



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ευφυή Συστήματα Διαχείρισης Υποδομών Νεφελώδους Υπολογισμού

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ανδρέας Π. Καψάλης

Αθήνα, Απρίλιος 2017



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ευφυή Συστήματα Διαχείρισης Υποδομών Νεφελώδους Υπολογισμού

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ανδρέας Π. Καψάλης

Συμβουλευτική Επιτροπή: **Ιάκωβος Στ. Βενιέρης, Καθηγητής Ε.Μ.Π**
Δήμητρα-Θεοδώρα Ι. Κακλαμάνη, Καθηγήτρια Ε.Μ.Π
Νικόλαος Κ. Ουζούνογλου, Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την

.....
Ι. Στ. Βενιέρης
Καθηγητής Ε.Μ.Π

.....
Δ.-Θ. Ι. Κακλαμάνη
Καθηγήτρια Ε.Μ.Π

.....
Ν. Κ. Ουζούνογλου
Καθηγητής Ε.Μ.Π

.....
Σ. Χ. Παπαβασιλείου
Καθηγητής Ε.Μ.Π

.....
Ν. Γ. Κοζύρης
Καθηγητής Ε.Μ.Π

.....
Θ. Α. Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

.....
Χ. Ζ. Πατρικάκης
Αν. Καθηγητής ΤΕΙ
Πειραιά

Αθήνα, Απρίλιος 2017

.....
Ανδρέας Π. Καψάλης
Διδάκτωρ Ε.Μ.Π.

Copyright © Ανδρέας Π. Καψάλης, 2017.

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

Σε όσους με στήριξαν σε αυτή τη προσπάθεια.

Περίληψη

Οι ραγδαίες εξελίξεις στο τομέα των πληροφοριακών συστημάτων, καθώς και η ανάπτυξη νέων προτύπων και τεχνολογιών στους τομείς επικοινωνιών και υλισμικού, έχουν δημιουργήσει νέα είδη υπηρεσιών που βρίσκουν εφαρμογή σε όλους τους τομείς της καθημερινότητας. Η όλο και πιο διαδεδομένη χρήση των υπηρεσιών αυτών συμβάλει στην παραγωγή τεράστιου όγκου δεδομένων και μετά-δεδομένων των οποίων η ανάλυση και επεξεργασία απαιτεί αποδοτικά και ευέλικτα μοντέλα υπολογισμού.

Την τελευταία δεκαετία η ανάγκη αυτή έχει καλυφθεί από τα υπολογιστικά «νέφη», ένα μοντέλο που προσφέρει υπολογιστικούς πόρους, πλατφόρμες και εφαρμογές με τη μορφή ευέλικτων υπηρεσιών. Το μοντέλο υπολογιστικών νεφών είναι πια στις μέρες μας ευρέως διαδεδομένο, και αποτελεί την εξ'ορισμού επιλογή για πολλές μορφές υπηρεσιών. Αυτή όμως η επιτυχία του μοντέλου καθώς επίσης και το γεγονός ότι έχει υιοθετηθεί καθολικά, δημιουργεί νέες ανάγκες, εφόσον η ραγδαία ανάπτυξη νέων υπηρεσιών και νέων τεχνολογιών έχουν ως αποτέλεσμα τη δημιουργία και την ανταλλαγή ακόμα μεγαλύτερου όγκου δεδομένων. Συνεπώς, δημιουργούνται απαιτήσεις για περισσότερους υπολογιστικούς πόρους, μεγαλύτερη ευελιξία στην παροχή τους και εξασφάλιση της ομαλής και συνεχούς λειτουργίας τους. Ένα χαρακτηριστικό παράδειγμα είναι η έκρηξη συστημάτων που διαχειρίζονται δίκτυα ευφυών συσκευών, ενεργοποιώντας το λεγόμενο Διαδίκτυο των Αντικειμένων (Internet of Things – IoT). Για την αποτελεσματική κάλυψη των απαιτήσεων αυτών, δημιουργείται η ανάγκη για τη εύρεση νέων λύσεων, οι οποίες θα έχουν ως στόχο να ικανοποιήσουν τα εξής:

- Υποστήριξη νέων επιπέδων διαλειτουργικότητας για την συνεχή και αποδοτική κάλυψη των ολοένα αυξανόμενων επιπέδων ζήτησης.
- Ευφυής και αποδοτική διαχείριση υποδομών υλισμικού.
- Υποστήριξη ευέλικτων μοντέλων και τεχνικών κατανομής εργασιών και δεδομένων.

Στο πλαίσιο της παρούσας διατριβής προτείνονται αρχιτεκτονικές ευφυών συστημάτων διαχείρισης υποδομών νεφελώδους υπολογισμού, οι οποίες θα καλύπτουν όλα τα πρακτικά και τεχνικά ζητήματα που προκύπτουν για την κάλυψη των προαναφερθεισών αναγκών. Οι αρχιτεκτονικές προτάσεις στο σύνολο τους προσφέρουν μία ολοκληρωμένη λύση ώστε να επιτρέψει σε παρόχους υπηρεσιών υπολογιστικού νέφους να επιτύχουν καλύτερα επίπεδα ποιότητας υπηρεσίας (QoS), επίτευξη καλύτερου επιπέδου τήρησης συμβολαίων χρήσης (SLAs) και αποδοτικότερη διαχείριση των πόρων

υλισμικού με τη χρησιμοποίηση εναλλακτικών τεχνολογιών εικονικοποίησης όπως αυτής των περιεκτών λογισμικού.

Λέξεις κλειδιά: Υπολογιστικά νέφη, ομιχλώδης υπολογισμός, διαδίκτυο των νεφών, διαδίκτυο των αντικειμένων, περιέκτες λογισμικού

Abstract

Rapid advancements in the field of ICT systems, as well as the development of new standards and technologies in the fields of communications and hardware, have created new types of services that apply in various sectors of everyday life. The ever increasing use of these types of services contributes to the production of massive amounts of data and metadata. Processing and analysis of such data requires new, efficient and flexible models of computing.

Over the last decade these requirements have been met by the Cloud Computing paradigm, which offers computing resources, platforms and applications in the form of services. The Cloud Computing paradigm is, nowadays, widely adopted and provides the by-default choice of development for various types of software applications. The success of this model, however, creates ever new needs, since this burst of service offerings results in the production and exchange of even more massive of amounts of data. As a result, new requirements rise that include the need for more resources from the end-user's point of view, and the need for better management of these resources from the provider's point of view. One characteristic example, is of course, the rise of IoT (Internet of Things) applications and platforms that manage and handle huge amounts of produced data from IoT systems. To better cover those needs, new solutions must be provided in order to satisfy the following:

- Support for new layers of interoperability for the continuous and effective coverage of the ever increasing computational resource demand
- Intelligent and efficient resource management of datacenters
- Support for flexible models and techniques of task allocation and execution

The research work presented in this thesis proposes a set of architectural solutions for intelligent systems for the management of Cloud Computing infrastructures, which will cover all practical and technical matters, to achieve the satisfaction of the aforementioned needs and requirements. The overall architecture attempts to create an integrated solution that will allow Cloud Service Providers to achieve better levels of QoS, SLAs and resource management by realizing and adopting innovative models and standards, such as the Fog Computing paradigm, the cooperative model of Cloud Federations and novel virtualization techniques such software containers.

Keywords: *Cloud computing, fog computing, cloud federation, internet of things, software containers*

Ευχαριστίες

Το παρόν κείμενο ολοκληρώνει την ερευνητική μου πορεία κατά τη διάρκεια των διδακτορικών μου σπουδών. Αποτελεί μία προσωπική δικαίωση των προσπαθειών μου, αλλά και μία δικαίωση των ανθρώπων που με εμπιστεύθηκαν και με στήριξαν. Κατά τη παρουσία μου στο Εργαστήριο Ευφώνων Επικοινωνιών και Δικτύων Ευρείας Ζώνης του ΕΜΠ, είχα τη τιμή και την ευτυχία να συνεργαστώ με εξαιρετους ανθρώπους, τους οποίους νιώθω και την ανάγκη να ευχαριστήσω.

Καταρχήν, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντά μου, κύριο Ιάκωβο Βενιέρη, Καθηγητή ΕΜΠ, για την ευκαιρία που μου έδωσε ώστε να εργαστώ σε αυτό το πολύ ενδιαφέρον ερευνητικό πεδίο, την πίστη του σε εμένα, αλλά και την άρτια και εποικοδομητική καθοδήγηση, ώστε να φτάσω στον τελικό στόχο. Θα ήθελα να ευχαριστήσω και την κυρία Δήμητρα-Θεοδώρα Κακλαμάνη, Καθηγήτρια ΕΜΠ, για την άψογη συνεργασία που είχαμε και τις καίριες συμβουλές που μου έδινε, αλλά και τον κύριο Νικόλαο Ουζούνoglou, Καθηγητή ΕΜΠ, για την στήριξη και τη καθοδήγηση σε αυτή τη διαδρομή. Επίσης, ένα θερμό ευχαριστώ στους κυρίους Χαράλαμπο Πατρικάκη, Αν. Καθηγητή ΤΕΙ Πειραιά, και Συμεών Παπαβασιλείου, Καθηγητή ΕΜΠ, για την άψογη συνεργασία και τις συμβουλές που μου παρείχαν. Θα ήθελα να ευχαριστήσω τον κύριο Νεκτάριο Κοζύρη, Καθηγητή ΕΜΠ, και την κυρία Θεοδώρα Βαρβαρίγου, Καθηγήτρια ΕΜΠ, για τη τιμή που μου έκαναν και δέχθηκαν να συμμετέχουν στην επταμελή επιτροπή μου.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τους συνεργάτες μου. Θεωρώ τον εαυτό μου τυχερό και ευτυχή που αποτέλεσα μέρος της συγκεκριμένης ομάδας του εργαστηρίου. Ευχαριστώ τη Δέσποινα και τη Μαριέλα για τη πορεία και τη συνεργασία που είχαμε κατά τη διάρκεια των ερευνητικών προγραμμάτων, αλλά και την όμορφη παρέα που κάναμε. Ευχαριστώ τον Παναγιώτη για την άψογη ερευνητική συνεργασία που είχαμε, αλλά και τις ευχάριστες ποδοσφαιρικές αναλύσεις κάθε Δευτέρας. Ευχαριστώ, επίσης, και τους Πέτρο και Μανώλη, δύο εξαιρετους συνεργάτες και ανθρώπους, που αν και δεν συνεργαστήκαμε άμεσα, η βοήθεια τους ήταν άμεση όταν την χρειάστηκα. Τέλος, θερμά ευχαριστώ στη Μαρία, τη Σοφία και τον Παναγιώτη Γκόνη για την άψογη συνεργασία. Εύχομαι σε όλους καλή τύχη και σταδιοδρομία. Ελπίζω να υπήρξα αντάξιος συνεργάτης τους.

Στην πορεία μου αυτή μεγάλο στήριγμα υπήρξαν οι δικοί μου άνθρωποι. Θέλω λοιπόν να ευχαριστήσω την Αθηνά, που ήταν εκεί όταν τη χρειάστηκα, από την αρχή της διαδρομής, σε κάθε εμπόδιο και κάθε δυσκολία. Της οφείλω πολλά και ελπίζω να φανώ

αντάξιος της. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου Παναγιώτη και Ειρήνη, αλλά και την αδερφή μου Λιάνα, για τις αξίες, την στήριξη και την αγάπη που μου έδωσαν καθ' όλη τη διάρκεια της ζωής μου και με διαμόρφωσαν σαν άνθρωπο. Ελπίζω η ολοκλήρωση αυτής της προσπάθειας να αποτελέσει μία δικαίωση για όλους.

Πίνακας Περιεχομένων

Εισαγωγή	19
1.1 Σύγχρονες ανάγκες υπηρεσιών, υπολογιστικών υποδομών & εφαρμογών	20
1.1.1 Εφαρμογές και υπηρεσίες Διαδικτύου των Αντικειμένων	20
1.1.2 Υποδομές, υπηρεσίες και εφαρμογές υπολογιστικών «νεφών»	22
1.2 Στόχοι της διατριβής	24
Τρέχουσα κατάσταση και υπάρχουσες ανάγκες	27
2.1 Υποστηρικτικές τεχνολογίες	27
2.1.1 Υπηρεσιοστρεφείς & Κατανεμημένες Αρχιτεκτονικές	27
2.1.2 Τεχνολογίες σημασιολογικού ιστού	29
2.1.3 Τεχνολογίες εικονικοποίησης	33
2.2 Σύγχρονα θέματα, αρχιτεκτονικές & τεχνολογίες υποδομών υπολογιστικών νεφών 36	
2.2.1 Κύρια χαρακτηριστικά	37
2.2.2 Μοντέλα υπηρεσιών	37
2.2.3 Μοντέλα ανάπτυξης	39
2.2.4 Συμβόλαια χρήσης υπηρεσιών νέφους	40
2.3 Θέματα ποιότητα παροχής υπηρεσιών (QoS) σε συστήματα και υπηρεσίες υπολογισμού	46
2.3.1 Υπάρχουσες λύσεις	46
2.3.2 Τρέχοντα θέματα	48
2.3.3 Πρακτικές περιπτώσεις χρήσης	50
2.4 Θέματα υπηρεσιών και συστημάτων υπολογισμού με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA)	53
2.4.1 Υπάρχουσες λύσεις	54
2.4.2 Τρέχοντα θέματα	59
2.4.3 Πρακτικές περιπτώσεις χρήσης	60
2.5 Θέματα ενεργειακής αποδοτικότητας σε περιβάλλοντα υπολογιστικών νεφών 62	
2.5.1 Υπάρχουσες λύσεις	65
2.5.2 Τρέχοντα θέματα	72
2.5.3 Πρακτικές περιπτώσεις χρήσης	74
Αρχιτεκτονικές Ευφυών Συστημάτων Διαχείρισης Υποδομών Νεφελώδους Υπολογισμού	78
3.1 Περιγραφή και αρχιτεκτονική προσέγγιση	78
3.2 Συστήματα «νεφελώδους» και «ομιχλώδους» υπολογισμού με έμφαση στη ποιότητα υπηρεσίας	80

3.2.1 Επίπεδο συσκευών.....	81
3.2.2 Επίπεδο δρομολόγησης.....	81
3.2.3 Επίπεδο ομίχλης.....	82
3.2.4 Επίπεδο νέφους.....	83
3.2.5 Πρότυπο μηνύματος «ομίχλης».....	84
3.3 Συστήματα «ομόσπονδων» δικτύων υποδομών υπολογιστικών νεφών με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA).....	86
3.3.1 Μοντέλο διαχείρισης πληροφορίας.....	88
3.3.2 Πρότυπο αιτήματος χρήσης υπηρεσίας.....	88
3.3.3 Υπηρεσίας ομοσπονδίας.....	89
3.3.4 Πράκτορας αντιστοίχισης.....	89
3.3.5 Υπηρεσία συναλλαγών.....	89
3.3.6 Πράκτορας κατάταξης.....	89
3.3.7 Υπηρεσία ευρετηρίου.....	90
3.3.8 Αλγόριθμος κατάταξης.....	90
3.4 Συστήματα διαχείρισης πόρων και υποδομών υπολογιστικών νεφών με έμφαση στην ενεργειακή αποδοτικότητα.....	94
3.4.1 Υπηρεσία πελάτη.....	95
3.4.2 Υπηρεσία εποπτείας.....	95
3.4.3 Υπηρεσία κατανομής πόρων.....	96
3.4.4 Υπηρεσία επιπέδου υποδομής.....	97
3.4.5 Επισημάνσεις σχετικά με τη προτεινόμενη λύση.....	98
Αξιολόγηση αρχιτεκτονικών προτάσεων και πειραματική διαδικασία.....	101
4.1 Υποστηρικτικά εργαλεία για τη ανάπτυξη των προσομοιώσεων.....	101
4.1.1 Πλατφόρμα προσομοιώσεων Cloudsim.....	101
4.1.2 Πλατφόρμα προσομοιώσεων CloudsimEx.....	105
4.2 Συστήματα «νεφελώδους» και «ομιχλώδους» υπολογισμού με έμφαση στη ποιότητα υπηρεσίας.....	106
4.2.1 Περιγραφή πειραματικής διαδικασίας.....	106
4.2.2 Παρουσίαση αποτελεσμάτων πειραματικής διαδικασίας.....	110
4.3 Συστήματα «ομόσπονδων» δικτύων υποδομών υπολογιστικών νεφών με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA).....	113
4.4 Συστήματα διαχείρισης πόρων και υποδομών υπολογιστικών νεφών με έμφαση στην ενεργειακή αποδοτικότητα.....	114
4.4.1 Περιγραφή πειραματικής διαδικασίας.....	114
4.4.2 Παρουσίαση αποτελεσμάτων πειραματικής διαδικασίας.....	118
Συμπεράσματα και προτάσεις για μελλοντική έρευνα.....	122

5.1	Συστήματα «νεφελώδους» και «ομιχλώδους» υπολογισμού με έμφαση στη ποιότητα υπηρεσίας.....	122
5.2	Συστήματα «ομόσπονδων» δικτύων υποδομών υπολογιστικών νεφών με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA).....	124
5.3	Συστήματα διαχείρισης πόρων και υποδομών υπολογιστικών νεφών με έμφαση στην ενεργειακή αποδοτικότητα	126
	Βιβλιογραφία	129
Παράρτημα	
	137
A.1	Παράδειγμα μηνύματος «ομίχλης».....	137
A.2	Προσομοιώσεις αποστολής μηνύματος «ομίχλης»	138
A.3	Αλγόριθμος βαθμολόγησης του διαμεσολαβητή «ομοσπονδίας»	138
A.4	Παράδειγμα αιτήματος «ομοσπονδίας»	139
	Δημοσιεύσεις.....	141

Ευρετήριο Εικόνων

Εικόνα 1 Οντολογία της πλατφόρμας mOSAIC αναφορικά με το απαιτούμενο μοντέλο ανάπτυξης [28].....	33
Εικόνα 2 Σύγκριση αρχιτεκτονικής περιεκτών λογισμικού και εικονικών μηχανημάτων	36
Εικόνα 3 Μοντέλο IaaS όπως αυτό παρέχεται από τη σουίτα λογισμικού αρχιτεκτονικής Openstack [9]	38
Εικόνα 4 Αρχιτεκτονική απεικόνιση της πλατφόρμας AppEngine της Google [40]	39
Εικόνα 5 Αξιολόγηση των συμβολαίων χρήσης μεταξύ πελάτη και παρόχου [48]	46
Εικόνα 6 Παράδειγμα αρχιτεκτονικής «ομιχλώδους» υπολογισμού [49]	47
Εικόνα 7 Αρχιτεκτονική συστήματος ευφούς σηματοδότησης (STL) [57]	52
Εικόνα 8 Αρχιτεκτονική Intercloud του προτύπου P2302 [7]	55
Εικόνα 9 Γενική αρχιτεκτονική της πλατφόρμας mOSAIC [28]	57
Εικόνα 10 Κατηγοριοποίηση «ομόσπονδων» δικτύων νεφών με βάση την αρχιτεκτονική ανάπτυξης [69].....	57
Εικόνα 11 «Ομόσπονδη» ταυτοποίηση υπηρεσιών AWS μέσω της υπηρεσίας ταυτοποίησης Azure Active Directory.....	61
Εικόνα 12 Απόδοση ισχύος εξυπηρετητή IBM x3550 M4 ¹²	64
Εικόνα 13 Ενεργειακή αποδοτικότητα του αλγορίθμου Ant Colony σε σχέση με τον First-Fit Decreasing [81].....	69
Εικόνα 14 Παράδειγμα κωδικοποίησης λύσης ενός γενετικού αλγορίθμου	70
Εικόνα 15 Το υπολογιστικό κέντρο υπηρεσιών «νέφους» και οι τρεις άξονες επέκτασης	78
Εικόνα 16 Προτεινόμενη αρχιτεκτονική της συνεργατικής πλατφόρμας ομιχλώδους υπολογισμού.....	81
Εικόνα 17 Πρότυπο μηνύματος «ομίχλης»	85
Εικόνα 18 Αρχιτεκτονική διαμεσολαβητή ομοσπονδίας υποδομών υπολογιστικών νεφών	88
Εικόνα 19 Δομικές μονάδες της πλατφόρμας Cloudsim [113].....	102
Εικόνα 20 Απόκριση της διαδικασίας βαθμολόγησης έναντι σε διαφορετικά σενάρια αριθμού παρόχων και συναλλαγών.....	114
Εικόνα 21 Συνολική κατανάλωση ενέργειας ανάλογα με την εφαρμοσμένη στρατηγική κατανομής	118

Ευρετήριο Πινάκων

Πίνακας 1 Απαιτήσεις ισχύος ενός εξυπηρετητή IBM x3550 M4 ανάλογα με τη χρησιμοποίηση πόρων συστήματος.....	63
Πίνακας 2 Περιγραφή μετρήσεων προσομοιώσεων	109
Πίνακας 3 Παρουσίαση αποτελεσμάτων προσομοιώσεων.....	111
Πίνακας 4 Διαμόρφωση σεναρίων προσομοιώσεων πλατφόρμας υβριδικής εικονικοποίησης.....	117
Πίνακας 5 Παρουσίαση αποτελεσμάτων προσομοιώσεων σχετικά με την κατανάλωση ενέργειας, παραβίασης SLA και μέσο αριθμό ενεργών εξυπηρετητών κατά τη διάρκεια μίας εκτέλεσης του προσομοιωτή.....	119

Κεφάλαιο 1

Εισαγωγή

Οι ραγδαίες εξελίξεις στο τομέα των πληροφοριακών συστημάτων, καθώς και η ανάπτυξη νέων προτύπων και τεχνολογιών στους τομείς επικοινωνιών και υλισμικού, έχουν δημιουργήσει νέα είδη υπηρεσιών που βρίσκουν εφαρμογή σε όλους τους τομείς της καθημερινότητας. Η όλο και πιο διαδεδομένη χρήση των υπηρεσιών αυτών συμβάλει στην παραγωγή τεράστιου όγκου δεδομένων και μετά-δεδομένων των οποίων η ανάλυση και επεξεργασία απαιτεί αποδοτικά και ευέλικτα μοντέλα υπολογισμού.

Την τελευταία δεκαετία η ανάγκη αυτή έχει καλυφθεί από τα υπολογιστικά «νέφη» (Cloud Computing), ένα μοντέλο που προσφέρει υπολογιστικούς πόρους, πλατφόρμες και εφαρμογές με τη μορφή ευέλικτων υπηρεσιών. Η μετάβαση στο μοντέλο των υπολογιστικών νεφών επιτρέπει την αποσύζευξη της ανάπτυξης εφαρμογών και υπηρεσιών με τη διαχείριση και συντήρηση της υποκείμενης υποδομής η οποία θα υποστηρίξει την λειτουργία της εφαρμογής ή υπηρεσίας. Με αυτό τον τρόπο οργανισμοί και επιχειρήσεις αφιερώνουν πόρους (χρόνος, χρήμα, ανθρώπινο δυναμικό κ.α.) στην βελτιστοποίηση της προσφερόμενης εφαρμογής/υπηρεσίας και αναθέτουν τις υποστηρικτικές λειτουργίες (φιλοξενία, συντήρηση υποδομής υλισμικού και λειτουργικών συστημάτων, αποσφαλμάτωση, επεκτασιμότητα κ.α.) σε τρίτους, σε παρόχους δηλαδή υπηρεσιών υπολογιστικού «νέφους».

Το μοντέλο υπολογιστικών νεφών είναι πια στις μέρες μας ευρέως διαδεδομένο, και αποτελεί την εξ'ορισμού επιλογή για χρήστες και οργανισμούς, οι οποίοι επιζητούν έναν εύκολο και ασφαλή τρόπο για την αποθήκευση των δεδομένων τους, την ανάπτυξη των εφαρμογών τους και την εκτέλεση των υπολογιστικών εφαρμογών τους. Η καθολική, όμως, υιοθέτηση του υπολογιστικού νέφους ως επιχειρησιακή διαδικασία¹, δημιουργεί νέες ανάγκες. Έχοντας ως δεδομένη τη δυνατότητα του «νέφους» να είναι σε θέση να υποστηρίξει τεράστιους φόρτους εργασιών, εκμεταλλευόμενο τη λεγόμενη «ελαστικότητα» του, οι χρήστες επικεντρώνονται στην ανάπτυξη των εφαρμογών και υπηρεσιών τους. Η ραγδαία εξέλιξη νέων μοντέλων και αρχιτεκτονικών εφαρμογών, απαιτεί περισσότερους πόρους (π.χ. το μοντέλο MapReduce [1] και η δημοφιλέστερη υλοποίηση του το Apache Hadoop [2] στηρίζονται στην ύπαρξη μίας κατανεμημένης συστοιχίας υπολογιστών) και δημιουργεί και διακινεί τεράστιους όγκους δεδομένων.

¹ <http://www.datacenterknowledge.com/archives/2016/11/11/2020-92-percent-data-center-traffic-will-cloud/>

Επίσης άρρηκτα συνδεδεμένο με την ύπαρξη του μοντέλου του υπολογιστικού «νέφους», είναι και το μοντέλο του Διαδικτύου των Αντικειμένων (Internet of Things – IoT), ένα σύνολο καινοτόμων τεχνολογιών στους τομείς του υλισμικού και λογισμικού. Το επαναστατικό αυτό μοντέλο, που επιτρέπει σε έξυπνες συσκευές να επικοινωνούν μεταξύ τους για την επίτευξη ενός συνολικού στόχου, έχει δημιουργήσει το έδαφος για τη δημιουργία νέων υπηρεσιών, εφαρμογών και πρωτοκόλλων. Αντίστοιχα όμως, δημιουργεί νέες απαιτήσεις για περισσότερους υπολογιστικούς πόρους, μεγαλύτερη ευελιξία στην παροχή τους και εξασφάλιση της ομαλής και συνεχούς λειτουργίας τους.

Για την αποτελεσματική κάλυψη των αναγκών αυτών, δημιουργείται η ανάγκη για νέα μοντέλα και τεχνικές διαχείρισης υποδομών που προσφέρουν υπηρεσίες «νέφους», τα οποία θα έχουν ως στόχο να ικανοποιήσουν τις εξής συγκεκριμένες ανάγκες:

- Υποστήριξη νέων επιπέδων διαλειτουργικότητας για την συνεχή και αποδοτική κάλυψη των ολοένα αυξανόμενων επιπέδων ζήτησης
- Ευφυής και αποδοτική διαχείριση φυσικών υποδομών
- Υποστήριξη ευέλικτων μεθόδων κατανομής εργασιών και ανάλυσης δεδομένων

Η παρούσα διδακτορική διατριβή προτείνει μία αρχιτεκτονική προσέγγιση η οποία θα καλύπτει όλα τα πρακτικά και τεχνικά ζητήματα που προκύπτουν για την κάλυψη των προαναφερθεισών αναγκών. Η αρχιτεκτονικές προτάσεις, στο σύνολο τους, προδιαγράφουν τις απαραίτητες λειτουργίες οι οποίες θα πρέπει να αναπτυχθούν και θα επιτρέπουν σε υποδομές υπολογιστικών νεφών να καλύψουν αυτές τις ανάγκες. Στη συνέχεια του παρόντος κεφαλαίου εξετάζονται οι σύγχρονες ανάγκες που ενέπνευσαν τη μελέτη και την ανάπτυξη των αρχιτεκτονικών προτάσεων που παρουσιάζονται στη διατριβή.

