decentralized or Collaborative Scheduling - Local-First Scheduling):

Στα κατανεμημένα συστήματα μεγάλης κλίμακας, οι κεντρικές αρχιτεκτονικές scheduling συχνά αντιμετωπίζουν προβλήματα επεκτασιμότητας και αυξημένης καθυστέρησης λόγω του όγκου των πληροφοριών που πρέπει να διαχειριστούν και του χρόνου που απαιτείται για τη λήψη και την υλοποίηση των αποφάσεων. Οι αποκεντρωμένες ή συνεργατικές προσεγγίσεις, όπως το "local-first scheduling", όπου οι αποφάσεις scheduling λαμβάνονται σε τοπικό επίπεδο από τους ίδιους τους κόμβους ή από μικρότερες ομάδες κόμβων, μπορούν να βελτιώσουν την επεκτασιμότητα, τη λανθάνουσα καθυστέρηση και την ανθεκτικότητα του συστήματος. Οι κόμβοι μπορούν να λαμβάνουν αποφάσεις με βάση τις τοπικές τους συνθήκες και να συνεργάζονται με γειτονικούς κόμβους για την επίτευξη ενός συνολικά αποτελεσματικού scheduling.

- Προληπτικές Αποφάσεις μέσω Τηλεμετρίας και Προβλέψεων (Leveraging Telemetry and Prediction for Proactive Decisions):

Η συλλογή και ανάλυση τηλεμετρικών δεδομένων σε πραγματικό χρόνο (π.χ., φόρτος CPU, χρήση μνήμης, δικτυακή κίνηση, θερμοκρασία) παρέχει κρίσιμες πληροφορίες για την τρέχουσα κατάσταση του συστήματος και την απόδοση των φόρτων εργασίας. Επιπλέον, η αξιοποίηση τεχνικών πρόβλεψης (π.χ., machine learning) για την πρόβλεψη της μελλοντικής ζήτησης πόρων ή της πιθανότητας αστοχιών επιτρέπει τη λήψη προληπτικών αποφάσεων scheduling και placement. Για παράδειγμα, ένας φόρτος εργασίας μπορεί να μετακινηθεί προληπτικά σε έναν λιγότερο φορτωμένο κόμβο πριν αυτός υπερφορτωθεί, ή ένας κόμβος που προβλέπεται να παρουσιάσει πρόβλημα μπορεί να αδειάσει από τους φόρτους εργασίας του.

- Δυναμική Εξισορρόπηση Καθυστέρησης, Κόστους και Αξιοπιστίας (Balancing Latency - Cost - Reliability in Real-Time):

Στα σύγχρονα κατανεμημένα συστήματα, συχνά υπάρχουν αντιφατικές απαιτήσεις όσον αφορά την καθυστέρηση (latency), το κόστος και την αξιοπιστία (reliability). Για παράδειγμα, η τοποθέτηση ενός φόρτου εργασίας σε μια περιοχή με χαμηλή καθυστέρηση μπορεί να είναι πιο ακριβή, ενώ η επιλογή ενός οικονομικότερου κόμβου μπορεί να οδηγήσει σε αυξημένη καθυστέρηση ή μειωμένη αξιοπιστία. Οι αποτελεσματικές στρατηγικές scheduling πρέπει να είναι σε θέση να εξισορροπούν δυναμικά αυτές τις αντιτιθέμενες απαιτήσεις σε πραγματικό χρόνο, λαμβάνοντας υπόψη τις τρέχουσες συνθήκες του συστήματος και τις προτεραιότητες των εφαρμογών. Αυτό μπορεί να περιλαμβάνει τη χρήση πολιτικών που επιτρέπουν την προσαρμογή των κριτηρίων scheduling ανάλογα με τις τρέχουσες ανάγκες και το κόστος.

Συνολικά, η ανάλυση τονίζει την ανάγκη για ευφυείς και δυναμικές προσεγγίσεις scheduling και placement που μπορούν να προσαρμόζονται στις διαρκώς μεταβαλλόμενες συνθήκες των σύγχρονων κατανεμημένων υπολογιστικών περιβαλλόντων. Η έμφαση στην προσαρμοστικότητα, την αποκέντρωση, την αξιοποίηση δεδομένων και την ικανότητα

εξισορρόπησης αντικρουόμενων στόχων αποτελούν τα θεμέλια για την ανάπτυξη αποτελεσματικών και αποδοτικών λύσεων διαχείρισης φόρτων εργασίας.

Επιπρόσθετα στα προαναφερθέντα ερευνητικά κενά, είναι σημαντικό να υπογραμμιστεί η σύνθετη και πολυδιάστατη φύση των προκλήσεων που σχετίζονται με την αποτελεσματική διαχείριση και τον προγραμματισμό εργασιών σε περιβάλλοντα edge computing.

- Περιορισμένη Ανάπτυξη AI-based Schedulers σε Edge Clusters:

Παρά τις σημαντικές δυνατότητες βελτιστοποίησης που έχουν επιδείξει οι τεχνικές TN σε κεντρικά περιβάλλοντα, η εφαρμογή και η εξέλιξη αντίστοιχων λύσεων για edge clusters βρίσκεται σε πρώιμο στάδιο. Αυτό οφείλεται στην ανομοιογένεια των συσκευών και πόρων, στους περιορισμούς υπολογιστικής ισχύος και ενέργειας, καθώς και στην ανάγκη για αποκεντρωμένη λήψη αποφάσεων σε πραγματικό χρόνο. Σημαντική ερευνητική πρόκληση αποτελεί η ανάπτυξη ευέλικτων και αυτόνομα προσαρμοζόμενων AI-based schedulers που θα λαμβάνουν υπόψη τις ιδιαίτερες απαιτήσεις και περιορισμούς των edge clusters.

- Ανάγκη για Ολοκληρωμένα Πλαίσια Benchmarking για Τοποθέτηση Ευαίσθητη στην Καθυστέρηση:

Η τοποθέτηση εφαρμογών και υπηρεσιών με ευαισθησία στην καθυστέρηση σε κατανεμημένα περιβάλλοντα όπως τα edge clusters απαιτεί αξιόπιστες και συγκρίσιμες μετρικές και εργαλεία αξιολόγησης. Η απουσία ενοποιημένων πλαισίων αναφοράς δυσχεραίνει τη σύγκριση αλγορίθμων και στρατηγικών τοποθέτησης, καθώς και την αντικειμενική αξιολόγηση της απόδοσής τους σε πραγματικά σενάρια χρήσης. Είναι απαραίτητη η ανάπτυξη τέτοιων πλαισίων που θα λαμβάνουν υπόψη κρίσιμες παραμέτρους όπως η καθυστέρηση δικτύου, η διαθεσιμότητα πόρων και οι απαιτήσεις ποιότητας υπηρεσίας (QoS) για την προώθηση της έρευνας και της ανάπτυξης.

Αναλυτικό Τελικό Συμπέρασμα:

Η αποτελεσματική ενορχήστρωση containerized εφαρμογών σε όλο το Cloud-to-Thing Continuum αποτελεί σύνθετη πρόκληση. Η επιτυχία έγκειται στην επιλογή του κατάλληλου μηχανισμού scheduling και placement για κάθε περίπτωση. Οι αναπτυσσόμενες τεχνικές πρέπει να είναι ευέλικτες και να προσαρμόζονται σε ποικίλες συνθήκες. Ο context-awareness είναι κρίσιμος για τη λήψη αποφάσεων τοποθέτησης και προγραμματισμού containers, λαμβάνοντας υπόψη το ευρύτερο περιβάλλον λειτουργίας (διαθέσιμοι πόροι, περιορισμοί δικτύωσης, απαιτήσεις εφαρμογών).

Η ενεργειακή αποδοτικότητα είναι όλο και πιο σημαντική, ειδικά σε περιβάλλοντα με περιορισμένους ενεργειακούς πόρους ή αυστηρές περιβαλλοντικές επιταγές. Οι μηχανισμοί ενορχήστρωσης πρέπει να ελαχιστοποιούν την κατανάλωση ενέργειας χωρίς να θέτουν σε κίνδυνο την απόδοση και την αξιοπιστία των εφαρμογών. Επιπλέον, η λειτουργική βιωσιμότητα των επιλεγμένων τεχνικών είναι απαραίτητη για τη μακροπρόθεσμη λειτουργία των συστημάτων, ακόμη και σε ετερογενή, περιορισμένα ή ασταθή περιβάλλοντα που συναντώνται συχνά στο Thing Continuum. Η ικανότητα αντιμετώπισης διακοπών, η αυτόματη ανάκαμψη και η ελαχιστοποίηση της ανθρώπινης παρέμβασης ενισχύουν τη λειτουργική βιωσιμότητα. Η τρέχουσα ερευνητική κατεύθυνση τονίζει ότι το μέλλον της

ενορχήστρωσης containerized εφαρμογών θα διαμορφωθεί από την ενσωμάτωση προγνωστικής ευφυΐας. Η ικανότητα πρόβλεψης μελλοντικών αναγκών και λήψης προληπτικών μέτρων για βελτιστοποίηση τοποθέτησης και προγραμματισμού θα βελτιώσει σημαντικά την απόδοση και την αποδοτικότητα. Η τάση προς την αποκέντρωση των συστημάτων ελέγχου, μεταφέροντας την ευθύνη λήψης αποφάσεων πιο κοντά στην άκρη του δικτύου, αναμένεται να ενισχύσει την ανθεκτικότητα και την ανταπόκριση σε τοπικές ανάγκες. Τέλος, η αυτονομία των συστημάτων ενορχήστρωσης, η ικανότητα λειτουργίας και προσαρμογής χωρίς συνεχή ανθρώπινη εποπτεία, θα είναι καθοριστική για την αποτελεσματική διαχείριση του αυξανόμενου αριθμού και της πολυπλοκότητας των containerized εφαρμογών στο Cloud-to-Thing Continuum. Η συνέργεια προγνωστικής ευφυΐας, αποκέντρωσης και αυτονομίας θα αποτελέσει τον πυρήνα της επόμενης γενιάς συστημάτων ενορχήστρωσης.

4.3 RQ3

4.3.1 Εισαγωγή

Η εννοχήστρωση containerized εφαρμογών στο συνεχώς εξελισσόμενο πλαίσιο του cloud-to-thing continuum αντιπροσωπεύει ένα πεδίο ύψιστης σημασίας για την σύγχρονη έρευνα και ανάπτυξη. Στόχος είναι η δημιουργία δυναμικών, ευφών και εξαιρετικά αποδοτικών μηχανισμών διαχείρισης που μπορούν να υλοποιήσουν τους διασκορπισμένους πόρους και τις ποικίλες υπηρεσίες σε ένα ετερογενές και πολυεπίπεδο ψηφιακό οικοσύστημα. Η εγγενής πολυπλοκότητα του εγχειρήματος πολλαπλασιάζεται καθώς οι εφαρμογές διευρύνουν την παρουσία τους από το κεντρικό cloud προς τα ενδιάμεσα στρώματα του fog computing και την ακραία υπολογιστική (edge computing), αγκαλιάζοντας έναν διαρκώς αυξανόμενο αριθμό ετερόκλητων συσκευών Internet of Things (IoT), κόμβων edge με ποικίλες δυνατότητες, αρχιτεκτονικών μικροϋπηρεσιών και περιβαλλόντων που χαρακτηρίζονται από περιορισμένους υπολογιστικούς και ενεργειακούς πόρους.

Σε αυτό το σύνθετο και δυναμικό τεχνολογικό υπόβαθρο, το τρίτο ερευνητικό ερώτημα (RQ3) αναδεικνύεται ως κομβικής σημασίας, εστιάζοντας στην ενδελεχή διερεύνηση των σημαντικότερων ανοιχτών προκλήσεων που εξακολουθούν να παρεμποδίζουν την πλήρη αξιοποίηση των δυνατοτήτων του cloud-to-thing continuum. Παράλληλα, το ερώτημα αυτό στοχεύει στην επισήμανση των νέων και αναδυόμενων ερευνητικών ευκαιριών που γεννιούνται από τη ραγδαία ενσωμάτωση προηγμένων τεχνολογιών τεχνητής νοημοσύνης (Artificial Intelligence – AI) και μηχανικής μάθησης (Machine Learning – ML) στις κρίσιμες διαδικασίες του scheduling (προγραμματισμός εργασιών), του autoscaling (αυτόματη προσαρμογή πόρων) και του deployment (ανάπτυξη και διάθεση εφαρμογών).[51]

Πεδίο Πρόκλησης	Περιγραφή	Επιπτώσεις
Ετερογένεια Υποδομών	Ποικιλία σε πόρους, αρχιτεκτονικές και δυνατότητες κόμβων edge/cloud	Δυσκολία στην κατανομή και βελτιστοποίηση εργασιών
Περιορισμένοι Πόροι	Edge/fog κόμβοι με χαμηλή CPU, RAM, μπαταρία	Ανάγκη για ελαφριούς και αποδοτικούς μηχανισμούς
Latency-sensitive Απαιτήσεις	Ιδιαίτερα αυστηρές ανάγκες καθυστέρησης για real-time εφαρμογές	Απαραίτητος ο edge-based προγραμματισμός
Κινητικότητα & Αστάθεια	Δυναμικές τοπολογίες, συχνές αποσυνδέσεις	Απαιτούνται adaptive, context-aware schedulers

Πολυπλοκότητα Δικτύου & Αλληλεξαρτήσεις	Overlapping τοπολογίες και πολυεπίπεδες υπηρεσίες	Δυσκολία στον αποδοτικό προγραμματισμό και εντοπισμό προβλημάτων
---	---	--

Πίνακας 4.3.1 Κύριες προκλήσεις ενορχήστρωσης σε περιβάλλοντα Edge/Fog/Cloud

Οι υφιστάμενες προκλήσεις εκτείνονται σε πολλαπλά επίπεδα, συμπεριλαμβάνοντας, μεταξύ άλλων:

- Διαχείριση Ετερογένειας και Περιορισμένων Πόρων:

Η ενορχήστρωση σε περιβάλλοντα με μεγάλη ποικιλία συσκευών και κόμβων, καθένας με διαφορετικές υπολογιστικές δυνατότητες, χωρητικότητα μνήμης, ενεργειακούς περιορισμούς και δυνατότητες δικτύωσης, απαιτεί εξελιγμένους μηχανισμούς για την αποτελεσματική κατανομή και διαχείριση των πόρων.

- Διασφάλιση Χαμηλής Ενδογενής Καθυστερήσης (Low Latency) και Υψηλής Απόκρισης:

Για εφαρμογές που είναι ευαίσθητες στον χρόνο, όπως η αυτόνομη οδήγηση, η βιομηχανική αυτοματοποίηση και η επαυξημένη/εικονική πραγματικότητα, η ελαχιστοποίηση της καθυστέρησης στην επεξεργασία και τη μετάδοση δεδομένων μεταξύ cloud, fog και edge είναι κρίσιμης σημασίας.

- Βελτιστοποίηση της Κινητικότητας και της Συνδεσιμότητας:

Η διαχείριση της κινητικότητας των συσκευών και των εφαρμογών, καθώς και η αντιμετώπιση των προκλήσεων που σχετίζονται με την αστάθεια και την περιορισμένη διαθεσιμότητα της δικτυακής σύνδεσης στο edge, αποτελούν σημαντικά εμπόδια.

- Ενίσχυση της Ασφάλειας και της Προστασίας των Δεδομένων:

Η διασφάλιση της ασφάλειας των εφαρμογών, των δεδομένων και των υποδομών σε ένα τόσο κατανεμημένο και εκτεταμένο περιβάλλον, καθώς και η συμμόρφωση με τους κανονισμούς προστασίας προσωπικών δεδομένων, απαιτούν καινοτόμες λύσεις.

- Αντιμετώπιση της Πολυπλοκότητας της Διαχείρισης και της Παρακολούθησης:

Η διαχείριση και η παρακολούθηση ενός μεγάλου αριθμού ετερογενών στοιχείων και διασυνδεδεμένων υπηρεσιών σε πολλαπλά επίπεδα αποτελεί μια εξαιρετικά σύνθετη πρόκληση.

4.3.2 Ευφυείς Μηχανισμοί Ενορχήστρωσης με AI και ML

Η αυξανόμενη ενσωμάτωση τεχνολογιών AI και ML υπόσχεται να φέρει επανάσταση στην αντιμετώπιση αυτών των προκλήσεων και να ανοίξει νέους ορίζοντες ερευνητικών ευκαιριών:

- Έξυπνο Scheduling Βασισμένο σε Προβλέψεις:

Αλγόριθμοι ML μπορούν να αναλύσουν ιστορικά δεδομένα και να προβλέψουν τις ανάγκες των εφαρμογών σε πόρους, καθώς και τις συνθήκες του περιβάλλοντος, επιτρέποντας την προληπτική και δυναμική κατανομή φόρτου εργασίας στους πιο κατάλληλους κόμβους.

- Αυτόματη Προσαρμογή Πόρων με Γνώμονα την Απόδοση:

Μοντέλα ΑΙ μπορούν να παρακολουθούν σε πραγματικό χρόνο την απόδοση των εφαρμογών και τη χρήση των πόρων, ενεργοποιώντας αυτόματες διαδικασίες autoscaling για την αποτελεσματική αντιμετώπιση των μεταβαλλόμενων απαιτήσεων.

- Ευφυής Ανάπτυξη και Τοποθέτηση Εφαρμογών:

Τεχνικές ML μπορούν να αξιολογήσουν πολλαπλά κριτήρια, όπως η διαθεσιμότητα πόρων, η εγγύτητα στους χρήστες, οι απαιτήσεις καθυστέρησης και το κόστος, για να καθορίσουν την ιδανική τοποθεσία ανάπτυξης για κάθε μικροϋπηρεσία ή container.

- Προγνωστική Συντήρηση και Ανίχνευση Ανωμαλιών:

Αλγόριθμοι ΑΙ μπορούν να αναλύσουν δεδομένα τηλεμετρίας και logs για να εντοπίσουν έγκαιρα πιθανές αστοχίες υλικού ή λογισμικού, καθώς και ανώμαλη συμπεριφορά που μπορεί να υποδηλώνει κυβερνοεπιθέσεις.

- Βελτιστοποίηση της Χρήσης Ενέργειας:

Μοντέλα ML μπορούν να αναπτύσσονται για την παρακολούθηση και την πρόβλεψη της κατανάλωσης ενέργειας σε ολόκληρο το continuum, προτείνοντας στρατηγικές για τη μείωση του ενεργειακού αποτυπώματος.

Συνολικά, το RQ3 αναδεικνύει την κρίσιμη ανάγκη για περαιτέρω έρευνα στην ανάπτυξη ευφών και αυτόνομων συστημάτων ενορχήστρωσης που μπορούν να αξιοποιήσουν πλήρως τις δυνατότητες του cloud-to-thing continuum, αντιμετωπίζοντας τις υφιστάμενες προκλήσεις και αξιοποιώντας τις αναδυόμενες ευκαιρίες που προσφέρουν οι τεχνολογίες ΑΙ και ML. Η επιτυχής αντιμετώπιση αυτού του ερευνητικού ερωτήματος θα οδηγήσει σε πιο αποδοτικές, αξιόπιστες και ευέλικτες λύσεις για την ανάπτυξη και τη διαχείριση κατανεμημένων εφαρμογών στο μέλλον.

Η αυτοματοποιημένη ενορχήστρωση containers αντιμετωπίζει προκλήσεις προσαρμοστικότητας και ευφυΐας, με τις υπάρχουσες μεθόδους να υστερούν. Η ενσωμάτωση ML/ΑΙ βρίσκεται σε ερευνητικό στάδιο, αφήνοντας ένα κενό. Η μελλοντική εξέλιξη θα επικεντρωθεί σε ευφυείς αλγορίθμους για δυναμική διαχείριση πόρων. Η ανάλυση στοχεύει στην κριτική αποτύπωση των προκλήσεων και στην ανάδειξη της αξίας της ΑΙ/ML για ευφύτερα, αποδοτικότερα και αυτόνομα προσαρμοζόμενα συστήματα ενορχήστρωσης, σκιαγραφώντας μελλοντικές ερευνητικές κατευθύνσεις.[52], [53]

Παρά την πρόοδο στην αυτοματοποίηση της ενορχήστρωσης containerized εφαρμογών, η αποτελεσματική διαχείρισή τους σε σύνθετα, κατανεμημένα και δυναμικά περιβάλλοντα παραμένει μια τεχνικά απαιτητική πρόκληση. Η πολυπλοκότητα εντείνεται λόγω της αρχιτεκτονικής μικροϋπηρεσιών και της διασποράς σε πολλαπλούς κόμβους με αλληλεξαρτήσεις. Σημαντικές προκλήσεις περιλαμβάνουν τη διαχείριση κατανεμημένων εφαρμογών, τη διασφάλιση δυναμικότητας και ελαστικότητας, την πολυεπίπεδη ασφάλεια, την αντιμετώπιση της πολυπλοκότητας δικτύων, την παρακολούθηση και διάγνωση προβλημάτων, τη διαχείριση του κύκλου ζωής των εφαρμογών και την ανάγκη για εξειδικευμένο προσωπικό. Η αντιμετώπιση αυτών των ζητημάτων είναι καθοριστική για την πλήρη αξιοποίηση της αυτοματοποίησης και της ενορχήστρωσης των σύγχρονων εφαρμογών. Η διαχείριση containerized εφαρμογών σε κατανεμημένα και δυναμικά περιβάλλοντα αποτελεί ένα συνεχώς εξελισσόμενο πεδίο έρευνας και ανάπτυξης. Οι σύγχρονες πλατφόρμες

ενορχήστρωσης, με το Kubernetes να κατέχει ηγετική θέση, αποτελούν θεμελιώδη εργαλεία για τη διαχείριση και την κλιμάκωση σύγχρονων, κατανεμημένων εφαρμογών. Παρέχουν ενσωματωμένους μηχανισμούς αυτόματης κλιμάκωσης (όπως ο Horizontal Pod Autoscaler - HPA) και προγραμματισμού εργασιών. Ωστόσο, αυτοί οι μηχανισμοί βασίζονται σε σχετικά απλές μετρικές, όπως η χρήση CPU ή μνήμης. Παρά την αποτελεσματικότητά τους σε βασικά σενάρια, οι υπάρχουσες πλατφόρμες ενορχήστρωσης επιδεικνύουν σημαντικούς περιορισμούς στην ευφυΐα και την προσαρμοστικότητά τους σε πιο πολύπλοκες και δυναμικές συνθήκες.

Κύριοι Περιορισμοί:

- Αδυναμία Έγκαιρης Πρόβλεψης και Ανταπόκρισης: Οι πλατφόρμες υστερούν στην ακριβή πρόβλεψη του φόρτου εργασίας, αντιδρώντας με καθυστέρηση στις μεταβολές, γεγονός που μπορεί να επηρεάσει αρνητικά την απόδοση και την εμπειρία του χρήστη. Η έγκαιρη αναγνώριση μειωμένου φόρτου θα επέτρεπε την αποδοτικότερη χρήση των πόρων και τη μείωση του κόστους.
- Έλλειψη Σύνθετων Κριτηρίων Βελτιστοποίησης: Οι μηχανισμοί κλιμάκωσης περιορίζονται σε λίγες βασικές μετρικές, παραβλέποντας κρίσιμες παραμέτρους όπως:
 - Latency: Δεν λαμβάνεται υπόψη για την προσαρμογή των πόρων.
 - Ενεργειακή Κατανάλωση: Δεν υπάρχει ενσωματωμένη λειτουργία βελτιστοποίησης.
 - Κόστος Cloud: Οι αποφάσεις κλιμάκωσης δεν συνυπολογίζουν τις τιμές των πόρων για τη μείωση του κόστους.
 - Αξιοπιστία: Οι απλοί μηχανισμοί ενδέχεται να μην είναι επαρκείς για σύνθετες καταστάσεις.

Οι τρέχουσες πλατφόρμες κλιμάκωσης αντιδρούν καθυστερημένα στην υπέρβαση των προκαθορισμένων ορίων, οδηγώντας σε πιθανά προβλήματα, ειδικά σε περιβάλλοντα με περιορισμένους πόρους, όπως τα edge nodes, όπου η άμεση ανταπόκριση στις αυξήσεις του φόρτου εργασίας είναι ζωτικής σημασίας. Η περιορισμένη ευφυΐα των σημερινών πλατφορμών ενορχήστρωσης εμποδίζει την αποτελεσματική διαχείριση και βελτιστοποίηση των κατανεμημένων εφαρμογών. Η ενσωμάτωση προηγμένων μηχανισμών πρόβλεψης και προληπτικών στρατηγικών κλιμάκωσης είναι απαραίτητη για τη βελτίωση της απόδοσης, της αξιοπιστίας και της αποδοτικότητας κόστους.

Η ενορχήστρωση σε ένα συνεχές υπολογιστικό περιβάλλον που εκτείνεται από το κεντρικό cloud έως τις ακραίες συσκευές (edge) παρουσιάζει ένα σύνολο μοναδικών και σύνθετων προκλήσεων, οι οποίες πηγάζουν κυρίως από την ετερογένεια των υποδομών και την εγγενή μεταβλητότητα και αστάθεια του περιβάλλοντος. Επιπλέον, η φύση των εφαρμογών που αναπτύσσονται σε αυτό το συνεχές δημιουργεί επικαλυπτόμενες τοπολογίες, επιτείνοντας την πολυπλοκότητα της διαχείρισης και του προγραμματισμού των εργασιών.

Η ετερογένεια υποδομών, με διαφορετικούς υπολογιστικούς κόμβους (ισχυροί servers στο cloud, περιορισμένοι σε πόρους κόμβοι στο edge με ποικίλες αρχιτεκτονικές) και ποικίλα interfaces επικοινωνίας (υψηλού εύρους/χαμηλής καθυστέρησης στο cloud, ασύρματα/IoT στο edge με διαφορετικά χαρακτηριστικά), αποτελεί βασική πρόκληση στην ενορχήστρωση cloud-edge και η διαχείρισή της είναι κρίσιμη για την απρόσκοπτη λειτουργία κατανεμημένων εφαρμογών. Η ενορχήστρωση στο edge περιβάλλον είναι δύσκολη λόγω

μεταβλητότητας, συχνών αποσυνδέσεων, περιορισμένων bandwidths και κινητικότητας χρηστών/workloads, απαιτώντας δυναμικούς και ευέλικτους μηχανισμούς. Οι σύγχρονες καταναεμημένες εφαρμογές, χρησιμοποιώντας τεχνολογίες cloud, fog και IoT, δημιουργούν περίπλοκες και αλληλεπικαλυπτόμενες τοπολογίες, γεγονός που καθιστά την ενορχήστρωση ιδιαίτερα απαιτητική παρά τα πλεονεκτήματα της μειωμένης καθυστέρησης και της τοπικής επεξεργασίας. Για την αποτελεσματική λειτουργία τους, απαιτούνται στρατηγικές προγραμματισμού που να λαμβάνουν υπόψη τις διασυνδέσεις (interface-aware) και να είναι ευαίσθητες στην καθυστέρηση (latency-sensitive), κατανέμοντας τις εργασίες με βάση τις απαιτήσεις καθυστέρησης και τις δυνατότητες των κόμβων (edge για άμεση αλληλεπίδραση, fog/cloud για υπολογιστική ισχύ). Εντούτοις, η αποτελεσματική ενορχήστρωση στο συνεχές cloud-edge αντιμετωπίζει σημαντικές προκλήσεις που σχετίζονται με την ετερογένεια των πόρων, τη μεταβλητότητα των συνθηκών, την πολυπλοκότητα των τοπολογιών και τις αυστηρές απαιτήσεις καθυστέρησης. Αυτό καθιστά την ανάπτυξη ευφών και δυναμικών μηχανισμών ενορχήστρωσης απαραίτητη. Η απουσία τέτοιων μηχανισμών οδηγεί αναπόφευκτα σε μη βέλτιστες κατανομές εργασιών, υποχρησιμοποίηση των διαθέσιμων πόρων και ανεπαρκές επίπεδο Ποιότητας Υπηρεσίας (QoS).