1.1 Σύγχρονες ανάγκες υπηρεσιών, υπολογιστικών υποδομών & εφαρμογών

1.1.1 Εφαρμογές και υπηρεσίες Διαδικτύου των Αντικειμένων

Ως Διαδίκτυο των Αντικειμένων (Internet of Things – IoT) ορίζεται το μοντέλο πλατφόρμας και υπηρεσιών λογισμικού που βασίζεται στην τη συλλογή και διακίνηση δεδομένων που παράγονται από «έξυπνες» συσκευές. Ως έξυπνες ορίζονται όλες οι συσκευές που έχουν δυνατότητες επικοινωνίας και συνδεσιμότητας με πρωτόκολλα χαμηλού (ZigBee, Bluetooth κ.α.) ή υψηλού επιπέδου (TCP/IP, HTTP κ.α.). Οι παραπάνω συσκευές μπορούν να οργανωθούν σε αυτόνομα δίκτυα και να ανταλλάσσουν δεδομένα

είτε απευθείας, είτε με τη βοήθεια ειδικών ελεγκτών (controllers [3]) που αναλαμβάνουν τη συλλογή των δεδομένων από το σύνολο των συσκευών. Οι περιπτώσεις χρήσης ποικίλλουν και καλύπτουν από οικιακές έως βιομηχανικές και ιατρικές ανάγκες για αυτοματισμό και εποπτεία του ελεγχόμενου περιβάλλοντος. Οι πλατφόρμες IoT ολοκληρώνουν τη λειτουργία των δικτύων έξυπνων συσκευών συλλέγοντας, αποθηκεύοντας και αναλύοντας τα παραγόμενα δεδομένα.

Ο όγκος και ο ρυθμός παραγωγής των δεδομένων των δικτύων έξυπνων συσκευών είναι αρκετά μεγάλος, ώστε να καθιστά απαγορευτική τη λειτουργία λύσεων IoT σε συμβατικά και μη ευέλικτα υπολογιστικά περιβάλλοντα. Τα υπολογιστικά νέφη, επί της ουσίας, ενεργοποίησαν την μαζική ανάπτυξη και υιοθέτηση του μοντέλου IoT, παρέχοντας λύσεις για τη αυξημένη ζήτηση υπολογιστικών πόρων, παραλληλοποίησης και ελαστικότητας για την γρήγορη και αποδοτική ανάλυση των δεδομένων. Οι πλατφόρμες IoT βρίσκουν εφαρμογή σε διάφορους τομείς (ιατρική, βιομηχανία, ενέργεια, οικιακοί αυτοματισμοί). Αυτό έχει σαν αποτέλεσμα την συνεχή παραγωγή δεδομένων τα οποία υπόκεινται σε πολλαπλά επίπεδα επεξεργασίας.

Η φύση των συστημάτων που ορίζονται από δίκτυα έξυπνων συσκευών, όπως π.χ. συστήματα ελέγχου περιβάλλοντος ή συστήματα παρακολούθησης ευπαθών ομάδων, απαιτεί την άμεση απόκριση της πλατφόρμας σε συγκεκριμένες μετρήσεις που υποδηλώνουν καταστάσεις κρισιμότητας. Όταν ένα κεντρικό υπολογιστικό περιβάλλον είναι υπεύθυνο για τη συλλογή και την επεξεργασία δεδομένων από πολλαπλές ροές ταυτόχρονα, είναι σύνηθες να παρατηρείται συμφόρηση τόσο στην δικτυακή όσο και στην υπολογιστική υποδομή, με αποτέλεσμα την καθυστέρηση στην επεξεργασία δεδομένων και αποστολή απάντησης στο υποκείμενο δίκτυο που ελέγχει τις συσκευές. Σε καταστάσεις υψηλής κρισιμότητας (π.χ. εντοπισμός πυρκαγιάς στον ελεγχόμενο χώρο, ανωμαλία στις μετρήσεις ενός ατόμου με συγκεκριμένες παθήσεις) η μη έγκυρη επεξεργασία και εξαγωγή αποφάσεων είναι μη αποδεκτή και καθιστά αδύνατη τη πρόληψη τους.

Παρ' ότι οι υποδομές και τεχνολογίες υπολογιστικού «νέφους» προσέφεραν το απαραίτητο περιβάλλον για την ανάπτυξη και λειτουργία εφαρμογών IoT, η ραγδαία έκρηξη τους, αλλά και οι συγκεκριμένες απαιτήσεις όσον αφορά τη ποιότητα λειτουργίας του περιβάλλοντος στο οποίο αποθηκεύονται και αναλύονται τα δεδομένα, δημιουργούν νέες ανάγκες για επεκτάσιμες υπηρεσίες υπολογισμού.

Η πρόσφατη ανάπτυξη ενός ενδιάμεσου αρχιτεκτονικού μοντέλου, το οποίο αναφέρεται και ως «ομιχλώδης» υπολογισμός (Fog Computing)², χαρακτηρίζεται από την γεωγραφική εγγύτητα ενός δικτύου, υπολογιστικά ικανών (computationally capable) συσκευών, σε σχέση με την πηγή των δεδομένων, δηλαδή το δίκτυο έξυπνων συσκευών. Σκοπός του είναι να προσφέρει υπηρεσίες υπολογισμού με το μικρότερο δυνατό κόστος, ώστε να εξυπηρετήσει ανάγκες αμεσότητας όσο αφορά την επεξεργασία ευαίσθητων και κρίσιμων δεδομένων, χρησιμοποιώντας την κεντρική υποδομή υπολογιστικού νέφους σαν αποθετήριο δεδομένων για αποθήκευση και μακροχρόνια επεξεργασία δεδομένων.

Οι ανάγκες που προκύπτουν κατά την μετάβαση αυτή συνοψίζονται στη δημιουργία αρχιτεκτονικών μοντέλων που θα επιτρέπουν την έξυπνη και ευέλικτη λειτουργία δομών «ομιχλώδους» υπολογισμού, ώστε να εξασφαλίζεται κάθε στιγμή η βέλτιστη ποιότητα υπηρεσίας (QoS) σε δίκτυα έξυπνων συσκευών.

1.1.2 Υποδομές, υπηρεσίες και εφαρμογές υπολογιστικών «νεφών»

Η λειτουργία των σύγχρονων υπολογιστικών κέντρων δεδομένων, τα οποία λειτουργούν στο όλα τα μοντέλα υπηρεσίας, στηρίζεται στην αποτελεσματική κατανομή των πόρων υλισμικού σε εικονικές συστοιχίες. Ένα τυπικό υπολογιστικό κέντρο αποτελείται από εκατοντάδες εξυπηρετητές (Servers). Η συνεχής λειτουργία ενός μεγάλου αριθμού τέτοιων μηχανημάτων μπορεί να επιφέρει τεράστιο κόστος στον πάροχο, κυρίως όσον αφορά την ενέργεια που καταναλώνεται. Μία κοινή πρακτική που μελετάται και ακολουθείται είναι η αυτοματοποιημένη κατανομή των εικονικών μηχανημάτων με σκοπό την βέλτιστη χρησιμοποίηση των πόρων υλισμικού (resource utilization) των διαθέσιμων εξυπηρετητών [4, 5]. Επί της ουσίας, πρόκειται για ένα πρόβλημα βελτιστοποίησης (bin packing problem) που έχει ως σκοπό την χρησιμοποίηση όσο το δυνατόν λιγότερων εξυπηρετητών για τη φιλοξενία του ιδίου αριθμού εικονικών μηχανημάτων, γεγονός το οποίο όχι μόνο θα μειώσει το κόστος συντήρησης για τον πάροχο, αλλά και θα εξασφαλίσει ότι το ενεργειακό αποτύπωμα από τη λειτουργία του υπολογιστικού κέντρου θα είναι μικρότερο και φιλικότερο προς το περιβάλλον.

Η ανάγκη για φιλικότερη προς το περιβάλλον λειτουργία του υπολογιστικού κέντρου, καθώς και η ανάγκη των παρόχων να προσφέρουν τους υπολογιστικούς πόρους όσο το δυνατόν πιο αποδοτικά έχουν οδηγήσει πολλούς παρόχους να υιοθετήσουν διαφορετικές τεχνολογίες σχετικά με την εικονικοποίηση των συστημάτων που

² http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

προσφέρονται προς τους χρήστες. Μία πιο ευέλικτη προσέγγιση είναι προσφορά πόρων υπό τη μορφή περιεκτών λογισμικού (Software Containers). Η ουσιαστική διαφορά τους με τη πιο «παραδοσιακή» και ευρέως διαδεδομένη τεχνολογία υπερεπόπτη είναι ότι οι περιέκτες λογισμικού δεν παρακρατούν μέρος των υπολογιστικών πόρων, όπως συμβαίνει στην περίπτωση των εικονικών μηχανημάτων (Virtual Machines – VMs). Στην ουσία μοιράζονται τον πυρήνα και τις κύριες λειτουργίες του φιλοξενούντος λειτουργικού συστήματος και είναι ιδανικά για να υποστηρίξουν μία μόνο εφαρμογή. Λόγω της ιδιαιτερότητας αυτής καταναλώνουν λιγότερους πόρους από τον εξυπηρετητή, σε αντίθεση με τα εικονικά περιβάλλοντα τεχνολογίας υπερεπόπτη, τα οποία επιβαρύνουν το κυρίως σύστημα καταναλώνοντας πόρους για τη λειτουργία ενός αυτόνομου λειτουργικού συστήματος.

Οι σύγχρονες τάσεις, απαιτούν την ύπαρξη λύσεων που θα επιτρέπουν σε υπολογιστικά κέντρα να χρησιμοποιούν και τις δύο τεχνολογίες εικονικοποίησης με τέτοιο τρόπο όπου θα ικανοποιούνται τόσο οι ανάγκες του παρόχου (κόστος, συντήρηση, πρόληψη και διαχείρισης σφαλμάτων κλπ.), αλλά και οι ανάγκες των χρηστών των υπηρεσιών «νέφους», όσον αφορά την ποιότητα της τελικής προσφερόμενης υπηρεσίας (QoS) μέσω πιο ανταγωνιστικών συμβολαίων χρήσης υπηρεσίας (Service Level Agreements - SLAs). Θα πρέπει επίσης, να υποστηρίζονται οι κατάλληλες αρχιτεκτονικές λειτουργίες που θα επιτρέπουν το διαχωρισμό και τη κατηγοριοποίηση των εφαρμογών και υπηρεσιών, με βάση το φύση λειτουργίας, και την αυτόματη κατανομή τους σε αντίστοιχους τύπους εικονικοποιημένων υπολογιστικών δομών (Software Containers ή VMs). Η διαδικασία αυτή απαιτεί την συνεργασία τεχνολογιών αυτοματοποίησης και διαμόρφωση λειτουργικών συστημάτων (Automation Tools [6]), καθώς και αλγορίθμων μηχανικής μάθησης (machine learning) για την ανάλυση ιστορικών δεδομένων χρήσης των υπηρεσιών και λειτουργίας της υποδομής.

Οι υπηρεσίες και εφαρμογές υπολογιστικών «νεφών» έχουν επεκταθεί σε σημείο ώστε να εξυπηρετούν πρακτικά το μεγαλύτερο μέρος των εφαρμογών που χρησιμοποιούνται σήμερα από έναν μέσο χρήστη. Ταυτόχρονα, του προσφέρουν την δυνατότητα να μεταφέρει το σύνολο της εργασίας του σε ένα απομακρυσμένο εικονικό περιβάλλον το οποίο έχει διαμορφωθεί αποκλειστικά σύμφωνα με τις ανάγκες του (μοντέλο υπηρεσίας IaaS).

Ο κατά προσέγγιση αριθμός τελικών χρηστών υπηρεσιών «νέφους» υπολογίζεται ότι μέχρι το 2020 θα αυξηθεί τόσο, ώστε το μοντέλο SaaS προβλέπεται να

αντικαταστήσει το 25% παραδοσιακών εφαρμογών (CRM, capital management)³. Αυτό μπορεί να οδηγήσει σε αδυναμία ικανοποίησης αιτημάτων για μίσθωση πόρων και υπηρεσιών, σε μείωση της ποιότητας διαφόρων υπηρεσιών, καθώς και σε αδυναμία τήρησης των συμφωνηθέντων SLAs. Επιπρόσθετα, τίθεται και το θέμα της κινητικότητας των πόρων, εφαρμογών αλλά και δεδομένων των χρηστών. Για παράδειγμα, ένα χρήστης ο οποίος μετακινείται σε μία άλλη ήπειρο θα απολαμβάνει χειρότερο επίπεδο υπηρεσίας καθώς η πρόσβαση στις υπηρεσίες ενός παρόχου θα είναι δυσκολότερη λόγω αυξημένης γεωγραφικής απόστασης, και χειρότερη ποιότητα δικτύου κλπ.

Τα παραπάνω θέματα, οδηγούν στην ανάγκη εύρεσης μίας μορφής διαλειτουργικότητας μεταξύ παρόχων υπηρεσιών υπολογιστικού νέφους. Λόγω της ύπαρξης πολλαπλών τεχνολογιών για την ανάπτυξη υποδομών υπολογιστικών «νεφών», καθώς και υποκείμενων τεχνολογιών και χαρακτηριστικών υποδομών, υπάρχει η ανάγκη προτυποποίησης της συνεργασίας δύο ή περισσότερων παρόχων, δημιουργώντας έτσι τις αναφερόμενες στη βιβλιογραφία «ομοσπονδίες» υπολογιστικών «νεφών» (Cloud Federations). Το υπό ανάπτυξη πρότυπο της IEEE P2302 [7] προδιαγράφει μία αρχιτεκτονική προσέγγιση αλλά και προτυποποίηση των υποκείμενων επικοινωνιών για την δημιουργία «ομόσπονδων» δικτύων παρόχων υπηρεσιών «νέφους».

Η ύπαρξη τέτοιων συνεργατικών δομών προϋποθέτει επίσης και την ύπαρξη ενδιάμεσων ελεγκτικών μηχανισμών διαμεσολάβησης (Brokering), ευρετηρίου υπηρεσιών (Service Registry), αιτημάτων και ελέγχου της ποιότητας των επί μέρους συναλλαγών μεταξύ των παρόχων. Μία οντότητα διαμεσολάβησης (Broker) θα πρέπει να εφαρμόζει αλγορίθμους αντιστοίχισης αιτημάτων σε παρόχους με τρόπο που θα εξασφαλίζει τη μέγιστο δυνατό όφελος για το σύνολο του συνεργατικού αυτού δικτύου.

1.2 Στόχοι της διατριβής

Τα σύγχρονα μοντέλα εφαρμογών και υπηρεσιών συνδέονται άρρηκτα με τις υπηρεσίες υπολογιστικού «νέφους» για λόγους που έχουν αναφερθεί και ανωτέρω. Τα ζητήματα που προκύπτουν από αυτή την έκρηξη των μοντέλων αυτών, των οποίων χαρακτηριστικό παράδειγμα είναι οι εφαρμογές IoT, είναι ικανά να οδηγήσουν σε συνθήκες όπου παραδοσιακές αρχιτεκτονικές ενδεχομένως να αδυνατούν να εξυπηρετήσουν τον ολοένα και πιο απαιτητικό φόρτο εργασίας που παράγεται. Αρκεί να αναλογιστεί κανείς πως ήδη πλατφόρμες IoT χρησιμοποιούνται για την ανάλυση

³ <http://www.gartner.com/newsroom/id/3119717>

δεδομένων που συλλέγονται από συσκευές που δύνανται να είναι διαμοιρασμένες κατά το εύρος μίας έκτασης που καλύπτει ολόκληρες πόλεις (π.χ. υπηρεσίες Smart Grid που προσφέρονται από παρόχους και διανομείς ηλεκτρικής ενέργειας [8]).

Η αποδοτική διαχείριση και ικανοποίηση των ζητημάτων αυτών απαιτούν μία καινοτόμα προσέγγιση στην αρχιτεκτονική των υπολογιστικών «νεφών», καθώς η λειτουργία τους αναμένεται να ξεφύγει από τα στενά όρια ενός ή πολλών υπολογιστικών κέντρων που ανήκουν σε ένα οργανισμό. Η φύση των υπηρεσιών που προσφέρει ένα υπολογιστικό κέντρο πρέπει να επεκταθεί με βάση τη συνεχή επίτευξη διαλειτουργικότητας με άλλες πλατφόρμες, υπηρεσίες αλλά και πρότυπα. Επιπρόσθετα, σύγχρονες λύσεις πρέπει να εφαρμοσθούν ώστε οι ήδη υπάρχουσες υπηρεσίες να είναι πιο ευέλικτες, αποδοτικές και εναρμονισμένες με τις σύγχρονες αυτές απαιτήσεις.

Σε αυτό το πλαίσιο η παρούσα διατριβή προτείνει λύσεις με τις οποίες ένα σύγχρονο υπολογιστικό κέντρο θα διαθέτει όλα τα εφόδια ώστε να υποστηρίξει όλες τις σύγχρονες μορφές υπηρεσιών και εφαρμογών. Η έμφαση δόθηκε στην ενσωμάτωση των πιο καινοτόμων τεχνολογιών αιχμής που δραστηριοποιούνται γύρω από τον τομέα των υπολογιστικών «νεφών». Η συνολική αρχιτεκτονική προσέγγιση επιχειρεί να επιτύχει τη βελτίωση των λειτουργικών χαρακτηριστικών ενός υπολογιστικού κέντρου και να προσθέσει τις απαραίτητες υπηρεσίες στην ήδη υπάρχουσα πλειάδα, ώστε να επεκτείνει τον τρόπο λειτουργίας του. Η δημιουργία συνεργατικών δικτύων υπολογιστικών «νεφών», η επέκταση των λειτουργιών στο επίπεδο του «ομιχλώδους» υπολογισμού αλλά και η ευέλικτη διαχείριση των υπάρχοντων πόρων του υπολογιστικού κέντρου, αποτελούν τους κύριους άξονες γύρω από τους οποίους αναπτύσσονται οι αρχιτεκτονικές βελτιώσεις που προτείνονται στην παρούσα διατριβή.

Με βάση αυτούς παραπάνω άξονες η παρούσα αρχιτεκτονική παραθέτει και εκτενή αξιολόγηση των εν λόγω προτάσεων, επιδιώκοντας να επικυρώσει την χρησιμότητα τους, σε σχέση με τους δείκτες αποδοτικότητας που μετριοούνται στα σύγχρονα συστήματα υπολογιστικών «νεφών»: την ποιότητα υπηρεσίας, την ενεργειακή αποδοτικότητα και τα επίπεδα τήρησης των συμβολαίων χρήσης (SLA).

Κεφάλαιο 2

Τρέχουσα κατάσταση και υπάρχουσες ανάγκες

Στο παρόν κεφάλαιο παρουσιάζεται η τρέχουσα κατάσταση στα πεδία που εξετάζει η διατριβή. Συγκεκριμένα, διερευνάται η πρόοδος από τεχνολογικής και ερευνητικής απόψεως σύγχρονων αρχιτεκτονικών, μοντέλων και προτύπων που παρουσιάζουν συνάφεια με τους τρεις άξονες γύρω από τους οποίους αναπτύσσονται οι αρχιτεκτονικές προτάσεις της διατριβής. Εξετάζονται υπάρχοντα συστήματα «ομιχλώδους» υπολογισμού, «ομοσπονδιών» υποδομών «νεφών», αλλά και τεχνικές και αρχιτεκτονικές ευφυούς διαχείρισης πόρων υποδομών με έμφαση στην ενεργειακή αποδοτικότητα. Η αναφορά τους στοχεύει αφενός στη παρουσίαση καινοτόμων ερευνητικών εργασιών και προτάσεων για τη βελτίωση συγκεκριμένων θεμάτων, και αφετέρου στη παρουσίαση υπαρχουσών συστημάτων που εφαρμόζονται σήμερα και προσφέρονται ως ώριμα και έτοιμα προϊόντα από παρόχους υπηρεσιών «νέφους» και συναφών υπηρεσιών.

Τέλος, θεωρήθηκε αναγκαίο να παρουσιαστούν όλες οι υποστηρικτικές τεχνολογίες, επάνω στις οποίες βασίστηκαν οι αρχιτεκτονικές προτάσεις της παρούσας διατριβής. Οι εν λόγω τεχνολογίες καλύπτουν τους τομείς των υπηρεσιών διαδικτύου, σημασιολογικού διαδικτύου, τεχνικών εικονικοποίησης. Παρουσιάζονται επίσης όλα τα κύρια χαρακτηριστικά των υπολογιστικών νεφών, ως μοντέλων υπηρεσιών, και τα είδη ανάπτυξης που εφαρμόζονται έως σήμερα.

2.1 Υποστηρικτικές τεχνολογίες

2.1.1 Υπηρεσιοστρεφείς & Κατανεμημένες Αρχιτεκτονικές

2.1.1.1 Αρχιτεκτονικές REST

Όλα τα μοντέλα ευέλικτων και επεκτάσιμων υπηρεσιών υπολογισμού («νεφελώδεις» & «ομιχλώδεις») στηρίζονται στην λειτουργία αυτόνομων μονάδων λογισμικού, οι οποίες ονομάζονται υπηρεσίες λογισμικού. Οι αρχιτεκτονικές που στηρίζονται σε υπηρεσίες λογισμικού αναφέρονται κυρίως ως υπηρεσιοστρεφείς. Σήμερα, οι αρχιτεκτονικές REST (Representational State Transfer) αποτελούν την πιο διαδεδομένη επιλογή ανάπτυξης υπηρεσιοστρεφών πλατφορμών. Οι υπηρεσίες REST χρησιμοποιούν το πρωτόκολλο HTTP για να προδιαγράψουν μία σειρά από καταστάσεις ή λειτουργίες. Κάθε λειτουργία χαρακτηρίζεται από ένα HTTP URI και μία ή περισσότερες

μεθόδους ανταλλαγής δεδομένων του πρωτοκόλλου HTTP (GET, POST, PUT, DELETE). Η προτυποποίηση με βάση το πρωτόκολλο HTTP επιτρέπει μεγάλα επίπεδα ευελιξίας, καθώς η επικοινωνία μέσω μηνυμάτων HTTP δημιουργεί ένα αφαιρετικό επίπεδο όπου διαφορετικές υπηρεσίες μπορούν να επικοινωνήσουν αγνοώντας παντελώς τα χαρακτηριστικά ή μία της άλλης, όπως γλώσσα και πλατφόρμα ανάπτυξης.

Εξαιτίας αυτής της ευελιξίας, οι υπηρεσιοστρεφείς αρχιτεκτονικές αποτέλεσαν τη καθιερωμένη προσέγγιση για την ανάπτυξη συστημάτων υποδομής υπολογιστικών «νεφών». Ένα παράδειγμα πλατφόρμας ανάπτυξης υπολογιστικού «νέφους» είναι το Openstack [9], το οποίο αποτελείται από μία συλλογή υπηρεσιών οι οποίες αναλαμβάνουν την ευθύνη να φέρνουν εις πέρας διαφορετικές δομικές λειτουργίες. Για παράδειγμα, η υπηρεσία αυθεντικοποίησης Keystone [10] αποτελεί το κεντρικό σημείο εισόδου στην πλατφόρμα, ελέγχοντας τους ρόλους, τα δικαιώματα πρόσβασης και τα διακριτικά ασφαλείας κάθε οντότητας, είτε αυτή είναι ένας απλός χρήστης είτε μία άλλη υπηρεσία.

Το παραπάνω παράδειγμα καθιστά πιο προφανή τη χρησιμότητα υπηρεσιοστρεφών αρχιτεκτονικών για την υποστήριξη υποδομών υπολογιστικών «νεφών», επιτρέποντας υψηλά επίπεδα επεκτασιμότητας, κλιμακωσιμότητας αλλά και διαλειτουργικότητα με κάθε τύπου ετερογενή συστήματα.

2.1.1.2 Αρχιτεκτονικές Δημοσίευσης/Συνδρομής (Pub/Sub)

Οι αρχιτεκτονικές REST, παρότι αποτελούν μία από τις πιο διαδεδομένες προσεγγίσεις για την ανάπτυξη υπηρεσιοστρεφών εφαρμογών, χαρακτηρίζονται από ένα σημαντικό μειονέκτημα. Η επικοινωνία μεταξύ των υπηρεσιών είναι μονόδρομη, με αποτέλεσμα όταν ένας χρήστης ή εφαρμογή καλεί μία υπηρεσία για την εκτέλεση μίας μεθόδου, δεν μπορεί να ξέρει παρά μόνον ότι η κλήση έχει γίνει και τα δεδομένα έχουν αποσταλεί επιτυχώς.

Η ανάγκη για ασύγχρονη επικοινωνία μεταξύ εφαρμογών και υπηρεσιών, οδήγησε στη δημιουργία αρχιτεκτονικών ανταλλαγής μηνυμάτων με ανάλογο ασύγχρονο τρόπο. Η αρχιτεκτονική Δημοσίευσης/Συνδρομής (Publish/Subscribe) αποτελεί ένα μοντέλο όπου εφαρμογές και υπηρεσίες συνδέονται σε ένα κοινό τόπο ή αλλιώς «θέμα» (topic). Όλες οι εφαρμογές ή υπηρεσίες που συνδέονται σε ένα κοινό θέμα ονομάζονται συνδρομητές. Κάθε οντότητα μπορεί να αποστείλει ένα μήνυμα, αναλαμβάνοντας έτσι το ρόλο εκδίδουσας οντότητας (Publisher). Το μήνυμα αποστέλλεται σε όλες τις εγγεγραμμένες οντότητες του θέματος με πλήρως ασύγχρονο τρόπο. Κάθε συνδρομητής είναι ταυτόχρονα και εκδότης, δημιουργώντας έτσι ένα περιβάλλον ανταλλαγής μηνυμάτων, το οποίο είναι πλήρως παραμετροποιήσιμο και

επεκτάσιμο. Για την επιτυχή ανταλλαγή και δρομολόγηση μηνυμάτων κάθε θέμα διατηρεί ένα διαμεσολαβητή (Broker) ο οποίος διατηρεί μία λίστα των συνδρομητών, ο οποίος αναλαμβάνει τη διαχείριση του θέματος, εξασφαλίζοντας ότι κάθε συνδρομητής θα πληρεί συγκεκριμένα ποιοτικά χαρακτηριστικά που θα καθορίζουν τη παρουσία του και θα εξασφαλίζουν τη ομαλή ανταλλαγή των μηνυμάτων.

Παράδειγμα αρχιτεκτονικής Pub/Sub αποτελεί το πρωτόκολλο επικοινωνίας MQTT [11], το οποίο είναι προσαρμοσμένο για την υποστήριξη δικτύων έξυπνων συσκευών, όπου μπορεί να παρατηρηθούν φαινόμενα υψηλής καθυστέρησης δικτύου (high latency).

2.1.2 Τεχνολογίες σημασιολογικού ιστού

Στις μέρες μας, όπου η εξέλιξη του Παγκόσμιου Ιστού είναι πλέον δεδομένη και όπου οι τεχνολογικές εξελίξεις γεννούν όλο και περισσότερες πηγές δεδομένων, η ανάγκη μίας προηγμένης προσέγγισης διαχείρισης της πληροφορίας είναι αναγκαία. Ταυτόχρονα, η ενσωμάτωση του υπερμεγέθους συνόλου των δεδομένων κρίνεται αναγκαία, καθώς με τον τρόπο αυτό είναι δυνατή η σφαιρική κατανόηση της κατανομημένης πληροφορίας και οι σχέσεις που επικρατούν μεταξύ των ποικίλων τμημάτων της [12]. Μία λύση, η οποία διασφαλίζει την αποδοτική διαχείριση και ενσωμάτωση ετερογενών δεδομένων και, κατ' επέκταση, την αποτελεσματική διασύνδεση εφαρμογών και συστημάτων, είναι η σημασιολογική περιγραφή των υποκείμενων δεδομένων και διαδικασιών. Αυτό συνεπάγεται την περιγραφή των προηγούμενων έτσι, ώστε το νόημά τους να είναι κατανοητό από μηχανές.