Η δημιουργία προηγμένων ευφών ενορχηστρωτών, οι οποίοι βασίζονται σε τεχνικές Μηχανικής Μάθησης (ML), αντιμετωπίζει μια σειρά θεμελιωδών προκλήσεων. Οι προκλήσεις αυτές σχετίζονται άμεσα με την ικανότητα αποτελεσματικής παρατήρησης των καταναεμημένων συστημάτων και την επακόλουθη εκπαίδευση ισχυρών μοντέλων για πρόβλεψη και λήψη αποφάσεων. Οι περιορισμοί αυτοί μπορούν να κατηγοριοποιηθούν σε τρεις κύριες κατηγορίες:

1. **Επάρκεια και Ποιότητα Δεδομένων Παρατήρησης:** Η ανάπτυξη αποτελεσματικών μοντέλων ML για ευφείς ενορχηστρωτές εξαρτάται άμεσα από τη συλλογή μεγάλου όγκου ποιοτικών δεδομένων παρατήρησης. Αυτό περιλαμβάνει την ενσωμάτωση ετερογενών πηγών δεδομένων και την αντιμετώπιση τεχνικών δυσκολιών για τη δημιουργία ενός συνεκτικού και αντιπροσωπευτικού συνόλου δεδομένων. Η ανεπάρκεια ποιοτικών δεδομένων αποτελεί σημαντικό εμπόδιο, καθώς περιορίζει την κατανόηση της συμπεριφοράς του συστήματος και, κατ' επέκταση, την αποτελεσματικότητα της ενορχήστρωσης.
2. **Προκλήσεις Καταναεμημένης Εκπαίδευσης Μοντέλων:** Η εκπαίδευση των μοντέλων ML σε πραγματικό χρόνο ή απευθείας στο edge, αν και απαραίτητη για την άμεση αντίδραση και προσαρμογή στις μεταβαλλόμενες συνθήκες, αντιμετωπίζει περιορισμούς λόγω των περιορισμένων υπολογιστικών πόρων στους εν λόγω κόμβους. Επιπλέον, η εξεύρεση αποδοτικών καταναεμημένων αλγορίθμων εκπαίδευσης είναι κρίσιμη για την υλοποίηση τέτοιων συστημάτων.
3. **Δυσκολίες στην Ενσωμάτωση Τεχνικών Διατήρησης της Ιδιωτικότητας:** Η εφαρμογή τεχνικών διατήρησης της ιδιωτικότητας, όπως η Ομοσπονδιακή Μάθηση (Federated Learning), παρουσιάζει ιδιαίτερες δυσκολίες στα edge περιβάλλοντα. Αυτές οφείλονται κυρίως στους περιορισμένους όγκους τοπικών δεδομένων που είναι διαθέσιμοι σε κάθε κόμβο, στην αστάθεια των δικτυακών συνδέσεων και στην ετερογένεια των δεδομένων μεταξύ των κόμβων.

Συνοψίζοντας, οι περιορισμοί που αφορούν την παρατήρηση των καταναεμημένων συστημάτων και την εκπαίδευση των ευφών μοντέλων συνιστούν σημαντικά εμπόδια για την ανάπτυξη αποτελεσματικών ευφών ενορχηστρωτών. Η αντιμετώπιση αυτών των

προκλήσεων απαιτεί την ανάπτυξη καινοτόμων λύσεων και περαιτέρω έρευνα τόσο στον τομέα της Μηχανικής Μάθησης όσο και στα καταναμεμημένα συστήματα, προκειμένου να αξιοποιηθεί πλήρως το δυναμικό τους για την αυτοματοποίηση και τη βελτιστοποίηση σύνθετων υποδομών.

Η ενορχήστρωση καταναμεμημένων πόρων (cloud, fog, IoT) εμποδίζεται από την απουσία ενοποιημένων προτύπων και διαλειτουργικότητας. Η διαχείριση ροών εργασίας και εφαρμογών σε ετερογενή, απομονωμένα εργαλεία, χωρίς κοινά APIs ή προδιαγραφές, δυσχεραίνει την ανάπτυξη και διαχείριση σύνθετων εφαρμογών. Υπάρχει έλλειψη μιας τυποποιημένης γλώσσας ενορχήστρωσης για deployments σε edge και cloud, ενώ οι υπάρχουσες λύσεις είναι πλατформο-εξαρτημένες, περιορίζοντας τη διαλειτουργικότητα και την ευελιξία στην επιλογή πόρων. Επιπλέον, η κατακερματισμένη διαχείριση μικροϋπηρεσιών εμποδίζει την ενιαία εποπτεία, την αντιμετώπιση προβλημάτων και τη βέλτιστη κατανομή εργασιών και πόρων.[53], [54]

Η απουσία προτύπων και διαλειτουργικότητας στην ενορχήστρωση πολυεπίπεδων υπολογιστικών περιβαλλόντων αποτελεί μια σημαντική πρόκληση. Η αντιμετώπισή της απαιτεί ανοικτά πρότυπα και κοινά εργαλεία για την απρόσκοπτη συνεργασία μεταξύ τεχνολογιών, η οποία είναι κρίσιμη για την αξιοποίηση καταναμεμημένων αρχιτεκτονικών και την επιτάχυνση της καινοτομίας.

Η αποτελεσματική ενορχήστρωση containers έρχεται αντιμέτωπη με αντικρουόμενες απαιτήσεις ποιότητας υπηρεσίας (QoS), όπου η βελτιστοποίηση μιας πτυχής μπορεί να επηρεάσει αρνητικά μια άλλη. Η διατήρηση μιας ισορροπίας είναι ζωτικής σημασίας για την αποδοτικότητα των καταναμεμημένων εφαρμογών.

Βασικές αντικρουόμενες απαιτήσεις περιλαμβάνουν:

- Μείωση Κόστους έναντι Υψηλής Αξιοπιστίας/Διαθεσιμότητας: Η επιδίωξη της οικονομίας μπορεί να οδηγήσει σε λιγότερους ή λιγότερο αξιόπιστους πόρους, ενώ η εξασφάλιση υψηλής αξιοπιστίας απαιτεί πλεονασμό και αυτόματη ανάκαμψη, αυξάνοντας το κόστος. Είναι απαραίτητη μια ισορροπία που να βασίζεται στις απαιτήσεις uptime και στον προϋπολογισμό, με δυναμική κλιμάκωση και γεωγραφικά καταναμεμημένα deployments να αποτελούν βοηθητικές λύσεις.
- Ελαχιστοποίηση Latency έναντι Βέλτιστης Χρήσης Πόρων: Η ελαχιστοποίηση της καθυστέρησης απαιτεί ανάπτυξη κοντά στους χρήστες ή χρήση αποκλειστικών πόρων, κάτι που ενδέχεται να οδηγήσει σε υπο-αξιοποίηση των πόρων. Αντίστοιχα, η βέλτιστη χρήση πόρων μέσω πυκνοποίησης μπορεί να αυξήσει την καθυστέρηση. Απαιτείται μια στρατηγική κατανομής πόρων που να ικανοποιεί τις συμφωνίες επιπέδου υπηρεσιών (SLAs) απόδοσης χωρίς να οδηγεί σε σπατάλη πόρων.
- Ενεργειακή Απόδοση έναντι Υψηλής Απόδοσης Εφαρμογής: Η ενεργειακή απόδοση μπορεί να επιτευχθεί μέσω της ενοποίησης φόρτων εργασίας ή της μείωσης της συχνότητας λειτουργίας, αλλά αυτό μπορεί να επηρεάσει την απόδοση κατά τη διάρκεια περιόδων αιχμής. Συνεπώς, απαιτούνται ευφυείς μηχανισμοί διαχείρισης πόρων που προσαρμόζονται στις ανάγκες της εφαρμογής, διατηρώντας παράλληλα την ενεργειακή απόδοση.

Η ικανότητα ενός συστήματος ενορχήστρωσης containers να διαχειρίζεται αποτελεσματικά αυτές τις αντισταθμίσεις (trade-offs) είναι καθοριστικής σημασίας για την επιτυχία των σύγχρονων cloud-native εφαρμογών. Οι σύγχρονοι ενορχηστρωτές παρέχουν ποικίλους

μηχανισμούς και πολιτικές που επιτρέπουν στους χρήστες να καθορίζουν τις προτεραιότητές τους και να επιτυγχάνουν την επιθυμητή ισορροπία μεταξύ αυτών των αντικρουόμενων στόχων.

Τα υπάρχοντα μοντέλα scaling και scheduling συχνά αδυνατούν να διαχειριστούν ταυτόχρονα πολλαπλούς περιορισμούς QoS, όπως καθυστέρηση, απώλεια πακέτων και διαθεσιμότητα. Επιπλέον, στερούνται ευέλικτων μηχανισμών σταθμισμένης βελτιστοποίησης, οδηγώντας σε υπο-οπτιμαλ αποφάσεις όπου η βελτιστοποίηση ενός QoS μπορεί να θυσιάσει άλλα. Η ανάγκη για μοντέλα που λαμβάνουν υπόψη σύνθετους περιορισμούς QoS και προσφέρουν σταθμισμένη βελτιστοποίηση είναι κρίσιμη για την αποδοτικότητα των κατανεμημένων συστημάτων. Η ενорχήστρωση containerized εφαρμογών ενσωματώνει ευφυή χαρακτηριστικά μέσω τεχνικών AI/ML για προβλεπτική λήψη αποφάσεων, προσαρμοστικότητα και βελτιστοποίηση πόρων με βάση πολλαπλά κριτήρια QoS.

Η ενισχυτική μάθηση (RL) αποτελεί ένα ισχυρό εργαλείο για την αυτοματοποιημένη διαχείριση κατανεμημένων συστημάτων. Αντί για στατικές πολιτικές, η RL επιτρέπει στον ενорχηστρωτή να μαθαίνει μέσω δοκιμών, αξιολογώντας τις επιπτώσεις των ενεργειών του (π.χ., scaling, κατανομή πόρων) και προσαρμόζοντας τις αποφάσεις του για βελτιστοποίηση. Ο ενорχηστρωτής (agent) παρατηρεί την κατάσταση του περιβάλλοντος, επιλέγει ενέργειες και λαμβάνει μια ανταμοιβή ανάλογα με την επιτυχία σε σχέση με έναν στόχο (π.χ., κόστος, απόκριση, αξιοπιστία). Μέσω επαναλαμβανόμενων αλληλεπιδράσεων, ο agent διαμορφώνει μια βέλτιστη πολιτική για τη λήψη αποφάσεων ενорχήστρωσης.[40]

4.3.3 Εφαρμογή της Ενισχυτικής Μάθησης (RL) στην Κλιμάκωση Containers

Η χρήση της ενισχυτικής μάθησης (RL) στην ενорχήστρωση containers αναδεικνύεται ως μια πολλά υποσχόμενη προσέγγιση για την αυτοματοποιημένη και βελτιστοποιημένη διαχείριση cloud-native εφαρμογών. Σύμφωνα με το άρθρο των Wang et al. (2023) με τίτλο "Container Scaling Strategy Based on Reinforcement Learning", ένα τυπικό πλαίσιο για τη μοντελοποίηση της ενорχήστρωσης μέσω RL είναι το Markov Decision Process (MDP). Ένα MDP χαρακτηρίζεται από τα ακόλουθα στοιχεία:

- **Σύνολο Καταστάσεων (States):** Αυτές αναπαριστούν τις πιθανές καταστάσεις του συστήματος, όπως το τρέχον φορτίο εργασίας κάθε container, η χρήση CPU και μνήμης σε κάθε κόμβο, οι χρόνοι απόκρισης των υπηρεσιών και ενδεχομένως μετρικές κόστους.
- **Σύνολο Ενεργειών (Actions):** Αυτές περιλαμβάνουν τις διαθέσιμες ενέργειες που μπορεί να εκτελέσει ο ενорχηστρωτής. Παραδείγματα είναι η αύξηση ή μείωση του αριθμού των αντιγράφων ενός container (οριζόντια κλιμάκωση), η προσαρμογή των πόρων (CPU, μνήμη) που διατίθενται σε ένα container (κάθετη κλιμάκωση) ή η μετακίνηση containers σε διαφορετικούς κόμβους.
- **Συνάρτηση Μετάβασης (Transition Function):** Αυτή καθορίζει την πιθανότητα μετάβασης από μια δεδομένη κατάσταση σε μια επόμενη, μετά την εκτέλεση μιας συγκεκριμένης ενέργειας. Λαμβάνει υπόψη την εγγενή αβεβαιότητα και τη δυναμική των κατανεμημένων συστημάτων.
- **Συνάρτηση Ανταμοιβής (Reward Function):** Αυτή ορίζει την άμεση ανταμοιβή που λαμβάνει ο agent όταν εκτελεί μια ενέργεια σε μια συγκεκριμένη κατάσταση. Η συνάρτηση ανταμοιβής είναι ζωτικής σημασίας για την καθοδήγηση της διαδικασίας

- μάθησης προς τους επιθυμητούς στόχους.
- Συντελεστής Απόσβεσης (Discount Factor): Αυτός καθορίζει την αξία που αποδίδεται στις μελλοντικές ανταμοιβές σε σύγκριση με τις άμεσες ανταμοιβές.

Για την επίλυση του MDP και την εξεύρεση της βέλτιστης πολιτικής κλιμάκωσης, χρησιμοποιούνται τεχνικές ενισχυτικής μάθησης, με το Q-learning να μαθαίνει μια συνάρτηση Q-value που αξιολογεί τη συνολική ανταμοιβή για κάθε δυνατή ενέργεια σε κάθε κατάσταση. Η διαδικασία περιλαμβάνει τη συνεχή ανανέωση των Q-values μέσω μιας ισορροπίας μεταξύ εξερεύνησης και εκμετάλλευσης. Ένα σημαντικό πρόβλημα είναι η διαχείριση του δυνητικά μεγάλου χώρου καταστάσεων-ενεργειών, για την οποία αξιοποιούνται τεχνικές όπως η παραμετροποίηση, η αφαίρεση και μέθοδοι προσέγγισης συναρτήσεων όπως τα Δίκτυα Βαθιάς Μάθησης Q (DQN). Ο απώτερος σκοπός είναι η εκπαίδευση ενός agent κλιμάκωσης ικανού να λαμβάνει αυτόνομες αποφάσεις σχετικά με την κατανομή και την προσαρμογή των πόρων, προβλέποντας τις ανάγκες και αποτρέποντας τη συμφόρηση για τη διασφάλιση της βέλτιστης απόδοσης.

Στην ενορχήστρωση containers, η συνάρτηση ανταμοιβής βελτιστοποιεί αντικρουόμενες παραμέτρους: κόστος (μέσω ευφυσούς scaling), καθυστέρηση (μέσω τοποθέτησης, δικτύωσης και caching) και αξιοπιστία (μέσω αναπαραγωγής, αυτόματης επανεκκίνησης, load balancing και backup). Η ισορροπία αυτών είναι κρίσιμη. Η ελαχιστοποίηση του κόστους μπορεί να αυξήσει την καθυστέρηση, ενώ η μείωση της καθυστέρησης ή η ενίσχυση της αξιοπιστίας μπορεί να αυξήσουν το κόστος. Ο σχεδιασμός αποτελεσματικής συνάρτησης ανταμοιβής απαιτεί κατανόηση των επιχειρησιακών προτεραιοτήτων και των tradeoffs μεταξύ αυτών, καθοδηγώντας τον αλγόριθμο ενορχήστρωσης στη βέλτιστη ισορροπία, λαμβάνοντας υπόψη τις δυναμικές συνθήκες και τις απαιτήσεις των εφαρμογών. Η συνεχής αξιολόγηση και προσαρμογή της είναι απαραίτητη.[55]

4.3.4 Συγκριτική Αξιολόγηση Τεχνικών και Αποτελεσμάτων

Αναλυτικά Αποτελέσματα:

- Αύξηση Απόδοσης: Παρατηρήθηκε μια σημαντική αύξηση της απόδοσης της τάξεως του 20%. Αυτό το ποσοστό αντιπροσωπεύει μια ουσιαστική βελτίωση στην παραγωγικότητα και την αποτελεσματικότητα του συστήματος ή της διαδικασίας που εξετάστηκε. Η αύξηση αυτή μπορεί να αποδοθεί σε διάφορους παράγοντες, οι οποίοι χρήζουν περαιτέρω ανάλυσης για την πλήρη κατανόησή τους.
- Βελτιωμένη Απόκριση HPA: Επιτεύχθηκε ταχύτερη απόκριση από το Horizontal Pod Autoscaler (HPA). Η μειωμένη καθυστέρηση στην απόκριση του HPA υποδηλώνει μια πιο ευέλικτη και δυναμική διαχείριση των πόρων, επιτρέποντας στο σύστημα να αντιδρά πιο άμεσα στις μεταβολές του φόρτου εργασίας.
- Καλύτερη Προσαρμογή σε Μεταβολές Φόρτου Εργασίας: Το σύστημα ή η διαδικασία κατέδειξε βελτιωμένη ικανότητα προσαρμογής σε διακυμάνσεις του φόρτου εργασίας. Αυτό σημαίνει ότι μπορεί να διαχειρίζεται αποτελεσματικά περιόδους αυξημένης ζήτησης χωρίς υποβάθμιση της απόδοσης και να μειώνει την κατανάλωση πόρων κατά περιόδους χαμηλής δραστηριότητας, οδηγώντας σε βελτιστοποίηση της λειτουργίας.

Ανάλυση Πλεονεκτημάτων:

1. Ευελιξία: Προσαρμογή χωρίς κανόνες, μάθηση από δεδομένα σε απρόβλεπτες συνθήκες.
2. Συνεχής Μάθηση: Online βελτίωση, αναγνώριση μοτίβων και προσαρμογή στρατηγικών σε άγνωστα περιβάλλοντα.

Περιορισμοί:

- Υψηλή πολυπλοκότητα: Απαιτούνται εξειδικευμένες γνώσεις για σχεδιασμό, αλγορίθμους και παραμέτρους. Χρονοβόρα και απαιτητική εκπαίδευση με μεγάλο όγκο δεδομένων.
- Εξάρτηση από σήματα ανταμοιβής: Η αποτελεσματικότητα εξαρτάται από την ποιότητα των σημάτων. Ελλιπή σήματα οδηγούν σε ανεπιθύμητες συμπεριφορές. Κρίσιμη η ακριβής διαμόρφωση συναρτήσεων ανταμοιβής.

Προβλεπτική Κλιμάκωση με Federated Learning

Η ομοσπονδιακή μάθηση επιτρέπει σε κατανεμημένους κόμβους IoT/edge να εκπαιδεύουν αυτόνομα μοντέλα μηχανικής μάθησης τοπικά, αποφεύγοντας τη μεταφορά ευαίσθητων δεδομένων. Εφαρμόζεται στην προληπτική αυτόματη κλιμάκωση πόρων σε cloud (IEEE, 2023), όπου τοπικά εκπαιδευμένα LSTM μοντέλα προβλέπουν φόρτο εργασίας και οι παράμετροί τους συγχωνεύονται, δημιουργώντας ένα βελτιωμένο καθολικό μοντέλο με προστασία ιδιωτικότητας και μειωμένη λανθάνουσα κατάσταση. Αυτή η προσέγγιση προσφέρει βελτιωμένη ακρίβεια προβλέψεων, προστασία ιδιωτικότητας, μειωμένη λανθάνουσα κατάσταση και επεκτασιμότητα, καθιστώντας την ομοσπονδιακή μάθηση σημαντική για ευφρείς και προστατευμένες εφαρμογές IoT/edge. Η προληπτική αυτόματη κλιμάκωση φόρτου εργασίας είναι ένα χαρακτηριστικό παράδειγμα των δυνατοτήτων της.

Αποτελέσματα:

- 23% μείωση του latency σε εφαρμογές web
- 17% μείωση στην κατανάλωση πόρων
- Ιδιωτικότητα διατηρείται, καθώς τα raw δεδομένα δεν μεταφέρονται

Πλεονεκτήματα:

- Προληπτική δράση αντί για αντιδραστική
- Κατάλληλο για privacy-sensitive σενάρια

Περιορισμοί:

- Χρονοβόρα διαδικασία συγχρονισμού μοντέλων
- Εξαρτάται από την ομοιογένεια των τοπικών δεδομένων[52]

4.3.5 Ρόλος της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης στην Ενορχήστρωση

Στα κατανεμημένα συστήματα με πληθώρα αισθητήρων, IoT, fog και cloud, η διαχείριση και ο προγραμματισμός εργασιών είναι σύνθετη πρόκληση λόγω αβεβαιότητας και ασάφειας δεδομένων. Οι τεχνικές soft computing, ιδιαίτερα τα Fuzzy Rule-Based Systems (FRBSs), αποτελούν ισχυρή προσέγγιση για λήψη αποφάσεων προγραμματισμού βάσει ασαφών δεδομένων, συνήθων σε δυναμικά περιβάλλοντα fog και IoT. Η διαχείριση αβεβαιότητας και ασάφειας οδηγεί σε ευέλικτους και προσαρμοστικούς μηχανισμούς προγραμματισμού που ανταποκρίνονται στις μεταβαλλόμενες συνθήκες.

Τα edge περιβάλλοντα, όπως τα δίκτυα fog και IoT, παρουσιάζουν μοναδικές προκλήσεις που καθιστούν την χρήση παραδοσιακών μεθόδων scheduling δύσκολη. Συγκεκριμένα:

- Αβεβαιότητα Δεδομένων: Οι αισθητήρες IoT συχνά παράγουν θορυβώδη ή ελλιπή δεδομένα. Επιπλέον, η ποιότητα της σύνδεσης δικτύου μεταξύ των συσκευών και των κόμβων fog μπορεί να είναι ασταθής, οδηγώντας σε καθυστερήσεις ή απώλεια δεδομένων.
- Ασαφείς Παράμετροι: Κριτήρια όπως η "εγγύτητα" ενός κόμβου ή η "χαμηλή κατανάλωση ενέργειας" είναι συχνά ποιοτικά και όχι ποσοτικά, καθιστώντας δύσκολη την άμεση χρήση αυστηρών αλγορίθμων προγραμματισμού.
- Δυναμική Τοπολογία: Η τοπολογία των edge δικτύων μπορεί να αλλάζει δυναμικά καθώς νέες συσκευές προστίθενται ή αποσυνδέονται, ή καθώς η ποιότητα των συνδέσεων μεταβάλλεται. Οι μηχανισμοί scheduling πρέπει να είναι σε θέση να προσαρμόζονται σε αυτές τις αλλαγές σε πραγματικό χρόνο.
- Πολλαπλά Κριτήρια: Οι αποφάσεις scheduling στα edge περιβάλλοντα συχνά πρέπει να λαμβάνουν υπόψη πολλαπλά, και ενδεχομένως αντικρουόμενα, κριτήρια όπως ο χρόνος εκτέλεσης, η κατανάλωση ενέργειας, η διαθεσιμότητα πόρων, η ασφάλεια και η καθυστέρηση δικτύου.

Τα Fuzzy Rule-Based Systems (FRBSs) προσφέρουν ένα ευέλικτο πλαίσιο για την αντιμετώπιση προκλήσεων, μοντελοποιώντας αβεβαιότητα μέσω ασαφών κανόνων "ΑΝ (συνθήκη) ΤΟΤΕ (ενέργεια)" για αποφάσεις προγραμματισμού που προσομοιάζουν την ανθρώπινη λογική. Σε άρθρο του MDPI (2020), η fuzzy logic χρησιμοποιείται για Interface-Aware Scheduling λαμβάνοντας υπόψη την τοπολογία δικτύου και ενσωματώνοντας soft constraints (χρόνος, ενέργεια, προσβασιμότητα) μέσω ασαφών κανόνων για ευέλικτη διαχείριση πόρων σε cloud-fog-IoT περιβάλλοντα.

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Η εφαρμογή των fuzzy rule-based systems στο scheduling σε edge περιβάλλοντα αντιπροσωπεύει μια πολλά υποσχόμενη κατεύθυνση για την αντιμετώπιση των εγγενών προκλήσεων αυτών των συστημάτων. Η ικανότητα των FRBSs να χειρίζονται ασαφή και αβέβαια δεδομένα, να ενσωματώνουν ποιοτικές πληροφορίες και να επιτρέπουν την ευέλικτη διαχείριση πολλαπλών κριτηρίων τα καθιστά ένα ισχυρό εργαλείο για την ανάπτυξη πιο έξυπνων και προσαρμοστικών μηχανισμών προγραμματισμού.

Μελλοντικές ερευνητικές κατευθύνσεις θα μπορούσαν να επικεντρωθούν στην περαιτέρω βελτίωση των FRBSs για scheduling σε edge περιβάλλοντα, όπως:

- Ανάπτυξη πιο εξελιγμένων μεθόδων για τη δημιουργία και τη βελτιστοποίηση των ασαφών κανόνων.
- Εξερεύνηση της ενσωμάτωσης των FRBSs με άλλες τεχνικές τεχνητής νοημοσύνης, όπως η μηχανική μάθηση, για την αυτόματη προσαρμογή των ασαφών κανόνων στις μεταβαλλόμενες συνθήκες του περιβάλλοντος.
- Αξιολόγηση της αποτελεσματικότητας των FRBSs σε πραγματικά edge περιβάλλοντα και σε μεγαλύτερη κλίμακα.
- Διερεύνηση της χρήσης FRBSs για την αντιμετώπιση πρόσθετων προκλήσεων scheduling, όπως η διαχείριση της κινητικότητας των συσκευών και η εξασφάλιση της ασφάλειας και της ιδιω

Πλεονεκτήματα:

- Ιδανικό για περιβάλλοντα υψηλής μεταβλητότητας
- Δεν απαιτεί μεγάλα σύνολα εκπαίδευσης

Περιορισμοί:

- Δυσκολία στη δημιουργία και συντήρηση του fuzzy ruleset
- Περιορισμένη υιοθέτηση σε συστήματα παραγωγής

4.3.6 Προηγμένες Τεχνικές και Αλγόριθμοι για Αυτοπροσαρμοζόμενους Ενορχηστρωτές

Η αξιοποίηση των Graph Neural Networks (GNNs) παρέχει μια ισχυρή μέθοδο για την ανάλυση και διαχείριση σύνθετων ιεραρχικών και δικτυωμένων σχέσεων που αναπτύσσονται μεταξύ διαφόρων υπηρεσιών και των υποκείμενων πόρων τους. Τα GNNs, με την ικανότητά τους να επεξεργάζονται δομές γραφημάτων, επιτρέπουν τη μοντελοποίηση των αλληλεξαρτήσεων, τη διάδοση πληροφοριών μεταξύ συνδεδεμένων κόμβων και την εξαγωγή σημαντικών συμπερασμάτων για τον προγραμματισμό και την κατανομή πόρων.

Στο επιστημονικό άρθρο με τίτλο "Microservice Workflow Scheduling with Resource Configuration Model under Deadline and Reliability Constraints", που δημοσιεύτηκε στο MDPI το 2025, παρουσιάζεται μια καινοτόμος προσέγγιση για τον προγραμματισμό ροών εργασίας μικροϋπηρεσιών, λαμβάνοντας υπόψη περιορισμούς χρονικών προθεσμιών και απαιτήσεων αξιοπιστίας, παράλληλα με τη διαχείριση της διαμόρφωσης των πόρων.

Στην παρούσα έρευνα αναπτύχθηκε ένα Graph Attention Network (GAT) για την εκμάθηση βέλτιστων στρατηγικών ανάθεσης πόρων σε μικροϋπηρεσίες σύνθετων ροών εργασίας, λαμβάνοντας υπόψη τις εξαρτήσεις και τις απαιτήσεις απόδοσης, αξιοπιστίας και κόστους. Επιπλέον, προτείνεται ο RMWS scheduler, ένας μηχανισμός προγραμματισμού που αξιοποιεί τις πληροφορίες του GAT για την ανάθεση και τον προγραμματισμό μικροϋπηρεσιών, επιτυγχάνοντας ισορροπία μεταξύ αξιοπιστίας και κόστους στην αποτελεσματική διαχείριση σύνθετων συστημάτων μικροϋπηρεσιών.

Βελτιωμένα Αποτελέσματα:

- **Σημαντική Επίτευξη Μείωσης Κόστους:** Από την αξιολόγηση του προτεινόμενου συστήματος χρονοπρογραμματισμού προέκυψε εντυπωσιακή μείωση κόστους, που φτάνει έως και το 44,5% συγκριτικά με τα συμβατικά συστήματα που χρησιμοποιήθηκαν ως σημείο αναφοράς. Το αξιοσημείωτο αυτό πλεονέκτημα οφείλεται στην βελτιστοποιημένη αξιοποίηση των διαθέσιμων πόρων, την αποφυγή καθυστερήσεων και την πιο αποτελεσματική διαχείριση των εργασιών.
- **Εξασφάλιση Υψηλής Αξιοπιστίας σε Περιβάλλοντα Κρίσιμων Χρονικών Απαιτήσεων:** Το σύστημα επιδεικνύει εξαιρετικά υψηλά επίπεδα αξιοπιστίας κατά την εκτέλεση ροών εργασιών με αυστηρές χρονικές προθεσμίες. Η ικανότητά του να τηρεί σταθερά τα χρονοδιαγράμματα και να ολοκληρώνει τις εργασίες εντός των προκαθορισμένων χρονικών ορίων τονίζει την ισχυρή του φύση και την καταλληλότητά του για εφαρμογές όπου ο χρόνος αποτελεί καθοριστικό παράγοντα επιτυχίας.

Μελλοντικές Κατευθύνσεις και Ερευνητικές Ευκαιρίες

Η αυξανόμενη πολυπλοκότητα των σύγχρονων υπολογιστικών περιβαλλόντων καθιστά τις στατικές μεθόδους ενορχήστρωσης ανεπαρκείς, οδηγώντας την ερευνητική κοινότητα στην αναγνώριση της ανάγκης για ευφυή και αυτοπροσαρμοζόμενα συστήματα. Η ανάπτυξη context-aware συστημάτων που επιτρέπουν τη δυναμική βελτιστοποίηση του scheduling και του scaling κρίνεται επιτακτική.

Παρά την ενσωμάτωση τεχνολογιών AI/ML, η ευρεία εφαρμογή τους παραμένει περιορισμένη, γεγονός που υποδεικνύει την ύπαρξη σημαντικών τεχνικών προκλήσεων. Η μελλοντική έρευνα αναμένεται να επικεντρωθεί στην ανάπτυξη εξελιγμένων αλγορίθμων AI/ML, σε νέες αρχιτεκτονικές, στην πρόβλεψη φόρτου εργασίας, στην αυτοματοποιημένη διαχείριση πόρων και σε εργαλεία παρακολούθησης. Στόχος είναι η δημιουργία ευφών συστημάτων ενορχήστρωσης ικανών να ανταποκριθούν στις απαιτήσεις των μελλοντικών δυναμικών περιβαλλόντων.

Η ενορχήστρωση containers στο συνεχές cloud-to-thing απαιτεί θεμελιώδεις αλλαγές προκειμένου να ανταποκριθεί στις σύνθετες απαιτήσεις απόδοσης που προκύπτουν σε ετερογενή και δυναμικά περιβάλλοντα. Οι παραδοσιακές προσεγγίσεις αποδεικνύονται ανεπαρκείς για την αποτελεσματική διαχείριση όλων των κρίσιμων παραμέτρων των σύγχρονων κατανεμημένων εφαρμογών, συμπεριλαμβανομένης της multi-objective και QoS-aware ενορχήστρωσης.

Είναι επιτακτική η βελτιστοποίηση πολλαπλών, συχνά αντικρουόμενων, κριτηρίων απόδοσης, όπως:

- **Χρόνος απόκρισης (latency):** Κρίσιμο για εφαρμογές IoT, βιομηχανικής αυτοματοποίησης και διαδραστικών υπηρεσιών. Η γεωγραφική κατανομή και η πολυπλοκότητα των δικτύων αυξάνουν την πρόκληση.
- **Ενεργειακή κατανάλωση:** Σημαντική για συσκευές edge με περιορισμένη ενέργεια. Η ενορχήστρωση πρέπει να την ελαχιστοποιεί χωρίς θυσία της απόδοσης.
- **Οικονομικό κόστος (leasing):** Η ενορχήστρωση πρέπει να λαμβάνει υπόψη τις τιμολογιακές πολιτικές του cloud και να ελαχιστοποιεί το κόστος.

- Αξιοπιστία και διαθεσιμότητα: Η συνεχής λειτουργία είναι υψίστης σημασίας. Η ενορχήστρωση πρέπει να εξασφαλίζει ανθεκτικότητα και αυτόματη ανάκαμψη.
- Τοποθεσία δεδομένων (data locality): Ρυθμιστικοί ή τεχνικοί περιορισμοί επιβάλλουν την αποθήκευση και επεξεργασία δεδομένων σε συγκεκριμένες περιοχές. Η ενορχήστρωση πρέπει να το λαμβάνει υπόψη.
- Για την αντιμετώπιση πολύπλευρων προκλήσεων, η υιοθέτηση προηγμένων τεχνικών βελτιστοποίησης για ταυτόχρονη διαχείριση πολλαπλών και αντιφατικών κριτηρίων είναι αναγκαία. Τα multi-objective optimization μοντέλα, όπως οι Pareto-optimal reinforcement learning agents (εκπαίδευση μέσω ενισχυτικής μάθησης για Pareto-optimal λύσεις που αντιπροσωπεύουν ισορροπία μεταξύ κριτηρίων) και τα Multi-criteria fuzzy systems (μοντελοποίηση αβεβαιότητας και υποκειμενικότητας για αποφάσεις ενορχήστωσης με βάση τις προτεραιότητες των κριτηρίων μέσω ασαφών κανόνων), αναδεικνύονται ελπιδοφόρες. [56]

4.3.7 Θεμελιώδεις Προϋποθέσεις για την Υλοποίηση AI στα Edge/Fog Περιβάλλοντα

Η αποτελεσματική λειτουργία μοντέλων Μηχανικής Μάθησης (ML) σε κόμβους edge και fog, οι οποίοι βρίσκονται πιο κοντά στην πηγή των δεδομένων, επιφέρει σημαντικά πλεονεκτήματα όσον αφορά τη μειωμένη καθυστέρηση και την επεξεργασία δεδομένων σε πραγματικό χρόνο. Ωστόσο, η ανάπτυξη και η διαχείριση αυτών των μοντέλων επιβάλλει την αντιμετώπιση συγκεκριμένων προκλήσεων και την ικανοποίηση ορισμένων κρίσιμων προϋποθέσεων:

Βασικές Προϋποθέσεις:

- Μικρό Αποτύπωμα Πόρων (RAM, CPU): Λόγω των περιορισμένων πόρων που συνήθως διαθέτουν οι edge και fog κόμβοι (σε σύγκριση με τα κεντρικά data centers), τα αναπτυσσόμενα ML μοντέλα οφείλουν να έχουν minimal απαιτήσεις σε μνήμη (RAM) και υπολογιστική ισχύ (CPU). Αυτός ο περιορισμός καθιστά απαραίτητη τη χρήση ελαφριών αρχιτεκτονικών μοντέλων και τεχνικών συμπίεσης.
- Επεξήγηση Αποφάσεων (Explainability) για Διαφάνεια και Συμμόρφωση (π.χ. GDPR): Στα πλαίσια της διασφάλισης της διαφάνειας και της συμμόρφωσης με κανονισμούς όπως ο Γενικός Κανονισμός Προστασίας Δεδομένων (GDPR), είναι ζωτικής σημασίας η ικανότητα των ML μοντέλων να παρέχουν εξηγήσεις για τις αποφάσεις τους. Η επεξηγησιμότητα βοηθά στην κατανόηση της λογικής πίσω από τις προβλέψεις και στην αντιμετώπιση πιθανών προκαταλήψεων ή σφαλμάτων.
- Real-time Inference με Ελάχιστη Καθυστέρηση: Ένας από τους βασικούς λόγους για την ανάπτυξη ML μοντέλων σε edge/fog κόμβους είναι η δυνατότητα πραγματοποίησης συμπερασμάτων (inference) σε πραγματικό χρόνο, με την ελάχιστη δυνατή καθυστέρηση. Αυτή η δυνατότητα είναι κρίσιμη για εφαρμογές που απαιτούν άμεση ανταπόκριση, όπως συστήματα αυτόνομης οδήγησης, βιομηχανική αυτοματοποίηση και ανίχνευση ανωμαλιών σε πραγματικό χρόνο.

4.3.8 Αρχιτεκτονικά Μοντέλα για Ευφυή Ενορχήστρωση στο Cloud-to-Thing Continuum

Για την περαιτέρω ανάπτυξη των ML μοντέλων σε edge/fog περιβάλλοντα, μελλοντική έρευνα μπορεί να επικεντρωθεί σε TinyML τεχνικές (Quantized NN, Pruning) για περιορισμένους πόρους, Explainable AI για διαφανή edge scheduling αποφάσεων, και εκπαίδευση edge μοντέλων σε πραγματικό χρόνο μέσω continual learning για προσαρμοστικότητα. Η επιτυχής αξιοποίηση απαιτεί ολιστική προσέγγιση λαμβάνοντας υπόψη πόρους, επεξηγησιμότητα και real-time απόκριση.

Η διαχείριση πολύπλοκων καταναμημένων συστημάτων και cloud απαιτεί αυτόνομους orchestrators για βελτίωση απόδοσης, αξιοπιστίας και ανθεκτικότητας. Ο βρόχος MAPE-K, με ευφυείς τεχνικές, επιτυγχάνει αυτονομία.

Βασικές Λειτουργίες Έξυπνου MAPE-K:

- Συνεχής Παρακολούθηση: Συλλογή κρίσιμων μετρήσεων από υποδομή, φόρτους εργασίας και QoS σε πραγματικό χρόνο.
- Ανάλυση με ML/AI: Αναγνώριση προτύπων, ανίχνευση ανωμαλιών και πρόβλεψη τάσεων για εξαγωγή συμπερασμάτων.
- Προσαρμοστικές Ενέργειες (Plan & Execute): Δυναμική ανακατανομή φόρτων, αυτόματη κλιμάκωση, βελτιστοποίηση και αυτοθεραπεία βάσει ανάλυσης.
- Συνεχής Ανατροφοδότηση Γνώσης: Αποθήκευση και αξιοποίηση γνώσης για συνεχή βελτίωση μοντέλων ML/AI, κανόνων και στρατηγικών ενορχήστρωσης.

Περαιτέρω Ερευνητικές Κατευθύνσεις:

Η εξέλιξη των έξυπνων βρόχων MAPE-K περιλαμβάνει την ενσωμάτωση agents ενισχυτικής μάθησης για πιο σύνθετες αποφάσεις και συστημάτων federated learning για εκπαίδευση καταναμημένων μοντέλων ML/AI με προστασία ιδιωτικότητας. Η ενίσχυση της αυτονομίας των orchestrators είναι κρίσιμη για τη διαχείριση σύνθετων καταναμημένων συστημάτων, καθώς οι παραδοσιακές προσεγγίσεις υστερούν στην αντιμετώπιση των προκλήσεων των cloud, fog και IoT αρχιτεκτονικών. Η έρευνα σε ML/AI και καταναμημένους αλγορίθμους υπόσχεται βελτιώσεις στην αυτονομία και την προσαρμογή των orchestrators.

Ένα θεμελιώδες στοιχείο για την επιτυχή ανάπτυξη και διαχείριση εφαρμογών σε cloud-fog-IoT περιβάλλοντα είναι η ενδεδειγμένη κατανόηση των διαφορετικών τύπων συνδέσεων που εμπλέκονται. Αυτές περιλαμβάνουν, μεταξύ άλλων:

- Συνδέσεις Cloud-to-Edge: Αυτές οι συνδέσεις αφορούν την επικοινωνία μεταξύ των κεντρικών υποδομών cloud και των συσκευών ή κόμβων fog που βρίσκονται πιο κοντά στην άκρη του δικτύου. Η φύση αυτών των συνδέσεων μπορεί να ποικίλλει σημαντικά ανάλογα με την απόσταση, την υποδομή δικτύωσης και τις απαιτήσεις εύρους ζώνης και καθυστέρησης της εφαρμογής.
- Συνδέσεις Edge-to-IoT: Αυτές οι συνδέσεις αφορούν την επικοινωνία μεταξύ των κόμβων fog και των πληθωρικών συσκευών IoT. Οι τεχνολογίες επικοινωνίας εδώ μπορεί να είναι εξαιρετικά ποικίλες, συμπεριλαμβανομένων πρωτοκόλλων όπως Bluetooth, Zigbee, LoRaWAN, καθώς και πιο παραδοσιακών όπως Ethernet ή Wi-Fi.

Οι περιορισμοί ισχύος, η διαλειτουργικότητα και η ασφάλεια είναι συχνά κρίσιμα ζητήματα σε αυτό το επίπεδο.

- Συνδέσεις μεταξύ κόμβων Edge: Σε ορισμένες περιπτώσεις, μπορεί να απαιτείται άμεση επικοινωνία μεταξύ διαφορετικών κόμβων fog για τη συνεργασία στην επεξεργασία δεδομένων ή την εκτέλεση λειτουργιών.
- Εσωτερικές συνδέσεις εντός κόμβων: Ακόμη και εντός ενός κόμβου fog ή μιας συσκευής IoT, υπάρχουν εσωτερικές συνδέσεις μεταξύ διαφορετικών υποσυστημάτων που πρέπει να λαμβάνονται υπόψη.

Η κατανόηση των χαρακτηριστικών κάθε είδους σύνδεσης, όπως το διαθέσιμο εύρος ζώνης, η αναμενόμενη καθυστέρηση, η αξιοπιστία και το κόστος, είναι ζωτικής σημασίας για τον σχεδιασμό ανθεκτικών και αποδοτικών κατανεμημένων εφαρμογών.

Η αποτελεσματική διαχείριση εφαρμογών σε cloud-fog-IoT απαιτεί προσαρμοσμένες στρατηγικές εντοπισμού, καθώς οι παραδοσιακές μέθοδοι δεν ανταποκρίνονται στις προκλήσεις αυτών των κατανεμημένων τοπολογιών. Οι ευέλικτες στρατηγικές πρέπει να λαμβάνουν υπόψη την καθυστέρηση, το εύρος ζώνης, την κινητικότητα, την αξιοπιστία, την ασφάλεια, τη διαχείριση πόρων και το κόστος, επιτρέποντας την τοποθέτηση εργασιών κοντά στην πηγή δεδομένων, τη βελτιστοποίηση της χρήσης του δικτύου, τη διαχείριση της κινητικότητας συσκευών, την αντιμετώπιση αστοχιών, την ενσωμάτωση μέτρων ασφαλείας, τη δυναμική κατανομή πόρων και τη μείωση του κόστους. Η ενσωμάτωση αυτών των παραμέτρων είναι απαραίτητη για την πλήρη αξιοποίηση των δυνατοτήτων των κατανεμημένων αρχιτεκτονικών. Οι μελλοντικοί schedulers πρέπει να είναι topology-aware, mobility-aware και context-aware για αποτελεσματικότερη διαχείριση των μικροϋπηρεσιών.[57]

Ένας topology-aware scheduler κατανοεί την τοπολογία δικτύου/υποδομής (διασυνδέσεις, latency, bandwidth). Επιλέγει την κατάλληλη διεπαφή δικτύου για κάθε μικροϋπηρεσία βάσει απαιτήσεων και συμφόρησης, τοποθετεί σχετικές μικροϋπηρεσίες κοντά για βελτιστοποίηση επικοινωνίας (μείωση latency, αύξηση απόδοσης, μείωση πόρων) και τοποθετεί αντίγραφα σε διαφορετικές ζώνες για ανθεκτικότητα/διαθεσιμότητα.

Σε κινητές συσκευές, ο προγραμματιστής λαμβάνει υπόψη την κινητικότητα και τις μεταβαλλόμενες συνθήκες δικτύωσης (πρόβλεψη κίνησης, δυναμική μετεγκατάσταση για χαμηλή καθυστέρηση/αξιόπιστη σύνδεση, διαχείριση αποσυνδέσεων).

- Context-aware Scheduling:

Ένας context-aware scheduler λαμβάνει υπόψη runtime δεδομένα και το περιβάλλον εκτέλεσης μικροϋπηρεσιών. Παρακολουθεί πόρους (CPU, μνήμη, δίσκος, δίκτυο), αναγνωρίζει τον φόρτο εργασίας, επιδιώκει ενεργειακή απόδοση, εφαρμόζει πολιτικές και αντιδρά σε συμβάντα (αστοχίες, αλλαγές ζήτησης, αναβαθμίσεις) για συνεχή λειτουργία και απόδοση. Οι μελλοντικοί schedulers θα είναι ευφυείς και προσαρμοστικοί, λαμβάνοντας αποφάσεις τοποθέτησης με βάση πολλούς παράγοντες (τοπολογία, κινητικότητα, πλαίσιο) για μέγιστη απόδοση, αξιοπιστία και αποδοτικότητα κατανεμημένων συστημάτων.

- Ενιαία αρχιτεκτονική για cross-layer και multi-domain orchestration:

Η διαχείριση εφαρμογών σε πολλαπλές υποδομές (δημόσια/ιδιωτικά νέφη, edges, ISPs, κέντρα δεδομένων) είναι σύνθετη και υπολείπεται ερευνητικά. Οι υπάρχουσες λύσεις δεν καλύπτουν επαρκώς την αποτελεσματική και συντονισμένη διαχείριση σε αυτό το πολυεπίπεδο περιβάλλον, εμποδίζοντας την αξιοποίηση των κατανεμημένων αρχιτεκτονικών και δημιουργώντας προκλήσεις στην απόδοση, την αξιοπιστία και την ασφάλεια.

Για την αντιμετώπιση των υπαρχόντων ερευνητικών κενών και τη βελτίωση της ενοποιημένης διαχείρισης εφαρμογών σε πολλαπλά επίπεδα και domains, η μελλοντική έρευνα θα μπορούσε να επικεντρωθεί στην ανάπτυξη και την εξέλιξη των ακόλουθων τομέων:

- **Οριζόντια ενορχήστρωση:** Δημιουργία μηχανισμών για απρόσκοπτη διαχείριση εφαρμογών σε αυτόνομα Kubernetes ή πολλαπλά clouds (federated Kubernetes, multi-cloud meshing). Έρευνα για διαχείριση ταυτότητας, δεδομένων, πολιτικών, τοποθέτησης φόρτου και επικοινωνίας μεταξύ clusters/clouds.
- **Κατακόρυφη ενορχήστρωση:** Ολιστική διαχείριση εφαρμογής, υποδομής και δικτύου. Ενοποίηση διαχείρισης αυτών των επιπέδων για βελτιστοποίηση. Έρευνα για τεχνικές cross-layer optimization (εφαρμογή, εικονικές μηχανές/containers, δίκτυο, αποθήκευση), όπως δυναμική προσαρμογή πόρων ή βελτιστοποίηση τοπολογίας δικτύου.
- **Προτυποποιημένα APIs και αφαιρέσεις:** Ανάπτυξη και καθιέρωση κοινών προτύπων και αφαιρέσεων για συνεπή ενορχήστρωση σε cloud, fog και IoT, ανεξάρτητα από πάροχο/τεχνολογία. Διευκόλυνση ανάπτυξης εργαλείων για διαχείριση εφαρμογών σε ετερογενή περιβάλλοντα (υπολογιστικοί πόροι, δικτύωση, αποθήκευση, ασφάλεια).

Η πρόοδος σε αυτούς τους ερευνητικούς τομείς θα συμβάλει στην υπέρβαση των υφιστάμενων εμποδίων και θα οδηγήσει στην ανάπτυξη πιο αποτελεσματικών και ευέλικτων λύσεων για τη συντονισμένη διαχείριση εφαρμογών που εκτείνονται σε πολλαπλά layers και domains.

Η προοπτική ενορχήστρωση εφαρμογών σε περιβάλλον containerized στο continuum cloud-to-thing επιτάσσει την υιοθέτηση ευφών, διαφανών, προγνωστικών και αυτορρυθμιζόμενων μηχανισμών, με τη χρήση τεχνολογιών Τεχνητής Νοημοσύνης/Μηχανικής Μάθησης και αρχιτεκτονικών με επίκεντρο το edge computing. Η διασφάλιση της ιδιωτικότητας και της Ποιότητας Υπηρεσιών (QoS) αναδεικνύεται ως καίριας σημασίας σε κατανεμημένα και διαρκώς μεταβαλλόμενα περιβάλλοντα. Η ερευνητική προσπάθεια θα εστιάσει στην ανάπτυξη ολοκληρωμένων λύσεων για την αποτελεσματική ενορχήστρωση σε αυτό το συνεχές, συνδυάζοντας αρμονικά την Τεχνητή Νοημοσύνη/Μηχανική Μάθηση, το edge computing, την προστασία της ιδιωτικότητας και την Ποιότητα Υπηρεσιών.

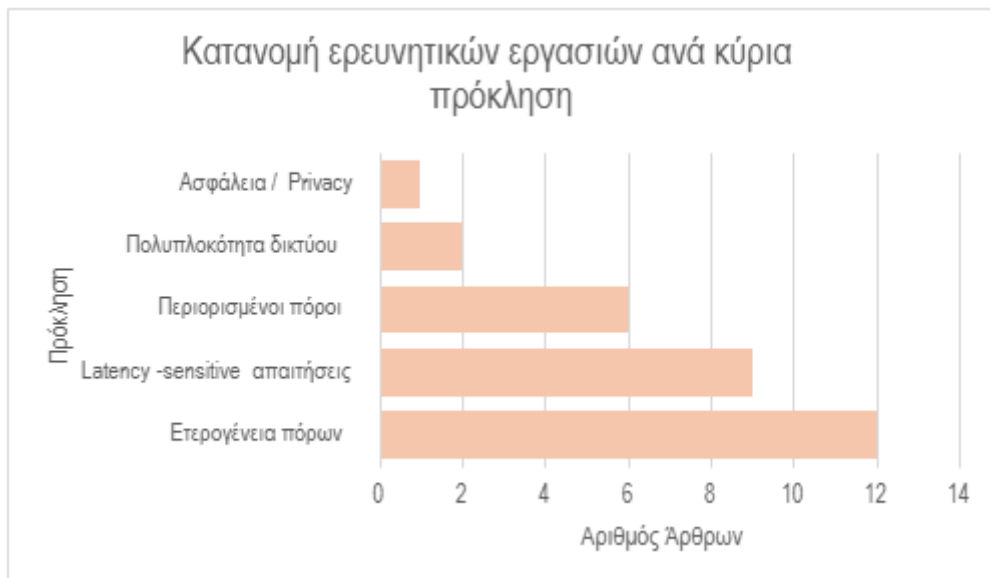
4.3.9 Συνοπτική Αξιολόγηση

Η εμβάθυνση στο τρίτο ερευνητικό ερώτημα αναδεικνύει μια κρίσιμη αλλαγή στην ενορχήστρωση containerized εφαρμογών, από το cloud έως το άκρο. Η παραδοσιακή, στατική και ανελαστική διαχείριση σταδιακά αντικαθίσταται από ευφυέστερα, αυτόνομα και δυναμικά συστήματα.

Η μετάβαση αυτή υπαγορεύεται από:

- Την αυξανόμενη ανομοιογένεια των υποδομών (cloud, edge data centers, IoT) με διαφορετικές δυνατότητες, bandwidth, αξιοπιστία και διαθεσιμότητα.
- Την εγγενή αβεβαιότητα στο edge λόγω αστάθειας δικτύου, περιορισμένων πόρων και απρόβλεπτων περιβαλλοντικών συνθηκών, καθιστώντας ανεπαρκή τα στατικά μοντέλα.
- Τις αυξημένες απαιτήσεις ποιότητας υπηρεσίας (QoS) των σύγχρονων εφαρμογών (χαμηλή καθυστέρηση, υψηλή διαθεσιμότητα), που απαιτούν ευφυείς μηχανισμούς ενορχήστρωσης για την αποφυγή υποβάθμισης της εμπειρίας του χρήστη.

Η εικόνα 4.3.1 συνοψίζει τις κύριες προκλήσεις που καταγράφηκαν στη βιβλιογραφία. Η ετερογένεια πόρων αναδεικνύεται ως η κυρίαρχη πρόκληση (40%), ενώ οι απαιτήσεις χαμηλής καθυστέρησης ακολουθούν (30%). Αντιθέτως, η ασφάλεια και ιδιωτικότητα καλύπτονται ελάχιστα (3%), αναδεικνύοντας ένα σαφές ερευνητικό κενό.



Εικόνα 4.3.1 Κατανομή των ερευνητικών εργασιών του RQ3 ανά κύρια πρόκληση που στοχεύουν να αντιμετωπίσουν οι προτεινόμενες τεχνικές ενορχήστρωσης.