Αυτή η θεμελιώδης ιδέα του σημασιολογικού εμπλουτισμού των δεδομένων οδήγησε στην εμφάνιση του Σημασιολογικού Ιστού (Semantic Web) [13]. Η ιδέα αυτή εισήχθη το 2001 από το δημιουργό του Παγκόσμιου Ιστού (World Wide Web), Tim Berners-Lee. Το όραμα του Berners-Lee προέβλεπε μία πιο ευέλικτη, ενσωματωμένη και αυτοπροσαρμοζόμενη έκδοση του Ιστού. Έκτοτε, ο οργανισμός W3C έχει αναπτύξει ένα σύνολο προτύπων και εργαλείων με σκοπό να υποστηρίξει το όραμα αυτό. Τα δημοσιευμένα πρότυπα περιλαμβάνουν το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework – RDF) [14], το Σχήμα Πλαισίου Περιγραφής Πόρων (Resource Description Framework Schema – RDFS) [15], τη Γλώσσα Οντολογιών Ιστού (Web Ontology Language - OWL) [16] και το Πρωτόκολλο SPARQL (SPARQL Protocol and Query Language for RDF) [17].

2.1.2.1 Σημασιολογικές δομές δεδομένων

Καθώς το RDF και το RDFS κέρδιζαν έδαφος ως γλώσσες μοντελοποίησης δεδομένων, εμφανίστηκε η ανάγκη για μία δομή δεδομένων σχεδιασμένη ειδικά για να διατηρεί σημασιολογικά δεδομένα και να παρέχει αποδοτικές μεθόδους εφαρμογής ερωτημάτων σε αυτά. Την ανάγκη αυτή κάλυψαν τα Αποθετήρια Τριπλετών (Triple Stores), μία δομή δεδομένων, η οποία είναι εμπνευσμένη από τη δομή του RDF. Τα αποθετήρια τριπλετών αποθηκεύουν γεγονότα εκφρασμένα ως τριπλέτες ή δηλώσεις (statements), οι οποίες έχουν τη μορφή «υποκείμενο -- κατηγορήμα -- αντικείμενο». Όπως και στο RDF, κάθε δομικό στοιχείο μίας τριπλέτας αναγνωρίζεται μοναδικά από ένα IRI, γεγονός το οποίο διευκολύνει την ενσωμάτωση καταναμημένων δεδομένων αποφεύγοντας τον κίνδυνο συγκρούσεων. Τα αποθετήρια τριπλετών χαρακτηρίζονται από τη δυνατότητά τους να φιλοξενούν αδόμητα δεδομένα αλλά και ημιδομημένα και δομημένα δεδομένα, ταυτόχρονα. Τα πιο αποδοτικά από αυτά μπορούν να αποθηκεύουν δισεκατομμύρια γεγονότων απαιτώντας έναν μόνο εξυπηρετητή ενώ το κόστος τους είναι συνήθως μικρότερο σε σύγκριση με μία σχεσιακή βάση δεδομένων της αντίστοιχης τάξης μεγέθους. Το χαμηλότερο κόστος, η εξοικονόμηση χρόνου και τα υψηλότερα επίπεδα ελαστικότητας και ακρίβειας αποτελούν ορισμένους από τους λόγους για τους οποίους τα αποθετήρια τριπλετών έχουν διεισδύσει στα συστήματα πολλών επιχειρήσεων.

Πρόσβαση στα οντολογικά δεδομένων των αποθετηρίων τριπλετών δίνεται μέσω της SPARQL [17], της γλώσσας ερωτημάτων για οντολογικά δεδομένα. Η γλώσσα SPARQL φέρει αρκετές ομοιότητες με την Turtle και την SQL (Structured Query Language), τη γλώσσα ερωτημάτων για σχεσιακά δεδομένα. Προσφέρει λειτουργίες ερωτημάτων SELECT, επιστρέφοντας ένα σύνολο δηλώσεων οι οποίες ταιριάζουν με το μοτίβο που παρατέθηκε ως μέρος του ερωτήματος. CONSTRUCT, η οποία επιστρέφει έναν RDF γράφο που σχηματίζεται από τις τριπλέτες που ικανοποιούν ένα μοτίβο γράφου που συνοδεύει το ερώτημα. ASK, ένα ερώτημα που απαντά θετικά (true) ή αρνητικά (false) ανάλογα με το αν το μοτίβο του ερωτήματος έχει λύση, χωρίς να επιστρέφει τη λύση αυτή καθ' αυτή. DESCRIBE, μία μορφή ερωτήματος που επιστρέφει όλες τις τριπλέτες που αναφέρονται σε ένα συγκεκριμένο IRI, είτε αυτό δρα ως υποκείμενο είτε ως αντικείμενο, δεδομένων ορισμένων περιορισμών που εκφράζονται μέσω του μοτίβου του ερωτήματος.

Όπως είναι προφανές, η γλώσσα SPARQL προσφέρεται μόνο για εφαρμογή ερωτημάτων με σκοπό την ανάγνωση δεδομένων και όχι τη μεταβολή τους. Στη δεύτερη περίπτωση, χρησιμοποιείται η γλώσσα SPARQL Update [18] [19], μία γλώσσα για την

ανανέωση RDF γράφων. Η SPARQL Update προδιαγράφει τις λειτουργίες INSERT, DELETE, DELETE/INSERT, LOAD και CLEAR σε ό,τι αφορά την ανανέωση ενός γράφου ή ενός αποθετηρίου γράφων. Το ερώτημα INSERT προσθέτει τριπλέτες σε ένα δεδομένο γράφο, το ερώτημα DELETE τις αφαιρεί από το γράφο αυτό, εάν υπάρχουν, ενώ το ερώτημα DELETE/INSERT χρησιμοποιείται συνήθως για λόγους ανανέωσης ενός γράφου. Η πράξη LOAD διαβάζει ένα RDF κείμενο από μία δεδομένη διεύθυνση (IRI) και, στη συνέχεια, εισάγει το γράφο του RDF κειμένου σε ένα αποθετήριο. Τέλος, εφαρμόζοντας το ερώτημα CLEAR σε ένα γράφο, όλες οι τριπλέτες του γράφου αυτού διαγράφονται.

2.1.2.2 Γλώσσα οντολογιών ιστού

Η OWL αποτελεί μία γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό. Λόγω των περιορισμών στη λειτουργικότητα του RDF-S, ο οργανισμός W3C εισήγαγε την πρώτη έκδοση του προτύπου της OWL το 2004 [16], ενώ το 2012 προτυποποιήθηκε η δεύτερη έκδοση της OWL (OWL 2) [20], η οποία είναι και η τρέχουσα. Η πρώτη έκδοση της OWL διατίθεται σε τρεις εκδόσεις, τις OWL Lite, OWL DL (Description Logic) και OWL Full, οι οποίες προσφέρουν διαφορετικές εκφραστικές δυνατότητες [21].

Η κύρια ιδέα της γλώσσας OWL δεν περιορίζεται απλά στην αναπαράσταση εννοιών και των συνδέσεών τους μέσω οντολογιών· στοχεύει στην ανάπτυξη οντολογιών, οι οποίες μπορούν να αξιοποιηθούν στα πλαίσια διαδικασιών λήψης αποφάσεων. Ταυτόχρονα, επωφελείται των δυνατοτήτων του RDF ώστε να επιτρέπει στις οντολογίες να κατανέμονται σε έναν αριθμό από συστήματα. Αυτό επιτυγχάνεται λόγω της ικανότητας οντολογιών που εκφράζονται μέσω της OWL να περιέχουν αναφορές σε όρους άλλων οντολογιών OWL [21].

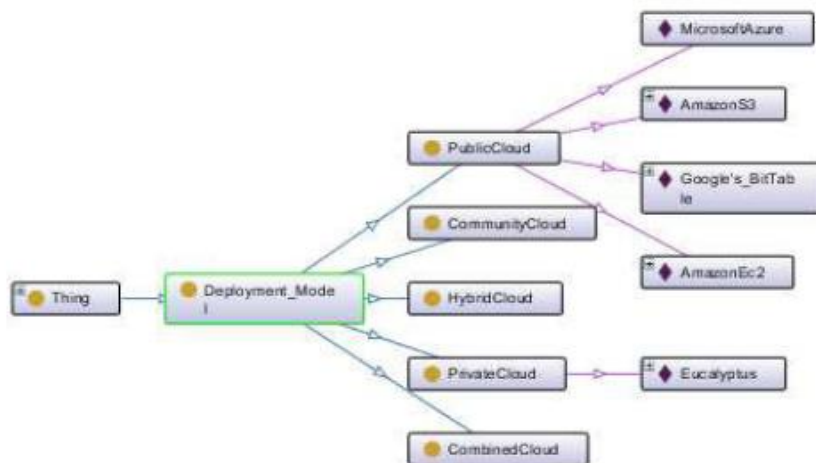
Η Γλώσσα Οντολογιών Ιστού προσφέρει πιο πλούσιες τεχνικές μοντελοποίησης εννοιών και των συσχετίσεών τους σε σχέση με το RDF-S. Για την ακρίβεια, η OWL χρησιμοποιείται για την ανάπτυξη οντολογιών, δηλαδή για τον ορισμό εννοιών και των συσχετίσεων που υπάρχουν μεταξύ τους. Βασίζεται στο RDF και το RDF-S, ενώ έχει εμπλουτιστεί ώστε να δίνει τη δυνατότητα δημιουργίας ακόμη πιο εκφραστικών λεξιλογίων. Χρησιμοποιώντας τον κατάλληλο όρο, είναι δυνατή η δημιουργία κλάσεων, ιδιοτήτων και στιγμιότυπων κλάσεων. Οι προαναφερθείσες ιδιότητες χωρίζονται σε ιδιότητες αντικειμένων (object property), οι οποίες συνδέουν στιγμιότυπα κλάσεων, και σε ιδιότητες δεδομένων (data property), οι οποίες αναθέτουν κάποια τιμή σε ένα στιγμιότυπο [22]. Ενώ το RDF-S χρησιμοποιείται για τον ορισμό κλάσεων, υποκλάσεων, υποιδιοτήτων, πεδίου ορισμού και εύρους τιμών [15], το εύρος των δυνατοτήτων της

γλώσσας OWL επεκτείνεται περαιτέρω, προσφέροντας λειτουργίες έκφρασης τομής και ένωσης κλάσεων και περιορισμών τιμών ή πληθικότητας.

Εκτός από την πλήρη έκδοση της OWL 2, ο οργανισμός W3C έχει δημοσιεύσει τρεις πιο περιορισμένες εκδόσεις της γλώσσας [23], κάθε μία από τις οποίες θυσιάζει κάποιες λειτουργίες του αρχικού συνόλου λειτουργιών ώστε να προσφέρει αυξημένα επίπεδα αποδοτικότητας σε κάποιες άλλες λειτουργίες. Συγκεκριμένα, η OWL 2 EL προσφέρεται για εφαρμογές που χρησιμοποιούν οντολογίες με μεγάλο αριθμό κλάσεων ή/και ιδιοτήτων. Ταυτόχρονα, με την OWL 2 EL η εφαρμογή κανόνων επαγωγικού συλλογισμού τόσο στα δεδομένα όσο και στο σχήμα της οντολογίας πραγματοποιείται σε πολυωνυμικό χρόνο.

Η γλώσσες οντολογιών ιστού και γενικά οι τεχνολογίες σημασιολογικού ιστού χρησιμοποιούνται ευρέως και για την αναπαράσταση δεδομένων και οντοτήτων σε υπηρεσίες υπολογιστικών νεφών. Η οντολογίες μαζί με τη δυνατότητα που προσφέρουν για λήψη αποφάσεων, μπορούν να έχουν ένα σημαντικό αντίκτυπο σε πολλά χαρακτηριστικά των μοντέλων «νεφελώδους» υπολογισμού. Οι οντολογίες μπορούν να ενεργοποιήσουν ευφυείς μηχανισμούς λήψης αποφάσεων για διάφορες υπηρεσίες προσαρμοσμένες σε μοντέλα υπολογιστικών νεφών. Επιπρόσθετα, δίνουν τη δυνατότητα για τη δημιουργία ενός κοινού πεδίου επικοινωνίας και αναπαράστασης δεδομένων, ώστε να επιτρέψουν την αποτελεσματική διαλειτουργικότητα μεταξύ ετερογενών παρόχων και συστημάτων υπολογιστικών νεφών. Υπηρεσίες που υποστηρίζουν σημασιολογικές και οντολογικές λειτουργίες προσφέρουν τα μέσα για την ευφυή επιλογή υπηρεσιών, με αυτοματοποιημένες μεθόδους, περιλαμβάνοντας ενιαία ευρετήρια, και υπηρεσίες διαμεσολάβησης.

Ένα σύστημα αναπαράστασης δεδομένων και οντοτήτων για υπηρεσίες «νέφους» είναι η οντολογία που δημιουργήθηκε στα πλαίσια του ερευνητικού προγράμματος mOSAIC [24], η οποία θα υποστηρίζει πολλαπλά πρότυπα (π.χ. OCCI [25], DMTF [26], NIST [27] κ.α.), πλατφόρμες και υπηρεσίες [28]. Μία από τις ανάγκες που εντοπίζεται ακόμα και σήμερα είναι η απουσία προτυποποίησης, ώστε να επιτρέπει στους χρήστες να χρησιμοποιούν υπηρεσίες διαμεσολάβησης που είναι σε θέση να επιλέξουν μέσα από μία πληθώρα ετερογενών υπηρεσιών «νέφους». Για αυτό το σκοπό, στα πλαίσια του ερευνητικού έργου προτείνεται μία σημασιολογική μηχανή που με τη βοήθεια μίας οντολογίας «νέφους» θα χρησιμοποιείται από το διαμεσολαβητή χρήστη για την εύρεση και επιλογή υπηρεσιών που να ικανοποιούν τις απαιτήσεις του χρήστη τόσο στη προσφερόμενη υπηρεσία, όσο και στα προσφερόμενα συμβόλαια χρήσης (SLA).



Εικόνα 1 Οντολογία της πλατφόρμας mOSAIC αναφορικά με το απαιτούμενο μοντέλο ανάπτυξης [28]

2.1.3 Τεχνολογίες εικονικοποίησης

Με τον όρο εικονικοποίηση, προσδιορίζεται το λογισμικό το οποίο ως κύριο σκοπό έχει τη δημιουργία ενός περιβάλλοντος στο οποίο περιλαμβάνεται ένα σύστημα ή μία συλλογή βιβλιοθηκών λογισμικού με σκοπό την φιλοξενία εφαρμογών. Η χρησιμότητα της εικονικοποίησης είναι η ανεξαρτητοποίηση της διαδικασίας της ανάπτυξης των εφαρμογών από το κυρίως λειτουργικό σύστημα και τους περιορισμούς που τυχόν αυτό επιβάλλει. Ο χρήστης μπορεί να παραμετροποιήσει το εικονικό περιβάλλον αποκλειστικά με βάση τις ανάγκες του, παρακάμπτοντας τεχνικές δυσκολίες, όπως π.χ. η διερεύνηση ζητημάτων συμβατότητας σε ένα σύστημα στο οποίο ενδεχομένως να μην έχει τη δυνατότητα και τα δικαιώματα να διαμορφώσει κατά τις ανάγκες του (συστήματα που φιλοξενούν πολλούς χρήστες). Το εικονικό περιβάλλον προσφέρει επίσης τη δυνατότητα για εύκολη μεταφορά (portability) και αντιγραφή του, ώστε να είναι διαμοιράσιμο σε άλλους χρήστες.

Στη παρούσα υποενότητα παρουσιάζονται οι δύο πιο διαδεδομένες τεχνολογίες εικονικοποίησης (virtualization). Η διερεύνηση τους στηρίχθηκε στο γεγονός ότι χρησιμοποιούνται κατά κόρον σήμερα, είτε ως άμεσα προσφερόμενες υπηρεσίες προς τους χρήστες (IaaS, CaaS), είτε ως υποστηρικτικές υποδομές για υπηρεσίες (PaaS και SaaS) προς τους χρήστες. Η πρώτη τεχνολογία εικονικοποίησης ονομάζεται και ως εικονικοποίηση υπερεπόπτη (hypervisor virtualization), και αποτελεί την παραδοσιακή μέθοδο δημιουργίας εικονικών συστημάτων τα οποία ονομάζονται εικονικά μηχανήματα (Virtual Machines – VMs). Τα εικονικά μηχανήματα αποτελούν το κύριο προσφερόμενο «προϊόν» σε υπηρεσίες IaaS, ενώ μέχρι πρόσφατα αποτελούσε την εξ' ορισμού μορφή δημιουργία υποδομής για την υποστήριξη υπηρεσιών PaaS και SaaS.

Η δεύτερη μορφή εικονικοποίησης περιλαμβάνει τη χρήση περιεκτών λογισμικού. Οι περιέκτες λογισμικού αποτελούν ευέλικτα πακέτα λογισμικού που περικλείουν μία εφαρμογή. Προσφάτως, έχουν ανέλθει ως μία πολύ αποδοτική λύση για την υποστήριξη υπηρεσιών PaaS και SaaS, ενώ παρέχονται και ως υπηρεσία απευθείας προς τους χρήστες (Container as a Service – CaaS).

2.1.3.1 Εικονικοποίηση υπερεπόπτη

Αποτελεί την παραδοσιακή προσέγγιση εικονικοποίησης συστημάτων, και χρησιμοποιείται κατά κόρον από όλους τους σύγχρονους παρόχους υπηρεσιών «νέφους». Για την δημιουργία του εικονικού συστήματος (VM) ο υπερεπόπτης δεσμεύει τους απαραίτητους πόρους από μία «δεξαμενή» του εξυπηρετητή (resource pool) μέσω του λειτουργικού του συστήματος. Η δεξαμενή αυτή πόρων περιλαμβάνει τον αριθμό των πυρήνων του επεξεργαστή, το διαθέσιμο μέγεθος μνήμης RAM, το διαθέσιμο μέγεθος σε αποθηκευτικό χώρο, αλλά και τους διαθέσιμους δικτυακούς πόρους (bandwidth – Mb/s). Στη συνέχεια δημιουργεί το κατάλληλο επίπεδο εικονικού υλισμικού και αναλαμβάνει την επικοινωνία μεταξύ εικονικού και φυσικού επιπέδου. Ο χρήστης του εικονικού αυτού μηχανήματος έχει την δυνατότητα να το χρησιμοποιήσει ως ένα κανονικό υπολογιστικό σύστημα, να εφαρμόσει το λειτουργικό σύστημα της επιλογής του. Επιπρόσθετα, έχει τη δυνατότητα επαναφοράς εικονικών αρχείων δίσκου (images), αλλά και να δημιουργήσει νέα, αποθηκεύοντας τη κατάσταση ενός τρέχοντος εικονικού μηχανήματος. Χαρακτηριστικά παραδείγματα αποτελούν οι υπερεπόπτες KVM [29], Xen [30] και Hyper-V [31].

Τα εικονικά συστήματα τύπου υπερεπόπτη διέπονται από συγκεκριμένα χαρακτηριστικά, τα οποία -ανάλογα με τη φύση της λειτουργίας- τους μπορούν να χαρακτηριστούν είτε ως θετικά είτε ως αρνητικά.

Το κύριο χαρακτηριστικό των εικονικών συστημάτων τύπου υπερεπόπτη είναι ότι απομονώνονται πλήρως από το υπερκείμενο λειτουργικό σύστημα. Το μόνο που γνωρίζει το λειτουργικό σύστημα του εξυπηρετητή που φιλοξενεί το εικονικό σύστημα είναι το αναγνωριστικό διεργασίας (PID) του υπερεπόπτη που έχει ανατεθεί για την λειτουργία του εικονικού συστήματος. Οι εσωτερικές διεργασίες του εικονικού μηχανήματος είναι μη προσβάσιμες και μη ορατές, χαρακτηριστικό το οποίο βελτιώνει την ασφάλεια της λειτουργίας του εικονικού μηχανήματος. Επίσης, λόγω του εικονικού στρώματος υλισμικού που δημιουργείται από τον υπερεπόπτη, οι βιβλιοθήκες αλλά και οι εφαρμογές που εγκαθίστανται στο εικονικό σύστημα μπορούν να είναι πλήρως ασύμβατες με το υπερκείμενο λειτουργικό σύστημα. Για παράδειγμα, ένα εικονικό

σύστημα μπορεί να έχει εγκατεστημένο λειτουργικό σύστημα Linux, όταν το υπερκείμενο λειτουργικό σύστημα που το φιλοξενεί είναι Windows.

Η δυνατότητα αυτή απομόνωσης, όμως, καθιστά τα εικονικά συστήματα υπερεπόπτη λιγότερο αποδοτικά όσον αφορά τη κατανάλωση πόρων του υπερκείμενου συστήματος. Η ανάγκη για την ύπαρξη ξεχωριστού λειτουργικού συστήματος, οδηγεί σε επιπρόσθετη κατανάλωση επεξεργαστικής ισχύος, μνήμης και αποθηκευτικού χώρου, ενώ το κόστος αυτό αυξάνεται λόγω των στρωμάτων επικοινωνίας που επιβάλλει ο υπερεπόπτης μεταξύ φυσικού και εικονικού υλισμικού [32].

2.1.3.2 Περιέκτες λογισμικού

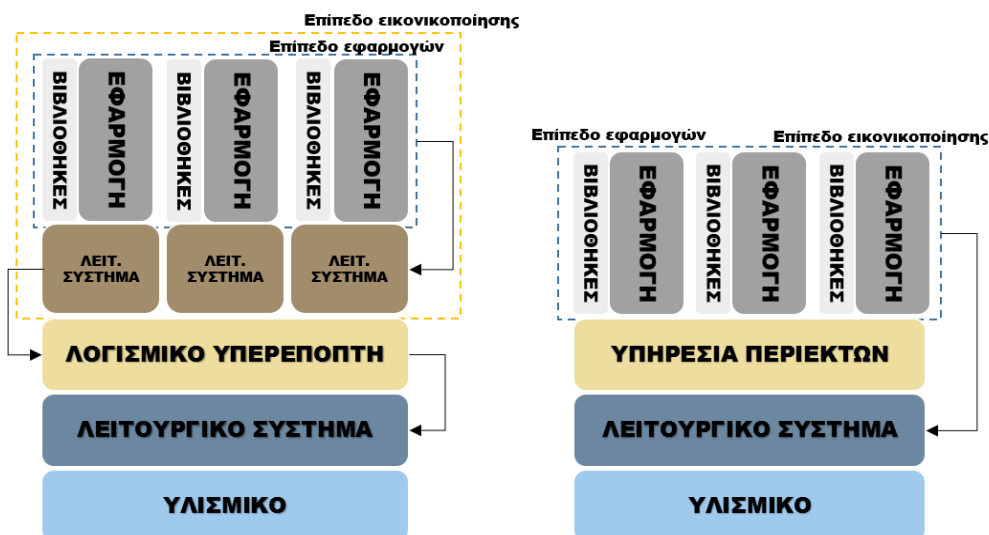
Η εικονικοποίηση μέσω περιεκτών λογισμικού (Software Containers) αποτελεί ένα πιο εύηλο και περισσότερο αποδοτικό μοντέλο δημιουργίας εικονικών περιβαλλόντων για τη λειτουργία εφαρμογών. Το κύριο χαρακτηριστικό τους είναι ότι λειτουργούν μοιραζόμενοι το πυρήνα του υπερκείμενου λειτουργικού συστήματος για να παρέχουν ένα περιβάλλον το οποίο διαθέτει ανεξάρτητες βιβλιοθήκες και μπορεί να υποστηρίξει συγκεκριμένες εφαρμογές. Από τη μεριά του χρήστη το εικονικό αυτό περιβάλλον μπορεί να φαίνεται πλήρως απομονωμένο και ανεξάρτητο, αλλά όλες οι διεργασίες που εκτελούνται εντός τους είναι προσβάσιμες και ορατές από το υπερκείμενο λειτουργικό σύστημα.

Η έλλειψη ανάγκης για τη δημιουργία εικονικού υλισμικού (hardware emulation), σημαίνει ότι οι περιέκτες δεν επιβαρύνουν καθόλου το λειτουργικό σύστημα στο οποίο φιλοξενούνται. Λόγω του χαρακτηριστικού αυτού, η έναρξη και η αρχικοποίηση της λειτουργίας τους είναι άμεση [33]. Επίσης, η κατανάλωση πόρων περιορίζεται μόνο σε όσους χρειάζονται οι περικλειόμενες εφαρμογές κάθε περιέκτη.

Από την άλλη, η έλλειψη επιπέδου απομόνωσης καθιστά τους περιέκτες λογισμικού περισσότερο ευάλωτες σε θέματα ασφαλείας καθώς λάθος χειρισμός από διαχειριστές μπορεί να οδηγήσει σε σφάλματα ή και απότομο τερματισμό των λειτουργιών τους [34]. Επιπλέον ζητήματα ασφαλείας εγείρονται και από το γεγονός ότι τα δεδομένα χρηστών και εφαρμογών δεν αποθηκεύονται σε ξεχωριστό χώρο του δίσκου που έχει δεσμευθεί, αλλά σε ένα κοινό χώρο (layered filesystem), καθιστώντας τα έτσι περισσότερο ευάλωτα. Τέλος, το γεγονός ότι οι περιέκτες λογισμικού μοιράζονται το πυρήνα του υπερκείμενου λειτουργικού συστήματος, τις περιορίζει όσον αφορά την υποστήριξη συγκεκριμένων εφαρμογών και βιβλιοθηκών.

Σύγχρονα συστήματα θυρίδων λογισμικού όπως τα Docker [35] και LXD [36], χρησιμοποιούνται συχνά σήμερα για την ανάπτυξη ευέλικτων και επεκτάσιμων

εφαρμογών και υπηρεσιών που θα μπορούν να αναπτύσσονται από ομάδες πολλαπλών χρηστών. Επίσης, σύγχρονες τεχνολογίες επεκτασιμότητας και ενορχήστρωσης (π.χ. Kubernetes [37]), καθιστούν τους περιέκτες λογισμικού ως μία πολύ ελκυστική λύση για τις σύγχρονες ανάγκες εφαρμογών και υπηρεσιών που λειτουργούν στα πλαίσια υποδομών υπολογιστικών «νεφών».



Εικόνα 2 Σύγκριση αρχιτεκτονικής περιεκτών λογισμικού και εικονικών μηχανημάτων

2.2 Σύγχρονα θέματα, αρχιτεκτονικές & τεχνολογίες υποδομών υπολογιστικών νεφών

Τα υπολογιστικά «νέφη» ως μοντέλο είναι, στις μέρες μας, το πιο ευρέως διαδεδομένο μοντέλο για την παροχή ευέλικτων, ισχυρών και κλιμακώσιμων υπηρεσιών υπολογισμού. Το κύριο χαρακτηριστικό των υπηρεσιών νέφους είναι ότι απαλλάσσουν το χρήστη από τη εγκατάσταση και συντήρηση ακριβού και περίπλοκου εξοπλισμού υλισμικού και λογισμικού. Ο χαρακτηρισμός «νέφος» και «νεφελώδης» προκύπτει από τη φύση λειτουργίας των υπηρεσιών, χαρακτηρίζοντας έτσι τον τρόπο με τον οποίο ο τελικός χρήστης έχει πρόσβαση στις υπηρεσίες αυτές. Η «νεφελώδης» μορφή των υπηρεσιών αυτών αποσυνδέει το σημείο πρόσβασης από τη πραγματική τοποθεσία μίας εφαρμογής/υπηρεσία.

Τα υπολογιστικά νέφη έχουν επιτύχει να προσφέρουν υπό τη μορφή υπηρεσιών όλα τις μορφές υπολογισμού που συναντώνται σήμερα στους τομείς των πληροφοριακών συστημάτων. Ένας πάροχος μπορεί να προσφέρει πόρους υλισμικού, υπολογιστικές πλατφόρμες και εφαρμογές σε μορφή υπηρεσίας, τις οποίες ο χρήστης μπορεί να χρησιμοποιήσει απομακρυσμένα από το περιβάλλον εργασίας του. Οι μορφές υπηρεσίας κατηγοριοποιούνται ανάλογα με το προσφερόμενο υπολογιστικό πόρο/υπηρεσία.