Στο πλαίσιο αυτό, η ενσωμάτωση τεχνητής νοημοσύνης (AI) και μηχανικής μάθησης (ML) θεωρείται στρατηγικά αναγκαία. Οι AI/ML αλγόριθμοι προσφέρουν τη δυνατότητα δυναμικής ανάλυσης δεδομένων σε πραγματικό χρόνο, πρόβλεψης πιθανών αστοχιών και αυτόνομης λήψης αποφάσεων για ανακατανομή πόρων, scaling και αυτο-ίαση, διασφαλίζοντας έτσι την επιθυμητή ποιότητα υπηρεσίας σε ιδιαίτερα απαιτητικά περιβάλλοντα.

Παρά την πρόοδο, η ευρεία υιοθέτηση εξελιγμένων τεχνικών (όπως ενισχυτική μάθηση, federated learning, fuzzy logic και γραφικά νευρωνικά δίκτυα) παραμένει περιορισμένη. Οι προκλήσεις περιλαμβάνουν την υψηλή υπολογιστική απαίτηση, τη δυσκολία επεξηγησιμότητας των αποφάσεων, την προσαρμογή σε περιορισμένους πόρους και τα ελλιπή πρότυπα διαλειτουργικότητας.

Η επόμενη γενιά των orchestrators θα βασίζεται σε:

- Πολυκριτηριακούς και context-aware μηχανισμούς scheduling, ικανούς να λαμβάνουν αποφάσεις σε πραγματικό χρόνο με πλήρη περιβαλλοντική επίγνωση.
- Ελαφριές και επεξηγήσιμες AI λύσεις, κατάλληλες για λειτουργία σε περιορισμένα περιβάλλοντα.
- Συνεργατικές και ομοσπονδιακές αρχιτεκτονικές, που θα υποστηρίζονται από κοινά APIs και πρότυπα για το cloud-fog-IoT continuum.

Η μετάβαση αυτή σηματοδοτεί μια βαθιά αλλαγή παραδείγματος, η οποία απαιτεί ολιστική επαναξιολόγηση των αρχών διαχείρισης κατανεμημένου λογισμικού, με έμφαση στην αυτοματοποίηση, την επεκτασιμότητα, την ανθεκτικότητα και τη βιώσιμη χρήση των πόρων. Η πρόοδος στον τομέα αυτό συνδέεται άρρηκτα με τη μακροπρόθεσμη βιωσιμότητα των ψηφιακών υποδομών, καθώς η έξυπνη διαχείριση θα αποτελέσει βασικό εργαλείο για τη βελτιστοποίηση απόδοσης και την υπεύθυνη κατανάλωση πόρων.

Η ανάπτυξη αυτών των λύσεων απαιτεί:

- Συνεχή επένδυση σε έρευνα και καινοτομία.
- Διεπιστημονική συνεργασία μεταξύ ακαδημαϊκών, βιομηχανίας και φορέων.
- Ένα ευέλικτο και προοδευτικό ρυθμιστικό πλαίσιο, που θα επιτρέπει την ασφαλή και υπεύθυνη εφαρμογή των νέων τεχνολογιών.

5. *Συζήτηση & Συγκριτική Αξιολόγηση*

5.1 *Συγκριτική Ανάλυση Πλατφορμών και Τεχνικών Ενορχήστρωσης*

Η ενορχήστρωση containerized εφαρμογών στο Cloud-to-Thing Continuum αποτελεί μια κρίσιμη πτυχή της σύγχρονης αρχιτεκτονικής πληροφορικής, απαιτώντας την επιλογή μιας ισχυρής και ευέλικτης πλατφόρμας. Η εν λόγω επιλογή πρέπει να ανταποκρίνεται αποτελεσματικά στις ποικίλες ανάγκες υποδομής, καθώς και στις αυστηρές απαιτήσεις απόδοσης, ασφάλειας και ευελιξίας που επιβάλλει το περιβάλλον Cloud-to-Thing. Μέσω μιας συστηματικής ανασκόπησης των διαθέσιμων ερευνών και μελετών, αναδείχθηκαν διάφορες πλατφόρμες ενορχήστρωσης που έχουν αποδειχθεί ιδιαίτερα δημοφιλείς και αποτελεσματικές στον τομέα.

Συγκεκριμένα, οι Kubernetes ξεχωρίζουν ως η πλέον διαδεδομένη και ευρέως χρησιμοποιούμενη πλατφόρμα, προσφέροντας εκτεταμένες δυνατότητες διαχείρισης, κλιμάκωσης και αυτοματοποίησης για containerized εφαρμογές. Επιπλέον, το Docker Swarm παρουσιάζει μια απλή και εύχρηστη εναλλακτική λύση, ιδανική για μικρότερες εγκαταστάσεις ή για περιβάλλοντα που απαιτούν γρήγορη εγκατάσταση και λειτουργία. Το OpenShift, βασισμένο στην τεχνολογία Kubernetes, προσφέρει μια ολοκληρωμένη πλατφόρμα για ανάπτυξη, λειτουργία και διαχείριση εφαρμογών στο cloud, ενσωματώνοντας εργαλεία για CI/CD και monitoring. Το Apache Mesos παρέχει ένα ισχυρό πλαίσιο για τη διαχείριση και ενορχήστρωση εφαρμογών σε μεγάλα clusters, ενώ το Nomad προσφέρει μια απλή και ευέλικτη λύση για deployment εφαρμογών σε διάφορα περιβάλλοντα. Τέλος, οι ελαφριές υλοποιήσεις όπως το K3s, προορίζονται για edge computing ή IoT περιβάλλοντα, όπου απαιτείται μειωμένο αποτύπωμα πόρων και γρήγορη εκκίνηση. Η επιλογή της κατάλληλης πλατφόρμας ενορχήστρωσης εξαρτάται από τις συγκεκριμένες ανάγκες και απαιτήσεις κάθε έργου, καθώς και από την πολυπλοκότητα και την κλίμακα της εφαρμογής.

Η σύγκριση των πλατφορμών πραγματοποιείται με βάση τα εξής κριτήρια:

Scalability: ικανότητα διαχείρισης δυναμικών φορτίων και ανάπτυξης σε πολλαπλούς κόμβους.

Υποστήριξη edge/fog: καταλληλότητα για λειτουργία σε περιβάλλοντα με περιορισμένους πόρους ή αυξημένες απαιτήσεις latency.

Δυνατότητες ενσωμάτωσης AI/ML: ύπαρξη υποδομών για αυτόματη προσαρμογή (autoscaling/scheduling) με χρήση τεχνητής νοημοσύνης.

Fault tolerance & reliability: ικανότητα αυτόματης ανάκαμψης, ανακατανομής φορτίου και αποφυγής σημείων αποτυχίας.

Χρήση στην πράξη (production readiness): βαθμός υιοθέτησης σε εμπορικά ή βιομηχανικά περιβάλλοντα.

Πλατφόρμα	Scalability	Υποστήριξη Edge/Fog	AI/ML Capabilities	Fault Tolerance	Εφαρμογή στην πράξη
Kubernetes	Πολύ υψηλή	Μέτρια (με K3s/Knative)	Μέσω Kubeflow, KEDA	Πολύ υψηλή	Βιομηχανικά πρότυπα
Docker Swarm	Μέτρια	Περιορισμένη	Μη ενσωματωμένη	Μέτρια	Απλές cloud εφαρμογές
OpenShift	Υψηλή	Μέτρια	Εμπορικά εργαλεία ML (Red Hat AI)	Πολύ υψηλή	Enterprise περιβάλλοντα
Apache Mesos	Υψηλή	Περιορισμένη	Όχι native	Υψηλή	Παλαιότερα deployments
Nomad	Υψηλή	Υψηλή (modular)	Προαιρετικά μέσω plugin	Υψηλή	Αναδυόμενη χρήση σε edge
K3s	Μέτρια	Πολύ υψηλή	Περιορισμένη υποστήριξη	Μέτρια	Κατάλληλο για IoT/edge

Πίνακας 5.1.1 – Συγκριτική αξιολόγηση πλατφορμών ενορχήστρωσης containerized

Το Kubernetes, όντας η πιο ώριμη και ευρέως διαδεδομένη λύση ενορχήστρωσης container, παρουσιάζει μια ολοκληρωμένη σουίτα χαρακτηριστικών και δυνατοτήτων. Η αρχιτεκτονική του επιτρέπει την αυτόματη κλιμάκωση, την αυτο-επούλωση, και την ευέλικτη διαχείριση εφαρμογών που βασίζονται σε container. Επιπλέον, η δυνατότητα επέκτασης μέσω custom resources και operators το καθιστά εξαιρετικά προσαρμόσιμο σε διάφορες ανάγκες. Ενδεικτικά, το KEDA παρέχει μηχανισμούς για event-driven autoscaling, ενώ το Kubeflow διευκολύνει τη δημιουργία και διαχείριση pipelines μηχανικής μάθησης.

Ωστόσο, η πολυπλοκότητα του Kubernetes μπορεί να αποδειχθεί αποθαρρυντική, ειδικά για ομάδες με περιορισμένη εμπειρία σε συστήματα ενορχήστρωσης. Η εγκατάσταση, ρύθμιση, και συντήρηση ενός Kubernetes cluster απαιτεί σημαντική τεχνογνωσία και πόρους. Επιπροσθέτως, η υψηλή κατανάλωση πόρων του Kubernetes το καθιστά λιγότερο κατάλληλο για περιβάλλοντα με περιορισμένους υπολογιστικούς πόρους, όπως τα edge deployments. Το K3s, που αναπτύχθηκε για να αντιμετωπίσει αυτές τις προκλήσεις, αποτελεί μια ελαφριά έκδοση του Kubernetes, σχεδιασμένη για περιβάλλοντα edge και IoT. Έχει απλοποιημένη αρχιτεκτονική και μειωμένο footprint, καθιστώντας το ιδανικό για συσκευές με περιορισμένη ισχύ. Παρόλα αυτά, η απλοποίηση έχει ως τίμημα την αφαίρεση ορισμένων λειτουργιών, όπως προηγμένες δυνατότητες fault tolerance και scheduling, που υπάρχουν στο πλήρες

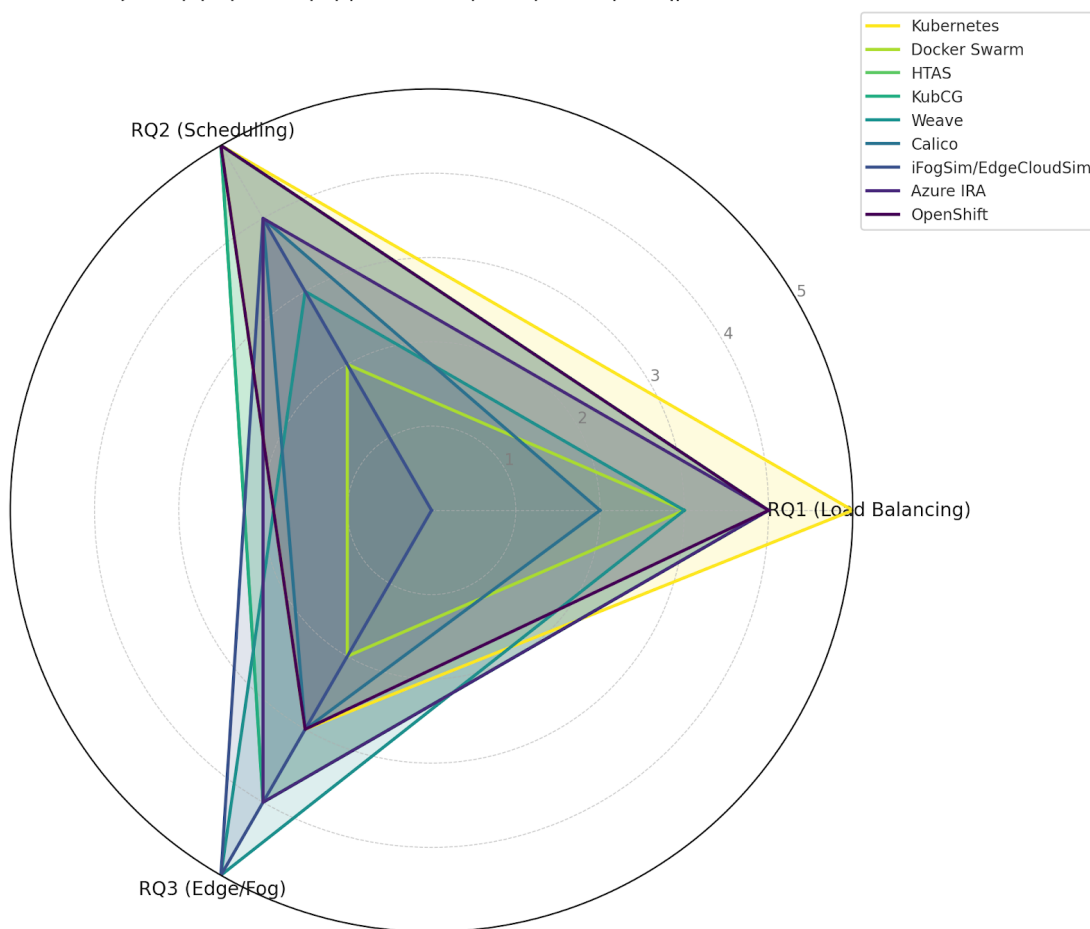
Kubernetes. Συνεπώς, η επιλογή μεταξύ Kubernetes και K3s εξαρτάται από τις συγκεκριμένες απαιτήσεις και περιορισμούς του εκάστοτε έργου.

Η πλατφόρμα Nomad της HashiCorp, αν και νεότερη άφιξη στον χώρο της ενορχήστρωσης εφαρμογών, έχει προσελκύσει σημαντική προσοχή και ενδιαφέρον. Αυτό οφείλεται στον ευέλικτο και modular σχεδιασμό της, που επιτρέπει την εύκολη ενσωμάτωση με διάφορες τεχνολογίες όπως service meshes για διαχείριση της επικοινωνίας μεταξύ υπηρεσιών και secret management tools για την ασφαλή αποθήκευση και διαχείριση κωδικών και άλλων ευαίσθητων πληροφοριών. Επιπλέον, η Nomad ξεχωρίζει για την ικανότητά της να ανταποκρίνεται αποτελεσματικά σε ετερογενή και κατανεμημένα περιβάλλοντα, όπου υπάρχουν ποικίλες υποδομές και απαιτήσεις. Δεν περιορίζεται μόνο στην ενορχήστρωση containers, αλλά υποστηρίζει επίσης virtual machines, binary applications, ακόμα και serverless workloads, προσφέροντας μεγάλη ευελιξία και προσαρμοστικότητα. Αυτή η ευρεία γκάμα υποστήριξης την καθιστά ιδιαίτερα ελκυστική για οργανισμούς με σύνθετες και διαφορετικές ανάγκες ενορχήστρωσης.

Αντιθέτως, πλατφόρμες όπως το Docker Swarm και το Apache Mesos υπολείπονται σε δυνατότητες επεκτασιμότητας και ενσωμάτωσης ML τεχνικών. Το OpenShift, ως πλατφόρμα διαχείρισης container, θεμελιώνεται στην ισχυρή βάση του Kubernetes, προσφέροντας παράλληλα μια σειρά από εμπορικά εργαλεία και λειτουργίες που στοχεύουν στη βελτίωση της διαχείρισης, της ασφάλειας και της ανάπτυξης εφαρμογών σε κλίμακα. Αυτές οι προσθήκες ενισχύουν τις δυνατότητες του Kubernetes, καθιστώντας το OpenShift μια ελκυστική επιλογή για επιχειρήσεις που απαιτούν αυξημένο έλεγχο και ευκολία στη διαχείριση των εφαρμογών τους. Παρόλα αυτά, η εμπορική φύση του OpenShift συνεπάγεται αυξημένο κόστος αδειοδότησης, γεγονός που ενδέχεται να αποτελέσει ανασταλτικό παράγοντα για μικρότερες επιχειρήσεις ή οργανισμούς με περιορισμένο προϋπολογισμό. Επιπλέον, η ενσωμάτωση των πρόσθετων εργαλείων και λειτουργιών μπορεί να οδηγήσει σε μεγαλύτερη πολυπλοκότητα, απαιτώντας εξειδικευμένες γνώσεις και αυξημένη προσπάθεια για τη διαχείριση και συντήρηση του συστήματος. Εν ολίγοις, το OpenShift παρέχει ισχυρές δυνατότητες για διαχείριση και ανάπτυξη εφαρμογών σε επίπεδο επιχείρησης, αλλά η πολυπλοκότητα και το κόστος αδειοδότησης αποτελούν σημαντικά σημεία που πρέπει να ληφθούν υπόψη κατά την αξιολόγησή του.

Συνολικά, η επιλογή πλατφόρμας ενορχήστρωσης εξαρτάται άμεσα από το σημείο του Continuum στο οποίο εφαρμόζεται: το cloud ευνοεί πλατφόρμες όπως το Kubernetes και το OpenShift, ενώ το edge απαιτεί ελαφριές, αποκεντρωμένες ή προσαρμοσμένες λύσεις, όπως το K3s ή το Nomad.

Αξιολόγηση Πλατφορμών ανά Ερευνητικό Ερώτημα (RQ)



Εικόνα 5.1.1 Αξιολόγηση πλατφορμών ανά Ερευνητικό ερώτημα (RQ)

Παρατηρήσεις από το διάγραμμα:

- **Kubernetes**, **HTAS** και **OpenShift** έχουν υψηλές επιδόσεις και στους 3 άξονες.
- **iFogSim/EdgeCloudSim** διακρίνονται κυρίως για την ανάλυση στο RQ3, χρήσιμο για edge/fog scenarios.
- **Docker Swarm** και **Calico** υστερούν σε τουλάχιστον έναν από τους τρεις άξονες, κυρίως λόγω περιορισμένων δυνατοτήτων για scheduling και heterogeneity.

5.2 Ερμηνεία των Ευρημάτων

Η ενδελεχής ανάλυση και η εμπεριστατωμένη σύγκριση των διαφόρων τεχνικών και πλατφορμών ενορχήστρωσης containerized εφαρμογών, όπως αυτές αναπτύσσονται και λειτουργούν στο πολύπλοκο και δυναμικό οικοσύστημα του Cloud-to-Thing Continuum, ανέδειξε μια σειρά από κρίσιμες διαπιστώσεις. Οι διαπιστώσεις αυτές δεν αφορούν μόνο τις τρέχουσες τεχνολογικές εξελίξεις και τα επιτεύγματα στον τομέα της ενορχήστρωσης, αλλά επίσης υπογραμμίζουν και αναδεικνύουν τα εννοιολογικά όρια και τις προκλήσεις που αντιμετωπίζουν οι σημερινές προσεγγίσεις.

Συγκεκριμένα, παρατηρείται ότι η ενορχήστρωση εφαρμογών σε αυτό το συνεχώς εξελισσόμενο περιβάλλον απαιτεί μια ισορροπία μεταξύ της ευελιξίας και της κλιμακωσιμότητας που προσφέρουν οι τεχνολογίες containerization και της ανάγκης για αποτελεσματική διαχείριση και συντονισμό των εφαρμογών σε πολλαπλά και διασυνδεδεμένα επίπεδα, από το cloud μέχρι τις edge συσκευές. Επιπλέον, οι σημερινές προσεγγίσεις συχνά παλεύουν με την πολυπλοκότητα της διαχείρισης δεδομένων, την ασφάλεια και την διαλειτουργικότητα μεταξύ διαφορετικών πλατφορμών και τεχνολογιών. Αυτές οι διαπιστώσεις υπογραμμίζουν την ανάγκη για συνεχή έρευνα και ανάπτυξη, καθώς και για την εξεύρεση νέων και καινοτόμων λύσεων που θα αντιμετωπίζουν τις προκλήσεις και θα αξιοποιούν τις δυνατότητες που προσφέρει το Cloud-to-Thing Continuum.

Πρώτον, διαπιστώνεται σαφής τεχνολογική κυριαρχία του Kubernetes, το οποίο λειτουργεί ως de facto πρότυπο σε cloud περιβάλλοντα. Η ωριμότητα του οικοσυστήματος, η υποστήριξη από κορυφαίους παρόχους cloud και η ευρεία συμβατότητα με MLOps εργαλεία (όπως Kubeflow και KServe) επιβεβαιώνεται από πληθώρα εργασιών. Παρ' όλα αυτά, η πολυπλοκότητα διαχείρισης και οι αυξημένες απαιτήσεις υποδομής καθιστούν το Kubernetes λιγότερο κατάλληλο για ακραίες edge περιπτώσεις, στοιχείο που έχει οδηγήσει στην ανάπτυξη ελαφρών υλοποιήσεων όπως το K3s.[25], [27], [58]

Η αξιοποίηση της τεχνητής νοημοσύνης για δυναμικό scheduling και autoscaling, όπως προκύπτει από την υπάρχουσα βιβλιογραφία, βρίσκεται σε ένα αρκετά πρώιμο στάδιο εξέλιξης, ιδίως όταν εξετάζονται οι εφαρμογές σε πραγματικά περιβάλλοντα edge. Αυτό το γεγονός συνεπάγεται ότι η πρακτική εφαρμογή των σχετικών τεχνολογιών είναι ακόμα περιορισμένη, κυρίως λόγω της πολυπλοκότητας και των ειδικών απαιτήσεων που επιβάλλουν τα edge computing περιβάλλοντα.

Συγκεκριμένα, έρευνες και μελέτες, όπως αυτές των Watada et al. και των Baresi et al., έχουν εστιάσει στη δυνατότητα χρήσης adaptive control μηχανισμών και συστημάτων αυτορρύθμισης πόρων που βασίζονται σε feedback loops, κυρίως σε πλατφόρμες container clusters. Αυτές οι προσεγγίσεις θεωρητικά επιτρέπουν την δυναμική διαχείριση των πόρων και την προσαρμογή στις μεταβαλλόμενες απαιτήσεις των εφαρμογών. Ωστόσο, παρά τις θεωρητικές δυνατότητες, υπάρχει σημαντική έλλειψη σαφών ενδείξεων για την πρακτική υλοποίηση αυτών των τεχνολογιών σε πραγματικά περιβάλλοντα edge. Αυτό είναι ιδιαίτερα εμφανές σε περιπτώσεις που αφορούν low-power κόμβους, όπου η διαχείριση των πόρων είναι εξαιρετικά κρίσιμη λόγω των περιορισμένων ενεργειακών αποθεμάτων, καθώς και σε εφαρμογές Containers που είναι ευαίσθητες σε θέματα latency, όπου η καθυστέρηση μπορεί να επηρεάσει σημαντικά την απόδοση.[59], [60]

Επιπλέον, η κυριαρχία κεντροποιημένων αρχιτεκτονικών ενορχήστρωσης, όπου το scheduling βασίζεται σε centralized control planes, θέτει σοβαρούς περιορισμούς ως προς τη λειτουργία σε αποκεντρωμένα ή intermittently connected περιβάλλοντα. Όπως αναφέρεται στην εργασία των Edwin Sturris και Kulikova Orchestrating Hybrid η ανάγκη για cloud orchestration frameworks που ενσωματώνουν μηχανισμούς τοπικού ελέγχου και προσαρμοστικότητας αναδεικνύεται ως κρίσιμη για το hybrid/edge μοντέλο. Η απουσία τυποποιημένων μεθόδων benchmarking για συστήματα edge/fog, τα οποία χαρακτηρίζονται από διακυμάνσεις στη συνδεσιμότητα, περιορισμένους πόρους και υψηλή μεταβλητότητα φορτίου, αποτελεί σημαντικό εμπόδιο. Αυτή η έλλειψη περιορίζει την εξαγωγή γενικών συμπερασμάτων από τις μελέτες και αμφισβητεί την εγκυρότητα των συγκρίσεων μεταξύ

διαφορετικών λύσεων. Είναι κρίσιμης σημασίας να αναπτυχθούν κοινά αποδεκτά πειραματικά σενάρια και μετρικές αξιολόγησης, καθώς αυτό έχει αναδειχθεί ως θεμελιώδης πρόκληση για την εξέλιξη του τομέα των Containers στο μέλλον. Χωρίς τέτοιες τυποποιημένες προσεγγίσεις, η σύγκριση και η αξιολόγηση των επιδόσεων και της αποτελεσματικότητας των διάφορων edge/fog συστημάτων παραμένει προβληματική, εμποδίζοντας την περαιτέρω πρόοδο και την αξιοποίηση των δυνατοτήτων τους.[16], [61]

Στο κεφάλαιο αυτό, εξετάζονται διεξοδικά οι παρατηρήσεις σχετικά με την ασυνέχεια μεταξύ της ερευνητικής θεωρίας και της πραγματικότητας της βιομηχανικής πρακτικής. Ειδικότερα, γίνεται αναφορά σε ένα σημαντικό χάσμα που αφορά τη μεταφορά πρωτοποριακών τεχνικών, όπως οι εννοχρηστωτές βασισμένοι σε Ενισχυτική Μάθηση (RL) ή οι κατανεμημένοι πράκτορες Τεχνητής Νοημοσύνης (AI), από το εργαστήριο στην παραγωγική διαδικασία. Η δυσκολία αυτή εδράζεται σε κρίσιμα ζητήματα όπως η διασφάλιση της ασφάλειας των συστημάτων, η ερμηνευσιμότητα των αποφάσεων που λαμβάνονται από τα μοντέλα, καθώς και η ομαλή ενσωμάτωση των νέων τεχνολογιών με τις ήδη υπάρχουσες υποδομές, οι οποίες συχνά είναι παλαιότερης τεχνολογίας (legacy). Η ανάλυση εστιάζει στη διαπίστωση ότι ενώ η έρευνα προωθεί συνεχώς νέες μεθόδους και αλγορίθμους, η βιομηχανία αντιμετωπίζει προκλήσεις στην εφαρμογή τους λόγω των παραπάνω περιορισμών. Αυτό οδηγεί σε μια καθυστέρηση στην υιοθέτηση των πιο καινοτόμων λύσεων και στην εκμετάλλευση των πλήρων δυνατοτήτων τους στην πράξη, επηρεάζοντας ενδεχομένως την ανταγωνιστικότητα και την αποδοτικότητα των επιχειρήσεων. Επιπλέον, η έλλειψη πρακτικών λύσεων για την αντιμετώπιση των θεμάτων ασφάλειας και ερμηνευσιμότητας μπορεί να εμποδίσει την ευρεία αποδοχή τέτοιων τεχνολογιών, καθιστώντας αναγκαία την ανάπτυξη εξειδικευμένων εργαλείων και πρακτικών που θα γεφυρώσουν το χάσμα μεταξύ θεωρίας και πράξης.