2.2.1 Κύρια χαρακτηριστικά

Πέρα από την παροχή των μορφών υπηρεσίας που αναφέρονται στην υποενότητα 2.2.2, οι υπηρεσίες νέφους χαρακτηρίζονται και από διαδικασίες που εγγυόνται τη συνεχή διαθεσιμότητα, βέλτιστη ποιότητα υπηρεσίας και ασφάλειας των εφαρμογών, δεδομένων και υπηρεσιών στα οποία έχει πρόσβαση ο τελικός χρήστης. Τα κύρια χαρακτηριστικά που εντοπίζονται είναι τα εξής [38]:

- **Υπηρεσία κατά παραγγελία.** Ο χρήστης αποκτά πρόσβαση σε πόρους και εφαρμογές, με βάση τις ανάγκες του και όταν αυτός το επιθυμεί, αυτόματα χωρίς την επέμβαση ανθρώπινου παράγοντα
- **Ευρεία δικτυακή προσβασιμότητα.** Οι υπηρεσίες είναι διαθέσιμες διαμέσου ενός προσωπικού δικτύου (ή του διαδικτύου) και είναι προσβάσιμες μέσω πρότυπων μηχανισμών και πρωτοκόλλων, τα οποία υποστηρίζονται από πληθώρα συσκευών
- **Συγκέντρωση πόρων.** Οι διαθέσιμοι πόροι του παρόχου συγκεντρώνονται σε μία κοινή «δεξαμενή» ώστε να εξυπηρετούν πολλαπλούς χρήστες, δημιουργώντας ένα μοντέλο «πολλαπλής ενοικίασης» (multi-tenancy), προσδιορίζοντας έτσι την δυνατότητα της υποκείμενης υποδομής να διαθέτει πόρους δυναμικά στους χρήστες ανάλογα με τις ανάγκες τους
- **Ελαστικότητα.** Ορίζεται ως η δυνατότητα των υπηρεσιών και πόρων να κλιμακώνουν αυτόματα ανάλογα με τις ανάγκες του χρήστη, αλλά και της υπηρεσίας/εφαρμογής. Η κλιμάκωση είναι αυξομειούμενη, δηλαδή υπάρχει η δυνατότητα για προσθήκη πόρων όταν π.χ. παρατηρείται αυξημένη ζήτηση για μία υπηρεσία ή αφαίρεση όταν είναι αναγκαίο να γίνει συντήρηση και εξοικονόμηση πόρων

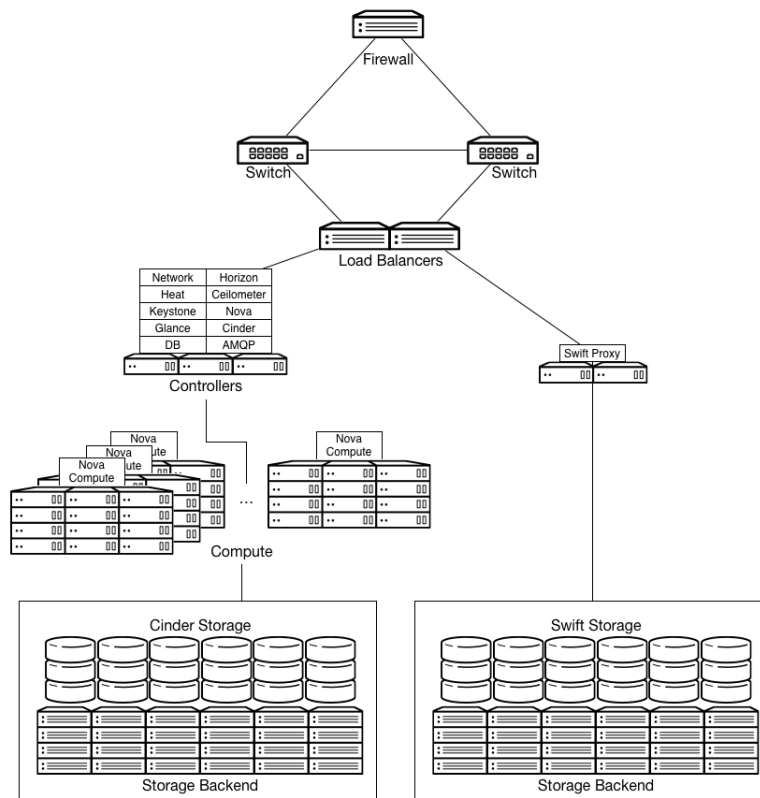
2.2.2 Μοντέλα υπηρεσιών

Κατά το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των Η.Π.Α (National Institute of Standards and Technology – NIST [27]), εντοπίζονται τρεις βασικές μορφές υπηρεσιών νέφους [38].

2.2.2.1 Υποδομή υπολογισμού ως υπηρεσία (*Infrastructure as a Service – IaaS*)

Ορίζεται ως η δυνατότητα που προσφέρεται στον τελικό χρήστη να χρησιμοποιεί πόρους υλισμικού (επεξεργαστή, μνήμη, σκληρό δίσκο) ώστε να εγκαταστήσει και να εκτελέσει λογισμικό (εφαρμογές, υπηρεσίες, λειτουργικά συστήματα). Ο χρήστης δεν έχει έλεγχο του υποκείμενου συστήματος ή υποδομής, αλλά ελέγχει το λειτουργικό

σύστημα, αποθηκευτικό χώρο και εφαρμογές που οριοθετούνται από ένα περιβάλλον. Οι πρόσβαση στους πόρους μπορεί να γίνει άμεσα, δηλαδή να παρέχεται στον χρήστη απευθείας σύνδεση σε έναν εξυπηρετητή (Metal as a Service – MaaS), είτε έμμεσα, μέσω ενός εικονικού περιβάλλοντος που συναντάται επίσης και ως εικονικό μηχάνημα (Virtual Machine). Οι χρήστες μπορούν να χρησιμοποιούν τα περιβάλλοντα αυτά (εικονικά ή μη) συνδεδεμένοι προς αυτά απομακρυσμένα (SSH, RDP, VNC, HTTP). Χαρακτηριστικό παράδειγμα υπηρεσίας IaaS αποτελεί η υπηρεσία EC2 (Elastic Computing 2) της Amazon [39]. Επίσης, παράδειγμα αρχιτεκτονικής IaaS παρουσιάζεται και στην Εικόνα 3.

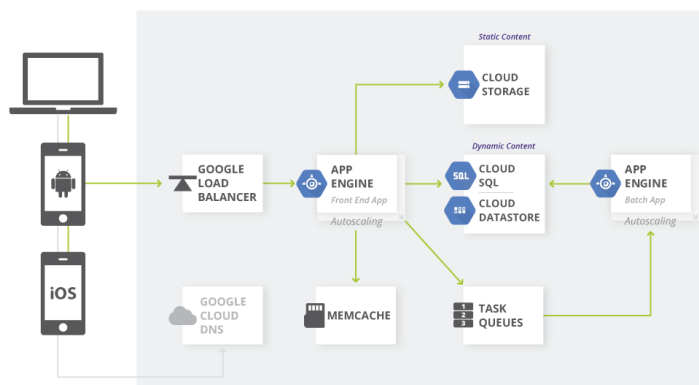


Εικόνα 3 Μοντέλο IaaS όπως αυτό παρέχεται από τη σουίτα λογισμικού αρχιτεκτονικής Openstack [9]

2.2.2.2 Πλατφόρμα υπολογισμού ως υπηρεσία (Platform as a Service – PaaS)

Ορίζεται ως δυνατότητα που παρέχεται προς τον χρήστη να αναπτύσσει εφαρμογές και υπηρεσίες, χρησιμοποιώντας γλώσσες προγραμματισμού, βιβλιοθήκες, εργαλεία και προγραμματιστικές διεπαφές (APIs) που προσφέρονται και υποστηρίζονται από τον πάροχο της υπηρεσίας. Η εγκατάσταση, η λειτουργία και η διαθεσιμότητα της υπηρεσίας/εφαρμογής του χρήστη αναλαμβάνεται από τον πάροχο. Ο χρήστης δεν έχει έλεγχο του υποκείμενου υποδομής που φιλοξενεί την υπηρεσία (λειτουργικό σύστημα, δίκτυο κλπ.), διατηρεί όμως τη δυνατότητα να επεμβαίνει στη

εφαρμογή του και να προσαρμόζει το περιβάλλον φιλοξενίας (hosting environment) κατά τις προτιμήσεις του. Χαρακτηριστικό παράδειγμα υπηρεσίας PaaS αποτελεί η πλατφόρμα ανάπτυξης εφαρμογών AppEngine της Google (Εικόνα 4) [40].



Εικόνα 4 Αρχιτεκτονική απεικόνιση της πλατφόρμας AppEngine της Google [40]

2.2.2.3 Υπολογιστική εφαρμογή ως υπηρεσία (Software as a Service – SaaS)

Ορίζεται ως η δυνατότητα που παρέχεται προς τον χρήστη να χρησιμοποιήσει εφαρμογές που αναπτύσσονται, εγκαθίστανται και προσφέρονται απ' ευθείας από τον πάροχο της υπηρεσίας. Οι εφαρμογές είναι προσβάσιμες από πολλαπλές συσκευές είτε μέσω ενός φυλλομετρητή διαδικτύου (web browser) είτε μίας εφαρμογής πελάτη που την εγκαθιστά ο χρήστης στην συσκευή του. Ο χρήστης δεν έχει έλεγχο του υποκείμενου υποδομής που φιλοξενεί την υπηρεσία (λειτουργικό σύστημα, δίκτυο κλπ.), και περιορίζεται στις επιλογές και ρυθμίσεις που προσφέρονται και σχετίζονται αποκλειστικά με ζητήματα χρήσης της εφαρμογής. Χαρακτηριστικό παράδειγμα SaaS είναι η σουίτα Office365 της Microsoft [41].

2.2.3 Μοντέλα ανάπτυξης

Τα υπολογιστικά νέφη κατηγοριοποιούνται ανάλογα με τον τρόπο με τον οποίο εγκαθίστανται και προσφέρουν τις υπηρεσίες τους προς τους τελικούς χρήστες. Σύμφωνα με το NIST [27], εντοπίζονται τέσσερις βασικές κατηγορίες ανάπτυξης υπολογιστικών νεφών:

- **Ιδιωτικό νέφος.** Η υποδομή που υποστηρίζει το επίπεδο των υπηρεσιών αναπτύσσεται και λειτουργεί εντός των ορίων ενός οργανισμού. Οι υπηρεσίες νέφους περιορίζονται προς τα μέλη του οργανισμού και δεν είναι προσβάσιμες εκτός του δικτύου που ανήκει στον οργανισμό. Δεν

υπάρχει φυσικά η δυνατότητα πρόσβασης των υπηρεσιών από το δημόσιο διαδίκτυο

- **Κοινοτικό νέφος.** Αποτελεί μία μορφή υπολογιστικών νεφών που προσφέρουν τις υπηρεσίες τους σε μία κλειστή ομάδα χρηστών που ανήκουν σε δύο ή περισσότερους οργανισμούς. Η υποκείμενη υποδομή ανήκει και συντηρείται από ένα ή περισσότερους οργανισμούς που ανήκουν στη κοινότητα
- **Δημόσιο νέφος.** Αποτελεί το τύπο νέφους που η υπηρεσίες του είναι διαθέσιμες και προσβάσιμες από το ευρύ κοινό. Η υποδομή μπορεί να ανήκει σε οποιαδήποτε ομάδα οργανισμών. Η πρόσβαση στις υπηρεσίες γίνεται συνήθως μέσω του δημόσιου διαδικτύου
- **Υβριδικό νέφος.** Αποτελεί έναν συνδυασμό των ανωτέρω, υποστηρίζονται είτε το δημόσιο είτε το ιδιωτικό μοντέλο ανάλογα με την ομάδα χρηστών και των υπηρεσιών

2.2.4 Συμβόλαια χρήσης υπηρεσιών νέφους

Η μετάβαση στο μοντέλο του νεφελώδους υπολογισμού προϋποθέτει ένα σαφές και καλά ορισμένο πλαίσιο που να καλύπτει όλα τα πρακτικά ζητήματα που προκύπτουν μεταξύ καταναλωτή/πελάτη και παρόχου. Σύμφωνα με το Συμβούλιο Καταναλωτών Προτύπων Νέφους (Cloud Standards Consumer Council) [42], τα συμβόλαια χρήσης είναι αρχεία ή συμφωνίες που ορίζουν ακριβώς τη σχέση μεταξύ καταναλωτή – παρόχου [43]. Καθώς το πεδίο των υπολογιστικών νεφών είναι ακόμα σε εξέλιξη, παρατηρείται ασυμβατότητα σε πολλές περιπτώσεις μεταξύ των απαιτήσεων του χρήστη και των πραγματικών συνθηκών παροχής υπηρεσίας από την πλευρά του παρόχου. Για παράδειγμα, σε ένα συμβόλαιο χρήσης μπορεί να μην ορίζεται η γεωγραφική διασπορά των δεδομένων του χρήστη, κάτι το οποίο να αποτελέσει ανασταλτικό για τον χρήστη λόγω των περιορισμών που επιβάλλονται από τα νομικά πλαίσια κάθε χώρας (π.χ. Η.Π.Α ή Ε.Ε) σχετικά με τη εξαγωγή συγκεκριμένου τύπου δεδομένων σε τρίτες χώρες [43].

Επίσης, συχνά παρατηρούνται διενέξεις μεταξύ πελατών και παρόχων. Μεγάλοι οργανισμοί ή εταιρίες δεν είναι ιδιαίτερα ευέλικτοι όσον αφορά τα εν λόγω συμβόλαια, ενώ μικρότεροι πάροχοι αναγκάζονται, για λόγους ανταγωνιστικότητας, να προσφέρουν μεγαλύτερη ευελιξία στα παρεχόμενα συμβόλαια, τείνουν όμως προς το να υπόσχονται περισσότερα από όσα δύνανται πραγματικά να προσφέρουν⁴. Συνεπώς, είναι αναγκαία η δημιουργία μίας προτυποποιημένης διαδικασίας η οποία να διασφαλίζει τα ελάχιστα

⁴ <https://www.cloudendure.com/blog/5-things-they-never-told-you-about-cloud-downtime/>

αναγκαία χαρακτηριστικά που θα πρέπει να περιγράφονται σε ένα συμβόλαιο χρήσης, αλλά και να προτείνει μεθόδους για την αξιολόγηση τους. Ως αποτέλεσμα, οι πελάτες θα έχουν στα χέρια τους τα απαραίτητα εργαλεία ώστε να συγκρίνουν και τελικώς να επιλέξουν τον πάροχο ή παρόχους για την κατανάλωση υπηρεσιών νέφους.

Σύμφωνα με την αναφορά του Συμβούλιο Καταναλωτών Προτύπων Νέφους (CSCC), ένα συμβόλαιο υπηρεσιών νέφους αποτελείται από τρία μέρη [43]:

- Σύμβαση πελάτη
- Πολιτική ορθής χρήσης
- Σύμβαση επιπέδου υπηρεσίας

Η *Σύμβαση Πελάτη* (Customer Agreement) επί της ουσίας περιγράφει τις σχέσεις μεταξύ πελάτη και παρόχου στο σύνολο τους κατά της διάρκεια του συμβολαίου, περιγράφοντας τους ρόλους, τις υποχρεώσεις και την εκτέλεση των διαδικασιών που πρέπει να αναλάβει κάθε εμπλεκόμενος.

Η *Πολιτική ορθής χρήσης* (Acceptable Use Policy) περιγράφει τους περιορισμούς που επιβάλλει ο πάροχος σχετικά με τη χρήση των υπηρεσιών. Ένα χαρακτηριστικό παράδειγμα αποτελεί η πολιτική χρήσης υπηρεσιών αποθήκευσης δεδομένων, όπου ο πάροχος συνήθως επιβάλλει περιορισμούς σχετικά με τη αποθήκευση δεδομένων που αποκτήθηκαν παράνομα (παράνομη μεταφόρτωση δεδομένων που προστατεύονται από πνευματικά δικαιώματα) ή περιέχουν υλικό που εμπίπτει εντός του κοινού ποινικού κώδικα.

Η *Σύμβαση επιπέδου υπηρεσίας* (Service Level Agreement) περιγράφει συγκεκριμένα χαρακτηριστικά που διέπουν τις υπηρεσίες που προσφέρει ο πάροχος προς τον πελάτη. Τα SLA δεν πρέπει να συγχέονται με τη ποιότητα υπηρεσίας (QoS), η οποία επί της ουσίας θεωρείται μέρος του SLA. Τα SLA οριοθετούν τις προσδοκίες του πελάτη σχετικά με τις παρεχόμενες υπηρεσίες και θέτουν ορισμένα πρόστιμα παραβίασης των ορίων αυτών. Για παράδειγμα, ένα χαρακτηριστικό ενός SLA είναι η προδιαγραφή της διαθεσιμότητας των υπολογιστικών πόρων μίας IaaS υπηρεσίας νέφους ως ποσοστό επί του χρόνου χρήσης. Μεγάλοι πάροχοι υπόσχονται ένα ποσοστό έως και 99.95% διαθεσιμότητας επί του συνόλου της διάρκειας της υπηρεσίας, ενώ την ίδια στιγμή ορίζουν τα πρόστιμα που θα κληθούν να πληρώσουν προς τον πελάτη, σε περίπτωση που η διαθεσιμότητα πέσει κάτω από 99.5% και 99% [44].

Η εγγύηση της υπηρεσίας ορίζει τους στόχους που σχετίζονται με μία ή περισσότερες υπηρεσίες. Αυτές οι εγγυήσεις απόδοσης δεσμεύουν τον πάροχο ώστε να επιτύχουν μία αποδεκτή επίδοση σχετικά με τις υπηρεσίες που παρέχουν. Σε περίπτωση

αποτυχίας, όπως αναφέρθηκε και παραπάνω, ο πάροχος θα κληθεί να αποδώσει πρόστιμα ανάλογα με το είδος του χαρακτηριστικού και εύρος μη συμμόρφωσης. Τα πρόστιμα αυτά συνήθως μεταφράζονται σε επιστροφή μέρους του κόστους χρήσης υπηρεσίας προς το πελάτη (Credit refund). Τυπικά παραδείγματα εγγυήσεων επίδοσης που σχετίζονται με τα συμβόλαια ή συμβάσεις χρήσης (SLA) αποτελούν τα παρακάτω χαρακτηριστικά [45]:

- **Διαθεσιμότητα**, που ορίζει το ποσοστό του συμφωνηθέντος χρόνου χρήσης, κατά το οποίο μία υπηρεσία νέφους πρέπει να είναι διαθέσιμη προς το πελάτη. Το χαρακτηριστικό της διαθεσιμότητας καλύπτει όλα τα μοντέλα των υπηρεσιών, είτε αναφερόμαστε σε πόρους (εικονικά μηχανήματα, αποθηκευτικό χώρο - IaaS), πλατφόρμες (PaaS) ή εφαρμογές (SaaS). Μπορεί να εκφράζεται ως ποσοστό επί του συνόλου των πόρων, για παράδειγμα ο πάροχος εγγυάται τη διαθεσιμότητα σε ποσοστό 99.95% όλων των εικονικών μηχανημάτων σε μία υπηρεσία IaaS, ή αντίθετα να εκφράζει τη διαθεσιμότητα ως ποσοστό για ένα μόνο πόρο ή είδος πόρων. Αξίζει να σημειωθεί, ότι στις περισσότερες περιπτώσεις ο πάροχος εξαιρεί από τη καταμέτρηση του χρόνου λειτουργίας των υπηρεσιών, τα χρονικά διαστήματα κατά οποία οι υπηρεσίες δεν είναι διαθέσιμες λόγω εργασιών συντήρησης στην υποκείμενη υποδομή
- **Χρόνος απόκρισης**, που συναντάται συνήθως (αλλά όχι αποκλειστικά) στα μοντέλα PaaS και SaaS, και εγγυάται τον ελάχιστο χρόνο απόκρισης της εφαρμογής ή υπηρεσίας. Τυπικά αφορά το χρόνο επεξεργασίας επεξεργασία ενός αιτήματος, και ο τρόπος υπολογισμού του ποικίλλει ανά πάροχο. Συνήθως, από της μεριά του παρόχου, ο χρόνος απόκρισης εκφράζεται ως το χρόνο που απαιτείται για την επεξεργασία ενός αιτήματος από τη στιγμή που εισέρχεται στο δίκτυο του έως τη στιγμή που αυτό θα εξέλθει του δικτύου. Πολλοί πάροχοι εκφράζουν ρητά τις εξαιρέσεις στον υπολογισμό του χρόνου απόκρισης, για καταστάσεις που είναι εκτός της δικαιοδοσίας και της ευθύνης τους, όπως στη προκειμένη περίπτωση τη συμπεριφορά του δικτύου (συμφόρηση, καθυστέρηση) του χρήστη ή οποιωνδήποτε ενδιάμεσων δικτύων. Για τον καταναλωτή αντίθετα, ο χρόνος απόκρισης αφορά το χρόνο που απαιτείται για την επεξεργασία ενός αιτήματος από τη στιγμή που θα το στείλει ο ίδιος, μέχρι να λάβει απάντηση από την υπηρεσία ή εφαρμογή νέφους. Ο τρόπος μέτρησης του χρόνου απόκρισης είναι συνήθως ανασταλτικός

παράγοντας για πολλούς χρήστες. Για την εγγυημένη προσφορά χρόνου απόκρισης, πολλοί πάροχοι προσφέρουν απευθείας σύνδεση (direct physical link) μεταξύ του δικτύου του χρήστη και του δικτύου του υπολογιστικού κέντρου του παρόχου.

- **Ρυθμός διακίνησης (throughput)**, το οποίο, εκφράζει τον αριθμό των αιτημάτων του χρήστη προς μία υπηρεσία, πόρο ή εφαρμογή νέφους σε μία μονάδα χρόνου. Ένα SLA μπορεί να ορίσει ένα μέγιστο όριο στον αριθμό των αιτημάτων που μπορούν επιτυχώς να επεξεργαστούν εντός μίας μονάδας χρόνου, τα οποία θα έχουν επίσης μία εγγύηση χρόνου απόκρισης. Σε περίπτωση που ο καταναλωτής αποστείλει περισσότερα αιτήματα από όσα ορίζει το όριο της σύμβασης, αυτό μπορεί να αποτελέσει παραβίαση της συμβάσης πολιτικής ορθής χρήσης που περιεγράφηκε παραπάνω, και ως αποτέλεσμα να εξαιρεθεί από τον υπολογισμό του ωφέλιμου ρυθμού διακίνησης ή του χρόνου απόκρισης

Ένα συμβόλαιο χρήσης πρέπει να ξεκάθαρα να αναφέρει πόσο συχνά συλλέγονται δεδομένα σχετικά με την επίδοση του παρόχου γύρω από τις ανωτέρω μετρικές. Η επιτροπή σχετικά με τις υπηρεσίες «νέφους» της ομάδας λειτουργικών συστημάτων (Open Systems Group – OSG [46]) του οργανισμού προτύπων αξιολόγησης απόδοσης (Standard Performance Evaluation Corporation – SPEC [47]), εργάζεται προς της δημιουργία ενιαίων συγκριτικών μεθόδων αξιολόγησης (benchmarks) σχετικά με την απόδοση παρόχων IaaS υπηρεσιών.

Αντίστοιχα με τα χαρακτηριστικά απόδοσης, που περιβάλλουν τις υπηρεσίες νέφους, υπάρχουν και τα χαρακτηριστικά που καλύπτουν την διαχείριση των δεδομένων του χρήστη που ενδεχομένως να παραχθούν ή να μεταφερθούν από και προς την υποδομή του παρόχου κατά τη διάρκεια χρήσης των υπηρεσιών νέφους. Η εγγυήσεις σχετικά με τα δεδομένα καλύπτουν ζητήματα όπως [45]:

- **Είδη δεδομένων που αποθηκεύονται**, που καθορίζει τη συμπεριφορά του χρήστη σχετικά με τα δεδομένα που επιλέγει να αποθηκεύσει στο «νέφος». Η επιλογή αυτή συνήθως καθοδηγείται από τη φύση τους, δηλαδή το πόσο ευαίσθητα ή προσωπικά είναι, και από την αρχιτεκτονική του δικτύου και της υποδομής του παρόχου. Οι οδηγίες αποθήκευσης, αλλά και οι πολιτικές χρήσης των υπηρεσιών δεδομένων ενός παρόχου πρέπει να επικοινωνούνται με σαφή τρόπο προς τον χρήστη. Για παράδειγμα, ο πάροχος θα πρέπει να ενημερώνει το χρήστη για την ύπαρξη αλλά και τη συχνότητα λειτουργίας ελεγκτικού

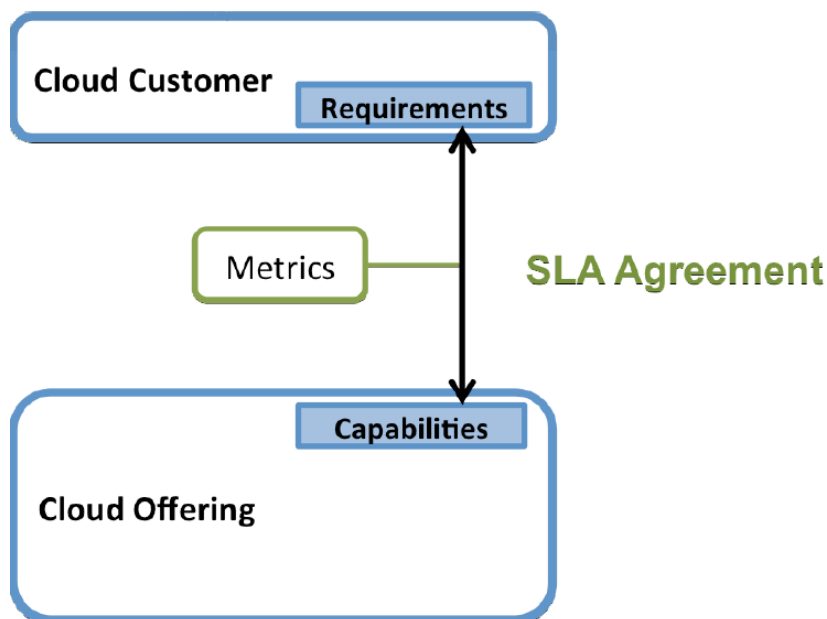
μηχανισμού που θα βεβαιώνει ότι τα δεδομένα δεν παραβιάζουν κανονισμούς και νομοθεσίες στις οποίες υπόκειται ο πάροχος. Αντίστοιχα, και από τη μεριά του ο πελάτης μπορεί να ορίσει ότι επιθυμεί να διενεργεί τακτικό έλεγχο μέσω τρίτου (third-party data auditing) ώστε να διασφαλίσει την ακεραιότητα και την ασφάλεια των δεδομένων του

- ***Το που αποθηκεύονται τα δεδομένα***, όπου καθορίζεται η επιθυμία από τη μεριά του χρήστη, και η δυνατότητα από της μεριά του παρόχου, για την ακριβή τοποθεσία αποθήκευσης των δεδομένων. Μεγάλοι πάροχοι υπηρεσιών νέφους, στη προσπάθειά τους να καλύψουν μία όσο το δυνατόν μεγαλύτερη βάση χρηστών, παρέχουν τις υπηρεσίες του μέσω πολλαπλών υπολογιστικών κέντρων τα οποία είναι γεωγραφικά διασκορπισμένα. Αν δεν ορισθεί ρητά από το χρήστη η εξ' ορισμού μέθοδος αποθήκευσης των δεδομένων μπορεί να περιλαμβάνει τη κατάτμηση τους και «φυσική» αποθήκευση σε πολλαπλά υπολογιστικά κέντρα τα οποία μπορεί να βρίσκονται σε διαφορετικές τοποθεσίες. Ο πελάτης θέλει να έχει γνώση της πραγματικής τοποθεσίας στην οποία αποθηκεύονται τα δεδομένα, για διάφορους λόγους, όπως π.χ. για τη συμμόρφωση με νομοθετικά πλαίσια που μπορεί να ισχύουν στη χώρα προέλευση. Επίσης, οι λόγοι μπορεί να σχετίζονται αποκλειστικά με την απόδοση, καθώς η εγγύτητα της τοποθεσίας των δεδομένων επηρεάζει το πόσο γρήγορα μπορεί ο χρήστης να τα ανακτήσει. Συγκεκριμένες απαιτήσεις σχετικά με την ακριβή τοποθεσία των δεδομένων, μπορούν να επιφέρουν διαφορετικές πολιτικές τιμολόγησης προς το πελάτη από το πάροχο της υπηρεσίας
- ***Εγγυήσεις σχετικά με την ακεραιότητα των δεδομένων***, όπου ο χρήστης ορίζει και απαιτεί συγκεκριμένες πολιτικές που αποσκοπούν τη επίτευξη ενός επιπέδου ασφάλειας των δεδομένων του. Μία τέτοια πολιτική είναι ο βαθμός αντιγραφής των δεδομένων. Η γενικότερη ύπαρξη λειτουργιών εφεδρείας δεδομένων (redundancy) παρέχεται με διαφορετικές μεθόδους ανάλογα με τις απαιτήσεις του πελάτη και τις δυνατότητες του παρόχου. Ο πελάτης μπορεί να ορίσει το αριθμό των αντιγράφων που πρέπει να παράγοντα για κάθε τύπο δεδομένων που αποθηκεύει σε μία υπηρεσία νέφους. Αντίστοιχα, μπορεί να ορίσει το πώς τα αντίγραφα αυτά κατανέμονται, δηλαδή το αν επιθυμεί τα αντίγραφα να κατανέμονται εντός ενός υπολογιστικού κέντρου ή

αποθηκεύονται σε πολλαπλά γεωγραφικά διασκορπισμένα υπολογιστικά κέντρα. Κάθε πολιτική εφεδρείας έχει τα πλεονεκτήματα και τις τεχνικές δυσκολίες για τους χρήστες και τους παρόχους και όπως και στην περίπτωση των απαιτήσεων του χρήστη σχετικά με την τοποθεσία αποθήκευσης των δεδομένων, διαφορετικές απαιτήσεις σχετικά με τη μέθοδο εφεδρείας μπορούν να επιβάλλουν διαφορετικές πολιτικές τιμολόγησης της υπηρεσίας. Επιπρόσθετες εγγυήσεις παρέχονται και στη περίπτωση καταστροφών και απώλειας δεδομένων των χρήστη, όπως ο χρόνος ανάκτησης των δεδομένων σε περίπτωση π.χ. βλάβης υλισμικού ή αποζημίωσης προς το χρήστη σε περίπτωση που η ασφάλεια και η ακεραιότητα των δεδομένων τεθεί σε κίνδυνο από κακόβουλες επιθέσεις τρίτων

Η εγγύηση συμμόρφωσης των παρόχων προς τα συμφωνηθέντα SLA, προϋποθέτει, από τη μεριά τους, την ύπαρξη προτυποποιημένης μεθόδου ορισμού, εποπτείας, συλλογής δεδομένων και αξιολόγησης συγκεκριμένων μετρικών. Όπως φαίνεται και στην Εικόνα 5, με αυτό τον τρόπο γίνεται ευκολότερη η σύγκριση μεταξύ παρόχων, αλλά και η δημιουργία SLAs, προς όφελος πάντα του καταναλωτή [48]. Η έλλειψη μίας τέτοια μεθόδου δημιουργεί σύγχυση στους καταναλωτές καθώς αδυνατούν να κατανοήσουν ακριβώς τους ορισμούς των χαρακτηριστικών, αλλά και το πώς επηρεάζουν τη μορφή των υπηρεσιών που θα λάβουν. Υπάρχουν, πολλές περιπτώσεις όπου η πραγματική απόδοση ενός παρόχου σχετικά με τα SLA δεν συμφωνεί με αυτή την οποία διαφημίζει και υπόσχεται.