5.3 Επιπτώσεις στην Πράξη

Η συστηματική ανάλυση της βιβλιογραφίας ανέδειξε ένα ευρύ φάσμα κρίσιμων τεχνολογικών πτυχών, αρχιτεκτονικών προσεγγίσεων και επιχειρησιακών στρατηγικών που διαδραματίζουν καθοριστικό ρόλο στη μεγιστοποίηση της αποτελεσματικότητας των τεχνικών εννοχρήστρωσης εφαρμογών που βασίζονται σε containers, στο πλαίσιο του συνεχούς Cloud-to-Thing. Τα πορίσματα αυτής της ερευνητικής προσπάθειας φέρουν σημαντικές επιπτώσεις και προεκτάσεις για την πρακτική ανάπτυξη και εφαρμογή λύσεων σε πολλαπλά επίπεδα. Πιο συγκεκριμένα, επηρεάζουν τόσο την οργάνωση και λειτουργία υποδομών DevOps, όσο και την ανάπτυξη και διαχείριση περιβαλλόντων που αξιοποιούν την τεχνητή νοημοσύνη και τη μηχανική μάθηση. Η μελέτη υπογραμμίζει την πολυπλοκότητα και την πολυδιάστατη φύση της εννοχρήστρωσης containers σε αυτό το εξελισσόμενο οικοσύστημα, καθιστώντας σαφές ότι απαιτείται μια ολιστική και προσαρμοστική προσέγγιση για την επίτευξη βέλτιστων αποτελεσμάτων.

Επιπτώσεις για DevOps περιβάλλοντα

Η κυριαρχία του Kubernetes σε περιβάλλοντα cloud είναι αδιαμφισβήτητη και υπογραμμίζει ότι η πλατφόρμα αυτή αναδεικνύεται ως η πλέον προτιμητέα τεχνολογική επιλογή για τη δημιουργία και διαχείριση κλιμακούμενων, fault-tolerant και declarative container clusters. Η ικανότητα του Kubernetes να ενσωματώνεται απρόσκοπτα με CI/CD pipelines, να υποστηρίζει αρχιτεκτονικές GitOps, και η ευρεία υποστήριξή του από ενεργές κοινότητες και κορυφαίους cloud παρόχους, το καθιστούν ιδανικό για τη θεμελίωση και λειτουργία

σύγχρονων, κεντροποιημένων cloud-native υποδομών. Πιο συγκεκριμένα, το Kubernetes επιτρέπει στις επιχειρήσεις να αυτοματοποιήσουν την ανάπτυξη, την κλιμάκωση και τη διαχείριση εφαρμογών σε containers με τρόπο αποτελεσματικό και ελαστικό. Αυτό σημαίνει ότι οι εφαρμογές μπορούν να ανταποκριθούν γρήγορα στις μεταβαλλόμενες απαιτήσεις, να διασφαλίσουν υψηλή διαθεσιμότητα και να ελαχιστοποιήσουν τον χρόνο διακοπής λειτουργίας σε περίπτωση προβλημάτων. Επιπλέον, η υποστήριξη για GitOps αρχιτεκτονικές εξασφαλίζει ότι οι αλλαγές στην υποδομή και τις εφαρμογές ελέγχονται και παρακολουθούνται με τρόπο διαφανή και ελεγχόμενο, μέσω του Git, ενισχύοντας την ασφάλεια και τη σταθερότητα.

Η ευρεία αποδοχή του Kubernetes και η δημιουργία μιας ισχυρής κοινότητας γύρω από αυτό, έχουν οδηγήσει στην ανάπτυξη πληθώρας εργαλείων και επεκτάσεων, που διευρύνουν τις δυνατότητές του και απλοποιούν τη χρήση του. Τέλος, η υποστήριξη από όλους τους μεγάλους cloud παρόχους εξασφαλίζει ότι οι επιχειρήσεις μπορούν να υλοποιήσουν τις λύσεις τους με βάση το Kubernetes, ανεξάρτητα από το cloud περιβάλλον που χρησιμοποιούν. Ωστόσο, σε σενάρια υποδομών κοντά στο άκρο του δικτύου (fog/edge), όπου επικρατούν περιορισμοί σε υπολογιστικούς πόρους, συνδεσιμότητα και ενεργειακή κατανάλωση, διαπιστώνεται η ανάγκη για **λιγότερο απαιτητικές, αποκεντρωμένες λύσεις**. Πλατφόρμες όπως το K3s (lightweight Kubernetes) ή το HashiCorp Nomad εμφανίζονται ως τεχνικά επαρκείς επιλογές, προσφέροντας λειτουργική πληρότητα με μειωμένο λειτουργικό αποτύπωμα. Οι παραπάνω λύσεις επιτρέπουν την υλοποίηση αποκεντρωμένων μοντέλων ενορχήστρωσης, κατάλληλων για γεωγραφικά διασκορπισμένες, μη συνεχώς συνδεδεμένες συστοιχίες κόμβων.

MLOps και Αυτοματοποιημένες Ροές Εργασίας Βασισμένες στην Τεχνητή Νοημοσύνη

Η αποτελεσματική εφαρμογή τεχνικών ενορχήστρωσης στον τομέα της μηχανικής μάθησης απαιτεί ευέλικτη και προσαρμοστική διαχείριση του κύκλου ζωής των μοντέλων. Αυτό περιλαμβάνει την εκπαίδευση, την εξυπηρέτηση και την επανεκπαίδευση μοντέλων σε ένα περιβάλλον αυξημένης πολυπλοκότητας και με ποικίλους ετερογενείς πόρους. Η χρήση εργαλείων όπως το Kubeflow, το KServe και το KEDA, ενσωματωμένα στο οικοσύστημα του Kubernetes, δημιουργεί το απαραίτητο πλαίσιο για τη διαχείριση αυτοματοποιημένων ροών MLOps, τόσο θεσμικά όσο και τεχνικά. Ωστόσο, η προσαρμογή αυτών των ροών σε περιβάλλοντα edge/fog παραμένει μια πρόκληση. Αυτό οφείλεται στην έλλειψη ενσωματωμένης υποστήριξης για χαμηλή καθυστέρηση, χαμηλή υπολογιστική πολυπλοκότητα και τοπική αυτονομία. Ως εκ τούτου, η ανάγκη για την ενσωμάτωση εξηγήσιμων και ελαφρών μοντέλων, προσαρμοσμένων σε υποδομές με περιορισμούς, είναι κρίσιμη για την ευρεία και πρακτική εφαρμογή τους.

Στρατηγικές υλοποίησης και αρχιτεκτονικές προσεγγίσεις

Τα ευρήματα της ανασκόπησης καταδεικνύουν ότι η υιοθέτηση υβριδικών μοντέλων ενορχήστρωσης, τα οποία συνδυάζουν τοπική (edge-level) ευφυΐα και παρακολούθηση με κεντροποιημένο έλεγχο και ενοποίηση, μπορεί να λειτουργήσει ως τεχνολογική γέφυρα ανάμεσα στις απαιτήσεις του Continuum. Συγκεκριμένα:

- Η στατική προ-τοποθέτηση (pre-deployment) μονάδων σε κόμβους με fallback σε cloud orchestrator εξασφαλίζει στοιχειώδη ανθεκτικότητα.
- Η χρήση fault-tolerant distributed storage (π.χ. GlusterFS) συνεισφέρει στη διατήρηση της διαθεσιμότητας δεδομένων, ακόμα και υπό συνθήκες τοπικής αποσύνδεσης
- Η ενσωμάτωση συστημάτων παρατήρησης και αυτοματισμού (π.χ. Prometheus, Grafana, Loki, Alertmanager) επιτρέπει την οικοδόμηση κλειστών κυκλωμάτων ελέγχου (closed feedback loops) για runtime προσαρμογή.

Σε επιχειρησιακό επίπεδο, η ανάδειξη αυτών των ευρημάτων υπογραμμίζει την κρίσιμη ανάγκη για τους οργανισμούς να ενσωματώσουν προηγμένες, πολυεπίπεδες αρχιτεκτονικές ενορχήστρωσης. Αυτές οι αρχιτεκτονικές οφείλουν να σχεδιαστούν με γνώμονα την εναρμόνιση με αυστηρές προδιαγραφές ασφάλειας, διασφαλίζοντας την προστασία των δεδομένων και την ακεραιότητα των συστημάτων σε όλο το εύρος του Cloud-to-Thing φάσματος. Επιπλέον, είναι απαραίτητο να επιτυγχάνεται πλήρης διαλειτουργικότητα μεταξύ των διαφόρων στοιχείων και υπηρεσιών, προωθώντας την απρόσκοπτη ανταλλαγή πληροφοριών και τη συνεργασία μεταξύ διαφορετικών συστημάτων και πλατφορμών. Τέλος, η αυτονομία σε κάθε σημείο του φάσματος είναι ζωτικής σημασίας, επιτρέποντας στα συστήματα να λειτουργούν αποτελεσματικά και να λαμβάνουν αποφάσεις αυτόνομα, χωρίς συνεχή ανθρώπινη παρέμβαση, κάτι που ενισχύει την ευελιξία και την αποτελεσματικότητα των επιχειρησιακών διαδικασιών.

5.4 Περιορισμοί της Ανασκόπησης

Παρά τη συστηματική μεθοδολογική προσέγγιση που ακολουθήθηκε για την ανάκτηση, επιλογή και ανάλυση των σχετικών μελετών, η παρούσα ανασκόπηση υπόκειται σε συγκεκριμένους περιορισμούς που επηρεάζουν, σε ορισμένο βαθμό, την εγκυρότητα και τη γενικευσιμότητα των συμπερασμάτων της.

Περιορισμοί πηγών και πεδίου αναζήτησης

Η επισκόπηση περιορίστηκε σε επιστημονικά άρθρα από ακαδημαϊκές βάσεις δεδομένων (IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, MDPI, Wiley), αποκλείοντας μη ακαδημαϊκές πηγές (white papers, blog posts, industry benchmarks, κυβερνητικές αναφορές) που συχνά περιέχουν πρόσφατες πρακτικές στην ενορχήστρωση containerized εφαρμογών σε edge/fog περιβάλλοντα. Ενδέχεται να έχουν παραλειφθεί καινοτόμες βιομηχανικές εφαρμογές χωρίς ακαδημαϊκή δημοσίευση. Ο χρονικός περιορισμός (2015–2025) επίσης θέτει όρια, αποκλείοντας πιθανές παλαιότερες συνεισφορές από το grid computing ή τους πρώτους cloud orchestrators.

Ετερογένεια των μελετών

Διαπιστώνεται έντονη ετερογένεια στη φύση, τη μεθοδολογία και την τεχνική αρτιότητα των επιλεγμένων μελετών. Πολλά άρθρα περιορίζονται σε θεωρητικές προσεγγίσεις ή συνοπτικές αναλύσεις προτεινόμενων αρχιτεκτονικών, χωρίς τη συνοδεία λεπτομερών πειραματικών

αποτελεσμάτων ή συγκριτικής αξιολόγησης. Αυτό καθιστά δυσχερή τη διασταύρωση και τη συνεπή κατηγοριοποίηση των ευρημάτων βάσει κοινών μεταβλητών και μετρικών.

Απουσία τυποποιημένων μετρικών και πλαισίων αξιολόγησης

Ένας από τους πλέον κρίσιμους περιορισμούς αφορά την έλλειψη καθολικά αποδεκτών benchmarks και μετρικών αξιολόγησης. Οι περισσότερες μελέτες υιοθετούν διαφορετικά σενάρια, παραμέτρους και μετρικές (π.χ. latency, throughput, resource utilization), γεγονός που καθιστά δυσχερή την εξαγωγή συγκρίσιμων ή στατιστικά συνεπών συμπερασμάτων. Επίσης, απουσιάζει ένα κοινό πειραματικό υπόβαθρο για την αξιολόγηση ενορχηστρωτών σε edge/fog περιβάλλοντα, υπό πραγματικές συνθήκες (π.χ. δικτυακή αστάθεια, ενεργειακοί περιορισμοί, heterogeneous hardware).

Περιορισμένη αναπαραγωγιμότητα και διαφάνεια δεδομένων

Ένα από τα πιο κρίσιμα ζητήματα που αναδεικνύονται κατά την ανάλυση της υπάρχουσας βιβλιογραφίας είναι ο σημαντικός περιορισμός στην αναπαραγωγιμότητα των πειραμάτων. Η δυνατότητα αναπαραγωγής αποτελεσμάτων είναι θεμελιώδης για την επιστημονική πρόοδο και την εμπιστοσύνη στα ευρήματα. Ωστόσο, στην παρούσα περίπτωση, παρατηρείται ότι σε μεγάλο βαθμό αυτή η δυνατότητα υπονομεύεται. Ελλείψεις στην Παροχή Δεδομένων και Πληροφοριών

Συγκεκριμένα, διαπιστώνεται ότι σε πολλές περιπτώσεις οι συγγραφείς των δημοσιεύσεων αποτυγχάνουν να παρέχουν επαρκή στοιχεία που θα επέτρεπαν σε άλλους ερευνητές να αναπαράγουν τα πειράματα. Αυτό περιλαμβάνει:

- Μη διαθεσιμότητα datasets: Συχνά, τα σύνολα δεδομένων (datasets) που χρησιμοποιήθηκαν για τις αναλύσεις και τα πειράματα δεν είναι δημόσια διαθέσιμα. Αυτό καθιστά αδύνατο για άλλους ερευνητές να επαληθεύσουν τα αποτελέσματα με τα ίδια δεδομένα.
- Απουσία κώδικα: Ο κώδικας που χρησιμοποιήθηκε για την επεξεργασία των δεδομένων και την εφαρμογή των μεθόδων σπανίως δημοσιεύεται ή παρέχεται. Αυτό σημαίνει ότι ακόμη και αν τα δεδομένα ήταν διαθέσιμα, η ακριβής διαδικασία ανάλυσης δεν μπορεί να αναπαραχθεί.
- Ελλιπής περιγραφή της πειραματικής υποδομής: Οι λεπτομέρειες σχετικά με τις πειραματικές συνθήκες, το υλικό, τον εξοπλισμό και τις παραμέτρους συχνά παραλείπονται ή περιγράφονται με ασαφή τρόπο. Αυτό δυσκολεύει την ακριβή αναπαραγωγή του πειραματικού σχεδιασμού.

Συνέπειες και Προκλήσεις

Η έλλειψη αναπαραγωγιμότητας δημιουργεί σημαντικά προβλήματα. Πρώτον, εμποδίζει την επαλήθευση και την κριτική αξιολόγηση των ευρημάτων. Χωρίς τη δυνατότητα να επαναληφθούν τα πειράματα, η εγκυρότητα των αποτελεσμάτων δεν μπορεί να επιβεβαιωθεί ανεξάρτητα. Δεύτερον, καθυστερεί την πρόοδο της έρευνας. Η αδυναμία αξιοποίησης και επέκτασης των προηγούμενων εργασιών λόγω της έλλειψης διαφάνειας και πληροφοριών επιβραδύνει την ανάπτυξη του πεδίου. Τέλος, υπονομεύει την εμπιστοσύνη στην επιστημονική διαδικασία. Όταν τα αποτελέσματα δεν μπορούν να αναπαραχθούν,

δημιουργούνται αμφιβολίες σχετικά με την αξιοπιστία και την αντικειμενικότητα της έρευνας.

Συνοψίζοντας, η ανάγκη για ενίσχυση της αναπαραγωγιμότητας στην έρευνα είναι κρίσιμη. Οι ερευνητές θα πρέπει να ενθαρρύνονται να παρέχουν ανοιχτά datasets, κώδικα και λεπτομερείς περιγραφές των πειραματικών διαδικασιών, ώστε να διασφαλίζεται η εγκυρότητα και η αξιοπιστία των αποτελεσμάτων τους.

Επιλογή και αξιολόγηση άρθρων

Παρά την εφαρμογή αυστηρών κριτηρίων ένταξης/αποκλεισμού και τη διπλή διαδικασία ελέγχου (screening τίτλου/περίληψης και πλήρους κειμένου), η τελική επιλογή των άρθρων εμπεριέχει αναπόφευκτα στοιχεία υποκειμενικότητας. Αυτό ισχύει ιδιαίτερα σε περιπτώσεις όπου η καταλληλότητα είναι οριακή ή όταν η σύνδεση με συγκεκριμένα ερευνητικά ερωτήματα δεν είναι ξεκάθαρη. Κατά συνέπεια, ενδέχεται να υπάρχουν μεροληψίες στην επιλογή.

Η μελέτη, παρά τους αναφερόμενους περιορισμούς, προσφέρει σημαντική συνεισφορά, η οποία όμως χρειάζεται να αξιολογηθεί με προσοχή και εντός ενός καθορισμένου πλαισίου. Είναι σημαντικό να τονιστεί ότι οι περιορισμοί δεν υπονομεύουν την αξία των συμπερασμάτων, αλλά υποδεικνύουν τις κατευθύνσεις για μελλοντική έρευνα. Για να ενισχυθεί η εγκυρότητα και η εφαρμοσιμότητα των αποτελεσμάτων, είναι απαραίτητη η διεξαγωγή περαιτέρω ερευνών σε αυστηρά ελεγχόμενες πειραματικές συνθήκες. Αυτό θα εξασφαλίσει μεγαλύτερη ακρίβεια και αξιοπιστία στα ευρήματα. Επιπλέον, η ενσωμάτωση δεδομένων από την πραγματική βιομηχανική πρακτική θα προσδώσει έναν πρακτικό χαρακτήρα στην έρευνα, καθιστώντας τα συμπεράσματα πιο άμεσα εφαρμόσιμα στον επιχειρησιακό τομέα. Η σύνδεση της ακαδημαϊκής έρευνας με την εμπειρία του κλάδου θα οδηγήσει σε πιο ολοκληρωμένες και ρεαλιστικές λύσεις. Με αυτόν τον τρόπο, η μεταφραστική ισχύς της έρευνας θα αυξηθεί σημαντικά, συμβάλλοντας στην ανάπτυξη πρακτικών και αποτελεσματικών στρατηγικών για τις επιχειρήσεις.

5.5 Προτάσεις για Μελλοντική Έρευνα

Η συστηματική ανασκόπηση που διεξήχθη ανέδειξε μια σειρά από σημαντικές προόδους στον τομέα της ενορχήστρωσης εφαρμογών που βασίζονται σε containers, ιδίως στο πλαίσιο του Cloud-to-Thing Continuum. Αυτή η περιοχή παρουσιάζει μια δυναμική εξέλιξη, με την ενσωμάτωση τεχνολογιών cloud και IoT (Internet of Things) να δημιουργεί νέες δυνατότητες για διαχείριση και λειτουργία εφαρμογών σε ένα ευρύ φάσμα συσκευών και υπολογιστικών περιβαλλόντων. Ωστόσο, παρά τις προόδους, η ανασκόπηση αποκάλυψε επίσης μια πληθώρα από ανοιχτά ζητήματα και σημαντικές τεχνολογικές προκλήσεις. Αυτά τα ζητήματα περιλαμβάνουν την ασφάλεια, την επεκτασιμότητα, τη διαλειτουργικότητα και τη διαχείριση πόρων, καθώς και την ανάγκη για πιο αποτελεσματικές μεθόδους αυτοματοποίησης και διαχείρισης του κύκλου ζωής των εφαρμογών. Αυτές οι προκλήσεις δεν αποτελούν απλώς εμπόδια, αλλά μπορούν να θεωρηθούν και ως ευκαιρίες για μελλοντική έρευνα και ανάπτυξη. Με την εστίαση στην αντιμετώπιση αυτών των ζητημάτων, η επιστημονική κοινότητα και η βιομηχανία μπορούν να συμβάλουν στην περαιτέρω εξέλιξη του τομέα, οδηγώντας σε πιο αξιόπιστες, αποτελεσματικές και ασφαλείς λύσεις για την ενορχήστρωση containerized εφαρμογών στο Cloud-to-Thing Continuum.

Ανάπτυξη αποκεντρωμένων και edge-native αρχιτεκτονικών ενορχήστρωσης

Παρά την εξέλιξη των υφιστάμενων πλατφορμών (π.χ. Kubernetes, OpenShift), παρατηρείται απουσία ενοποιημένων μοντέλων ενορχήστρωσης που να σχεδιάζονται εξ αρχής για κατανεμημένα, ετερογενή, resource-constrained περιβάλλοντα, όπως το fog και το edge. Προτείνεται η διερεύνηση αποκεντρωμένων αρχιτεκτονικών, βασισμένων σε peer-to-peer μηχανισμούς, agent-based σχεδίαση ή even-driven συνεργατικές δομές, ικανές να λειτουργούν αυτόνομα και χωρίς συνεχή εξάρτηση από κεντρικά control planes.

Ενσωμάτωση explainable AI (XAI) τεχνικών στον προγραμματισμό πόρων

Αν και οι μηχανισμοί τεχνητής νοημοσύνης (π.χ. reinforcement learning, neural scheduling policies) παρουσιάζουν ενθαρρυντικά αποτελέσματα σε ελεγχόμενα πειραματικά περιβάλλοντα, η υιοθέτησή τους σε πραγματικά συστήματα παραμένει περιορισμένη. Η χρήση επεξηγήσιμων (explainable) και ελαφρών ML μοντέλων μπορεί να ενισχύσει την αποδοχή τους από διαχειριστές υποδομών, καθώς και να βελτιώσει τη διαφάνεια στη λήψη αποφάσεων σε συστήματα υψηλής κρισιμότητας, όπως αυτά του edge computing.

Καθιέρωση καθολικών μετρικών αξιολόγησης και benchmarking frameworks

Η συγκριτική μελέτη των πλατφορμών ενορχήστρωσης καταδεικνύει ένα κρίσιμο κενό στην τυποποίηση της αξιολόγησής τους, ιδίως όταν πρόκειται για περιβάλλοντα edge/fog. Συγκεκριμένα, διαπιστώνεται η απουσία ολοκληρωμένων και τυποποιημένων πλαισίων αξιολόγησης που να είναι ικανά να αποτυπώσουν με ακρίβεια τις ιδιαιτερότητες αυτών των περιβαλλόντων. Χαρακτηριστικά όπως η μεταβλητή καθυστέρηση, οι περιορισμοί στην κατανάλωση ενέργειας ή διακοπτόμενη ή αποσπασματική συνδεσιμότητα, καθώς και η ανάγκη για τοπική αυτονομία και ανεξαρτησία δεν λαμβάνονται επαρκώς υπόψη στα υπάρχοντα εργαλεία αξιολόγησης.

Κατά συνέπεια, αναδεικνύεται επιτακτική η ανάγκη για τη σχεδίαση και την υιοθέτηση εξειδικευμένων benchmarks, τα οποία θα εστιάζουν σε σενάρια που προσομοιώνουν την πραγματική χρήση σε συνθήκες cloud-to-edge. Αυτά τα benchmarks θα πρέπει να ενσωματώνουν ένα ευρύ φάσμα μετρήσεων, όπως η καθυστέρηση ή διακύμανση της καθυστέρησης, η διαθεσιμότητα, ο χρόνος ανάκαμψης, η κατανάλωση ενέργειας, καθώς και το κόστος ανάπτυξης και διαχείρισης μοντέλων. Η ανάπτυξη τέτοιων εξειδικευμένων εργαλείων αξιολόγησης είναι απαραίτητη για την ορθή επιλογή και βελτιστοποίηση των πλατφορμών ενορχήστρωσης για edge/fog deployments, διασφαλίζοντας την αποτελεσματική λειτουργία και την αποδοτικότερη εκμετάλλευση των πόρων σε αυτά τα σύνθετα και απαιτητικά περιβάλλοντα.

Συνδυασμός ενορχήστρωσης εφαρμογών και δεδομένων

Η πλειονότητα των ερευνητικών εργασιών που έχουν διεξαχθεί μέχρι σήμερα στον τομέα της ενορχήστρωσης, φαίνεται να εστιάζει κατά κύριο λόγο στην ενορχήστρωση των εφαρμογών (application-level containers). Αυτή η εστίαση, αν και κρίσιμη για την εύρυθμη λειτουργία των εφαρμογών, συχνά παραμελεί σε σημαντικό βαθμό την πολυπλοκότητα και τις προκλήσεις που αφορούν την αλληλεπίδραση με τους μηχανισμούς διανομής, συγχρονισμού και αποθήκευσης δεδομένων. Αυτό είναι ιδιαίτερα αισθητό όταν τα δεδομένα διακινούνται μεταξύ του cloud και των αισθητήρων (edge devices) και αντίστροφα, μια διαδικασία που εμπεριέχει σημαντικές προκλήσεις σε θέματα latency, ασφάλειας και συνέπειας δεδομένων. Η

ερευνητική κοινότητα καλείται να διευρύνει τους ορίζοντες της και να στραφεί σε πιο ολοκληρωμένες προσεγγίσεις. Μια τέτοια προσέγγιση θα μπορούσε να είναι η ανάπτυξη και η μελέτη co-orchestration μοντέλων. Σε αυτά τα μοντέλα, η ενορχήστρωση των εφαρμογών και των δεδομένων θα πραγματοποιείται συγχρονισμένα. Αυτό σημαίνει ότι οι αποφάσεις για την εκτέλεση και την κλιμάκωση των εφαρμογών θα λαμβάνονται παράλληλα με τις αποφάσεις για τη διαχείριση και τη μεταφορά των δεδομένων. Επιπλέον, τα μοντέλα αυτά θα πρέπει να είναι προσαρμόσιμα, δηλαδή να μπορούν να αλλάζουν τη συμπεριφορά τους ανάλογα με τις μεταβαλλόμενες συνθήκες του περιβάλλοντος (π.χ. διαθεσιμότητα bandwidth, φόρτος εργασίας, ενεργειακή κατανάλωση). Τέλος, η ανάπτυξη context-aware μηχανισμών είναι απαραίτητη, ώστε το σύστημα ενορχήστρωσης να μπορεί να αντιλαμβάνεται και να αξιοποιεί πληροφορίες από το περιβάλλον (context) για να λαμβάνει τις καλύτερες δυνατές αποφάσεις.

Με αυτό τον τρόπο, η έρευνα μπορεί να συμβάλει στην ανάπτυξη πιο αποδοτικών, αξιόπιστων και ευέλικτων συστημάτων ενορχήστρωσης που θα ανταποκρίνονται στις αυξανόμενες ανάγκες των σύγχρονων εφαρμογών και υποδομών cloud-sensor. Η εστίαση στην ολοκληρωμένη διαχείριση εφαρμογών και δεδομένων, με παράλληλη προσαρμογή στις συνθήκες του περιβάλλοντος, αποτελεί κλειδί για την επίτευξη αυτού του στόχου.

Δοκιμές σε πραγματικά σενάρια χρήσης (real-world validation)

Η αξιολόγηση πολλών προτεινόμενων λύσεων περιορίζεται σε θεωρητικό επίπεδο ή σε δοκιμές που διεξάγονται σε ελεγχόμενα, τεχνητά περιβάλλοντα. Αυτό δημιουργεί ένα κενό μεταξύ της θεωρίας και της πραγματικής εφαρμογής. Για να γεφυρωθεί αυτό το χάσμα και να διασφαλιστεί η πλήρης κατανόηση των δυνατοτήτων και των περιορισμών αυτών των προτάσεων, είναι απαραίτητη η υλοποίηση πιλοτικών δοκιμών σε πραγματικά, δυναμικά περιβάλλοντα. Τέτοια περιβάλλοντα περιλαμβάνουν, ενδεικτικά, τις έξυπνες πόλεις, όπου η συνδεσιμότητα και η διαχείριση δεδομένων είναι κρίσιμες, το βιομηχανικό IoT, όπου η αξιοπιστία και η ασφάλεια είναι πρωταρχικής σημασίας, την ιατρική τηλεπαρακολούθηση, όπου η ακρίβεια και η αδιάλειπτη λειτουργία είναι ζωτικής σημασίας, και τα δίκτυα οχημάτων (V2X), όπου η ταχύτητα και η αλληλεπίδραση σε πραγματικό χρόνο είναι απαραίτητες. Η διεξαγωγή πιλοτικών δοκιμών σε αυτά τα ποικίλα περιβάλλοντα θα επιτρέψει μια ολοκληρωμένη και εμπειριστατωμένη αξιολόγηση όχι μόνο της τεχνικής αρτιότητας των προσεγγίσεων, αλλά και της επιχειρησιακής τους βιωσιμότητας. Δηλαδή, θα αποδειχθεί αν οι προτάσεις μπορούν να λειτουργήσουν αποτελεσματικά και αποδοτικά υπό πραγματικές συνθήκες, να ανταποκριθούν στις απαιτήσεις και τις προκλήσεις του κάθε περιβάλλοντος και να προσφέρουν ουσιαστική αξία στους χρήστες και τους οργανισμούς.