Η ανάγκη για μία καθολική και προτυποποιημένη μεθοδολογία αξιολόγησης παρόχων, είναι εμφανής επίσης και στα ομόσπονδα δίκτυα νεφών όπου το ρόλο του πελάτη αναλαμβάνει ένας πάροχος υπηρεσιών νέφους. Η επιτυχία του εν λόγω μοντέλου (Cloud Federations) βασίζεται σε μεγάλο βαθμό στην δυνατότητα των παρόχων να εξυπηρετούν όχι μόνο τις απαιτήσεις σε υπηρεσίες και πόρους, αλλά να μπορούν να φέρνουν εις πέρας και τα SLA που κληρονομούν από τους χρήστες των παρόχων-καταναλωτών.



Εικόνα 5 Αξιολόγηση των συμβολαίων χρήσης μεταξύ πελάτη και παρόχου [48]

2.3 Θέματα ποιότητα παροχής υπηρεσιών (QoS) σε συστήματα και υπηρεσίες υπολογισμού

Το μοντέλο του νεφελώδους υπολογισμού προτάθηκε για να αντιμετωπίσει συγκεκριμένα πρακτικά προβλήματα τα οποία δημιουργήθηκαν από τη ραγδαία ανάπτυξη συστημάτων και τεχνολογιών IoT. Το κύριο χαρακτηριστικό τους είναι η προσφορά υπολογιστικών πόρων για την άμεση επεξεργασία δεδομένων που παράγονται από δίκτυα IoT. Η εξασφάλιση της ποιότητας υπηρεσίας βασίζεται στο γεγονός ότι τα δίκτυα «ομίχλης» βρίσκονται στο πολύ εγγύς περιβάλλον των δικτύων συσκευών. Με αυτό τον τρόπο δημιουργείται ένα ενδιάμεσο επίπεδο μεταξύ των πηγών δεδομένων και της πλατφόρμας που συνήθως βρίσκεται σε κεντρικές υποδομές IaaS.

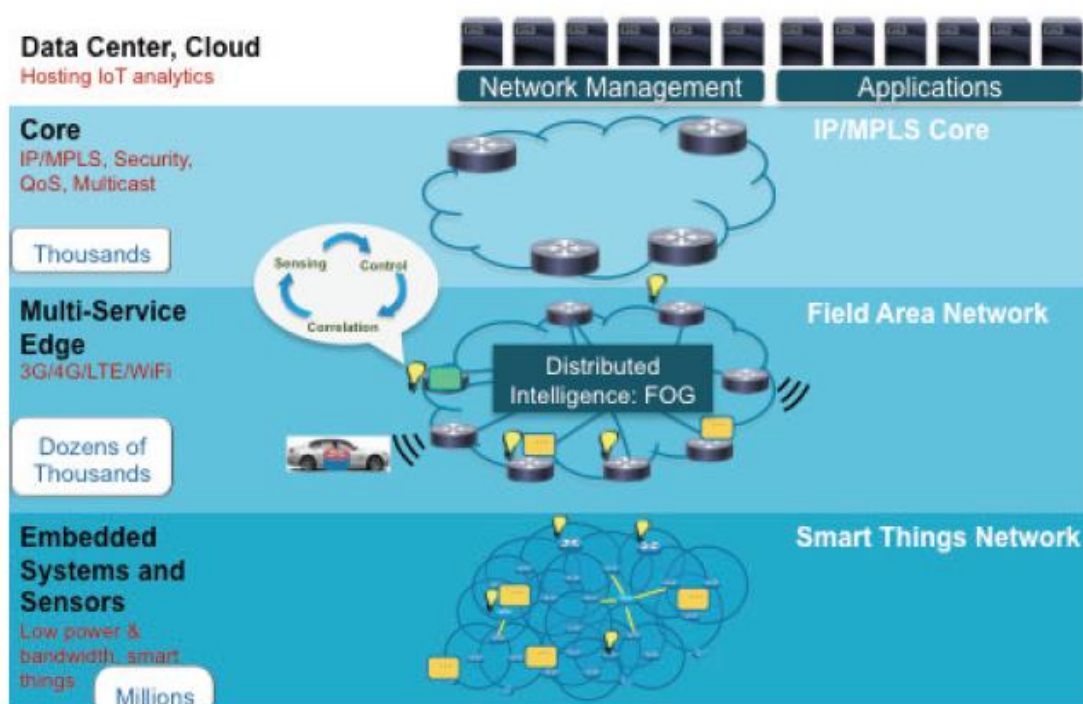
2.3.1 Υπάρχουσες λύσεις

Η υπάρχουσα αρχιτεκτονική προσέγγιση συστημάτων «ομιχλώδους» υπολογισμού, επικεντρώνεται στη δημιουργία δικτύων από συσκευές που βρίσκονται εντός του ή σε πολύ μικρή απόσταση από το περιβάλλον των δικτύων συσκευών που παράγουν δεδομένα σε ένα IoT σύστημα. Κάθε συσκευή που αποτελεί μέρος ενός δικτύου «ομίχλης» θα πρέπει εκτός από το να λάβει και να αποθηκεύσει προσωρινά τα δεδομένα, να εκτελέσει μία συγκεκριμένη μορφή εργασίας επάνω σε αυτά. Αυτές οι εργασίες ρουτίνας (tasks) συνήθως εφαρμόζουν συγκεκριμένους βασικούς αλγορίθμους στατιστικής ανάλυσης και μηχανικής μάθησης ώστε να ελέγξουν εάν κάποια δεδομένα εμπίπτουν εντός αυστηρά ορισμένων φυσιολογικών τιμών. Στις περιπτώσεις χρήσης συστημάτων IoT όπου έξυπνες συσκευές με τα κατάλληλα αισθητήρια όργανα ελέγχουν

συγκεκριμένα χαρακτηριστικά χρηστών, άλλων αντικειμένων ή του υπερκείμενου περιβάλλοντος, είναι σύνηθες τα παραγόμενα δεδομένα από τα αισθητήρια όργανα να αναλύονται για εντοπισμό τυχόν ανωμαλιών. Μία τέτοια διαδικασία θα πρέπει να διεκπεραιωθεί εγγυημένα και το συντομότερο δυνατόν, λόγω της κρισιμότητας για τη άμεση λήψη αποφάσεων σε συστήματα IoT. Σύμφωνα με την Cisco λοιπόν οι αρχιτεκτονικές «ομιχλώδους» υπολογισμού προσφέρουν τα εξής⁵:

- Ελαχιστοποίηση δικτυακής καθυστέρησης
- Αντιμετώπιση ζητημάτων ασφαλείας που σχετίζονται με το χειρισμό των IoT δεδομένων
- Αξιόπιστη λειτουργία των συστημάτων που στηρίζονται σε συσκευές IoT
- Συλλογή και διασφάλιση των δεδομένων σε μία ευρεία γεωγραφικά περιοχή

Στην Εικόνα 6 παρουσιάζεται μία βασική αρχιτεκτονική συστήματος «ομίχλης»:



Εικόνα 6 Παράδειγμα αρχιτεκτονικής «ομιχλώδους» υπολογισμού [49]

2.3.1.1 Αρχιτεκτονικές βελτιώσεις

Τα δεδομένα που δρομολογούνται προς τα συστήματα «ομιχλώδους» υπολογισμού πρέπει να αναλυθούν με τον πιο αποδοτικό και άμεσο τρόπο. Ένα πρακτικό ζήτημα που προκύπτει είναι η ανάγκη για τη κατανομή των δεδομένων και των

⁵ http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

διεργασιών σε συσκευές που θα προχωρήσουν στην επεξεργασία. Καθώς η επεξεργασία των δεδομένων αποτελεί επιπλέον επεξεργαστικούς πόρους πέρα από αποθηκευτικό χώρο, είναι απαραίτητο να παρέχεται η ανάλογη λειτουργικότητα στο δίκτυο «ομίχλης» που να μπορεί να καταναίμει τις επιμέρους εργασίες (tasks) ανάλογα με το κόστος σε χρόνο και σε πόρους.

Η χρησιμοποίηση δικτύων ορισμένων από λογισμικό (Software Defined Networks – SDNs) επιτρέπει την εισαγωγή λογικής στο σημείο άφιξης και συλλογής των δεδομένων [50], όπου μία συμβατή συσκευή SDN θα μπορεί να δρομολογεί δεδομένα και να αναθέτει εργασίες ανάλογα με το είδος και τον όγκο των δεδομένων στις επιμέρους συσκευές του δικτύου «ομίχλης». Η παραπάνω λύση χρησιμοποιεί το πρωτόκολλο MQTT ανταλλαγή των μηνυμάτων Fog σύμφωνα με το μοντέλο Pub/Sub.

Για τη κατανομή της κάθε IoT εργασίας (task) στο κατάλληλο τερματικό «ομίχλης» ανάλογα με τα στατικά (συνολικοί πόροι, αρχιτεκτονική υλισμικού) αλλά και δυναμικά χαρακτηριστικά (διαθέσιμοι πόροι, θέση στο δίκτυο, διαθέσιμη μπαταρία) του, έχει προταθεί η χρησιμοποίηση μοντέλων πρόβλεψης [51]. Τα εν λόγω μοντέλα χρησιμοποιούν μεθόδους μηχανικής μάθησης (machine learning) για να αναλύσουν τη συμπεριφορά χρηστών σε δίκτυα IoT και να προβλέψουν προσεγγιστικά την ανάγκη σε πόρους κάθε εργασίας που θα εισέλθει στο ενδιάμεσο δίκτυο «ομίχλης». Η εν λόγω προσέγγιση αντιλαμβάνεται το αρχιτεκτονικό επίπεδο Fog ως ένα υπολογιστικό κέντρο μικρότερης κλίμακας (Micro Datacenter), το οποίο εξυπηρετεί εγγύς γεωγραφικά δίκτυα IoT. Αυτή η υπόθεση επεκτείνεται στη δημιουργία ενός επιπλέον επίπεδου εικονικοποίησης ώστε οι IoT εργασίες να ανατίθενται σε εικονικά τερματικά.

Τέλος, μία ενδιαφέρουσα προσέγγιση που εντοπίζεται στη βιβλιογραφία [52], προτείνει τη δημιουργία προφίλ υπηρεσιών που προσφέρονται από τερματικά σε ένα δίκτυο «ομίχλης». Με αυτό τον τρόπο, ανάλογα με το είδος της εργασίας που πρέπει να εφαρμοστεί σε δεδομένα που παράγονται από δίκτυα IoT, αυτή ανατίθεται σε τερματικά τα οποία έχουν χαρακτηριστεί ότι μπορούν να τη διεκπεραιώσουν. Με αυτό τον τρόπο οι διαχειριστές του συστήματος μπορούν, μελετώντας τα χαρακτηριστικά των αλγορίθμων ανάλυσης των δεδομένων, αλλά των χαρακτηριστικών των συσκευών να ορίσουν προφίλ συσκευών ανάλογα με τα χαρακτηριστικά τους και να περιορίσουν τις υπολογιστικές τους δυνατότητες σε συγκεκριμένες εφαρμογές ή ομάδες εφαρμογών.

2.3.2 Τρέχοντα θέματα

Οι αξιόπιστη λειτουργία των πλατφορμών IoT είναι άρρηκτα συνδεδεμένη με τις υπηρεσίες νέφους καθώς δίνουν τη δυνατότητα για τη κάλυψη των αναγκών που

δημιουργούνται από τη χρήση και τη λειτουργίας τους. Καθώς οι πλατφόρμες IoT βρίσκουν εφαρμογή σε πολλαπλούς τομείς του ερευνητικού και βιομηχανικού τομέα, η λειτουργία τους στηρίζει διάφορες επιχειρησιακές διαδικασίες ενός οργανισμού ή επιχείρησης. Για παράδειγμα, ένα χαρακτηριστικό παράδειγμα είναι η χρήση συσκευών και πλατφορμών IoT για τη αυτοματοποίηση της διαδικασίας παραγωγής σε ένα εργοστάσιο, που εντοπίζονται στις πρωτοβουλίες Industry 4.0 [53] [54] και Factories of the Future [55]. Σε αυτές τα παραδείγματα οι πλατφόρμες IoT είναι υπεύθυνες για τη διαχείριση όλων των διαδικασιών της παραγωγής αναλύοντας δεδομένα που συλλέγονται από το περιβάλλον και τους χρήστες. Η λήψη αποφάσεων γίνεται με βάση την ανάλυση που πραγματοποιείται, και εκτός από αξιόπιστη παραγωγή γνώσης από τα δεδομένα αυτά, υπάρχει ξεκάθαρη ανάγκη για τη αμεσότητα στη ανάλυση των δεδομένων με το λιγότερη δυνατή χρήση πόρων.

Η αμεσότητα στη λήψη των αποφάσεων δημιουργεί νέα ζητήματα τα οποία χρειάζονται να καλυφθούν για να επιτραπεί η συνεχής ανάπτυξη και υιοθέτηση λύσεων IoT. Οι παραδοσιακές προσεγγίσεις όπου οι υπολογιστικές ανάγκες συστημάτων IoT καλύπτονται από ένα κεντρικό «νέφος», ενδέχεται στο μέλλον να μην επαρκεί για την κάλυψη αποτελεσματική κάλυψη τους. Εκτός από την ενδεχόμενη έλλειψη σε πόρους, μία κεντρική πλατφόρμα επεξεργασίας μπορεί να μην είναι σε θέση να ικανοποιήσει τις ανάγκες αμεσότητας λόγω συγκεκριμένων δικτυακών χαρακτηριστικών (π.χ. latency). Συγκεκριμένα οι ανάγκες που εγείρουν είναι οι εξής:

- Κατανομή πόρων υπολογισμού με βάση τη φύση των δεδομένων και των αλγορίθμων ανάλυσης
- Υπηρεσίες και δυνατότητες υπολογισμού και επεξεργασίας «πρωτογενών» δεδομένων με την ελάχιστη δυνατή καθυστέρηση. Το χαρακτηριστικό αυτό κρίνεται ιδιαίτερα κρίσιμο σε περιπτώσεις όπου η ανάλυση των δεδομένων οδηγεί σε μία λήψη απόφασης που θα προωθηθεί στη συνέχεια σε συστήματα ελέγχου (actuators)
- Επικοινωνία και πραγματικό χρόνο με ασύγχρονο τρόπο. Τα παραδοσιακά μοντέλα «νεφελώδους» υπολογισμού υποστηρίζουν επικοινωνία σύγχρονου τύπου (π.χ. HTTP REST). Αυτό καθιστά αδύνατη την αμφίδρομη επικοινωνία σε πραγματικό χρόνο, και επιφέρει επιπλέον κατανάλωση πόρων από τη μεριά των καταναλωτών των υπηρεσιών καθώς είναι αναγκασμένοι να εγκαθιδρύουν μία νέα σύνδεση κάθε φορά που χρειάζεται να επικοινωνήσουν με τις υπηρεσίες

- Δυνατότητες δημιουργίας δυναμικών «ομοσπονδιών» υπηρεσιών υπολογισμού, με έμφαση τη συνεχή εξασφάλιση παρουσίας επαρκών υπολογιστικών πόρων για να εξυπηρετήσουν τα αυξομειούμενα φορτία εργασίας

Σε πολλά από αυτά τα ζητήματα έχει προταθεί ένα μοντέλο υπολογισμού, το οποίο έχει ονομασθεί ως «ομιχλώδης» υπολογισμός ώστε να χαρακτηρίσει ένα ενδιάμεσο επίπεδο ανάμεσα στη πηγή των δεδομένων και το επίπεδο νέφους, όπου βρίσκονται οι υπηρεσίες υπολογισμού. Ο F. Bonomi είναι από τους πρώτους, όπου στη εργασία του [49], σκιαγράφησε το πλαίσιο λειτουργίας του ενδιάμεσου αυτού μοντέλου.

Το αρχιτεκτονικό μοντέλο του «ομιχλώδους» υπολογισμού μπορεί να απελευθερώσει πολλές δυνατότητες λύσεων IoT, εξασφαλίζοντας την αμεσότητα στην συλλογή και την επεξεργασία των δεδομένων σε πραγματικό χρόνο. Παρόλα αυτά για την περαιτέρω συνειδητοποίηση των δυνατοτήτων που έχουν οι υπηρεσίες IoT στη καθημερινότητα μας, είναι αναγκαία η επέκταση του ενδιάμεσου αυτού αρχιτεκτονικού επιπέδου με τρόπο τέτοιο ώστε υπάρξουν οι προϋποθέσεις για μία ολοκληρωμένη συλλογή πόρων από διάφορες ετερογενείς πηγές. Οι συσκευές και τα τερματικά που θα μετέχουν σε ένα περιβάλλον «ομίχλης», μπορούν, και θα πρέπει να χαρακτηρίζονται από πλήρη ετερογένεια σχετικά με τη φύση τους, αλλά και τη σχετική τους θέση. Η υποστήριξη κινητικότητας αποτελεί μία πρόκληση, ώστε να συμπεριληφθούν πάσας φύσεως φορητές και προσωπικές συσκευές, ώστε να αξιοποιηθούν οι σύγχρονες δυνατότητες του υλισμικού τους και των πολλαπλών πρωτοκόλλων συνδεσιμότητας που διαθέτουν.

Η παρούσα διατριβή προτείνει μία ολοκληρωμένη αρχιτεκτονική λύση που επεκτείνει τη λειτουργία του μοντέλου λειτουργίας μία πλατφόρμας «νέφους» IoT στο επίπεδο «ομίχλης» με σκοπό να εξασφαλίσει την καλύτερη δυνατή ποιότητα υπηρεσίας. Σε αυτό το πλαίσιο προτείνονται λύσεις για την υποστήριξη κινητών συσκευών, δημιουργίας συνεργατικών δικτύων «ομίχλης», αλλά και για την κατανομή του φόρτου εργασίας με τρόπο που εξασφαλίζει την ελάχιστη δυνατή καθυστέρηση στην εκτέλεση των εργασιών.

2.3.3 Πρακτικές περιπτώσεις χρήσης

Καθώς το μοντέλο του «ομιχλώδους» υπολογισμού είναι ακόμα σε ερευνητικό στάδιο, δεν υπάρχει δηλαδή κάποιο υλοποίηση σε βαθμό που να στηριχθεί στη βιομηχανία, είναι δύσκολο να εντοπιστούν πρακτικές περιπτώσεις χρήσης όπου το μοντέλο του «ομιχλώδους» υπολογισμού να βρίσκεται σε εφαρμογή. Οι ιδανικές

περιπτώσεις χαρακτηρίζονται από την ανάγκη για την ύπαρξη ευφυούς λογικής υπολογισμού στο εγγύς περιβάλλον της πηγής των δεδομένων όπου η ελάχιστη δυνατή καθυστέρηση είναι κρίσιμης σημασίας. Επίσης, η γεωγραφική διασπορά όπου η συνδεσιμότητα δεν είναι εξασφαλισμένη και η παραγωγή τεράστιων όγκων δεδομένων τα οποία δεν είναι πρακτικό να μεταφέρονται στο επίπεδο «νέφους», είναι δύο ακόμα χαρακτηριστικά, που θα στοιχειοθετούσαν μία περίπτωση χρήσης για ένα επίπεδο «ομίχλης»⁶.

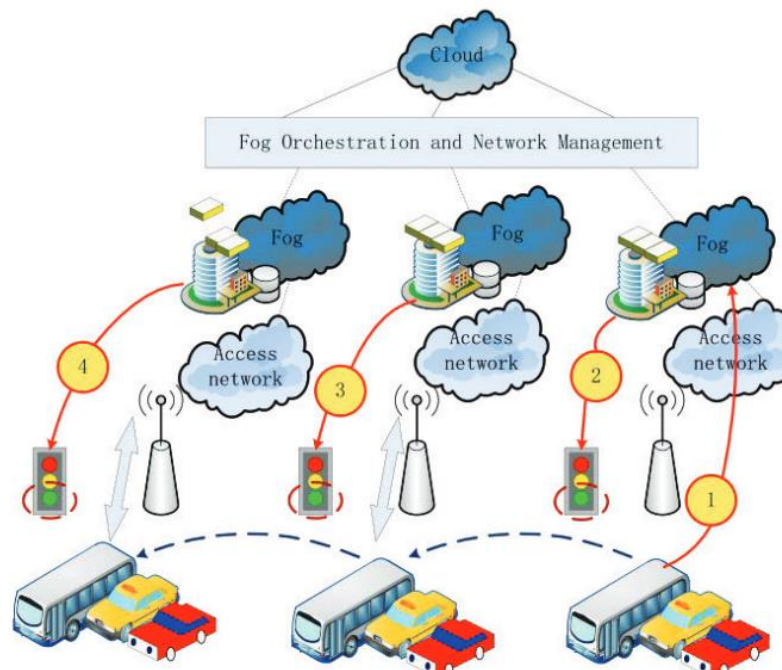
Σήμερα, υπάρχουν ήδη περιπτώσεις συστημάτων που στηρίζονται σε μεγάλο βαθμό στη συλλογή δεδομένων από συσκευές IoT. Ταυτόχρονα, το εύρος της λειτουργίας τους συντελεί στη δημιουργία και τη διακίνηση τεραστίου όγκου δεδομένων. Πράγματι, οι πλατφόρμες IoT είναι ένας από τους κύριους ενεργοποιητικούς παράγοντες του φαινομένου Big Data, καθώς τα δεδομένα που παράγονται από τα συστήματα IoT, ικανοποιούν και τις τέσσερις συνθήκες χαρακτηρισμού (Volume, Velocity, Veracity, Variety)⁷.

Τα συστήματα IoT ρύθμισης οδικής κυκλοφορίας (Smart Traffic Lights), τα οποία στηρίζονται στη λειτουργία «έξυπνων» σηματοδοτών. Τα συστήματα STL, αποσκοπούν στη δυναμική ρύθμιση της κυκλοφοριακής ροής σε δρόμους μαζικής και ταχείας κυκλοφορίας, εγκαθιστώντας ένα δίκτυο αισθητήριων οργάνων σε διασταυρώσεις. Τα αισθητήρια αυτά όργανα μετρούν την απόσταση και την ταχύτητα των οχημάτων καθώς προσεγγίζουν τη διασταύρωση από όλες τις πλευρές, και ανιχνεύουν τη παρουσία πεζών, αναβατών ποδηλάτων, αλλά και οχημάτων έκτακτης ανάγκης, όπως ασθενοφόρα και πυροσβεστικά οχήματα. Μετά την κατάλληλη ανάλυση των δεδομένων αυτών, αποστέλλουν τα κατάλληλα μηνύματα στους φωτεινούς σηματοδότες (actuators), ώστε να ρυθμίσουν την κυκλοφορία. Τα συστήματα STL στοχεύουν σε α) αποφυγή ατυχημάτων, β) διατήρηση σταθερής ροής της κίνησης και γ) συλλογή συναφών δεδομένων για την αξιολόγηση και τη βελτίωση του συστήματος [56]. Η χρησιμότητα ενός επιπέδου «ομίχλης» είναι εμφανής, καθώς η λήψη αποφάσεων πρέπει να γίνεται άμεσα, ιδίως σε έκτακτες καταστάσεις όπως αντιμετώπιση ατυχημάτων, κάτι που προϋποθέτει την άμεση ανάλυση των δεδομένων με τη μικρότερη δικτυακή καθυστέρηση. Επίσης, η αναμενόμενη συλλογή μεγάλου όγκου δεδομένων για την ιστορική ανάλυση της κυκλοφορίας σε ένα εκτεταμένο εύρος χρόνου, καθιστά μη βέλτιστη τη πρακτική αποστολής τους σε ένα κεντρικό αποθετήριο. Η συλλογή μέρους των δεδομένων στο επίπεδο «ομίχλης» για προσωρινή αποθήκευση (caching) ή για μία

⁶ <http://www.networkworld.com/article/3147085/internet-of-things/which-iot-applications-work-best-with-fog-computing.html>

⁷ <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>

πρωτογενή επεξεργασία, μπορεί να προσφέρει πολύ καλύτερη ποιότητα υπηρεσίας σε τέτοιου είδους συστήματα.