Πολιτικές ασφαλείας με επίγνωση τοποθεσίας και πολιτική δεδομένων (data-aware, geo-aware orchestration)

Η ενσωμάτωση μηχανισμών ασφαλείας σε ετερογενή περιβάλλοντα παραμένει ανεπαρκώς μελετημένη. Η μελλοντική έρευνα θα μπορούσε να εστιάσει σε πολιτικές δυναμικής ενορχήστρωσης που λαμβάνουν υπόψη απαιτήσεις εντοπισμού, προστασίας δεδομένων (data sovereignty), και κρυπτογραφικών πολιτικών προσαρμοσμένων ανάλογα με το γεωγραφικό και νομικό πλαίσιο. Καθίσταται σαφές ότι η ενορχήστρωση containerized εφαρμογών στο Cloud-to-Thing Continuum δεν αποτελεί απλώς τεχνικό πρόβλημα κατανομής πόρων, αλλά πολυπαραγοντικό ερευνητικό πεδίο, στο οποίο η υπολογιστική επιστήμη, η τεχνητή

νοημοσύνη, η μηχανική ασφάλειας και η θεωρία συστημάτων αλληλεπιδρούν. Η αντιμετώπιση των παραπάνω προκλήσεων δύναται να προωθήσει ουσιαστικά την επόμενη γενιά καταναεμημένων, έξυπνων και προσαρμοστικών υπολογιστικών συστημάτων.

6. Συμπεράσματα

6.1 Επισκόπηση της Μελέτης

Η παρούσα εργασία είχε ως βασικό αντικείμενο τη διερεύνηση του πεδίου της ενορχήστρωσης containerized εφαρμογών στο πλαίσιο του Cloud-to-Thing Continuum, μέσω της συστηματικής χαρτογράφησης της υφιστάμενης επιστημονικής βιβλιογραφίας. Το Cloud-to-Thing Continuum συνιστά ένα ολοένα και πιο διαδεδομένο υπολογιστικό υπόδειγμα, το οποίο περιλαμβάνει καταναεμημένες υποδομές από το κεντρικό cloud έως το άκρο του δικτύου (edge) και ενδιάμεσες υλοποιήσεις fog computing. Η ανάγκη για ευέλικτη, αποδοτική και ασφαλή διαχείριση containerized εφαρμογών εντός αυτού του ετερογενούς και πολυεπίπεδου πλαισίου καθιστά την ενορχήστρωση έναν πυρήνα τεχνικής πρόκλησης και καινοτομίας.

Η μελέτη στηρίχθηκε στη μεθοδολογική προσέγγιση της Συστηματικής Ανασκόπησης Βιβλιογραφίας (Systematic Literature Review – SLR), επιδιώκοντας την αντικειμενική και τεκμηριωμένη καταγραφή, ανάλυση και σύνθεση των υπαρχόντων ερευνητικών αποτελεσμάτων. Το βασικό ερευνητικό ερώτημα συνοψίστηκε ως εξής:

Πώς διαμορφώνεται το τοπίο των τεχνολογιών, εργαλείων, στρατηγικών και ανοιχτών προκλήσεων που σχετίζονται με την ενορχήστρωση containerized εφαρμογών στο Cloud-to-Thing Continuum;

Για την απάντηση αυτού του κεντρικού ερωτήματος, η εργασία οργανώθηκε γύρω από τρεις επιμέρους ερευνητικούς άξονες:

- **RQ1:** Ποιες τεχνικές, πλατφόρμες και αρχιτεκτονικές έχουν χρησιμοποιηθεί για την ενορχήστρωση containerized εφαρμογών στο Continuum;
- **RQ2:** Ποια είναι τα βασικά κριτήρια αξιολόγησης απόδοσης και ποιες μετρικές εφαρμόζονται για benchmarking;
- **RQ3:** Ποιες είναι οι βασικές ερευνητικές προκλήσεις, τεχνολογικοί περιορισμοί και ευκαιρίες καινοτομίας (ιδίως μέσω AI/ML) στον εν λόγω τομέα;

Η θεωρητική θεμελίωση της μελέτης αναπτύχθηκε στο Κεφάλαιο 1, όπου παρουσιάστηκαν οι βασικές έννοιες του cloud–fog–edge continuum, της containerization και των τεχνολογιών ενορχήστρωσης (Kubernetes, Docker Swarm, OpenShift, Mesos, Nomad, Rancher κ.ά.). Δόθηκε ιδιαίτερη έμφαση:

- στα πλεονεκτήματα και μειονεκτήματα των containerized workloads έναντι παραδοσιακών υποδομών,
- στη σημασία της ενορχήστρωσης για την επίτευξη αυτοματοποίησης, scaling και fault tolerance,
- και στη χρήση των containers στο πλαίσιο των CI/CD pipelines (DevOps) αλλά και της μηχανικής μάθησης (MLOps).

Στο Κεφάλαιο 3 περιγράφηκε αναλυτικά η μεθοδολογία SLR που ακολουθήθηκε. Ορίστηκαν με σαφήνεια τα κριτήρια ένταξης και αποκλεισμού (inclusion/exclusion), οι βάσεις δεδομένων αναζήτησης (IEEE, ACM, Springer, Elsevier, MDPI, Wiley, arXiv), καθώς και οι λογοτεχνικές στρατηγικές αναζήτησης (συνδυασμοί λέξεων-κλειδιών, χρονικά όρια 2015–2025, peer-reviewed εργασία). Ακολούθησε διπλή διαδικασία φιλτραρίσματος (screening τίτλων/περιλήψεων και πλήρους κειμένου), καταλήγοντας σε ένα σώμα 42 επιλεγμένων μελετών.

Στο Κεφάλαιο 4 πραγματοποιήθηκε θεματική ανάλυση των επιλεγμένων άρθρων με κατηγοριοποίηση των προσεγγίσεων που εντοπίστηκαν, ομαδοποιημένες ανά ερευνητικό ερώτημα. Η παρουσίαση περιλάμβανε:

- συγκριτικά στοιχεία για τις κυριότερες πλατφόρμες ενορχήστρωσης,
- καταγραφή των benchmarking μετρικών και παραμέτρων αξιολόγησης απόδοσης,
- και ταξινόμηση των βασικών περιορισμών και ερευνητικών κενού στην περιοχή, με έμφαση στις edge-native λύσεις και την ενσωμάτωση τεχνητής νοημοσύνης.

Τέλος, στα Κεφάλαια 5 και 6, συντέθηκε η συγκριτική και κριτική αποτίμηση των αποτελεσμάτων, καθώς και η εξαγωγή γενικών συμπερασμάτων, προτεινόμενων στρατηγικών και κατευθύνσεων για μελλοντική έρευνα.

6.2 Κύρια Συμπεράσματα

Η ανάλυση των 42 επιλεγμένων επιστημονικών μελετών που πληρούσαν τα προκαθορισμένα κριτήρια ένταξης της παρούσας συστηματικής ανασκόπησης ανέδειξε μια σειρά από καίρια ευρήματα σχετικά με την τρέχουσα κατάσταση, τις τεχνολογικές επιλογές και τις ερευνητικές τάσεις στο πεδίο της ενορχήστρωσης containerized εφαρμογών στο Cloud-to-Thing Continuum. Τα αποτελέσματα αποδίδονται στη συνέχεια ανά ερευνητικό ερώτημα:

RQ1: Τεχνικές, Πλατφόρμες και Αρχιτεκτονικές Ενορχήστρωσης

Η συντριπτική πλειονότητα των μελετών ανέδειξε το Kubernetes ως την επικρατούσα πλατφόρμα ενορχήστρωσης σε περιβάλλοντα cloud, κυρίως λόγω της ωριμότητας του οικοσυστήματός του, της ευρείας αποδοχής από τη βιομηχανία και της υποστήριξης εργαλείων παρακολούθησης και αυτοματοποίησης (e.g., Helm, Prometheus, ArgoCD). Παρά ταύτα, η εφαρμογή του Kubernetes σε fog/edge συστήματα παρουσιάζει σημαντικούς

περιορισμούς, σχετιζόμενους με την πολυπλοκότητα της εγκατάστασης, τις αυξημένες απαιτήσεις σε πόρους και την ανάγκη συνεχούς συνδεσιμότητας.

Στην αντίθετα όχθη, ελαφρύτερες και αποκεντρωμένες πλατφόρμες, όπως το K3s, το Nomad και σε μικρότερο βαθμό το Docker Swarm, καταγράφηκαν ως καταλληλότερες για υλοποιήσεις σε περιορισμένων πόρων περιβάλλοντα (e.g. IoT gateways, edge clusters), με έμφαση στην ευκολία διαχείρισης και την ταχεία ανάπτυξη σε απομονωμένα σημεία του δικτύου. Εντούτοις, η έλλειψη προηγμένων λειτουργιών scheduling, αυτοθεραπείας και native AI ενσωμάτωσης αποτελεί περιοριστικό παράγοντα για τη χρήση τους σε πολύπλοκες ροές εργασίας.

RQ2: Κριτήρια Αξιολόγησης και Benchmarking Πλατφορμών

Η αξιολόγηση των τεχνικών και εργαλείων ενορχήστρωσης βασίστηκε κυρίως σε μετρικές απόδοσης (performance) και ανθεκτικότητας (resilience). Οι πιο συχνά χρησιμοποιούμενες μετρικές περιλαμβάνουν:

- latency και response time, ειδικά σε real-time εφαρμογές στο edge,
- resource utilization (CPU, memory, network),
- scalability (horizontal/vertical), σε stress-test συνθήκες,
- availability και fault recovery time,
- deployment και convergence time.

Παρότι αρκετές μελέτες προτείνουν δικές τους μεθοδολογίες αξιολόγησης, δεν διαπιστώθηκε η ύπαρξη καθολικά αποδεκτών benchmarking πλαισίων ή πλήρως εναρμονισμένων σεναρίων αξιολόγησης. Η έλλειψη κοινής βάσης πειραματικής σύγκρισης περιορίζει τη δυνατότητα ουσιαστικής αξιολόγησης και γενίκευσης των αποτελεσμάτων.

RQ3: Ερευνητικές Προκλήσεις, Τάσεις και Τεχνολογικά Κενά

Η βιβλιογραφία καταγράφει ένα ευρύ φάσμα ανοικτών ζητημάτων και ερευνητικών προκλήσεων, με κυριότερες τις εξής:

- Απουσία αποκεντρωμένων (decentralized) αρχιτεκτονικών ενορχήστρωσης που να λειτουργούν αξιόπιστα υπό συνθήκες περιοδικής συνδεσιμότητας, χαμηλής διαθεσιμότητας πόρων και αυξημένου jitter.
- Περιορισμένη εφαρμογή τεχνικών τεχνητής νοημοσύνης (AI/ML) σε real-time autoscaling/scheduling μηχανισμούς, και σχεδόν μηδενική παρουσία explainable AI σε operational contexts.
- Ανεπαρκής αντιμετώπιση ζητημάτων ασφάλειας, ιδίως σε περιβάλλοντα χωρίς σταθερό κεντρικό έλεγχο, όπου οι πολιτικές ελέγχου πρόσβασης, πιστοποίησης και integrity διαχείρισης δεδομένων πρέπει να ενσωματώνονται στην ίδια την ενορχήστρωση.
- Ελλιπής υποστήριξη co-orchestration μοντέλων που λαμβάνουν υπόψη ταυτόχρονα την εφαρμογή και τα δεδομένα, γεγονός ιδιαίτερα σημαντικό για edge-to-cloud ροές

δεδομένων με ιδιαιτερότητες απορρήτου ή εντοπισμού (data residency).

Τέλος, διαπιστώνεται ένα σημαντικό χάσμα μεταξύ ακαδημαϊκής έρευνας και πρακτικής βιομηχανικής υλοποίησης. Πολλές προτεινόμενες λύσεις παραμένουν θεωρητικές, χωρίς σαφές πλάνο μεταφοράς σε production-grade περιβάλλοντα.

6.3 Συμβολή της Εργασίας

Η παρούσα διπλωματική εργασία φιλοδοξεί να αποτελέσει σημαντική συνεισφορά στην κατανόηση, χαρτογράφηση και ανάλυση των τεχνολογικών, ερευνητικών και αρχιτεκτονικών εξελίξεων που αφορούν την ενορχήστρωση containerized εφαρμογών εντός του Cloud-to-Thing Continuum. Η συμβολή της έγκειται σε πολλαπλά επίπεδα, τόσο θεωρητικά όσο και εφαρμοσμένα:

1. Συστηματική και τεκμηριωμένη επισκόπηση της υφιστάμενης βιβλιογραφία

Η εργασία αξιοποιεί τη μεθοδολογία της Systematic Literature Review (SLR), επιτρέποντας τη δομημένη και επαναλήψιμη καταγραφή των πιο σημαντικών ερευνητικών μελετών της τελευταίας δεκαετίας (2015–2025). Αντί της αποσπασματικής παρουσίας ερευνητικών τάσεων, η μελέτη προσφέρει μια ολιστική αποτύπωση του πεδίου, με κατηγοριοποίηση των προσεγγίσεων ανά τύπο πλατφόρμας, τεχνική στρατηγική, εφαρμοστικό περιβάλλον και βαθμό ωριμότητας υλοποίησης.

2. Ανάδειξη των περιορισμών των υφιστάμενων τεχνολογιών σε ετερογενή περιβάλλοντα

Η εργασία καταδεικνύει, μέσα από συγκριτική ανάλυση, ότι ενώ το Kubernetes επικρατεί ως πρότυπο σε cloud περιβάλλοντα, δεν υπάρχει ακόμη κοινώς αποδεκτή, ώριμη λύση για ενορχήστρωση σε edge/fog υποδομές με περιορισμούς σε latency, επεξεργαστική ισχύ και ενεργειακή κατανάλωση. Η επισήμανση της ανάγκης για edge-native ή decentralized orchestrators συνιστά σημαντικό εννοιολογικό βήμα προς την κατεύθυνση της αποσύνδεσης από centralized μοντέλα.

3. Χαρτογράφηση της τεχνολογικής διάδρασης μεταξύ DevOps και MLOps

Η εργασία συνδέει τον χώρο της ενορχήστρωσης με πρακτικές DevOps και MLOps, τεκμηριώνοντας την ανάγκη για ενοποίηση των pipelines ανάπτυξης, εκπαίδευσης και εξυπηρέτησης μοντέλων μηχανικής μάθησης. Ιδιαίτερη έμφαση δίνεται σε εργαλεία όπως το Kubeflow, το KServe και το KEDA, τα οποία επιτρέπουν τη διαχείριση AI εφαρμογών εντός orchestrated clusters, αναδεικνύοντας τον ρόλο της ενορχήστρωσης ως τεχνολογικού συνδετικού κρίκου μεταξύ εφαρμογών, δεδομένων και αλγορίθμων.

4. Επισήμανση ερευνητικών κενών και στρατηγικών κατευθύνσεων

Η εργασία εντοπίζει, με σαφήνεια και τεκμηρίωση, τις βασικές ελλείψεις στη διεθνή βιβλιογραφία: την απουσία standard benchmarking frameworks, την έλλειψη explainable AI στον προγραμματισμό πόρων, την περιορισμένη πρακτική εφαρμογή των περισσότερων προτάσεων σε πραγματικά περιβάλλοντα. Ως εκ τούτου, η μελέτη παρέχει συγκροτημένες

προτάσεις για μελλοντική έρευνα και διαμορφώνει ένα πλαίσιο στρατηγικής για το πώς μπορεί να εξελιχθεί το πεδίο προς παραγωγικά, αποκεντρωμένα και διαφανή υπολογιστικά μοντέλα.

5. Εκπαιδευτική και μεθοδολογική αξία

Τέλος, η εργασία μπορεί να λειτουργήσει ως πρότυπο εφαρμογής της μεθοδολογίας SLR στο πεδίο της υπολογιστικής υποδομής και του cloud/edge orchestration. Η σαφής διάρθρωση των ερωτημάτων, η τεκμηρίωση της διαδικασίας φιλτραρίσματος, η δημιουργία πινάκων σύγκρισης, και η θεματική σύνθεση ανά RQ καθιστούν την εργασία χρήσιμο υλικό για μελλοντικούς φοιτητές, ερευνητές και επαγγελματίες του κλάδου.

Συνολικά, η εργασία συνεισφέρει στη συστηματική κατανόηση ενός ταχύτατα εξελισσόμενου τομέα, ο οποίος επηρεάζει άμεσα τις αρχιτεκτονικές επιλογές και τη στρατηγική ανάπτυξης λογισμικού σε ένα παγκοσμίως καταναεμμένο και δυναμικό υπολογιστικό περιβάλλον.

6.4 Περιορισμοί της Μελέτης

Αν και η παρούσα εργασία υιοθέτησε αυστηρά επιστημονικά πρότυπα στο σχεδιασμό και την εφαρμογή της Συστηματικής Ανασκόπησης Βιβλιογραφίας (SLR), οφείλει να αναγνωρίσει και να καταγράψει τους περιορισμούς που διέπουν τη μεθοδολογία και ενδέχεται να επηρεάζουν την πληρότητα, τη γενικευσιμότητα και την εγκυρότητα των συμπερασμάτων.

Περιορισμός πηγών και χρονικού πλαισίου

Η ανασκόπηση περιορίστηκε σε άρθρα δημοσιευμένα σε έγκριτες επιστημονικές βάσεις δεδομένων (IEEE Xplore, ACM DL, SpringerLink, ScienceDirect, Wiley, MDPI, arXiv), αποκλείοντας πηγές grey literature, όπως white papers, τεχνικά blogs, industry reports ή εμπορικές παρουσιάσεις. Αν και αυτή η επιλογή διασφαλίζει την ακαδημαϊκή ποιότητα των πηγών, ενδέχεται να αποκλείει πρωτογενείς πρακτικές εφαρμογές ή τεχνολογίες αιχμής που δεν έχουν ακόμη τεκμηριωθεί σε επιστημονικά πλαίσια.

Επιπλέον, το χρονικό εύρος 2015–2025, μολονότι εύλογο για την ωριμότητα του πεδίου, πιθανώς παραλείπει παλαιότερες βάσεις που αφορούν πρώιμες μορφές υπολογιστικής ενορχήστρωσης (π.χ. grid orchestration, πρώιμα container stacks).

Ετερογένεια και ποικιλία των επιλεγμένων μελετών

Οι μελέτες που περιελήφθησαν στην ανασκόπηση χαρακτηρίζονται από μεγάλη ετερογένεια όσον αφορά τη θεματολογία, τη μεθοδολογία, τις τεχνικές προσεγγίσεις και το επίπεδο ωριμότητας. Πολλές εργασίες εστιάζουν σε αρχιτεκτονικές προτάσεις ή proof-of-concept υλοποιήσεις, χωρίς να συνοδεύονται από πλήρη πειραματική αξιολόγηση, γεγονός που καθιστά δυσχερή τη συγκριτική αποτίμηση.

Απουσία καθολικά αποδεκτών benchmarks και μεθόδων αξιολόγησης

Όπως προκύπτει και από την απάντηση στο RQ2, η έλλειψη εδραιωμένων, καθολικά αποδεκτών μετρικών απόδοσης και benchmarking πλαισίων περιορίζει τη δυνατότητα άμεσης

και έγκυρης σύγκρισης των αποτελεσμάτων μεταξύ διαφορετικών ερευνητικών έργων. Ως αποτέλεσμα, τα συμπεράσματα που προκύπτουν είναι σε μεγάλο βαθμό συγκριτικά εντός των μελετών, και όχι οριζόντια ως προς το σύνολο του πεδίου.

Περιορισμένη αναπαραγωγιμότητα και διαφάνεια

Ο βαθμός αναπαραγωγιμότητας των πειραμάτων που περιγράφονται στα άρθρα ποικίλει σημαντικά. Συχνά απουσιάζουν κρίσιμες λεπτομέρειες υλοποίησης (e.g., παραμετροποίηση containers, υποδομές δοκιμών, scripts, datasets), με αποτέλεσμα η ανεξάρτητη επαλήθευση των ευρημάτων να καθίσταται ανέφικτη ή ασαφής. Το φαινόμενο αυτό πλήττει τη διαφάνεια και την επιστημονική εγκυρότητα μεγάλου μέρους της τεχνικής βιβλιογραφίας.

Ενδεχόμενες προκαταλήψεις στην επιλογή και αξιολόγηση

Παρότι η διαδικασία επιλογής ακολούθησε προκαθορισμένα inclusion/exclusion criteria και βασίστηκε σε διπλό screening, παραμένει πιθανό να έχουν προκύψει υποκειμενικά στοιχεία αξιολόγησης, ιδίως σε περιπτώσεις άρθρων που κινούνται στα όρια καταλληλότητας. Επιπλέον, η θεματική εστίαση σε orchestration και containerization ενδέχεται να απέκλεισε συναφείς προσεγγίσεις που δίνουν έμφαση σε data pipelines, service mesh, ή decentralized infrastructure χωρίς ρητή αναφορά σε containers.

Συνολικά, οι παραπάνω περιορισμοί δεν αναιρούν τη χρησιμότητα και την επιστημονική αξία της μελέτης, αλλά καθορίζουν το πλαίσιο εντός του οποίου θα πρέπει να ερμηνεύονται τα αποτελέσματα. Η συμπερίληψη εναλλακτικών μορφών δεδομένων, η υιοθέτηση καθολικών προτύπων αξιολόγησης και η διερεύνηση της σύνδεσης με βιομηχανικές πρακτικές αποτελούν σημαντικές ευκαιρίες ενίσχυσης της εγκυρότητας σε μελλοντικές ερευνητικές προσπάθειες.

6.5 Μελλοντική Ενέργεια

Η ανάλυση της υφιστάμενης βιβλιογραφίας, όπως διαμορφώθηκε από την παρούσα συστηματική ανασκόπηση, κατέδειξε ένα πεδίο που, παρά τη ραγδαία τεχνολογική του πρόοδο, εξακολουθεί να χαρακτηρίζεται από κρίσιμα τεχνολογικά κενά, μεθοδολογικές ανεπάρκειες και ερευνητικές προκλήσεις, ειδικά όταν εξετάζεται υπό το πρίσμα της υλοποίησης σε edge και fog υποδομές. Στο πλαίσιο αυτό, προτείνονται συγκεκριμένες κατευθύνσεις για μελλοντική ερευνητική δραστηριότητα, οι οποίες προκύπτουν άμεσα από τα συμπεράσματα της εργασίας:

1. Ανάπτυξη αποκεντρωμένων (decentralized) και edge-native πλατφορμών ενορχήστρωσης

Η πλειονότητα των υφιστάμενων εργαλείων βασίζεται σε κεντρικοποιημένα μοντέλα ελέγχου, τα οποία παρουσιάζουν σημαντικούς περιορισμούς σε σενάρια υψηλής γεωγραφικής διασποράς, χαμηλής συνδεσιμότητας ή περιορισμένων υπολογιστικών πόρων. Η ανάπτυξη αποκεντρωμένων αρχιτεκτονικών ενορχήστρωσης, βασισμένων σε peer-to-peer μοντέλα, agent-based scheduling ή lightweight control planes, αποτελεί μια θεμελιώδη πρόκληση. Ιδιαίτερο ενδιαφέρον παρουσιάζει η ενσωμάτωση τεχνικών consensus ή distributed ledger για την υποστήριξη trust-aware orchestration στα άκρα του δικτύου.

2. Ενσωμάτωση επεξηγήσιμων και αποδοτικών τεχνικών τεχνητής νοημοσύνης

Παρόλο που τεχνικές όπως το deep reinforcement learning έχουν αρχίσει να ενσωματώνονται σε ερευνητικές πλατφόρμες ενορχήστρωσης, η έλλειψη επεξηγησιμότητας (explainability) και η υψηλή υπολογιστική απαίτηση περιορίζουν την πρακτική τους εφαρμογή. Προτείνεται η διερεύνηση ελαφρών, επεξηγήσιμων και ενεργειακά αποδοτικών αλγορίθμων που να ενσωματώνονται εγγενώς στον λήπτη αποφάσεων του orchestrator και να λαμβάνουν υπόψη συνθήκες πραγματικού χρόνου, κατανάλωση πόρων, και ποιότητα εμπειρίας (QoE).

3. Καθιέρωση ενιαίων, ανοιχτών benchmarking πλαισίων

Η ερευνητική κοινότητα στερείται κοινά αποδεκτών benchmarking προτύπων για την αξιολόγηση ενορχηστρωτών σε edge/fog περιβάλλοντα. Απαιτείται η δημιουργία τυποποιημένων σεναρίων αξιολόγησης που να λαμβάνουν υπόψη παραμέτρους όπως: μεταβλητή συνδεσιμότητα, κατανάλωση ενέργειας, κατανομή latency, resilience σε node failure, ταυτόχρονη διαχείριση δεδομένων και εφαρμογών. Η υλοποίηση τέτοιων πλαισίων θα ενίσχυε τη διαφάνεια, την αναπαραγωγικότητα και την συγκρισιμότητα των ερευνητικών έργων.

4. Συνδυασμός ενορχήστρωσης εφαρμογών και δεδομένων (application-data co-orchestration)

Η πλειονότητα των υφιστάμενων προσεγγίσεων εστιάζει στον προγραμματισμό εφαρμογών, αγνοώντας το γεγονός ότι σε edge/fog περιβάλλοντα η τοποθέτηση και διαχείριση των δεδομένων αποτελεί εξίσου κρίσιμο παράγοντα. Η μελλοντική έρευνα θα μπορούσε να εστιάσει σε συνεκτικές στρατηγικές co-orchestration, όπου η θέση των εφαρμογών, η κίνηση των δεδομένων και οι μηχανισμοί αποθήκευσης συντονίζονται για την επίτευξη χαμηλότερης καθυστέρησης, ενεργειακής αποδοτικότητας και συμμόρφωσης με πολιτικές δεδομένων (e.g. GDPR, data locality).