Εικόνα 7 Αρχιτεκτονική συστήματος ευφούς σηματοδότησης (STL) [57]

Τα συστήματα «επαυξημένης» πραγματικότητας (augmented reality – AR) και ανάλυσης εικόνας και βίντεο σε πραγματικό χρόνο, αποτελούν ένα ακόμα παράδειγμα απαιτητικών συστημάτων υπολογισμού. Όπως αναφέρεται και στις βιβλιογραφικές εργασίες των Yi et. al. [57] [58], οι εφαρμογές AR είναι πια ευρέως διαδεδομένες σε κινητές συσκευές (smartphones, tablets, smart glasses κ.α.), προβάλλοντας ένα εικονικό επίπεδο πληροφορίας στο πραγματικό κόσμο, μέσω της οθόνης της συσκευής του χρήστη. Παραδείγματα τέτοιων εφαρμογών αποτελούν τα Google Glass [59], Sony SmartEyeglass [60] και Microsoft HoloLens [61]. Οι εφαρμογές AR έχουν υψηλές απαιτήσεις σε υπολογιστικούς πόρους ώστε να αναλύουν την εικόνα σε πραγματικό χρόνο και μεγάλο δικτυακό εύρος ζώνης για τη μεταφορά των δεδομένων. Όπως είναι φυσικό, η παραμικρή καθυστέρηση (ακόμα και της τάξης των msec) στην ανάλυση των δεδομένων, θα μειώσει δραστικά την εμπειρία του χρήστη. Η υποστήριξη των εφαρμογών AR από το μοντέλο του «ομιχλώδους» υπολογισμού, μπορεί να συμβάλλει στην βελτίωση της ποιότητας της υπηρεσίας, μειώνοντας και το χρόνο επεξεργασίας και μεταφοράς των δεδομένων. Στην έρευνα των Ha et al. [62], παρουσιάζεται μία πλατφόρμα παροχής υπηρεσιών πληροφόρησης σε ενδυτές έξυπνες συσκευές Google Glass, οι οποία προωθεί μέρος της επεξεργασίας των δεδομένων σε κοντινά υπολογιστικά συστήματα (Cloudlets), και έτσι επιτυγχάνει χαμηλή δικτυακή καθυστέρηση από άκρη σε άκρη.

Τα συστήματα ανάλυσης δεδομένων διασυνδεδεμένων οχημάτων (connected vehicle), όλες οι πλατφόρμες ανάλυσης δεδομένων μεγάλης κλίμακος (Big Data), οι πλατφόρμες Smart Grid για την ευφυή ανάλυση των δεδομένων από τη κατανάλωση και διανομή ηλεκτρικής ενέργειας, αποτελούν χαρακτηριστικά παραδείγματα εφαρμογών που το μοντέλο του «ομιχλώδους» υπολογισμού, μπορεί να επιφέρει σημαντικές βελτιώσεις, επεκτείνοντας τη λειτουργία του υπολογιστικού «νέφους» στο εγγύς περιβάλλον του χρήστη και της πηγής δεδομένων για τη διατήρηση της υψηλότερης δυνατής ποιότητας υπηρεσίας (QoS).

2.4 Θέματα υπηρεσιών και συστημάτων υπολογισμού με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA)

Οι σύγχρονες ανάγκες των υπηρεσιών υπολογιστικών νεφών είναι σε θέση να δημιουργήσουν καινοτόμα μοντέλα διαχείρισης πόρων και λειτουργιών των υπολογιστικών κέντρων, που ανεξάρτητα με το μοντέλο της παρεχόμενης υπηρεσίας (IaaS, PaaS, SaaS) αποτελούν τη βασική δομή που υποστηρίζει την προσφερόμενη ως υπηρεσία υποδομή, πλατφόρμα ή εφαρμογή. Και καθώς το ζητούμενο που προκύπτει είναι η διαθεσιμότητα των πόρων ώστε να επιτυγχάνεται καλύτερη ποιότητα υπηρεσίας και αύξηση των βάσης των χρηστών, η διαχείριση τους προκρίνεται ως ένα μείζον θέμα για παρόχους υπηρεσιών υπολογισμού νέφους.

Κάθε πάροχος επιθυμεί να διαθέτει τη κατάλληλη υποδομή, τόσο σε μέγεθος όσο και σε τεχνολογία, κάτι το οποίο ενδεχομένως να μην είναι εφικτό. Το κόστος αγοράς, εγκατάστασης και συντήρησης είναι σημαντικό, ενώ υπάρχει και το ζήτημα της διαχείρισης βλαβών και σφαλμάτων τα οποία μπορούν να επηρεάσουν σημαντικά την ποιότητα της υπηρεσίας και να οδηγήσουν σε ποινές παραβίασης SLA (SLA penalties).

Η πρόοδος που έχει σημειωθεί στο τομέα των υπολογιστικών νεφών, καθοδήγησε την έρευνα προς τη δημιουργία ενός νέου τύπου συνεργατικών δικτύων τα οποία αποτελούνται από ανεξάρτητες υπολογιστικές υποδομές. Οι λεγόμενες ομοσπονδίες νέφους (Cloud Federations) στοχεύουν στη δημιουργία μίας αρχιτεκτονικής η οποία θα δίνει τη δυνατότητα σε παρόχους υπηρεσιών υπολογιστικού «νέφους» (Cloud Service Providers ή CSPs) να συνδέονται μεταξύ τους και να προσφέρουν τις υπηρεσίες τους ή να ενοικιάζουν τους πόρους τους, συνεργαζόμενοι προς το κοινό στόχο της αύξησης της ποιότητας της υπηρεσίας. Η αρχιτεκτονική δομή αυτή στοχεύει επίσης στη προσφορά υπηρεσιών ευρετηρίου, διευθυνσιοδότησης, και πλειστηριασμού δημιουργώντας ένα ολοκληρωμένο και διαλειτουργικό «διαδίκτυο των νεφών» (Internet of Clouds) [7].

Σημαντικό σημείο αναφοράς σε διάφορες εργασίες είναι η αποδοτική λειτουργία των υπηρεσιών που θα λειτουργούν ως διαμεσολαβητές ή Brokers. Συνήθως προσφέρουν τις εξής λειτουργίες:

- Υποδοχής και ανάλυσης αιτημάτων πελατών CSP
- Εύρεσης (filtering), βαθμολόγησης (rating) και επιλογής (selection) του καταλληλότερου παρόχου
- Καταλόγου των διαθέσιμων CSP μαζί με τα ποιοτικά και ποσοτικά χαρακτηριστικά τους. Ως ποσοτικά, μπορούν να θεωρηθούν εκείνα τα χαρακτηριστικά τα οποία προσδιορίζουν τους διαθέσιμους πόρους που προσφέρει η υποδομή και ποιοτικά τα χαρακτηριστικά τα οποία προσδιορίζουν σε ένα βαθμό τη ποιότητα υπηρεσίας.
- Αποθετηρίου δεδομένων που θα διατηρούν ιστορικό με προηγούμενες συναλλαγές και μετρήσεις σχετικά με τη τήρηση των συμφωνηθέντων SLAs

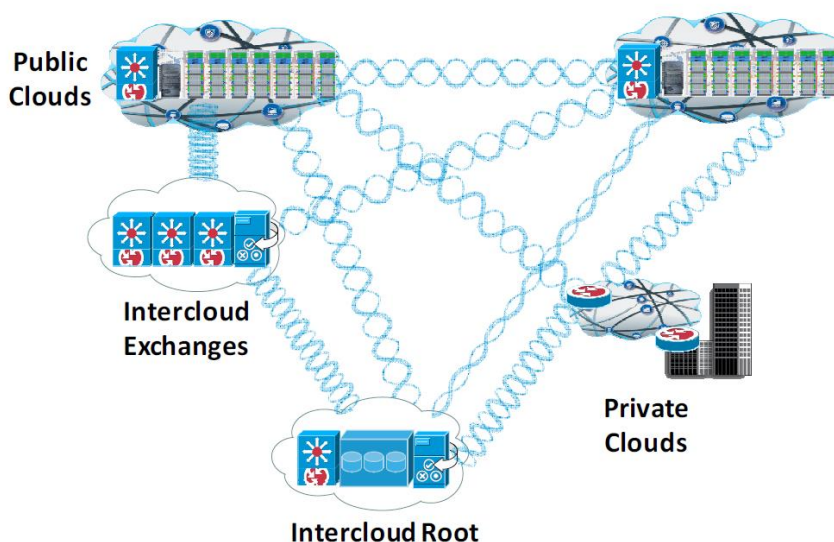
Είναι εμφανές ότι οι υπηρεσίες διαμεσολαβητή αποτελούν ένα ζωτικό τμήμα μίας αρχιτεκτονικής προσανατολισμένης σε ομοσπονδίες νεφών. Σε αυτή τη βάση λοιπόν, η παρούσα διατριβή παρουσιάζει μία αρχιτεκτονική προσέγγιση σχετικά με τη λειτουργία διαμεσολαβητή σε συνεργατικά δίκτυα «νεφών». Η καινοτομία της προσέγγισης εντοπίζεται στη χρησιμοποίηση αλγορίθμου βαθμολόγησης που χρησιμοποιείται σε ανταγωνιστικά παίγνια (π.χ. διαδικτυακά παιχνίδια πολλαπλών παιχτών), προσαρμοσμένο στη συνεργατική φύση του ενός «διαδικτύου νεφών».

2.4.1 Υπάρχουσες λύσεις

2.4.1.1 Πρότυπο IEEE P2302

Κύριος εκφραστής των αρχιτεκτονικών Intercloud είναι το εν ενεργεία πρότυπο P2302 της IEEE [7], όπου περιγράφεται η λειτουργία παρόχων υπηρεσιών υπολογιστικού νέφους εντός της ομοσπονδίας. Για την επίτευξη της διαλειτουργικότητας μεταξύ ετερογενών παρόχων, δηλαδή παρόχων που διαθέτουν εντελώς διαφορετικά είδη υπηρεσιών και APIs και ως συνέπεια δεν έχουν τη δυνατότητα να επικοινωνήσουν απευθείας, προτείνεται η δημιουργία των Intercloud Gateways, πυλών λογισμικού δηλαδή, που θα αποτελούν το σημείο επαφής κάθε παρόχου με το υπόλοιπο δίκτυο και τις οντότητες του υποστηρίζοντας πολλαπλές τεχνολογίες ανάπτυξης (π.χ. Amazon EC2 [39], Azure [63], και Google [64]) να παρέχει ένα επίπεδο αφαιρετικότητας όσον αφορά την επικοινωνία διαφορετικών παρόχων. Όσον αφορά το επίπεδο του διαμεσολαβητή το πρότυπο προτείνει τη δημιουργία δύο ξεχωριστών

οντοτήτων: α) του InterCloud Root και β) του InterCloud Exchange οι οποίες αναμένονται να λειτουργούν σε κάθε δίκτυο δημιουργώντας έτσι μία ιεράρχηση όσον αφορά τη επικοινωνία μεταξύ διαφορετικών ομόσπονδων δικτύων προσεγγίζοντας σε αντιστοιχία τη λειτουργία του Internet.



Εικόνα 8 Αρχιτεκτονική Intercloud του προτύπου P2302 [7]

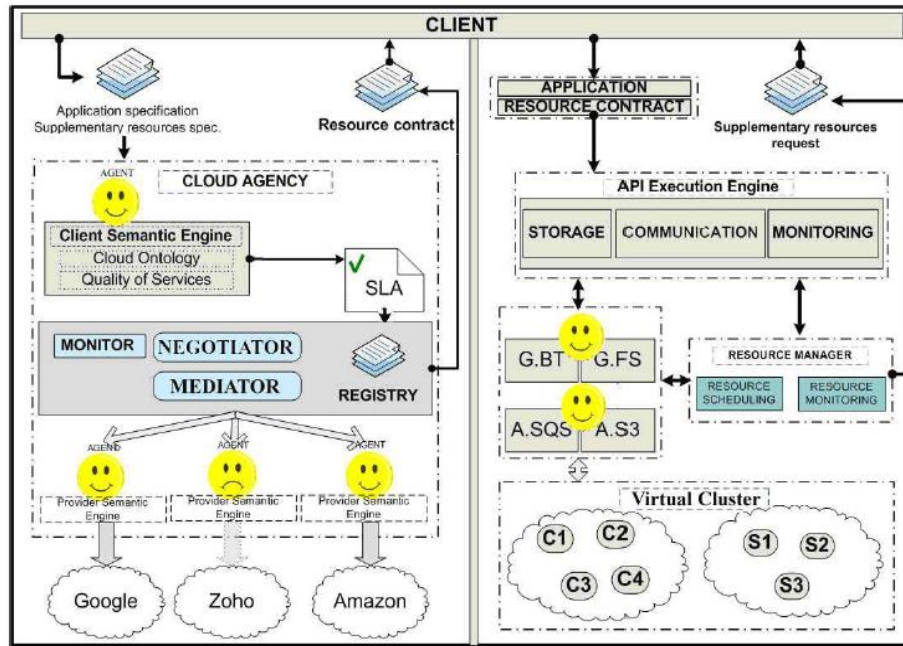
Η προσέγγιση του προτύπου P2302 ακολουθεί ένα κεντρικοποιημένο μοντέλο όπου κάθε CSP θα πρέπει να επικοινωνεί με τον Intercloud Root για την εύρεση παρόχων που θα ικανοποιήσουν τα αιτήματα του και εν συνεχεία με τον Exchange για τη δρομολόγηση της επικοινωνίας του με τον υποψήφιο πάροχο. Οι βασικές λειτουργίες ανατίθενται σε κεντρικές δομές που αναλαμβάνουν το κυριότερο μέρος των απαραίτητων διαδικασιών. Ως πρωτόκολλο επικοινωνίας έχει επιλεγθεί το XMPP [65], καθώς παρέχει τις απαραίτητες λειτουργίες επικοινωνίας μεταξύ οντοτήτων, αλλά και ελέγχου παρουσίας και ασφαλείας. Το πρωτόκολλο XMPP προτείνεται επίσης ως βάση για τη δημιουργία υπηρεσιών που θα χαρακτηρίζονται από τη ασύγχρονη φύση τους, όπως οι επεκτάσεις XEP-0244 και XWS4] [66] [67].

2.4.1.2 Λοιπές αρχιτεκτονικές προσεγγίσεις

Μία αντίστοιχη αρχιτεκτονική με αυτή που προτείνεται στο πρότυπο P2302 παρουσιάζεται στην εργασία [68]. Στην εν λόγω εργασία προτείνεται η δημιουργία ενδιάμεσων δομών που θα αναλαμβάνουν της διαδικασίες περάτωσης και ολοκλήρωσης των συναλλαγών μεταξύ δύο ή περισσότερων παρόχων. Σε αυτό το πλαίσιο παρουσιάζεται η λειτουργία του CloudExchange (CEX), το οποίο θα λειτουργεί σαν μία αγορά ή ανταλλακτήριο υπηρεσιών νέφους όπου κάθε υποδομή παρόχου θα δημοσιεύει και προσφέρει τις υπηρεσίες του. Αντίστοιχα, ο διαμεσολαβητής «νέφους» (Cloud

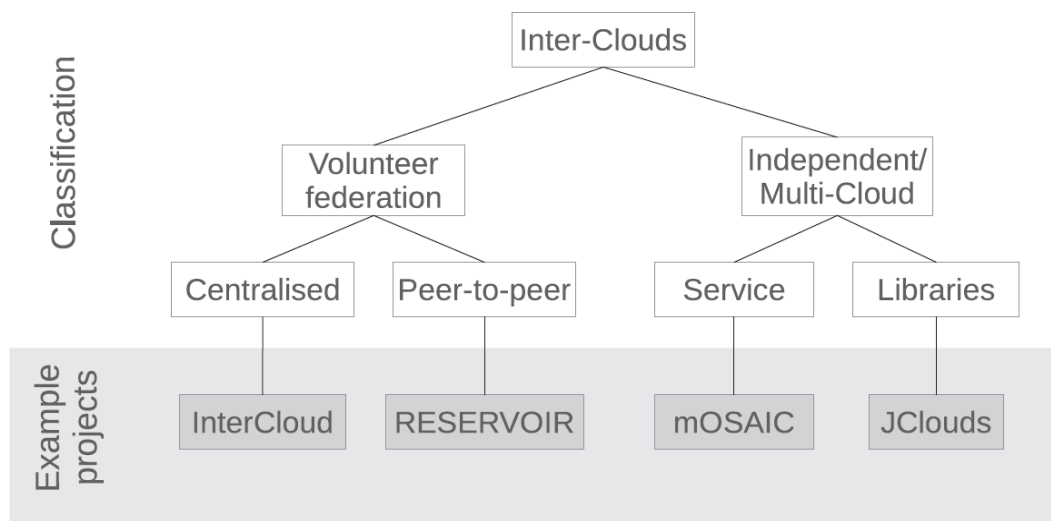
Broker) θα λειτουργεί με βάση τις απαιτήσεις κάθε παρόχου ώστε να εντοπίσει τις κατάλληλες υπηρεσίες και τον κατάλληλο πάροχο που θα τις προσφέρει. Τέλος, η υπηρεσία ενορχηστρωτή νέφους (Cloud Coordinator), θα λειτουργεί εντός της υποδομής του κάθε παρόχου όπου σε συνεργασία με το επίπεδο υπηρεσιών της υποδομής (IaaS API) θα αναλαμβάνει όλες τις λειτουργίες που σχετίζονται με την υποδομή (διαχείριση πόρων) και τη διαχείριση των πολιτικών συμμετοχής στο ομόσπονδο δίκτυο από άποψη SLA.

Αντίθετα με τις προσεγγίσεις που ακολουθούν με κεντρική και ιεραρχική δομή για τη λειτουργία των ομόσπονδων δικτύων, υπάρχουν και δίκτυα peer-to-peer (P2P) όπου οι προαναφερθείσες υπηρεσίες μεσάζοντος βρίσκονται σε κάθε πάροχο ο οποίος τις χρησιμοποιεί για να εντοπίσει και εγκαθιδρύσει μία συναλλαγή με έναν πάροχο. Μία τέτοιου τύπου προσέγγιση παρουσιάζεται στα πλαίσια του ερευνητικού προγράμματος mOSAIC [24], όπου προτείνονται λειτουργίες που δρουν στη πλευρά του τελικού χρήστη κυρίως και αναλαμβάνουν με βάση τα απαιτούμενα SLAs και πόρους να βρίσκουν τον κατάλληλο πάροχο. Συγκεκριμένα, για την πλευρά του χρήστη λειτουργεί το ένα πολυπρακτορικό σύστημα (Cloud Agency) το οποίο αναλαμβάνει τη διαπραγμάτευση των SLAs, την μεταφορά και εγκατάσταση των υπηρεσιών/εφαρμογών πελάτη, και τη εύρεση υπηρεσιών υποψήφιων παρόχων. Από τη πλευρά του παρόχου λειτουργεί ανάλογη υπηρεσία διαχείρισης πόρων για τη δημιουργία των εικονικών δομών που θα προσφερθούν στον τελικό χρήστη. Αν και η αρχιτεκτονική του mOSAIC δεν προορίζεται εξ' αρχής για ομοσπονδίες νεφών, η εν λόγω πλατφόρμα υπηρεσιών μπορεί να εξυπηρετήσει τέτοιου είδους ανάγκες αν θεωρηθεί ότι το ρόλο του τελικού χρήστη αναλαμβάνει ένας άλλος πάροχος υπηρεσιών νέφους.



Εικόνα 9 Γενική αρχιτεκτονική της πλατφόρμας mOSAIC [28]

Συνολικά, υπάρχουν διαφορετικές αρχιτεκτονικές που εξετάζουν την οργάνωση ενός ομόσπονδου δικτύου νεφών, οι οποίες εξετάζουν και συγκεκριμένα χαρακτηριστικά του δικτύου από αρχιτεκτονικής απόψεως. Συνοπτικά, η ταξινόμηση αυτών των προσεγγίσεων παρουσιάζεται στην Εικόνα 10:



Εικόνα 10 Κατηγοριοποίηση «ομόσπονδων» δικτύων νεφών με βάση την αρχιτεκτονική ανάπτυξης [69]

2.4.1.3 Αρχιτεκτονικές διαμεσολαβητή και αξιολόγησης παρόχων

Οι ομοσπονδίες νεφών και τα συνεργατικά δίκτυα, στηρίζονται κυρίως στις ενδιάμεσες υπηρεσίες που θα δρομολογήσουν την επικοινωνία και την ανταλλαγή υπηρεσιών ανάμεσα σε δύο ή περισσότερους παρόχους υπηρεσιών νέφους. Μία από τις

κύριες λειτουργίες που παρατηρείται σε όλες τις αρχιτεκτονικές προσεγγίσεις, είναι αυτή του διαμεσολαβητή, της υπηρεσίας δηλαδή που έχει την ευθύνη να λαμβάνει τις απαιτήσεις του πελάτη σε ένα προτυποποιημένο αίτημα και να βρίσκει τον κατάλληλο πάροχο, με τελικό σκοπό να εξασφαλίσει τη καλύτερη δυνατή ποιότητα υπηρεσίας και τα καλύτερα δυνατά επίπεδα τήρησης SLA.

Για την επιλογή του κατάλληλου παρόχου χρειάζεται η εφαρμογή αξιολόγησης των παρόχων, με βάση πάντα τις απαιτήσεις του τελικού χρήστη. Διάφορες μέθοδοι έχουν προταθεί ώστε να επιτυγχάνεται μία δυναμικού τύπου αξιολόγησης, η οποία να γίνεται με βάση συγκεκριμένους ποιοτικούς δείκτες. Παρ' όλο που οι λύσεις που παρουσιάζονται είναι ανεπτυγμένες γύρω από τον τελικό πελάτη/χρήστη, τα προτεινόμενα μοντέλα μπορούν να επεκταθούν σε περιπτώσεις χρήσης και σενάρια ενός δικτύου που απαρτίζεται από ανεξάρτητους παρόχους υπηρεσιών νέφους, μιας που κάθε πάροχος μπορεί να αναλάβει το ρόλο του πελάτη/τελικού χρήστη και το αντίστροφο. Η διαδικασία επιλογής του κατάλληλου παρόχου για να ικανοποιήσει ένα αίτημα μίας συνεργαζόμενης υποδομής, συνήθως επιβάλλει τη ταξινόμηση των διαθέσιμων παρόχων σύμφωνα με ποσοτικά ή ποιοτικά χαρακτηριστικά τα οποία καλύπτουν τις απαιτήσεις του αιτούντος παρόχου. Με βάση τα παραπάνω, προκύπτει ένα πολυκριτηριακό πρόβλημα, το οποίο σε αρκετές περιπτώσεις μπορεί να εξυπηρετηθεί μέσω της οντολογικής αναπαράστασης των δεδομένων αυτών στο επίπεδο του διαμεσολαβητή. Συγκεκριμένες εργασίες έχουν πραγματοποιηθεί για τη δημιουργία ενός ενιαίου μοντέλου το οποίο θα περιγράφει όλα τα ήδη δεδομένων αλλά και κανόνων που θα διέπουν κάθε οντότητα σε ένα «διαδίκτυο νεφών» [70] [28].

Η υιοθέτηση σημασιολογικών μοντέλων παρέχει χαρακτηριστική ευκολία στο φιλτράρισμα των υποψήφιων παρόχων, παρόλα αυτά είναι αναγκαία και επιπλέον ταξινόμηση σύμφωνα με τα χαρακτηριστικά και την απόδοση τους. Συγκεκριμένες μελέτες έχουν γίνει για τη βαθμολόγηση (rating) παρόχων σύμφωνα με την όσο το δυνατόν πιο έγκυρη μέτρηση των συμφωνηθέντων SLA. Στην εργασία [71] προτείνεται η αξιολόγηση των παρόχων σύμφωνα με δείκτες μέτρησης υπηρεσίας (Service Measurement Index – SMI), με τη εφαρμογή προσαρμοσμένων βαρών σε κάθε μετρήσιμο χαρακτηριστικό μέσω της μεθόδου Advance Hierarchy Process (AHP). Μία αντίστοιχη προσέγγιση [72], αξιολογεί τους παρόχους με βάση τη ομοιότητα των χαρακτηριστικών που προσφέρουν σε σχέση με τα αιτούμενα χαρακτηριστικά του πελάτη (η μέτρηση γίνεται μέσω της Ευκλείδειας απόστασης των λύσεων από την ιδανική).

Οι παραπάνω μέθοδοι δίνουν συγκεκριμένες λύσεις στο πολυκριτηριακό και πολλαπλών διαστάσεων πρόβλημα της αξιολόγησης και βαθμολόγησης παρόχων. Αξίζει

όμως να σημειωθεί ότι αυτά τα τελικά προτεινόμενα συστήματα δημιουργήθηκαν με επίκεντρο τον τελικό χρήστη, χωρίς να λαμβάνουν υπόψιν τα χαρακτηριστικά ενός ομόσπονδου δικτύου υπολογιστικών νεφών όπου τελικός χρήστης και πάροχος είναι έννοιες εναλλασσόμενες. Σε περίπτωση, επιπρόσθετα, που χρειάζεται να εξετάζεται και η συμπεριφορά των εμπλεκόμενων στο σύστημα τότε χρειάζεται μία αρκετά προσαρμόσιμη τεχνική αξιολόγησης των παρόχων.

2.4.2 Τρέχοντα θέματα

Η αποδοτική και σωστή λειτουργία ενός δικτύου παρόχων υπηρεσιών νέφους, απαιτεί την επίτευξη των παρακάτω στόχων:

- **Διαλειτουργικότητα**, η οποία απαιτεί μία προτυποποιημένη μορφή επικοινωνίας μεταξύ των οντοτήτων μίας ομοσπονδίας νεφών. Η ύπαρξη προτυποποίησης κρίνεται ιδιαιτέρως απαραίτητη ώστε να είναι δυνατή η επικοινωνία μεταξύ ετερογενών υποδομών
- **Αξιολόγηση/εποπτεία**, η οποία θα εξασφαλίζει ότι κάθε πάροχος σε ένα συνεργατικό δίκτυο, ιδιαίτερα όταν αυτό είναι εθελοντικού χαρακτήρα, θα αποδίδει σύμφωνα με τα απαιτούμενα και ότι η συνολική απόδοση και αξία του δικτύου θα κυμαίνεται σε υψηλά επίπεδα. Βασικό προαπαιτούμενο είναι η ύπαρξη ενός μηχανισμού εποπτείας που θα έχει τη δυνατότητα να αξιολογεί τις συναλλαγές μεταξύ δύο παρόχων, ανεξάρτητα με την αναφορά του καταναλωτή/πελάτη. Εν συνεχεία, με βάση την εποπτείας και την αξιολόγηση των συναλλαγών, θα ακολουθεί η αξιολόγηση των παρόχων με βάση την απόδοσή τους, και τη συμπεριφορά τους εντός του δικτύου (σε περίπτωση όπου το δίκτυο είναι εθελοντικού χαρακτήρα)
- **Ασφάλεια/εμπιστοσύνη**, η οποία θα εξασφαλίζει ότι η ανταλλαγή των οποιασδήποτε μορφής δεδομένων θα χαρακτηρίζεται από απόλυτη ασφάλεια και προστασία της ιδιωτικότητας. Οι ομοσπονδίες νεφών αποτελούνται από ανεξάρτητους παρόχους υπηρεσιών νέφους που ανήκουν σε διαφορετικούς οργανισμούς, οι οποίοι με τη σειρά τους διατηρούν τη δικιά τους βάση χρηστών. Συνεπώς, η συναλλαγές με τρίτους παρόχους για την κατανάλωση υπηρεσιών ενδέχεται να οδηγήσει στη μεταφορά δεδομένων σε υποδομές τρίτων, όπου ο ιδιοκτήτης οργανισμός δεν έχει κανένα έλεγχο. Ως εκ τούτου, προκύπτει και το ζήτημα της εμπιστοσύνης, ώστε κάθε πάροχος να είναι βέβαιος ότι οι

υπηρεσίες που καταναλώνει είναι υψηλής ποιότητας και διατηρείται η ακεραιότητα και η ιδιωτικότητα των δεδομένων

Τα παραπάνω αποτελούν ανοικτά ζητήματα που πρέπει να απαντηθούν ώστε δημιουργηθεί μία ολοκληρωμένη αρχιτεκτονική προσέγγιση που τα καλύπτει. Κυρίαρχης σημασίας αποτελεί η λειτουργία της υπηρεσίας διαμεσολαβητή (Broker), καθώς σε ένα κεντρικοποιημένο και ιεραρχικά δομημένο μοντέλο, θα είναι υπεύθυνος για τη ικανοποίηση όλων των παραπάνω ζητημάτων. Για να επιτευχθεί αυτό χρειάζεται μία προτυποποιημένη μέθοδος χαρακτηρισμού των αιτημάτων υπηρεσιών και των συμφωνηθέντων συμβολαίων χρήσης (SLA). Μέχρι στιγμής, κάθε πάροχος παρέχει ξεχωριστά μοντέλα και συμβόλαια χρήσης υπηρεσιών, τα οποία επιπρόσθετα προτυποποιούνται με διαφορετικό τρόπο.