5. Υλοποίηση και αξιολόγηση σε πραγματικά περιβάλλοντα

Η πλειονότητα των προτεινόμενων συστημάτων ενορχήστρωσης έχει αξιολογηθεί μόνο σε συνθετικά, εργαστηριακά ή προσομοιωμένα περιβάλλοντα, χωρίς επιβεβαίωση σε πραγματικές συνθήκες λειτουργίας. Επομένως, είναι απαραίτητη η διερεύνηση της τεχνικής και λειτουργικής βιωσιμότητας των πιο υποσχόμενων λύσεων σε context-specific εφαρμογές, όπως:

- Έξυπνες πόλεις (smart cities),
- Βιομηχανικά edge συστήματα (IIoT),
- Ενεργειακά αυτόνομα δίκτυα αισθητήρων,
- Υποδομές υγειονομικής φροντίδας (eHealth, telemonitoring),
- Κρίσιμα συστήματα μεταφορών (V2X, vehicular fog computing).

6. Συστήματα ασφάλειας και εμπιστοσύνης με επίγνωση τοποθεσίας και δεδομένων

Η μελλοντική έρευνα πρέπει να ενσωματώσει ασφαλείς πολιτικές ενορχήστρωσης που να λαμβάνουν υπόψη τη γεωγραφική θέση των κόμβων, τα χαρακτηριστικά των δεδομένων (ευαίσθητα, ανώνυμα, νομικά δεσμευμένα), και τα προφίλ αξιοπιστίας κάθε υποδομής. Ο σχεδιασμός geo-aware και data-aware orchestrators με ενσωματωμένα μοντέλα εμπιστοσύνης αποτελεί σημαντική προοπτική, ιδιαίτερα σε διεθνώς κατανεμημένα δίκτυα με νομικές και κανονιστικές διαφοροποιήσεις.

Η προώθηση των παραπάνω ερευνητικών κατευθύνσεων θα επιτρέψει τη μετάβαση από θεωρητικά μοντέλα σε πρακτικές, επεκτάσιμες και ασφαλείς λύσεις ενορχήστρωσης, ενισχύοντας τη δυνατότητα λειτουργίας έξυπνων, αυτόνομων και προσαρμοζόμενων συστημάτων σε κάθε επίπεδο του Cloud-to-Thing Continuum.

ΠΑΡΑΡΤΗΜΑ Α

Πίνακας Μεταδεδομένων Ερευνητικών Ερωτημάτων

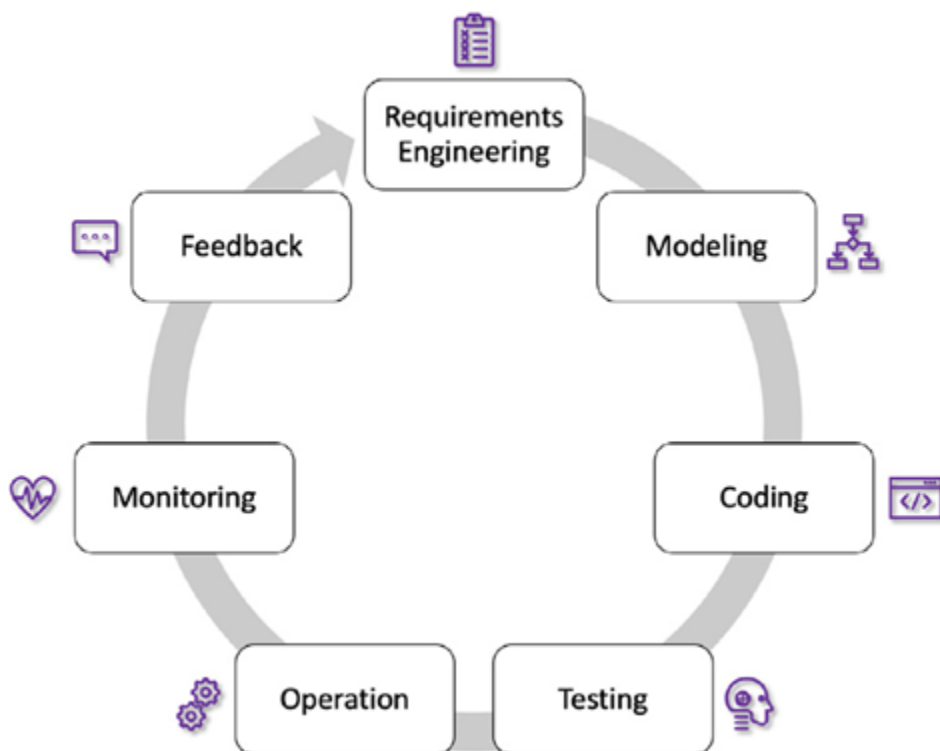
Authors	Title	RQ	Venue	Year	DOI
L. M. Saini et al.	KubeEdge for Scalable IoT Applications	RQ1	ACM	2024	10.1109/IC3I61595.2024.10828876
A. Marchese and O. Tomarchio	Enhancing the Kubernetes Platform with a Load-Aware Orchestration Strategy	RQ1	ACM	2025	10.1007/s42979-025-03712-z
A. Jeffery et al.	Rearchitecting Kubernetes for the Edge	RQ1	ResearchGate	2021	10.1145/3434770.3459730
E. Truyen et al.	Feasibility of container orchestration for adaptive performance isolation	RQ1	ResearchGate	2020	10.1145/3341105.3374034
N. Naydenov and S. Ruseva	Cloud Container Orchestration Architectures: a Systematic Mapping Study	RQ1	ResearchGate	2023	10.1109/INFOTEH57020.2023.10094059
G. M. Freire et al.	A Distributed Software Architecture for IoT: Container Orchestration Impact and Evaluation	RQ1	ScienceDirect	2024	10.1016/j.procs.2024.06.019
B. I. Ismail et al.	Evaluation of Docker as Edge computing platform	RQ1	SpringerLink	2016	10.1109/ICOS.2015.7377291

M. S. A. Muthanna and A. Tselykh	Development of Docker and Kubernetes Orchestration Platforms for Industrial IoT Service Migration	RQ1	Wiley	2022	10.1109/MoNeTec55448.2022.9960769
R. Figueiredo and K. Subratie	Demo: EdgeVPN.io: Open-source VPN for Seamless Edge Computing with Kubernetes	RQ1	MDPI	2020	10.1109/SEC50012.2020.00032
T. Gupta and S. Bhambhani	A Scalable Framework for IoT Applications in Cloud Environments	RQ1	CORE	2025	10.1109/ICSC64553.2025.10967655
R. Girau et al.	Definition and implementation of the Cloud Infrastructure for the integration of the Human Digital Twin in the Social Internet of Things	RQ1	CORE	2024	10.1016/j.comnet.2024.110632
F. Tusa and S. Clayman	End-to-end slices to orchestrate resources and services in the cloud-to-edge continuum	RQ1	Containers.pdf	2023	10.1016/j.future.2022.11026
S. Risco et al.	Rescheduling serverless workloads across the cloud-to-edge continuum	RQ1	ResearchGate	2024	10.1016/j.future.2023.12015
M. Schneider et al.	A Framework for Software Deployment in Manufacturing Environments	RQ1	ACM	2024	10.1016/j.procir.2024.10204
A. Yousefpour et al.	All one needs to know about fog computing and related edge computing paradigms: A complete survey	RQ1	ACM	2019	10.1016/j.sysarc.2019.02009
D. Loconte et al.	Expanding the cloud-to-edge continuum to the IoT in serverless federated learning	RQ1, RQ3	IEEE	2024	10.1016/j.future.2024.02024
Y. Pan et al.	A performance comparison of cloud-based container orchestration tools	RQ2	ACM	2019	10.1109/ICBK.2019.00033
R. Lakshminarayanan	Evaluation of Performance in Container Orchestration with Single-Board Computers in k3s Cluster	RQ2	Science Direct	2024	10.1109/ICCDA64887.2024.10867392
J. Entrialgo et al.	Joint Autoscaling of Containers and Virtual Machines for Cost Optimization	RQ2, RQ3	Science Direct	2024	10.1007/s10723-023-09732-4
N/A	CLOUD DATA SECURITY METHODS: KUBERNETES VS DOCKER SWARM	RQ2	Science Direct	2022	10.56726/irjmets32176
I. Vasireddy et al.	Kubernetes and Docker Load Balancing: State-of-the-Art Techniques and Challenges	RQ2	Wiley	2023	10.55524/ijirem.2023.106.7
D. Yakubov and D. Hästbacka	Comparative Analysis of Lightweight Kubernetes Distributions for Edge Computing	RQ2	Wiley	2025	10.1007/978-3-031-84617-5_7

R. Fayos-Jordan et al.	Performance comparison of container orchestration platforms with low cost devices in the fog	RQ2	arXiv	2020	10.1016/j.jnca.2020.102788
W. U. Zahra et al.	Analyzing the Limitations and Efficiency of Configuration Strategies in Hybrid Cloud Environments	RQ2, RQ3	arXiv	2024	10.56979/702/2024
D. Yakubov and D. Hästbacka	Comparative Analysis of Lightweight Kubernetes Distributions for Edge Computing	RQ2	arXiv	2025	10.1007/978-3-031-84617-5_7
I. Ahmad et al.	Container scheduling techniques: A Survey and assessment	RQ2	arXiv	2022	10.1016/j.jksuci.2021.03002
K. Sharma et al.	Performance Metrics Analysis for Container Scheduling in Cloud Environment	RQ2	arXiv	2024	10.1109/ACOIT62457.2024.10941176
M. Usman, S. Ferlin, and A. Brunstrom	Performance Analysis of Lightweight Container Orchestration Platforms for Edge-Based IoT Applications	RQ2	CORE	2024	10.1109/SEC62691.2024.00032
B. Ali et al.	GAIKube: Generative AI-based Proactive Kubernetes Container Orchestration Framework	RQ3	Science Direct	2024	10.1109/TCCN.2024.3508771
S. Ren et al.	Blockchain-Based Trusted Container Orchestration for Edge Computing	RQ3	Science Direct	2022	10.1109/ICTC55196.2022.9952673
M. Gaglianese et al.	Green Orchestration of Cloud-Edge Applications	RQ3	MDPI	2023	10.1109/SOSE58276.2023.00036
Y. Zhang et al.	Does Generative AI Generate Smells Related to Container Orchestration?	RQ3	ACM	2024	10.1145/3643991.3645079
Z. Zhong et al.	Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions	RQ3	ResearchGate	2022	10.1145/3510415
J. Dogani and F. Khunjush	Proactive auto-scaling in container-based edge computing using federated learning	RQ3	Science Direct	2024	10.1016/j.jpdc.2024.104837
E. Zeydan et al.	Intelligent Service Orchestration in Edge Cloud Networks	RQ3	Wiley	2021	10.1109/MNET.101.210214
R. P. De Prado et al.	Smart containers schedulers for microservices in cloud-fog-IoT networks	RQ3	MDPI	2020	10.3390/s20061714
W. Li et al.	Microservice Workflow Scheduling with a Resource Configuration Model	RQ3	Science Direct	2025	10.3390/s25041253

DevOps, MLOps & CI/CD

Η σύγχρονη ανάπτυξη λογισμικού και συστημάτων βρίσκεται αντιμέτωπη με ένα συνεχώς αυξανόμενο επίπεδο πολυπλοκότητας, που εκτείνεται από τη σύλληψη της ιδέας και την αρχική δημιουργία, μέχρι τη συνεχή διαχείριση, την παρακολούθηση και την εξέλιξη των παραγόμενων λύσεων. Η αδιάκοπη επιδίωξη για τη βελτίωση της ποιότητας, της λειτουργικότητας και της αξιοπιστίας των προϊόντων λογισμικού έχει οδηγήσει στην ευρεία και ταχεία υιοθέτηση σύγχρονων μεθοδολογιών συνεχούς μηχανικής λογισμικού και συστημάτων. Ένα εμβληματικό παράδειγμα αυτών των μεθοδολογιών αποτελούν οι πρακτικές **DevOps**, οι οποίες επιδιώκουν τη στενή συνεργασία και την ενοποίηση των ομάδων ανάπτυξης (Development) και λειτουργιών (Operations). Παράλληλα, παρατηρείται μια ραγδαία ενσωμάτωση της **Τεχνητής Νοημοσύνης (AI)**, και ειδικότερα της **Μηχανικής Μάθησης (ML)**, σε ποικίλα και κρίσιμα στάδια του συνολικού κύκλου ζωής της ανάπτυξης λογισμικού, με στόχο την αυτοματοποίηση, την πρόβλεψη και τη βελτιστοποίηση των διαδικασιών.

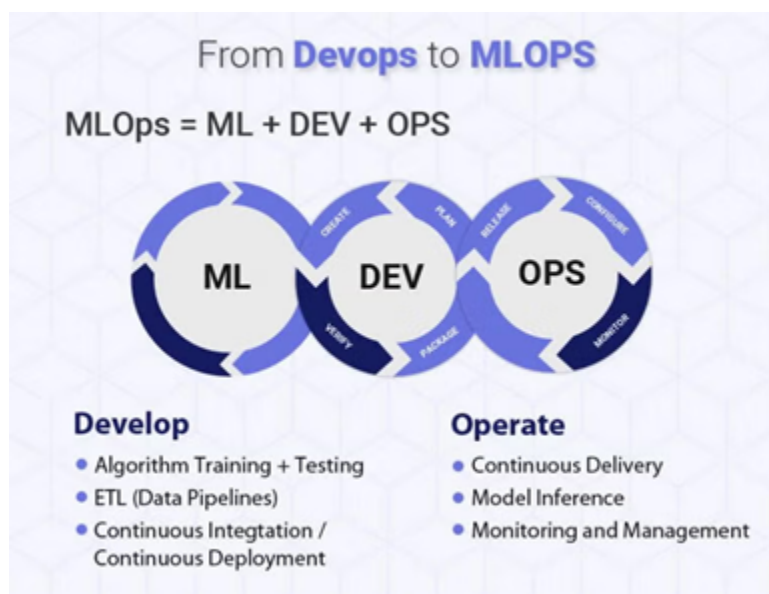


Εικόνα 1 – Κύκλος ζωής ανάπτυξης εφαρμογών με έμφαση στις φάσεις απαιτήσεων, μοντελοποίησης, υλοποίησης, δοκιμών, λειτουργίας, παρακολούθησης και ανατροφοδότησης.

Η συνεχής μηχανική λογισμικού αντιπροσωπεύει μια θεμελιώδη μετατόπιση παραδείγματος στον τρόπο με τον οποίο αντιλαμβανόμαστε και υλοποιούμε την ανάπτυξη λογισμικού. Δεν αποτελεί απλώς ένα σύνολο τεχνικών πρακτικών, αλλά μια ολιστική προσέγγιση που συνδέει άρρηκτα την επιχειρηματική στρατηγική ενός οργανισμού με τις αναπόφευκτα σύνθετες διαδικασίες ανάπτυξης και διάθεσης λογισμικού. Ο απώτερος στόχος είναι η παραγωγή ποιοτικού λογισμικού που όχι μόνο ανταποκρίνεται στις τρέχουσες ανάγκες των επιχειρήσεων και των τελικών χρηστών, αλλά είναι επίσης ευέλικτο και προσαρμόσιμο στις συνεχώς εξελισσόμενες απαιτήσεις της αγοράς και του ανταγωνισμού. Στο πλαίσιο αυτό, το κίνημα

των DevOps έρχεται συμπληρωματικά να ενισχύσει αυτήν τη φιλοσοφία, εστιάζοντας στην επίτευξη μιας απρόσκοπτης και συνεχούς ενοποίησης των δραστηριοτήτων που σχετίζονται με την ανάπτυξη λογισμικού και την επιχειρησιακή του υλοποίηση και λειτουργία στο παραγωγικό περιβάλλον. Αυτή η ισχυρή συνέργεια καλλιεργεί και ενισχύει τη συνεργασία, την επικοινωνία και την αμοιβαία κατανόηση μεταξύ των διαφόρων εμπλεκόμενων ομάδων, συμπεριλαμβανομένων των ομάδων ανάπτυξης, των ομάδων λειτουργιών, των ομάδων ασφάλειας και άλλων βασικών ενδιαφερομένων μερών. Ως αποτέλεσμα, επιταχύνονται σημαντικά οι κύκλοι επανάληψης (iteration cycles), βελτιώνεται η ροή ανατροφοδότησης (feedback loops) για την ταχύτερη αναγνώριση και επίλυση προβλημάτων και προωθείται ο εκτεταμένος αυτοματισμός σε ολόκληρο το pipeline παροχής λογισμικού (software delivery pipeline), από την αρχική δημιουργία κώδικα μέχρι την τελική διάθεση και παρακολούθηση της εφαρμογής.[62]

Η τεχνητή νοημοσύνη αναδεικνύεται ως ένας στρατηγικός παράγοντας που διεισδύει σε διάφορους επιχειρηματικούς τομείς και διαθέτει την επαναστατική δυνατότητα να αναμορφώσει ριζικά τις καθιερωμένες πρακτικές DevOps. Εργαλεία και τεχνικές που αξιοποιούν την τεχνητή νοημοσύνη προσφέρουν προηγμένες δυνατότητες, όπως η προγνωστική ανάλυση (predictive analytics) για την έγκαιρη πρόβλεψη πιθανών προβλημάτων, η αυτοματοποίηση επαναλαμβανόμενων και χρονοβόρων εργασιών, απελευθερώνοντας έτσι τους ανθρώπινους πόρους για πιο δημιουργικές και στρατηγικές δραστηριότητες, και η υποστήριξη για την έξυπνη λήψη αποφάσεων (intelligent decision-making) σε κάθε φάση του κύκλου ζωής του λογισμικού. Ένα χαρακτηριστικό και ολοένα και πιο διαδεδομένο παράδειγμα της συνένωσης της τεχνητής νοημοσύνης με τις πρακτικές DevOps είναι το AIOps (Artificial Intelligence for IT Operations). Το AIOps εφαρμόζει τις αρχές και τις τεχνικές της τεχνητής νοημοσύνης απευθείας στις λειτουργίες πληροφορικής, επιφέροντας σημαντικές βελτιώσεις σε τομείς όπως η αποτελεσματικότερη ανίχνευση και πρόληψη ευπαθειών ασφαλείας (security vulnerabilities), η βελτίωση της αποδοτικότητας και της ακρίβειας στον έλεγχο ποιότητας του λογισμικού (software testing), η αυτοματοποιημένη διαχείριση συμβάντων και η πρόβλεψη πιθανών διακοπών λειτουργίας, συμβάλλοντας έτσι στη διασφάλιση της συνεχούς και αδιάλειπτης λειτουργίας των κρίσιμων συστημάτων και εφαρμογών. Επιπλέον, η ενσωμάτωση πρακτικών CI/CD (Continuous Integration/Continuous Delivery), οι οποίες εστιάζουν στην αυτοματοποίηση της ενσωμάτωσης αλλαγών κώδικα και της συχνής παράδοσης λογισμικού, αποτελεί έναν κρίσιμο παράγοντα για την επιτυχία των DevOps και την αποτελεσματική αξιοποίηση των δυνατοτήτων της τεχνητής νοημοσύνης στον σύγχρονο κύκλο ζωής ανάπτυξης λογισμικού. Παράλληλα, στον αναδυόμενο τομέα της διαχείρισης μοντέλων μηχανικής μάθησης, οι αρχές και οι πρακτικές των DevOps επεκτείνονται και προσαρμόζονται, δημιουργώντας το πεδίο του MLOps (Machine Learning Operations), το οποίο στοχεύει στην αυτοματοποίηση και την τυποποίηση των διαδικασιών ανάπτυξης, δοκιμής και διάθεσης μοντέλων μηχανικής μάθησης, διασφαλίζοντας την αξιοπιστία και την αναπαραγωγιμότητα των λύσεων τεχνητής νοημοσύνης.



Εικόνα 2 – Μετάβαση από DevOps σε MLOps: ενσωμάτωση των διαδικασιών ανάπτυξης και λειτουργίας με τεχνικές μηχανικής μάθησης για αυτοματοποίηση και συνεχή μάθηση.

Ο ορισμός του DevOps αποτελεί αντικείμενο συζήτησης, με απόψεις που ποικίλλουν από το ότι πρόκειται για ένα πολιτισμικό κίνημα ταχείας ανάπτυξης λογισμικού έως έναν τίτλο θέσης που απαιτεί δεξιότητες τόσο στον προγραμματισμό όσο και στις λειτουργίες πληροφορικής. Η συνέργεια αυτή επιταχύνει τους κύκλους επανάληψης, βελτιώνει τους βρόχους ανατροφοδότησης για ταχεία επίλυση προβλημάτων και προάγει τον αυτοματισμό κατά μήκος του pipeline παροχής λογισμικού. Η τεχνητή νοημοσύνη ενσωματώνεται στρατηγικά σε συγκεκριμένους επιχειρηματικούς τομείς, κατέχοντας σημαντικές δυνατότητες αναδιαμόρφωσης των DevOps πρακτικών. Εργαλεία και τεχνικές που υποστηρίζονται από AI παρέχουν προγνωστική ανάλυση, αυτοματοποίηση επαναλαμβανόμενων εργασιών και έξυπνη λήψη αποφάσεων σε όλο τον κύκλο ζωής του λογισμικού. Ένα χαρακτηριστικό παράδειγμα αυτής της τάσης είναι η εμφάνιση του AIOps, που εφαρμόζει τεχνητή νοημοσύνη στις λειτουργίες IT, ενισχύοντας πτυχές όπως η ανίχνευση ευπαθειών ασφαλείας και η αποτελεσματικότητα στον έλεγχο λογισμικού. Ωστόσο, η εφαρμογή των DevOps παραμένει συχνά κατακεραματισμένη και αποσπασματική εντός οργανισμών. Το pipeline παροχής λογισμικού αντιμετωπίζεται συχνά ως μια αλληλουχία διακριτών και ανεξάρτητων φάσεων, γεγονός που εμποδίζει την πλήρη αξιοποίηση των δυνατοτήτων του. Καθώς οι DevOps και AI ωριμάζουν, γίνεται αντιληπτή η ανάγκη για μια πιο ολιστική και ολοκληρωμένη προσέγγιση, ικανή να διαχειριστεί την κλιμακούμενη πολυπλοκότητα της συνεχούς ανάπτυξης εξελιγμένων συστημάτων. Η συνεχής μηχανική σύνθετων συστημάτων στο πλαίσιο των DevOps απαιτεί την υιοθέτηση μεθόδων ενισχυμένων από AI και εξειδικευμένων εργαλείων, ικανά να εξάγουν χρήσιμη γνώση και αξιοποιήσιμες πληροφορίες από τα τεράστια streams γεγονότων και τον πλούτο σχεδιαστικής πληροφορίας που παράγεται κατά την ανάπτυξη και λειτουργία. Στόχος είναι να εξοπλιστούν οι ομάδες ανάπτυξης και λειτουργιών με ευφυή υποστήριξη για συνεχή βελτίωση συστημάτων, ταχύτερες και πιο αξιόπιστες αναπτύξεις, αποτελεσματική συνεργασία και σημαντική μείωση του downtime μέσω προγνωστικής συντήρησης και εντοπισμού ανωμαλιών. Σε αυτό το πλαίσιο, το Model-Driven Engineering

(MDE) αναδεικνύεται ως καίριο παράδειγμα για τη διαχείριση της πολυπλοκότητας των σύγχρονων συστημάτων, αυξάνοντας το επίπεδο αφαίρεσης χρησιμοποιώντας μοντέλα ως βασικές αναπαραστάσεις του συστήματος και του περιβάλλοντός του, διευκολύνοντας την επικοινωνία, την ανάλυση και την αυτόματη παραγωγή κώδικα. Η έλλειψη ολοκληρωμένων προσεγγίσεων που να συνδυάζουν συστηματικά τις αρχές του MDE, του DevOps και της AI με ενιαία εργαλεία αποτελεί εμπόδιο για την επίτευξη πραγματικά συνεχούς και έξυπνης μηχανικής λογισμικού. Το DevOps βασίζεται σε συνεχείς βρόχους ανατροφοδότησης, ευελιξία απέναντι στις μεταβαλλόμενες απαιτήσεις και στη βαθιά αυτοματοποίηση. Το AIOps βασίζεται σε AI και ML για να αντιμετωπίσει τις προκλήσεις του μοντέρνου IT, απαιτώντας ολοκλήρωση με το DevOps pipeline, με ροές προειδοποιήσεων, insights και προβλέψεων που θα ενισχύουν κρίσιμες λειτουργίες ανάπτυξης και λειτουργίας.[63]

DevOps Phases	AIDoArT Architecture Components
 Requirements Engineering	 AI for Requirement Engineering
 Modeling	 Data Representation, Model-based capability, AI for Modelling
 Coding	 AI for Code,
 Testing	 AI for Testing
 Operation	 Data management, Data Handling capability, Storage capability, Computation capability, Ingestion & Handling, Automation
 Monitoring	 Data collection, AI for Monitoring
 Feedback	 Engagement & Analysis, Accountability, Explainability

Εικόνα 3 – Αντιστοίχιση φάσεων του DevOps κύκλου με τα αντίστοιχα αρχιτεκτονικά στοιχεία της πλατφόρμας AIDoArT που αξιοποιούν τεχνητή νοημοσύνη.

Το DevOps θεμελιώνεται σε τρεις βασικούς πυλώνες που αλληλεπιδρούν και αλληλοσυμπληρώνονται για την επίτευξη της μέγιστης αποδοτικότητας και ευελιξίας στην ανάπτυξη και λειτουργία λογισμικού. Πρώτον, η αδιάλειπτη παροχή ανατροφοδότησης αποτελεί κεντρικό στοιχείο, διασφαλίζοντας ότι οι ομάδες ανάπτυξης και λειτουργίας λαμβάνουν συνεχείς πληροφορίες για την απόδοση, τα προβλήματα και τις ανάγκες των χρηστών. Αυτή η συνεχής ροή ανατροφοδότησης επιτρέπει την έγκαιρη αναγνώριση και επίλυση προβλημάτων, τη βελτίωση των διαδικασιών και την προσαρμογή των προϊόντων στις μεταβαλλόμενες απαιτήσεις. Δεύτερον, η προσαρμοστικότητα απέναντι σε μεταβαλλόμενες απαιτήσεις είναι ζωτικής σημασίας σε ένα δυναμικό περιβάλλον όπου οι ανάγκες των επιχειρήσεων και των χρηστών εξελίσσονται διαρκώς. Το DevOps προωθεί την ευελιξία και την ικανότητα των ομάδων να ανταποκρίνονται γρήγορα και αποτελεσματικά στις νέες απαιτήσεις, μειώνοντας τον χρόνο παράδοσης και αυξάνοντας την

ανταγωνιστικότητα. Τρίτον, η εκτεταμένη αυτοματοποίηση διαδραματίζει καθοριστικό ρόλο στην επιτάχυνση των διαδικασιών, τη μείωση των ανθρώπινων σφαλμάτων και την απελευθέρωση των ομάδων για να επικεντρωθούν σε πιο στρατηγικές εργασίες. Η αυτοματοποίηση καλύπτει ένα ευρύ φάσμα δραστηριοτήτων, από τη δημιουργία και τη δοκιμή κώδικα έως την ανάπτυξη και την παρακολούθηση των εφαρμογών.

Στον σύγχρονο ψηφιακό κόσμο, οι υποδομές πληροφορικής γίνονται ολοένα και πιο πολύπλοκες και ογκώδεις, δημιουργώντας νέες προκλήσεις για τη διαχείριση και τη λειτουργία τους. Το AIOps (Artificial Intelligence for IT Operations) αναδύεται ως μια καινοτόμος προσέγγιση που αξιοποιεί την Τεχνητή Νοημοσύνη και τη Μηχανική Μάθηση για την αντιμετώπιση αυτών των προκλήσεων. Μέσω της ανάλυσης μεγάλων όγκων δεδομένων από διάφορες πηγές, το AIOps είναι σε θέση να εντοπίζει ανωμαλίες, να προβλέπει πιθανά προβλήματα και να αυτοματοποιεί διαδικασίες που σχετίζονται με τη λειτουργία των IT συστημάτων. Ωστόσο, για να επιτευχθεί η πλήρης αξιοποίηση των δυνατοτήτων του AIOps, είναι απαραίτητη η ενσωμάτωσή του με τη ροή εργασιών του DevOps. Αυτή η συνέργεια επιτρέπει την απρόσκοπτη ροή πληροφοριών και την αυτοματοποίηση ενεργειών σε όλο τον κύκλο ζωής της ανάπτυξης και λειτουργίας λογισμικού. Συγκεκριμένα, η ενσωμάτωση του AIOps παρέχει στις ομάδες DevOps ροές ειδοποιήσεων σε πραγματικό χρόνο για πιθανά προβλήματα, εμπειριστατωμένες πληροφορίες και αναλύσεις για την κατανόηση των αιτιών των προβλημάτων και προβλέψεις για μελλοντικές τάσεις και κινδύνους. Αυτές οι δυνατότητες ενισχύουν κρίσιμες λειτουργίες ανάπτυξης και διαχείρισης, όπως η έγκαιρη ανίχνευση και επίλυση σφαλμάτων, η βελτιστοποίηση της απόδοσης των εφαρμογών, η πρόληψη διακοπών λειτουργίας και η αυτοματοποίηση επαναλαμβανόμενων εργασιών, οδηγώντας σε αυξημένη αποδοτικότητα, μειωμένο κόστος και βελτιωμένη εμπειρία χρήστη. Η συνδυαστική δύναμη του DevOps και του AIOps δημιουργεί ένα ευφύες και αυτοματοποιημένο περιβάλλον IT που είναι ικανό να ανταποκρίνεται αποτελεσματικά στις απαιτήσεις του σύγχρονου ψηφιακού μετασχηματισμού.[64]

Η υιοθέτηση του DevOps έχει σημαντικές επιπτώσεις στην οργανωσιακή κουλτούρα, απαιτώντας τη δημιουργία νέων ρόλων και την αναδιάρθρωση υφιστάμενων ομάδων. Η συνεπής ανάπτυξη ενός ψηφιακού προϊόντος δεν είναι απλώς μια τεχνική διαδικασία, αλλά μια ολιστική προσέγγιση που απαιτεί συνεχή επαγρύπνηση και αυστηρή εποπτεία σε όλα τα στάδια του κύκλου ζωής του. Η ασφαλής παρακολούθηση των επιδόσεων, της σταθερότητας και της διαθεσιμότητας είναι θεμελιώδους σημασίας για την έγκαιρη ανίχνευση και αντιμετώπιση προβλημάτων, διασφαλίζοντας έτσι μια απρόσκοπτη εμπειρία χρήσης. Παράλληλα, η τακτική και αυτοματοποιημένη ανάπτυξη νέων εκδόσεων, εμπλουτισμένων με νέες λειτουργίες και βελτιώσεις, αποτελεί κινητήρια δύναμη για την εξέλιξη του προϊόντος και την ανταπόκρισή του στις μεταβαλλόμενες ανάγκες της αγοράς και των χρηστών. Πέρα από την τεχνική αρτιότητα, η λειτουργικότητα του προϊόντος και η ικανοποίηση των πελατών κατέχουν περίοπτη θέση. Ένα προϊόν, όσο άρτια τεχνικά και αν είναι, δεν έχει αξία αν δεν ανταποκρίνεται στις προσδοκίες και τις ανάγκες των χρηστών του. Η συνεχής συλλογή και ανάλυση feedback από τους πελάτες, καθώς και η μέτρηση βασικών δεικτών ικανοποίησης, είναι απαραίτητες για την κατανόηση των αδυναμιών και των ευκαιριών βελτίωσης του προϊόντος.

Πέρα από την επιτάχυνση του κύκλου κυκλοφορίας και την ενίσχυση της εμπιστοσύνης, υπάρχουν και άλλα εξίσου σημαντικά κίνητρα που ωθούν τους οργανισμούς στην υιοθέτηση του DevOps. Μεταξύ αυτών περιλαμβάνονται η δυνατότητα για ταχύτερη και συχνότερη

ανάπτυξη λογισμικού, η οποία επιτρέπει την άμεση ανταπόκριση σε ανάγκες και ευκαιρίες. Επιπρόσθετα, το DevOps διευκολύνει την ταχύτερη λήψη ανατροφοδότησης από τους χρήστες και τα ενδιαφερόμενα μέρη, η οποία είναι απαραίτητη για τη συνεχή βελτίωση του λογισμικού. Ένας ακόμη κρίσιμος παράγοντας είναι η ενίσχυση της διαμοιραζόμενης γνώσης και η προώθηση της στενής συνεργασίας μεταξύ των ομάδων ανάπτυξης (Development) και λειτουργιών (Operations). Η βελτιωμένη επικοινωνία και η βαθύτερη κατανόηση των ρόλων και των προκλήσεων κάθε ομάδας οδηγεί στη δημιουργία πιο αξιόπιστου, ανθεκτικού και αποτελεσματικού λογισμικού. Η συνεργασία αυτή συμβάλλει στην άρση των εμποδίων και στην ομαλότερη ροή εργασίας κατά τη διάρκεια ολόκληρου του κύκλου ζωής του λογισμικού.

Τέλος, ένα ευρύ φάσμα τεχνολογικών και πολιτισμικών εργαλείων διαδραματίζει καθοριστικό ρόλο στην επιτυχή εφαρμογή του DevOps, ιδίως σε περιπτώσεις όπου οι ομάδες είναι γεωγραφικά κατακεκομμένες. Τα τεχνολογικά εργαλεία είναι απαραίτητα για την αυτοματοποίηση και την απλοποίηση των βασικών διαδικασιών που εμπλέκονται στην ανάπτυξη και τη διάθεση λογισμικού. Σε αυτά περιλαμβάνονται οι αγωγοί Συνεχούς Ενοποίησης/Συνεχούς Παράδοσης (CI/CD), οι οποίοι αυτοματοποιούν τη διαδικασία από την δημιουργία του κώδικα μέχρι την ανάπτυξή του σε παραγωγικό περιβάλλον. Επιπλέον, η χρήση πρακτικών Infrastructure as Code (IaC) επιτρέπει την διαχείριση και την παροχή υποδομών με αυτοματοποιημένο και επαναλήψιμο τρόπο. Η ενδελεχής παρακολούθηση (monitoring) και η καταγραφή (logging) των συστημάτων είναι επίσης κρίσιμες για την έγκαιρη ανίχνευση και αντιμετώπιση προβλημάτων. Παράλληλα με τα τεχνολογικά εργαλεία, τα πολιτισμικά εργαλεία και οι πρακτικές είναι εξίσου σημαντικά για την καλλιέργεια ενός συνεργατικού περιβάλλοντος. Η ενίσχυση της ανοικτής και αποτελεσματικής επικοινωνίας, η προώθηση της διαμοιραζόμενης γνώσης μεταξύ των μελών της ομάδας και η δημιουργία ενός κλίματος αμοιβαίου σεβασμού και εμπιστοσύνης είναι απαραίτητα στοιχεία για την επιτυχία του DevOps, ανεξάρτητα από τη φυσική τοποθεσία των ομάδων.[65]

Είναι σημαντικό να αναγνωρίσουμε ότι η διεξαγωγή έρευνας σχετικά με την υιοθέτηση του DevOps, ιδίως σε περιβάλλοντα όπου οι ομάδες είναι κατακεκομμένες γεωγραφικά, αντιμετωπίζει ορισμένες σημαντικές προκλήσεις όσον αφορά την εγκυρότητα των ευρημάτων. Η εγκυρότητα των συμπερασμάτων (conclusion validity), η οποία αφορά την ορθότητα των συμπερασμάτων που εξάγονται από την ανάλυση των δεδομένων, και η εγκυρότητα των κατασκευών (construct validity), η οποία σχετίζεται με το κατά πόσο οι χρησιμοποιούμενες μετρήσεις και έννοιες αποδίδουν αποτελεσματικά τα θεωρητικά κατασκευάσματα που μελετώνται, αποτελούν κρίσιμα ζητήματα. Ειδικότερα, η ακρίβεια με την οποία μετρώνται οι έννοιες που σχετίζονται με το DevOps και η επιλογή κατάλληλων και συστηματικών στρατηγικών για την αναζήτηση της σχετικής βιβλιογραφίας μπορούν να επηρεάσουν σημαντικά την εγκυρότητα της έρευνας. Για την αποτελεσματική αντιμετώπιση αυτών των πιθανών απειλών για την εγκυρότητα, οι ερευνητές εφαρμόζουν αυστηρές μεθοδολογίες που περιλαμβάνουν τη διαφανή και λεπτομερή καταγραφή όλων των βημάτων της ερευνητικής διαδικασίας, καθώς και τη σαφή αναφορά στα θεμελιώδη και αναγνωρισμένα έργα του επιστημονικού πεδίου του DevOps. Η διαφάνεια στη μεθοδολογία και η σύνδεση με την υπάρχουσα γνώση ενισχύουν την αξιοπιστία και την εγκυρότητα των ερευνητικών ευρημάτων.[62], [63]

Βιβλιογραφία

- [1] A. Malviya and R. K. Dwivedi, “A Comparative Analysis of Container Orchestration Tools in Cloud Computing,” in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, Mar. 2022, pp. 698–703. doi: 10.23919/INDIACom54597.2022.9763171.
- [2] J. Z. Tam *et al.*, “A Containerization Framework for Bioinformatics Software to Advance Scalability, Portability, and Maintainability,” in *ACM-BCB 2023 - 14th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, Association for Computing Machinery, Inc, Sep. 2023. doi: 10.1145/3584371.3612948.
- [3] S. K. Mondal, X. Wu, and R. Pan, “On Rapid Application Deployment with Kubernetes,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2024, pp. 76–80. doi: 10.1145/3700523.3700538.
- [4] A. S. Navarro, M. Garcia-Pineda, and J. Gutierrez-Aguado, “PodInsights: a millisecond pod metric collector for Kubernetes,” in *Proceedings of the 12th Euro-American Conference on Telematics and Information Systems, EATIS 2024*, Association for Computing Machinery, Inc, Mar. 2025. doi: 10.1145/3685243.3685281.
- [5] D. Maia, F. Correia, and P. Queiroz, “Configurational Patterns of Container Orchestration,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2024. doi: 10.1145/3698322.3698342.
- [6] R. Vaño, I. Lacalle, and C. E. Palau, “Federation of distributed domains in the Cloud-Edge-IoT Continuum,” in *Proceedings of the 2nd International Workshop on MetaOS for the Cloud-Edge-IoT Continuum*, New York, NY, USA: ACM, Mar. 2025, pp. 14–19. doi: 10.1145/3721889.3721923.
- [7] Z. Zhong, M. Xu, M. A. Rodriguez, C. Xu, and R. Buyya, “Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions,” *ACM Comput Surv*, vol. 54, no. 10, Jan. 2022, doi: 10.1145/3510415.

- [8] A. A. Moustafa, S. A. Elsayed, and H. S. Hassanein, "Community-Oriented Edge Computing Platform," *Comput Commun*, vol. 239, Jul. 2025, doi: 10.1016/j.comcom.2025.108179.
- [9] M. D. Hossain *et al.*, "The role of microservice approach in edge computing: Opportunities, challenges, and research directions," Dec. 01, 2023, *Korean Institute of Communications and Information Sciences*. doi: 10.1016/j.ict.2023.06.006.
- [10] R. Girau *et al.*, "Definition and implementation of the Cloud Infrastructure for the integration of the Human Digital Twin in the Social Internet of Things," *Computer Networks*, vol. 251, Sep. 2024, doi: 10.1016/j.comnet.2024.110632.
- [11] F. Tusa and S. Clayman, "End-to-end slices to orchestrate resources and services in the cloud-to-edge continuum," *Future Generation Computer Systems*, vol. 141, pp. 473–488, Apr. 2023, doi: 10.1016/j.future.2022.11.026.
- [12] S. Risco, C. Alarcón, S. Langarita, M. Caballer, and G. Moltó, "Rescheduling serverless workloads across the cloud-to-edge continuum," *Future Generation Computer Systems*, vol. 153, pp. 457–466, Apr. 2024, doi: 10.1016/j.future.2023.12.015.
- [13] M. Schneider, B. Götz, and T. Bauernhansl, "A Framework for Software Deployment in Manufacturing Environments," in *IFAC-PapersOnLine*, Elsevier B.V., 2024, pp. 1043–1048. doi: 10.1016/j.procir.2024.10.204.
- [14] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Sep. 01, 2019, *Elsevier B.V.* doi: 10.1016/j.sysarc.2019.02.009.
- [15] D. Loconte, S. Ieva, A. Pinto, G. Loseto, F. Scioscia, and M. Ruta, "Expanding the cloud-to-edge continuum to the IoT in serverless federated learning," *Future Generation Computer Systems*, vol. 155, pp. 447–462, Jun. 2024, doi: 10.1016/j.future.2024.02.024.
- [16] M. Yousif, "The Edge, the Cloud, and the Continuum," in *2022 Cloud Continuum, Cloud-Continuum 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/CloudContinuum57429.2022.00001.
- [17] I. Cilic, I. P. Zarko, and M. Kusek, "Towards service orchestration for the cloud-to-thing continuum," in *2021 6th International Conference on Smart and*

Sustainable Technologies, SpliTech 2021, Institute of Electrical and Electronics Engineers Inc., Sep. 2021. doi: 10.23919/SpliTech52315.2021.9566410.

- [18] *Proceedings : 2021 IEEE 23rd International Conference on High Performance Computing & Communications, 7th International Conference on Data Science & Systems, 19th International Conference on Smart City, 7th International Conference on Dependability in Sensor, Cloud & Big Data Systems & Applications : HPCC-DSS-SmartCity-DependSys 2021 : 20-22 December 2021, Haikou, Hainan, China*. IEEE Computer Society, Conference Publishing Services, 2021.
- [19] J. Harika, P. Baleeshwar, K. Navya, and H. Shanmugasundaram, “A Review on Artificial Intelligence with Deep Human Reasoning,” in *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 81–84. doi: 10.1109/ICAAIC53929.2022.9793310.
- [20] A. Ullah *et al.*, “Orchestration in the Cloud-to-Things compute continuum: taxonomy, survey and future directions,” Dec. 01, 2023, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1186/s13677-023-00516-5.
- [21] N. Naydenov and S. Ruseva, “Cloud Container Orchestration Architectures, Models and Methods: a Systematic Mapping Study,” in *2023 22nd International Symposium INFOTEH-JAHORINA, INFOTEH 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/INFOTEH57020.2023.10094059.
- [22] B. Oliveira *et al.*, “Function-as-a-Service for the Cloud-to-Thing Continuum: A Systematic Mapping Study Function-as-a-Service for the Cloud-to-Thing Continuum: A Systematic Mapping Study Function-as-a-Service for the Cloud-to-Thing continuum: a Systematic Mapping Study,” pp. 82–93, 2023, doi: 10.5220/0011982600003482i.
- [23] A. Carrera-Rivera, F. Larrinaga, and G. Lasa, “How-to conduct a systematic literature review: A quick guide for computer science research,” Nov. 01, 2022, *Elsevier B.V.* doi: 10.1016/j.compind.2022.103730.
- [24] Y. Pan, I. Chen, F. Brasileiro, G. Jayaputera, and R. Sinnott, “A performance comparison of cloud-based container orchestration tools,” in *Proceedings - 10th IEEE International Conference on Big Knowledge, ICBK 2019*, Institute of Electrical and Electronics Engineers Inc., Nov. 2019, pp. 191–198. doi: 10.1109/ICBK.2019.00033.

- [25] R. Lakshminarayanan, "Evaluation of Performance in Container Orchestration with Single-Board Computers in k3s Cluster," in *2024 2nd International Conference on Computing and Data Analytics, ICCDA 2024 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICCDA64887.2024.10867392.
- [26] B. Ali *et al.*, "GAIKube: Generative AI-based Proactive Kubernetes Container Orchestration Framework for Heterogeneous Edge Computing," *IEEE Trans Cogn Commun Netw*, 2024, doi: 10.1109/TCCN.2024.3508771.
- [27] L. M. Saini, P. Ajmani, V. Asha, K. Pithamber, and N. Sreevani, "KubeEdge for Scalable IoT Applications," in *Proceedings of International Conference on Contemporary Computing and Informatics, IC3I 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 899–905. doi: 10.1109/IC3I61595.2024.10828876.
- [28] J. Entrialgo, M. García, J. García, J. M. López, and J. L. Díaz, "Joint Autoscaling of Containers and Virtual Machines for Cost Optimization in Container Clusters," *J Grid Comput*, vol. 22, no. 1, Mar. 2024, doi: 10.1007/s10723-023-09732-4.
- [29] A. Marchese and O. Tomarchio, "Enhancing the Kubernetes Platform with a Load-Aware Orchestration Strategy," *SN Comput Sci*, vol. 6, no. 3, Mar. 2025, doi: 10.1007/s42979-025-03712-z.
- [30] A. Jeffery, H. Howard, and R. Mortier, "Rearchitecting Kubernetes for the Edge," in *EdgeSys 2021 - Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2021*, Association for Computing Machinery, Inc, Apr. 2021, pp. 7–12. doi: 10.1145/3434770.3459730.
- [31] S. Ren, C. Lee, and Z. Latif, "Blockchain-Based Trusted Container Orchestration for Edge Computing," in *International Conference on ICT Convergence*, IEEE Computer Society, 2022, pp. 88–92. doi: 10.1109/ICTC55196.2022.9952673.
- [32] E. Truyen, A. Jacobs, S. Verreydt, E. H. Beni, B. Lagaisse, and W. Joosen, "Feasibility of container orchestration for adaptive performance isolation in multi-tenant SaaS applications," in *Proceedings of the ACM Symposium on Applied Computing*, Association for Computing Machinery, Mar. 2020, pp. 162–169. doi: 10.1145/3341105.3374034.
- [33] N. Naydenov and S. Ruseva, "Cloud Container Orchestration Architectures, Models and Methods: a Systematic Mapping Study," in *2023 22nd International Symposium*

INFOTEH-JAHORINA, INFOTEH 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/INFOTEH57020.2023.10094059.

- [34] G. M. Freire, H. P. Curasma, and J. C. Estrella, “A Distributed Software Architecture for IoT: Container Orchestration Impact and Evaluation,” in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 224–231. doi: 10.1016/j.procs.2024.06.019.
- [35] M. Gaglianese, J. Soldani, S. Forti, and A. Brogi, “Green Orchestration of Cloud-Edge Applications: State of the Art and Open Challenges,” in *Proceedings - 17th IEEE International Conference on Service-Oriented System Engineering, SOSE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 250–261. doi: 10.1109/SOSE58276.2023.00036.
- [36] B. I. Ismail *et al.*, “Evaluation of Docker as Edge computing platform,” in *ICOS 2015 - 2015 IEEE Conference on Open Systems*, Institute of Electrical and Electronics Engineers Inc., Jan. 2016, pp. 130–135. doi: 10.1109/ICOS.2015.7377291.
- [37] “CLOUD DATA SECURITY METHODS: KUBERNETES VS DOCKER SWARM,” *International Research Journal of Modernization in Engineering Technology and Science*, Dec. 2022, doi: 10.56726/irjmets32176.
- [38] I. Vasireddy, G. Ramya, and P. Kandi, “Kubernetes and Docker Load Balancing: State-of-the-Art Techniques and Challenges,” *International Journal of Innovative Research in Engineering and Management*, vol. 10, no. 6, pp. 49–54, Dec. 2023, doi: 10.55524/ijirem.2023.10.6.7.
- [39] M. S. A. Muthanna and A. Tselykh, “Development of Docker and Kubernetes Orchestration Platforms for Industrial Internet of Things Service Migration,” in *4th International Science and Technology Conference “Modern Network Technologies 2022”, MoNeTec 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/MoNeTec55448.2022.9960769.
- [40] S. Rac and M. Brorsson, “Cost-Effective Scheduling for Kubernetes in the Edge-to-Cloud Continuum.”
- [41] R. Figueiredo and K. Subratie, “Demo: EdgeVPN.io: Open-source Virtual Private Network for Seamless Edge Computing with Kubernetes,” in *Proceedings - 2020 IEEE/ACM Symposium on Edge Computing, SEC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 190–192. doi: 10.1109/SEC50012.2020.00032.

- [42] D. Yakubov and D. Hästbacka, “Comparative Analysis of Lightweight Kubernetes Distributions for Edge Computing: Performance and Resource Efficiency,” 2025, pp. 81–95. doi: 10.1007/978-3-031-84617-5_7.
- [43] T. Goethals, “FLEDGE: Kubernetes Compatible Container Orchestration on Low-resource Edge Devices.”
- [44] R. Fayos-Jordan, S. Felici-Castell, J. Segura-Garcia, J. Lopez-Ballester, and M. Cobos, “Performance comparison of container orchestration platforms with low cost devices in the fog, assisting Internet of Things applications,” *Journal of Network and Computer Applications*, vol. 169, Nov. 2020, doi: 10.1016/j.jnca.2020.102788.
- [45] A. Čop, B. Bertalaníč, and C. Fortuna, “An Overview and Solution for Democratizing AI Workflows at the Network Edge,” Jul. 2024, doi: 10.1016/j.jnca.2025.104180.
- [46] Y. Zhang, R. Meredith, W. Reeves, J. Coriolano, M. A. Babar, and A. Rahman, “Does Generative AI Generate Smells Related to Container Orchestration?: An Exploratory Study with Kubernetes Manifests,” in *Proceedings - 2024 IEEE/ACM 21st International Conference on Mining Software Repositories, MSR 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 192–196. doi: 10.1145/3643991.3645079.
- [47] S. Kaiser, A. S,aman, S. Tosun, and T. Korkmaz, “Edge System Design Using Containers and Unikernels for IoT Applications.”
- [48] W. U. Zahra, M. Talha Amjad, A. Ahsan, and G. Mumtaz, “Analyzing the Limitations and Efficiency of Configuration Strategies in Hybrid Cloud Environments”, doi: 10.56979/702/2024.
- [49] D. Yakubov and D. Hästbacka, “Comparative Analysis of Lightweight Kubernetes Distributions for Edge Computing: Performance and Resource Efficiency,” 2025, pp. 81–95. doi: 10.1007/978-3-031-84617-5_7.
- [50] T. Gupta and S. Bhambhani, “A Scalable Framework for IoT Applications in Cloud Environments,” in *2025 10th International Conference on Signal Processing and Communication (ICSC)*, IEEE, Feb. 2025, pp. 22–29. doi: 10.1109/ICSC64553.2025.10967655.
- [51] Z. Zhong, M. Xu, M. A. Rodriguez, C. Xu, and R. Buyya, “Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions,” *ACM Comput Surv*, vol. 54, no. 10, Jan. 2022, doi: 10.1145/3510415.

- [52] J. Dogani and F. Khunjush, "Proactive auto-scaling technique for web applications in container-based edge computing using federated learning model," *J Parallel Distrib Comput*, vol. 187, May 2024, doi: 10.1016/j.jpdc.2024.104837.
- [53] P.-J. Maenhaut *et al.*, "Resource Management in a Containerized Cloud: Status and Challenges."
- [54] A. Rajuroy and M. Emmanuel, "Optimizing Resource Management and Scalability in Container Orchestration Platforms: A Comparative Study." [Online]. Available: <https://www.researchgate.net/publication/389853261>
- [55] E. Zeydan, J. Mangues-Bafalluy, and Y. Turk, "Intelligent Service Orchestration in Edge Cloud Networks," *IEEE Netw*, vol. 35, no. 6, pp. 126–132, Nov. 2021, doi: 10.1109/MNET.101.2100214.
- [56] R. P. De Prado, S. García-Galán, J. E. Muñoz-Expósito, A. Marchewka, and N. Ruiz-Reyes, "Smart containers schedulers for microservices provision in cloud-fog-IoT networks. challenges and opportunities," *Sensors (Switzerland)*, vol. 20, no. 6, Mar. 2020, doi: 10.3390/s20061714.
- [57] W. Li, X. Li, L. Chen, and M. Wang, "Microservice Workflow Scheduling with a Resource Configuration Model Under Deadline and Reliability Constraints," *Sensors (Basel)*, vol. 25, no. 4, Feb. 2025, doi: 10.3390/s25041253.
- [58] Z. Wang, M. Goudarzi, J. Aryal, and R. Buyya, "Container Orchestration in Edge and Fog Computing Environments for Real-Time IoT Applications."
- [59] I. Ahmad, M. G. AlFailakawi, A. AlMutawa, and L. Alsalman, "Container scheduling techniques: A Survey and assessment," Jul. 01, 2022, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2021.03.002.
- [60] K. Sharma, P. Khurana, and M. Kumar, "Performance Metrics Analysis for Container Scheduling in Cloud Environment," in *2024 Asian Conference on Intelligent Technologies, ACOIT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ACOIT62457.2024.10941176.
- [61] M. Usman, S. Ferlin, and A. Brunstrom, "Performance Analysis of Lightweight Container Orchestration Platforms for Edge-Based IoT Applications," in *Proceedings - 2024 IEEE/ACM Symposium on Edge Computing, SEC 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 321–332. doi: 10.1109/SEC62691.2024.00032.

- [62] R. Amaro, R. Pereira, and M. M. da Silva, “Mapping DevOps capabilities to the software life cycle: A systematic literature review,” Jan. 01, 2025, *Elsevier B.V.* doi: 10.1016/j.infsof.2024.107583.
- [63] M. N. Chowdary, S. Bussa, C. K. Chowdary, and M. Gupta, “Automated Pipeline for the Deployment using OpenShift,” in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 220–229. doi: 10.1016/j.procs.2022.12.025.
- [64] G. Turin, A. Borgarelli, S. Donetti, F. Damiani, E. B. Johnsen, and S. L. Tapia Tarifa, “Predicting resource consumption of Kubernetes container systems using resource models,” *Journal of Systems and Software*, vol. 203, Sep. 2023, doi: 10.1016/j.jss.2023.111750.
- [65] R. Grande, A. Vizcaíno, and F. O. García, “Is it worth adopting DevOps practices in Global Software Engineering? Possible challenges and benefits,” *Comput Stand Interfaces*, vol. 87, Jan. 2024, doi: 10.1016/j.csi.2023.103767.