Επίσης, παρ' όλο που έχουν διενεργηθεί συγκεκριμένες εργασίες επάνω στο ζήτημα της αξιολόγησης της υπηρεσίας με βάση την τήρηση των συμβολαίων SLA από τη πλευρά του παρόχου, δεν είναι προσαρμοσμένες σε ένα συνεργατικό περιβάλλον όπου κάθε χρήστης υπηρεσίας μπορεί να είναι και πάροχος ταυτόχρονα. Η ιδιότητα αυτή απαιτεί μία πιο συγκεκριμένη μέθοδο αξιολόγησης, η οποία, εκτός των χαρακτηριστικών των υποψήφιων παρόχων, θα λαμβάνει επίσης τα χαρακτηριστικά του πελάτη ως ισότιμου μέλους του δικτύου, ώστε να εξασφαλίζεται η μέγιστη αποδοτικότητα και δικαιοσύνη στο σύνολο του δικτύου.

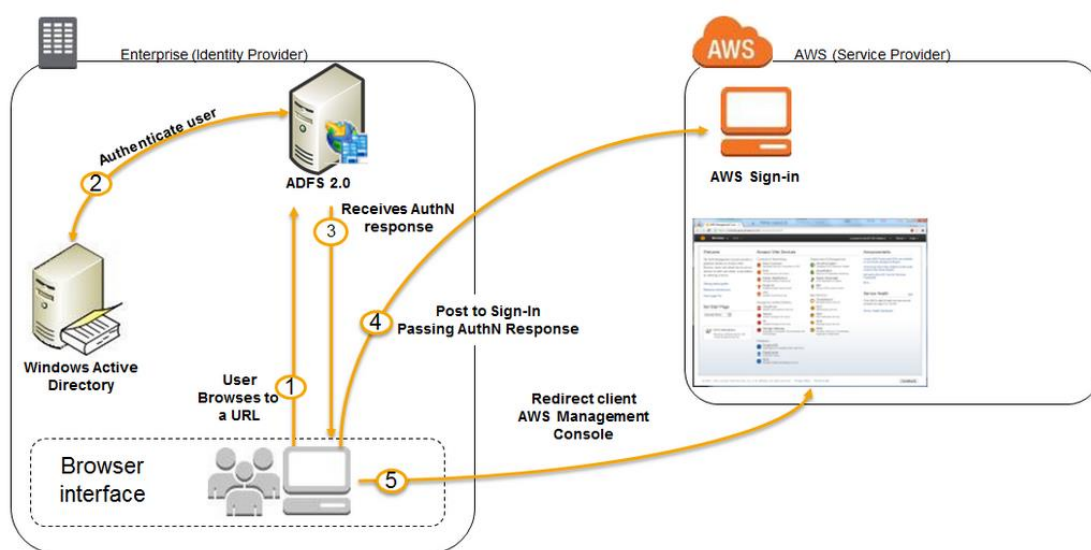
Μία από τις προτάσεις της παρούσας διατριβής είναι η λειτουργία ενός μηχανισμού αξιολόγησης παρόχων με βάση την απόδοση τους, καθώς επίσης και τη συνολική τους συμπεριφορά εντός του δικτύου. Η αρχιτεκτονική πρόταση παρουσιάζει την λειτουργία πλατφόρμας διαμεσολαβητή η οποία θα αναλαμβάνει να βρίσκει τους κατάλληλους παρόχους για κάθε πελάτη, λαμβάνοντας υπόψιν κάθε φορά, την απόδοση και συμπεριφορά και των δύο συμμετεχόντων.

2.4.3 Πρακτικές περιπτώσεις χρήσης

Η διαλειτουργικότητα μεταξύ υποδομών υπολογιστικών νεφών είναι ένα πεδίο έρευνας σχετικά νέο. Η προτυποποίηση της επικοινωνίας και η δημιουργία μίας αρχιτεκτονικής που θα υποστηρίζει όλες τις μορφές υπηρεσιών και ετερογενείς πλατφόρμες λογισμικού υπολογιστικού νέφους (π.χ. Openstack [9], OpenNebula [73]), απαιτεί μία πολύ λεπτομερή προσέγγιση.

Αν και η ετερογένεια δεν υποστηρίζεται, η δημιουργία δικτύων που αποτελούνται από υπολογιστικά κέντρα υπηρεσιών νέφους, ως επί το πλείστον, υποστηρίζεται από τους περισσότερους παρόχους και πλατφόρμες ανάπτυξης υπηρεσιών νέφους.

Για παράδειγμα η υπηρεσία AWS [74] της Amazon προσφέρει διεπαφές για σύνδεση πολλαπλών χρηστών σε ένα λογαριασμό AWS, μέσω της υπηρεσίας Azure Active Directory [75]. Συγκεκριμένα η υπηρεσία Διαχείρισης Ταυτοποίησης και Πρόσβασης (Identity and Access Management – IAM), προσφέρει τις διεπαφές για τη δημιουργία μίας πύλης πρόσβαση στις υπηρεσίες AWS, στην οποία θα έχουν πρόσβαση χρήστες από εξωτερικές οντότητες (που δεν διαθέτουν δηλαδή απευθείας πρόσβαση στις υπηρεσίες της Amazon). Με αυτή τη λειτουργία «ομόσπονδης» ταυτοποίησης (identity federation), δίνεται η δυνατότητα στους χρήστες ενός οργανισμού να αυθεντικοποιούνται σε οποιονδήποτε OpenID-συμβατό πάροχο ταυτοποίησης και να συνδέονται σε ένα κοινό λογαριασμό χρήστη υπηρεσιών AWS⁸.



Εικόνα 11 «Ομόσπονδη» ταυτοποίηση υπηρεσιών AWS μέσω της υπηρεσίας ταυτοποίησης Azure Active Directory⁹

Η πλατφόρμα διαχείρισης και ενορχήστρωσης υπηρεσιών και εφαρμογών για περιέκτες λογισμικού Kubernetes [37], δίνει τη δυνατότητα δημιουργίας συστοιχιών περιεκτών κατανεμημένων σε διαφορετικές υποδομές [76]. Η λειτουργία αυτή στηρίζεται σε δύο βασικούς πυλώνες:

- Συγχρονισμό πόρων και δεδομένων που σχετίζονται με τη χρήση των περιεκτών κατά μήκος των συστοιχιών
- Δυνατότητα ομαδοποίησης των συστοιχιών που βρίσκονται σε διαφορετικές περιοχές και ζώνες σε μία κοινή βάση ευρετηρίου, όπως π.χ. η δυνατότητα διαμόρφωσης εξυπηρετητών DNS και κατανεμητών

⁸ <https://aws.amazon.com/iam/details/manage-federation>

⁹ <https://aws.amazon.com/blogs/security/enabling-federation-to-aws-using-windows-active-directory-adfs-and-saml-2-0/>

φορτίου (load balancers), που θα προσφέρουν την λειτουργικότητα τους περικλείοντας όλες τις συστοιχίες ασχέτως της γεωγραφικής τους θέσης

Ένα ιδιαίτερο χαρακτηριστικό είναι ότι η πλατφόρμα Kubernetes υποστηρίζει την εν λόγω λειτουργικότητα άσχετα με τον πάροχο της υποδομής νέφους στον οποίο είναι εγκατεστημένη η συστοιχία περιεκτών. Έτσι δίνεται η δυνατότητα στους χρήστες για να αναπτύξουν τις εφαρμογές εξασφαλίζονται υψηλή διαθεσιμότητα και ποιότητα υπηρεσίας, ευκολία στη μετάβαση από μία υποδομή σε μία άλλη, και αποφεύγοντας τους περιορισμούς παρόχων IaaS (vendor lock-in) που επιβάλλονται στη μεταφορά εικονικών μηχανημάτων από μία γεωγραφική ζώνη σε μία άλλη.

Παρόλες τις ενδιαφέρουσες αρχιτεκτονικές προσεγγίσεις που επιτρέπουν τη διαλειτουργικότητα όσον αφορά τη χρήση υπηρεσιών νέφους, μία πραγματική περίπτωση δημιουργίας «ομόσπονδων» δικτύων από ανεξάρτητες υποδομές υπηρεσιών νέφους δεν υπάρχει σε βαθμός ωριμότητας που να εφαρμόζεται σε πραγματικές συνθήκες. Σε ερευνητικό επίπεδο μία τέτοια υποδομή επιτεύχθηκε στα πλαίσια του ερευνητικού προγράμματος BonFIRE [77]. Η πλατφόρμα BonFIRE αποτέλεσε μία πρώτη προσπάθεια δημιουργίας μίας αρχιτεκτονικής υποδομής που θα προσέφερε σε ερευνητές και χρήστες την δυνατότητα χρήσης υπηρεσιών IaaS από πολλαπλούς παρόχους και οργανισμούς που διατηρούν υπολογιστικές υποδομές από διάφορες χώρες της Ευρώπης. Η λειτουργία του στηρίχθηκε στην ανάπτυξη ενός διαμεσολαβητή (Broker) και της δημοσιοποίησης ενός API για την χρήση των υπηρεσιών διαφανώς ανεξάρτητα από τον πάροχο. Επιπρόσθετα, παρέχεται και μία κεντρική γραφική διεπαφή με τη μορφή εφαρμογής ιστού, προς τους χρήστες για περαιτέρω ευκολία χρήσης της πλατφόρμας BonFIRE.

2.5 Θέματα ενεργειακής αποδοτικότητας σε περιβάλλοντα υπολογιστικών νεφών

Οι προοπτικές που έχουν δημιουργήσει τα σύγχρονα μοντέλα παροχής υπηρεσιών υπολογισμού μέσα από σύγχρονα υπολογιστικά κέντρα, οδηγούν στην όλο και αυξανόμενη ζήτηση υπηρεσιών που θα καλύπτουν όλο και περισσότερα ζητήματα, πρακτικά και λειτουργικά, που σχετίζονται με τις σύγχρονες μορφές εφαρμογών και υπηρεσιών. Από τη πλευρά ενός παρόχου υπηρεσιών νέφους, η διατήρηση ανταγωνιστικότητας προϋποθέτει τη διατήρηση της ποιότητας της υπηρεσίας και της απόδοσης SLA σε υψηλά επίπεδα. Επιπρόσθετα, τα κύρια χαρακτηριστικά των υπολογιστικών νεφών, όπως η δυνατότητα ελαστικότητας και συνεχούς διαθεσιμότητας, προϋποθέτει την ύπαρξη ενός ικανοποιητικού «στόλου» από

εξυπηρετητές και δικτυακό εξοπλισμό που να εξασφαλίζει ότι το σύνολο των εφαρμογών ενός μεγάλου αριθμού χρηστών να λειτουργεί ομαλά.

Μέσα από τη χρήση και λειτουργία υπερσύγχρονων υπολογιστικών κέντρων, από παρόχους υπηρεσιών νέφους, προκύπτει το θέμα της ενεργειακής αποδοτικότητας το οποίο αποκτά ανησυχητικές διαστάσεις. Στο σύνολο τους, τα υπολογιστικά κέντρα καταναλώνουν το 2% της παγκόσμιας παραγόμενης ενέργειας, όταν την ίδια στιγμή η αεροναυτιλία καταναλώνει το 2%-2,5%¹⁰. Το τεράστιο αυτό ποσοστό κατανάλωσης ενέργειας, προκύπτει από τη συνεχή λειτουργία ενός μεγάλου αριθμού εξυπηρετητών. Παρ' όλα αυτά, οι ανάγκες των χρηστών δεν ανταποκρίνονται σε αυτό το προφίλ λειτουργίας. Η μέση χρησιμοποίηση των υπολογιστικών κέντρων παγκοσμίως κυμαίνεται από 10-15%. Αυτό σημαίνει, ότι κατά μέσο όρο για κάθε εξυπηρετητή που βρίσκεται σε λειτουργία, μόνο το 10-15% των πόρων του χρησιμοποιείται για την εξυπηρέτηση του φόρτου εργασίας χρηστών υπηρεσιών νέφους¹¹. Στην Πίνακα 1 και στην Εικόνα 12 φαίνεται οι απαιτήσεις σε ισχύ ενός εξυπηρετητή IBM x3550 M4, ανάλογα με τη χρησιμοποίηση του συστήματος σύμφωνα με τις μετρήσεις SPECpower_ssj2008 [78].

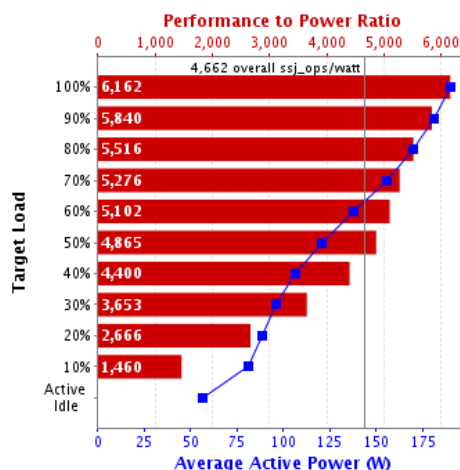
Πίνακας 1 Απαιτήσεις ισχύος ενός εξυπηρετητή IBM x3550 M4 ανάλογα με τη χρησιμοποίηση πόρων συστήματος¹²

Χρησιμοποίηση πόρων Συστήματος (%)	Ισχύς (W)
99,8	190
90,2	181
80	170
70,1	156
59,9	138
50	120
39,9	106
29,9	95,9
20,1	88,2
10,1	80,9
Άεργος (Idle)	56,5

¹⁰ <http://www.economist.com/technology-quarterly/2016-03-12/after-moores-law>

¹¹ http://www.computerweekly.com/news/450296750/Datacentre-energy-consumption-habits-overlooked-by-campaigners?utm_campaign=cw_uk&utm_medium=social&utm_source=twitter&utm_content=1464024680

¹² https://www.spec.org/power_ssj2008/results/res2012q3/power_ssj2008-20120807-00523.html



Εικόνα 12 Απόδοση ισχύος εξυπηρετητή IBM x3550 M4¹²

Όπως φαίνεται και στα παραπάνω γραφήματα, από άποψη ενεργειακής αποδοτικότητας, είναι προτιμότερο να λειτουργεί ένας εξυπηρετητής με φόρτο εργασίας που να ισοδυναμεί σε κατανάλωση του 80% των πόρων του, παρά δύο εξυπηρετητές που να έχουν φόρτο εργασίας που να καταναλώνει το 40% των πόρων τους.

Βεβαίως, με τη πρόοδο της τεχνολογίας υλισμικού, οι εξυπηρετητές θα είναι πιο ενεργειακά αποδοτικοί, απαιτώντας λιγότερη ισχύ σε κατάσταση αεργίας. Η λειτουργία, όμως, όσο το δυνατόν λιγότερων εξυπηρετητών, παραμένει ως η επικρατούσα στρατηγική για τους παρόχους υπηρεσιών νέφους. Συνεπώς, κύριος στόχος είναι η παροχή των ίδιων λειτουργιών και της ίδια ποιότητας υπηρεσίας χρησιμοποιώντας τον μικρότερο δυνατό αριθμό εξυπηρετητών.

Στο επίκεντρο των μοντέλο υπηρεσία νέφους βρίσκεται η λειτουργία εικονικών μηχανημάτων, τα οποία αποτελούν περιβάλλοντα τα οποία προσομοιώνουν ένα πλήρες σύστημα με το δικό του λειτουργικό χρησιμοποιώντας ένα μέρος των πόρων του εξυπηρετητή. Η τεχνική της εικονικοποίησης, επιτρέπει τη κατάτμηση των πόρων ενός εξυπηρετητή και τη προσφορά τους ως ανεξάρτητα λειτουργικά συστήματα κατάλληλα για οποιαδήποτε υπολογιστική χρήση. Τα εικονικά μηχανήματα, είτε ως το άμεσο προσφερόμενο μέσο σε μοντέλα IaaS, είτε ως υποστηρικτική υποδομή σε μοντέλα PaaS και IaaS, είναι η μορφή λογισμικού που λειτουργεί στους εξυπηρετητές ενός υπολογιστικού κέντρου.

Προκύπτει, συνεπώς, η ανάγκη για μία αποδοτική κατανομή των εικονικών μηχανημάτων ώστε να εξασφαλίζεται η αποδοτική λειτουργία των εξυπηρετητών. Επιπρόσθετα, εκτός από τη αποδοτική κατανομή των εικονικών μηχανημάτων, μία ιδιαίτερα ενδιαφέρουσα προσέγγιση εντοπίζεται στη χρησιμοποίηση μίας διαφορετικής

τεχνολογίας εικονικοποίησης. Οι περιέκτες λογισμικού (Software Containers) επιτρέπουν το ίδιο επίπεδο εικονικοποίησης προς τους χρήστες, καταναλώνοντας λιγότερους πόρους από το υπερκείμενο λειτουργικό σύστημα [79].

Παρ' όλο που η μέση χρησιμοποίηση των υπολογιστικών κέντρων παγκοσμίως παρατηρείται στο 10-15%, μεγάλοι πάροχοι υπηρεσιών νέφους, όπως η Netflix και η Google, έχουν καταφέρει να αυξήσουν τη μέση χρησιμοποίηση των υπολογιστικών τους κέντρων στα επίπεδα του 50-70%¹³. Αυτό σημαίνει πως είναι δυνατόν να υπάρξουν αποδοτικές και ανταγωνιστικές πολιτικές διαχείρισης του στρώματος λογισμικού ώστε να επιτευχθεί μία πιο οικολογική προσέγγιση στην πορεία και εξέλιξη των υπολογιστικών νεφών.

2.5.1 Υπάρχουσες λύσεις

Ένα συνήθης στόχος των παρόχων υπηρεσιών νέφους είναι η εξοικονόμηση ενέργειας επιτυγχάνοντας την ίδια ποιότητα υπηρεσία χρησιμοποιώντας όσον το δυνατόν λιγότερους εξυπηρετητές για την παροχή των υπηρεσιών αυτών. Ένα κλασσικό φαινόμενο το οποίο χαρακτηρίζει την μη-οικονομική λειτουργία των υπολογιστικών κέντρων είναι το φαινόμενο της υπό-χρησιμοποίησης πόρων, το οποίο παρατηρείται όταν ένας εξυπηρετητής λειτουργεί χρησιμοποιώντας μόνο ένα μικρό ποσοστό των πόρων. Στη προσπάθεια για την επίλυση του προβλήματος αυτού, η κατανομή των εικονικών μηχανημάτων με τέτοιο τρόπο ώστε να επιτυγχάνεται καλύτερη χρησιμοποίηση των εξυπηρετητών (50-60%), και συνεπώς να χρειάζεται να είναι όσο το δυνατόν λιγότεροι εξυπηρετητές ενεργοί κάθε στιγμή.

2.5.1.1 Μηχανισμοί κατανομής προσαρμοσμένοι στην ενεργειακή αποδοτικότητα

Το πρόβλημα κατανομής των εικονικών μηχανημάτων σε εξυπηρετητές εντοπίζεται να έχει πολλές ομοιότητες με το γνωστό μη-ντετερμινιστικό πολυωνυμικού χρόνου (NP-Hard) πρόβλημα Bin-Packing, όπου μία σειρά αντικειμένων πρέπει να κατανεμηθούν σε «δοχεία», με τέτοιο τρόπο ώστε κάθε στιγμή να ικανοποιούνται μία σειρά από συνθήκες (constraints) και στο τέλος να βελτιστοποιείται μία αντικειμενική συνάρτηση. Στη περίπτωση της προσανατολισμένης προς την ενεργειακή αποδοτικότητα κατανομή των εικονικών μηχανημάτων σε εξυπηρετητές, ο αντικειμενικός στόχος είναι η ελαχιστοποίηση των ενεργών εξυπηρετητών.

Στη λογική που περιεγράφηκε παραπάνω, οι χρονοπρογραμματιστές (schedulers) που αναλαμβάνουν τη κατανομή χρησιμοποιούν συγκεκριμένους

¹³ http://www.computerweekly.com/news/450296750/Datacentre-energy-consumption-habits-overlooked-by-campaigners?utm_campaign=cw_uk&utm_medium=social&utm_source=twitter&utm_content=1464024680

αλγορίθμους που επιλύουν το παραπάνω πρόβλημα. Ενώ ως κύριος στόχος παραμένει πάντα η ελαχιστοποίηση των ενεργών εξυπηρετητών, οι διαχειριστές της υποδομής θα πρέπει να λαμβάνουν υπόψιν τους και περαιτέρω παράγοντες όπως η πολυπλοκότητα και ο χρόνος εκτέλεσης του αλγορίθμου.

Μία από τις διαδεδομένες ομάδες αλγορίθμων είναι αυτή των άπληστων αλγορίθμων, οι οποίες χαρακτηρίζονται από την χαμηλή πολυπλοκότητα τους και την χαμηλή απαίτηση σε χρόνο όσον αφορά την εκτέλεση τους. Δύο χαρακτηριστικά παραδείγματα άπληστων αλγορίθμων είναι οι First-Fit και First-Fit Decreasing. Ο First-Fit ουσιαστικά καταναίμει τα εικονικά μηχανήματα στον πρώτο εξυπηρετητή που μπορεί να το φιλοξενήσει. Αντίστοιχα, ο First-Fit Decreasing ακολουθεί την ίδια διαδικασία, αφού πρώτα ταξινομήσει τα εικονικά μηχανήματα με βάση τις απαιτήσεις σε πόρους. Ο αλγόριθμος First-Fit Decreasing, παρουσιάζει ελαφρώς καλύτερα αποτελέσματα, καθώς η ταξινόμηση με βάση τις απαιτήσεις σε πόρους (κατά φθίνουσα σειρά) επιτρέπει μία πιο στοχευμένη κατανομή που επιτυγχάνει μία πιο οικονομική χρήση των εξυπηρετητών.

Συχνά, μέρος της διαδικασίας κατανομής αποτελεί και η μεταφορά ενός εικονικού μηχανήματος που βρίσκεται ήδη σε λειτουργία από έναν εξυπηρετητή σε έναν άλλο. Η διαδικασία της «μετανάστευσης» (migration) εικονικών μηχανημάτων είναι μία παράμετρος που πρέπει να λαμβάνεται υπόψιν κατά τη διαδικασία κατανομής [5]. Υπάρχουν δύο ήδη «μετανάστευσης»:

- *Μετανάστευση μετά από παύση λειτουργίας*, όπου περιλαμβάνει τη προσωρινή διακοπή λειτουργίας του εικονικού μηχανήματος και τη μεταφορά του στιγμιότυπου του στον νέο εξυπηρετητή. Αποτελεί την απλούστερη μορφή ανακατανομής, η οποία όμως επηρεάζει την απόδοση SLA, καθώς το εικονικό μηχανήμα θα παραμείνει εκτός λειτουργίας για όσο χρονικό διάστημα ανάλογο με το μέγεθος του στιγμιότυπου
- *«Ζωντανή» Μετανάστευση*, όπου το μηχανήμα μεταφέρεται χωρίς να διακοπεί η λειτουργία του. Στη πραγματικότητα, το στιγμιότυπο του εικονικού μηχανήματος αντιγράφεται στον νέο εξυπηρετητή. Όταν ολοκληρωθεί η αντιγραφή, διακόπτεται η λειτουργία του παλαιού εικονικού μηχανήματος και εκκινεί το νέο. Η «ζωντανή μετανάστευση» επιτρέπει μηδαμινή απουσία υπηρεσίας, επηρεάζει όμως την απόδοση του εικονικού μηχανήματος (performance degradation) κατά τη διαδικασία της αντιγραφής του στιγμιότυπου του στο νέο περιβάλλον

Οι παραπάνω αλγόριθμοι προσαρμόζονται ώστε να εξυπηρετήσουν τις ανάγκες της υποδομής. Σε απλές περιπτώσεις ακολουθείται η μέθοδος του στατικού κατωφλίου

χρησιμοποίησης (static utilization threshold), όπου ορίζεται το ανώτατο όριο χρησιμοποίησης πόρων ενός εξυπηρετητή, πέρα απ' το οποίο ο εξυπηρετητής θεωρείται ότι υπέρ-χρησιμοποιείται και αποκλείεται από οποιαδήποτε διαδικασία κατανομής εικονικών μηχανημάτων [5].

2.5.1.1.1 «Άπληστοι» αλγόριθμοι

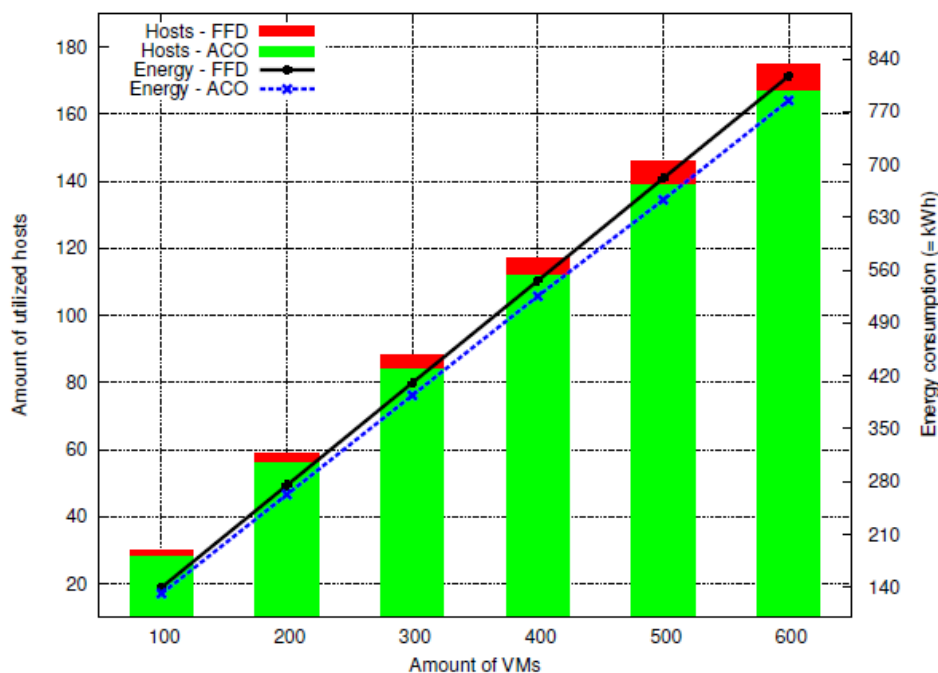
Πέρα από τον αλγόριθμο κατανομής, ιδιαίτερο ενδιαφέρον παρουσιάζουν και οι στρατηγικές που ορίζουν το πώς και το πότε πραγματοποιούνται οι κατανομές. Ένα κρίσιμο σημείο που επηρεάζει σημαντικά τον τρόπο λειτουργία ενός χρονοπρογραμματιστή υποδομής είναι ο ορισμός των σχετικών κριτηρίων. Συχνά, η διαδικασία της κατανομής εκκινεί έπειτα από συλλογή δεδομένων λειτουργίας της υποδομής, όπου με τη βοήθεια των κατάλληλων μηχανισμών εποπτείας, όταν παρατηρούνται φαινόμενα υπό η υπέρ-χρησιμοποίησης της υποδομής. Επιπρόσθετα, κριτήριο αποτελεί και η συμπεριφορά των εικονικών μηχανημάτων όσον αφορά τη ρεαλιστική κατανάλωση πόρων, όπου πάλι με τους κατάλληλους μηχανισμούς εποπτείας, η κατανομή γίνεται με βάση τη μέση χρησιμοποίηση πόρων ή τη μέγιστη χρησιμοποίηση πόρων που παρατηρήθηκε κατά τη διάρκεια μίας προκαθορισμένης χρονικής περιόδου [80].

2.5.1.1.2 Αλγόριθμος Ant Colony

Ενώ, οι άπληστοι αλγόριθμοι αποτελούν μία πολύ διαδεδομένη προσέγγιση, δεν φημίζονται για την ποιότητα των λύσεων που παράγουν. Όσον αφορά τη ενεργειακή αποδοτικότητα, παρατηρείται πως οι άπληστοι αλγόριθμοι σπαταλούν αρκετούς πόρους [81]. Μία εναλλακτική προσέγγιση του ζητήματος, παρατηρείται μέσα από τη πρόταση και τη μελέτη μέτα-ευριστικών αλγορίθμων. Οι μέτα-ευριστικοί αλγόριθμοι επιτυγχάνουν καλύτερα αποτελέσματα επιτρέποντας τη επεξεργασία πολλαπλών γειτονικών λύσεων με τη χρήση στοχαστικών διαδικασιών. Ο βασικός τρόπος λειτουργίας των αλγορίθμων αυτών είναι ο εξής: αρχικά παράγεται μία ομάδα λύσεων με τη χρήση ενός άπληστου αλγορίθμου. Στη συνέχεια, με τη χρήση στοχαστικών διαδικασιών εξετάζονται γειτονικές λύσεις και επιλέγονται οι καταλληλότερες που βελτιστοποιούν την αντικειμενική συνάρτηση. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να παρατηρηθεί σύγκλιση, όταν πια δηλαδή, οι νέες λύσεις δεν θα είναι ανώτερης ποιότητας από τις αμέσως προηγούμενες. Δύο ευρέως διαδεδομένες ομάδες αλγορίθμων που έχουν μελετηθεί και παρουσιαστεί στην σχετική βιβλιογραφία είναι οι αλγόριθμοι Ant Colony, και οι γενετικοί. Προκύπτουν από τη μελέτη και τη μοντελοποίηση διαδικασιών που έχουν παρατηρηθεί στη φύση (bio-inspired).

Η αρχική έκδοση του αλγορίθμου Ant Colony, προτάθηκε από τον Dorigo το 1992 [82] και αποτέλεσε μία καινοτόμο λύση στο πρόβλημα βελτιστοποίησης του πλανόδιου πωλητή (TSP). Ο αλγόριθμος μοντελοποιεί την λειτουργία αποικίας μυρμηγκιών κατά τη διαδικασίας εύρεσης τροφής. Βασικό στοιχείο του αλγορίθμου Ant Colony είναι η ανεξάρτητη λειτουργία μίας ομάδας «εργατών». Κάθε εργάτης αναλαμβάνει να παράξει μία λύση σχετικά με την κατανομή των εικονικών μηχανημάτων σε εξυπηρετητές [83] [81]. Όταν όλοι εργάτες ολοκληρώσουν τη εκτέλεση, από τη κύρια διαδικασία του αλγορίθμου, επιλέγεται η βέλτιστη λύση, στην οποία εναποτίθεται «φερορμόνη», ενισχύεται δηλαδή ένα ειδικό βάρος που οδηγεί τους «εργάτες» στο να προτιμήσουν αυτή τη λύση. Το δεύτερο χαρακτηριστικό του αλγορίθμου, εντοπίζεται στο στοχαστικό τρόπο με τον οποίο οι εργάτες επιλέγουν μέρη μίας λύσης. Ενώ ο καθοδηγητικός παράγοντας για την επιλογή εξυπηρετητών είναι η ποσότητα «φερορμόνης», μία τυχαία μεταβλητή μπορεί να οδηγήσει τον εργάτη να επιλέξει μία άλλη «οδό» για την κατανομή του εικονικού μηχανήματος, ώστε τελικά να οδηγηθεί σε μία γειτονική συνολική λύση. Με αυτό τον τρόπο αποκτά ο αλγόριθμος τον ευριστικό του χαρακτήρα. Για να επιτραπεί ακόμα μεγαλύτερη του χώρου λύσεων, η ποσότητα «φερορμόνης» εξασθενεί με την πάροδο επαναλήψεων της κύριας διαδικασίας του αλγορίθμου, καθιστώντας τη λύση λιγότερο «ελκυστική» με την πάροδο του χρόνου σε περίπτωση που δεν επιλέγεται ως βέλτιστη. Ο αλγόριθμος τερματίζει μετά από πεπερασμένο αριθμό εκτελέσεων της διαδικασίας εύρεσης λύσης από τους εργάτες ή εάν ο αλγόριθμος για συγκεκριμένο αριθμό επαναλήψεων έχει επέλθει σε σύγκλιση, όταν δηλαδή, δεν έχει βρεθεί μία ποιοτικότερη λύση. Ένα από τα κύρια πλεονεκτήματα του αλγορίθμου Ant Colony είναι το υψηλό επίπεδο παραμετροποίησης, όπου δίνεται η δυνατότητα στον διαχειριστή να ορίσει τον αριθμό επαναλήψεων και εργατών, τη ποσότητα και το ρυθμό εξασθένησης «φερορμόνης», τη βαθμό τυχειότητας στην επιλογή λύσεων από τους εργάτες, και το όριο σύγκλισης του αλγορίθμου. Έτσι, δίνεται η δυνατότητα ρύθμισης της λειτουργίας του ώστε να επιτυγχάνεται και ικανοποιητική ποιότητα λύσεων, αλλά και ικανοποιητικός χρόνος εκτέλεσης. Επίσης, καθώς η λειτουργία της ρουτίνας κάθε εργάτη είναι τελείως ανεξάρτητη από τη λειτουργία των υπολοίπων, δίνεται η δυνατότητα στον αλγόριθμο να εκτελεί παράλληλα τη λειτουργία των εργατών [84]. Αντίθετα, ακόμα και αν ρυθμιστεί με έμφαση στο χρόνο εκτέλεσης, ο αλγόριθμος Ant Colony είναι λιγότερο αποδοτικός όσον αφορά τη πολυπλοκότητα και τις απαιτήσεις σε επεξεργαστική ισχύ. Σε περιπτώσεις όπου ο αριθμός των εικονικών μηχανημάτων και εξυπηρετητών είναι αρκετά μεγάλος, ο μεγάλος χρόνος που θα απαιτηθεί για να επιτευχθεί σύγκλιση του αλγορίθμου, μπορεί να αποτελέσει αποτρεπτικό παράγοντα για τη επιλογή αλγορίθμου

κατανομής. Στην Εικόνα 13 φαίνεται η ενεργειακή αποδοτικότητα μίας έκδοσης του αλγορίθμου σε σχέση με τον «άπληστο» αλγόριθμο First-Fit Decreasing (FFD) [81].



Εικόνα 13 Ενεργειακή αποδοτικότητα του αλγορίθμου Ant Colony σε σχέση με τον First-Fit Decreasing [81]

2.5.1.1.3 Γενετικοί αλγόριθμοι

Η έτερη πιο διαδεδομένη ομάδα αλγορίθμων βελτιστοποίησης που ανήκει στην οικογένεια των μέτα-ευριστικών είναι οι γενετικοί αλγόριθμοι. Η πρώτη έκδοση γενετικού αλγορίθμου προτάθηκε από τον Holland το 1992 [85]. Η λειτουργία του αλγορίθμου προσεγγίζει τη φυσική διαδικασία της εξέλιξης μέσω της αναπαραγωγής και μετάλλαξης. Οι πιθανές λύσεις κωδικοποιούνται με τη μορφή σειράς χρωμοσωμάτων, ζεύγη δηλαδή εικονικών μηχανημάτων/εξυπηρετητών που παρομοιάζονται με μία αλληλουχία DNA. Αρχικά, ο αλγόριθμος χρησιμοποιεί μία αρχική οικογένεια λύσεων, οι οποίες έχουν παραχθεί συνήθως μέσω «άπληστων» αλγορίθμων (FF, FFD). Στη συνέχεια, ο αλγόριθμος επιλεγεί έναν προκαθορισμένο αριθμό ζευγών-γονέων. Ο αλγόριθμος έχει την τάση να επιλέγει τους πιο «ικανούς» γονείς (τις καλύτερες λύσεις), επιτρέποντας όμως την επιλογή λιγότερο ικανών για να επιτρέψει την καλύτερη δυνατή κάλυψη του χώρου των λύσεων. Από τα επιλεγμένα ζεύγη προκύπτουν απόγονοι, μέσω του συνδυασμού χρωμοσωμάτων του ενός γονέα με του άλλου. Οι απόγονοι, εφόσον αποτελούν εφικτές λύσεις, υπάρχει η πιθανότητα να μεταλλαχθούν, να υποστούν δηλαδή, μία τυχαία αλλαγή σε ένα χρωμόσωμα. Οι έγκυρες λύσεις-απόγονοι τοποθετούνται στην αρχική «οικογένεια» λύσεων και η διαδικασία επαναλαμβάνεται έως ότου να τερματίσει ο αλγόριθμος, οπότε και επιστρέφεται η καλύτερη λύση. Οι

γενετικοί αλγόριθμοι χαρακτηρίζονται από την αποδοτικότητα τους σε προβλήματα κατανομής που στοχεύουν στην ενεργειακή αποδοτικότητα. Όπως και στην περίπτωση του Ant Colony, οι γενετικοί αλγόριθμοι επιτρέπουν υψηλά επίπεδα παραμετροποίησης ώστε να βρεθεί η «χρυσή» τομή μεταξύ χρόνου εκτέλεσης και ποιότητας της τελικής λύσης. Στη βιβλιογραφία έχει μελετηθεί πληθώρα παραλλαγών γενετικών αλγορίθμων, όπου προτείνονται διαφορετικές τεχνικές κωδικοποίησης χρωμοσωμάτων [86] [87] [88], καθώς επίσης στρατηγικές αρχικοποίησης της οικογένειας των λύσεων [89], αλλά και παράλληλη και κατανεμημένη λειτουργία του αλγορίθμου [90].

1	2	3	4	5	6	7	8	9	← Εικονικό μηχάνημα
4	5	1	1	2	2	5	3	4	← Εξυπηρετητής

Εικόνα 14 Παράδειγμα κωδικοποίησης λύσης ενός γενετικού αλγορίθμου

2.5.1.2 Εξοικονόμηση πόρων με ευέλικτη εικονικοποίηση

Όπως περιεγράφηκε και στην προηγούμενη υποενότητα, εντοπίζεται στη βιβλιογραφία διάφορες τεχνικές κατανομής και οργάνωσης των εικονικών συστοιχιών που είτε προσφέρονται απευθείας στους χρήστες, είτε χρησιμοποιούνται ως βάση για υπηρεσίες PaaS και SaaS από τους παρόχους. Επιπρόσθετα, οι τεχνολογία υλισμικού καθώς και η τεχνολογίες υπερεπόπτη (KVM, Xen κλπ.) έχουν εξελιχθεί σε ένα ώριμο στάδιο όπου επιτρέπει καλύτερη αποδοτικότητα στη διαχείριση των πόρων συστήματος. Τα νέα μοντέλα επεξεργαστών, μνημών, σκληρών δίσκων και τροφοδοτικών απαιτούν λιγότερη ισχύ σε καταστάσεις αεργίας (idle state), ενώ η απαίτηση αυτή κλιμακώνει καλύτερα με την κατανάλωση της επεξεργαστικής ισχύος. Οι τεχνολογίες υπερεπόπτη από την άλλη, έχουν τη δυνατότητα να καταλαμβάνουν και να κατανέμουν απευθείας πόρους του συστήματος προς τα λειτουργικά συστήματα και επιμέρους διεργασίες των εικονικών μηχανημάτων, με τις τεχνικές pinning και pci passthrough, παρακάμπτοντας το ενδιάμεσο επίπεδο επικοινωνίας του υπερεπόπτη.

Ακόμα όμως και με την εμφανή πρόοδο των ανωτέρω τεχνολογιών, η χρήση των εικονικών μηχανημάτων σε εκτεταμένη κλίμακα δεν είναι αποδοτική. Η επιβάρυνση που επιβάλλει ο υπερεπόπτης στο υπερκείμενο σύστημα συσσωρευτικά αποτελεί έναν σημαντικό παράγοντα σπατάλης πόρων, ενώ η εν γένει χρήση των εικονικών μηχανημάτων απαιτεί χρήση επιπλέον αποθηκευτικού χώρου για τη διατήρηση ενός ολόκληρου συστήματος. Ένας επιπλέον λόγος σπατάλης πόρων είναι υπό-χρησιμοποίηση των εικονικών μηχανημάτων. Τα εικονικά μηχανήματα καταλαμβάνουν

πόρους επεξεργαστή (πυρήνες ή επιμέρους νήματα) και μνήμης, οι οποίοι δεν είναι διαθέσιμοι για τις υπόλοιπες διεργασίες του υπερκείμενου λειτουργικού συστήματος. Όταν αυτοί οι πόροι δεν χρησιμοποιούνται πλήρως από τις διεργασίες του εικονικού μηχανήματος, συσσωρευτικά ο εξυπηρετητής μπορεί να βρεθεί σε μία κατάσταση όπου μεγάλο μέρος των πόρων του είναι σε κατάσταση αεργίας.

Το φαινόμενο αυτό παρατηρείται σε υποδομές υπολογιστικών νεφών, λόγω του μεγάλου αριθμού εικονικών μηχανημάτων τα οποία λειτουργούν σε καθημερινή βάση. Επίσης, οι time-shared στρατηγικές που χρησιμοποιούνται για να επιτρέψουν τη κατάτμηση των πυρήνων ενός επεξεργαστή (συνήθως σε μία κλίμακα 16:1) ώστε να είναι δυνατόν να λειτουργούν περισσότερα εικονικά μηχανήματα ανά εξυπηρετητή, αναιρούν τα βελτιώσεις στις τεχνολογίες υπερεπόπτη (cpu pinning και pci passthrough).

Η λύση που φαίνεται ότι επικρατεί και προτείνει μία καινοτόμα λύση στο ζήτημα της σπατάλης πόρων, είναι η μετάβαση του μοντέλου εικονικοποίησης από τεχνολογία υπερεπόπτη (hypervisor-based) σε τεχνολογία περιεκτών (container-based). Οι περιέκτες λογισμικού αποτελούν (βλ. Κεφ. 2.1.3.2) μια πολύ ευέλικτη τεχνική εικονικοποίησης δημιουργώντας ένα εικονικό περιβάλλον που περιέχει τις απολύτως απαραίτητες εξαρτήσεις για τη λειτουργία μίας εφαρμογής. Το κύριο πλεονέκτημα των περιεκτών είναι ότι σε σχέση με τα εικονικά μηχανήματα υπερεπόπτη, η επιβάρυνση του συστήματος για τη λειτουργία του υπερεπόπτη (overhead) είναι μηδαμινή [79]. Σε σχέση με τα εικονικά μηχανήματα υπερεπόπτη, οι περιέκτες λογισμικού δεν χρησιμοποιούνται (προς το παρόν) για γενική χρήση, συνεπώς η χρήση του περιορίζεται στη δημιουργία υποδομής για υπηρεσίες PaaS και SaaS. Η μελέτη κατανομής και διαχείρισης περιεκτών λογισμικού σε υπηρεσίες PaaS μελετάται στην εργασία [4], όπου γίνεται χρήση ομάδας μέτα-ευριστικών αλγορίθμων για τη κατανομή των περιεκτών λογισμικού, με βάση την ενεργειακή αποδοτικότητα και τη απόδοση SLA. Η αποδοτική διαχείριση πόρων, ακόμα και όταν η εικονική υποδομή αποτελείται από περιέκτες λογισμικού, είναι ένα μείζον ζήτημα, κυρίως σε περιπτώσεις όπως αυτής Google, όπου περίπου 2 δις. περιέκτες¹⁴ εκκινούν καθημερινά για να υποστηρίξουν τις διάφορες υπηρεσίες PaaS που προσφέρει. Οι περιέκτες λογισμικού αποτελούν το κατάλληλο μέσο για τη λειτουργία κατανεμημένων ενορχηστρωμένων υπηρεσιών. Η χαμηλή κατανάλωση πόρων σε συνδυασμό με την σχετικά άμεση έναρξη λειτουργίας τους σε σχέση με τα παραδοσιακά εικονικά μηχανήματα υπερεπόπτη, έχει οδηγήσει στην υιοθέτηση νέων μηχανισμών ενορχήστρωσης όπως το Kubernetes [37] και το Mesos [91], οι οποίες παρέχουν μηχανισμούς κλιμάκωσης και αυτοίασης, αλλά και λειτουργικών συστημάτων, όπως το

¹⁴ <https://opensource.googleblog.com/2014/06/>

CoreOS [92]. Ακόμα, και όταν η υποδομή περιεκτών δεν λειτουργεί απευθείας στους εξυπηρετητές (native), αλλά σε ένα ενδιάμεσο στρώμα εικονικών μηχανημάτων υπερεπόπτη, η εξοικονόμηση πόρων είναι εμφανής. Όπως παρουσιάζεται σε μελέτη της VMware¹⁵, η λειτουργία εφαρμογών Web χρησιμοποιώντας περιέκτες λογισμικού επάνω από ένα στρώμα εικονικών μηχανημάτων, επιτυγχάνει παρόμοια απόδοση με περίπτωση των εφαρμογών σε περιέκτες που λειτουργούν απευθείας στους εξυπηρετητές.

2.5.2 Τρέχοντα θέματα

Η χρήση περιεκτών λογισμικού υιοθετείται όλο και περισσότερο ως μία λύση που μπορεί να προσφέρει περισσότερη ευελιξία στους παρόχους υπηρεσιών νέφους, καθώς στη βιβλιογραφία η υποστηρικτική λειτουργία των περιεκτών εντοπίζεται και ως CaaS (Container as a Service) [93] [94]. Ταυτόχρονα επιτρέπει την μείωση του κόστους λειτουργίας της υποδομής, και συμβάλει σε μία πιο οικολογικά-φιλική (eco-friendly) προσέγγιση του τρόπου λειτουργίας των υπολογιστικών κέντρων παγκοσμίως.

Παρόλα αυτά, η χρήση εικονικών μηχανημάτων τεχνολογίας υπερεπόπτη παραμένει αναγκαία σε πολλές περιπτώσεις. Μία βασική περίπτωση παραμένει αυτή του μοντέλου IaaS, όπου για γενική χρήση τα εικονικά μηχανήματα προσφέρουν μία φιλικότερη χρήση προς τον τελικό χρήστη. Αντίθετα οι περιέκτες λογισμικού είναι κατάλληλοι για τη φιλοξενία μίας υπηρεσίας ή εφαρμογής (π.χ. έναν εξυπηρετητή εφαρμογών Web, μία υπηρεσία SSH, έναν πράκτορα λογισμικού), χωρίς αυτό να αποκλείει ότι μπορούν να χρησιμοποιηθούν ως ένα ξεχωριστό λειτουργικό περιβάλλον.

Επίσης, είναι σαφές πως μία από τις διαφορές ανάμεσα στις τεχνολογίες (βλ. Κεφ. 2.1.2) εικονικοποίησης είναι το επίπεδο απομόνωσης των διεργασιών που προσφέρεται από τις τεχνολογίες υπερεπόπτη, καθώς τα εικονικά μηχανήματα αποτελούν ένα κλειστό περιβάλλον προς το υπόλοιπο σύστημα. Το ζήτημα αυτό της ασφάλειας αποτελεί ένα ρυθμιστικό παράγοντα για τον οποίο ακόμα και σήμερα, για θέματα ποιότητας υπηρεσίας και SLA, τα εικονικά μηχανήματα προτιμώνται έναντι των περιεκτών λογισμικού.

Η ευελιξία των περιεκτών λογισμικού θα μπορούσε να βελτιώσει το μοντέλο παροχής υπηρεσιών IaaS. Οι περιέκτες λογισμικού δεν καταλαμβάνουν εξ' αρχής πόρους από το σύστημα. Οι περιέκτες λογισμικού ενθυλακώνουν μόνο το απαραίτητο λογισμικό που απαιτείται από μία εφαρμογή, συνεπώς μπορεί να ειπωθεί πως, ουσιαστικά, οι

¹⁵ <https://blogs.vmware.com/performance/2015/04/scaling-web-2-0-applications-using-docker-containers-vsphere-6-0.html>

περιέκτες λογισμικού καταναλώνουν μόνο τους πόρους που απαιτούνται κάθε στιγμή από την εφαρμογή. Προκύπτει λοιπόν η ανάγκη για τα ακόλουθα:

- Εφαρμογή των κατάλληλων ορίων κατανάλωσης πόρων για τους περιέκτες λογισμικού, μία λειτουργία που ήδη υποστηρίζεται από διάφορες τεχνολογίες περιεκτών λογισμικού όπως το Docker με την χρήση των Cgroups¹⁶
- Για την εφαρμογή των κατάλληλων ορίων που αναφέρθηκαν παραπάνω, χρειάζεται κατάλληλος μηχανισμός πρόβλεψης του φόρτου εργασίας και της δικτυακής κίνησης που θα επιβάλλει η εφαρμογή και η χρήση στο υπερκείμενο σύστημα. Συγκεκριμένοι αλγόριθμοι μηχανικής μάθησης έχουν προταθεί στη βιβλιογραφία, οι οποίοι παράγουν μοντέλα κατανάλωσης πόρων σε περιβάλλοντα IaaS, που προσφέρουν εικονικά μηχανήματα [95] [96]
- Υπάρχει ανάγκη για τη κατάλληλη προτυποποίηση στιγμιότυπων και εικόνων που να υποστηρίζουν τις απαραίτητες λειτουργίες για μία φιλικότερη προς το τελικό χρήστη χρήση του περιέκτη. Περιέκτες που θα δίνουν τη δυνατότητα στο χρήστη να τα χειριστεί απομακρυσμένα, ενδεχομένως μία διαχειριστική πλατφόρμα διαδικτύου ή πρωτόκολλα απομακρυσμένης σύνδεση σε τερματικό του περιέκτη (π.χ. SSH)

Επιτακτική είναι επίσης η ανάγκη για τη διαμόρφωση των περιβαλλόντων οποιασδήποτε μορφής με αυτοματοποιημένο τρόπο. Τεχνολογίες όπως το π.χ. cloud-init [97] ή το Ansible [6], δίνουν τη δυνατότητα στον χρήστη να ορίσει συγκεκριμένες λειτουργίες που επιθυμεί να εκτελεστούν κατά τη έναρξη της λειτουργίας του εικονικού μηχανήματος/περιέκτη, όπως μεταφόρτωση λογισμικού και διαμόρφωση του λειτουργικού συστήματος του εικονικού μηχανήματος. Η χρήση μηχανισμών αυτοματισμού (automation tools), μπορεί να συμβάλλει στη κατανάλωση λιγότερων αποθηκευτικών πόρων.

Για τη περίπτωση όπου η χρήση των εικονικών μηχανημάτων είναι επιβεβλημένη, υπάρχει περιθώριο βελτίωσης των ήδη υπάρχουσών λύσεων προσαρμοσμένων προς τη εξοικονόμηση πόρων. Παρατηρείται, πολλές φορές, η ανάγκη μετανάστευσης των εικονικών μηχανημάτων από έναν εξυπηρετητή προς έναν άλλον, ώστε να μειωθεί ο αριθμός των ενεργών εξυπηρετητών του υπολογιστικού κέντρου. Η διαδικασία μετανάστευσης δεν αποτελεί μία ιδιαίτερα αποδοτική λύση, καθώς, στη

¹⁶ <https://docs.docker.com/engine/docker-overview/#the-underlying-technology>

περίπτωση που αυτή απαιτείται χωρίς την παύση λειτουργίας του μηχανήματος (live migration), τότε επιφέρει και επιπρόσθετη κατανάλωση πόρων, αλλά και μείωση της ποιότητας της υπηρεσίας για ένα μεγάλο χρονικό διάστημα.

Το μεγαλύτερο ποσοστό των εφαρμογών που εκτελείται σε περιβάλλοντα που προσφέρονται μέσω του μοντέλου IaaS, παρουσιάζουν μία τυχαία και «εκρηκτική» κατανομή του φόρτου εργασίας τους. Το φαινόμενο αυτό οδηγεί σε μια κατάσταση μαζικής υπό-χρησιμοποίησης των πόρων του εικονικού μηχανήματος και, ως επακόλουθο, του εξυπηρετητή που το φιλοξενεί.

Ως εκ τούτου, προκύπτει ανάγκη για το χαρακτηρισμό της εφαρμογής που εκτελείται σε ένα εικονικό μηχάνημα, είτε μέσω ανάλυσης των δεδομένων κατανάλωσης πόρων, είτε με απευθείας εποπτεία των διεργασιών του εικονικού μηχανήματος (εφόσον για λόγους ασφαλείας αυτό είναι αποδεκτό από τον χρήστη). Ο χαρακτηρισμός του εικονικού μηχανήματος, με βάση τη χρήση του μπορεί να επιτρέψει στη διαχείριση της υποδομής στο να προσφέρει το εικονικό περιβάλλον με τη μορφή περιέκτη.

Η παρούσα διατριβή, μελετά μία τέτοια λειτουργία στα πλαίσια της αρχιτεκτονικής πρότασης για υποστήριξη υβριδικής εικονικοποίησης σε υποδομές υπολογιστικών νεφών. Στην περίπτωση χρήσης που εξετάζεται, η μορφή του εικονικού μηχανήματος καθορίζεται από τη χρήση, εντός ενός προτυποποιημένου μηνύματος κατά τη διάρκεια της αίτησης υπηρεσίας. Μία λειτουργία αυτόματου χαρακτηρισμού της λειτουργίας του εικονικού μηχανήματος, σε συνδυασμό με τις απαιτήσεις του χρήστη, θα επέτρεπε περαιτέρω επίπεδα ευελιξίας στο συγκεκριμένο ζήτημα.

2.5.3 Πρακτικές περιπτώσεις χρήσης

Στη παρούσα περίοδο, οι τεχνολογίες που υποστηρίζουν και καθιστούν εφικτή τη παροχή υπηρεσιών νέφους έχουν φτάσει σε ένα ικανοποιητικό επίπεδο ωριμότητας. Η εμπειρία, η γνώση αλλά και οι νέες συναφείς τεχνολογίες επιτρέπουν τη πλήρη παραμετροποίηση και ευελιξία σε μία υπολογιστική υποδομή, επιτρέποντας έτσι στους παρόχους να εγγυούνται τη τήρηση των συμφωνηθέντων SLAs, ποιότητα υπηρεσίας και ασφάλεια.

Το ζήτημα, όμως, του ενεργειακού αποτυπώματος που επιφέρει η λειτουργία υπολογιστικών κέντρων τεράστιας κλίμακας, αποκτά βαρύνουσα σημασία. Αυτό έχει ως αποτέλεσμα μεγάλοι πάροχοι υπηρεσιών νέφους να προβαίνουν στις απαραίτητες ενέργειες για την αντιμετώπιση της υπάρχουσας κατάστασης, η οποία αυτή τη στιγμή δεν είναι ιδιαίτερα αποδοτική. Επιπρόσθετα, υπάρχει πρόδος όσον αφορά την ανάπτυξη εμπορικών συστημάτων προσαρμοσμένα στην ενορχήστρωση των υπηρεσιών

μία υποδομής νέφους ώστε να γίνεται αποδοτικότερη και οικονομικότερη χρήση των πόρων του υπολογιστικού κέντρου.

Χαρακτηριστικό είναι το φαινόμενο μετάβασης των υπηρεσιών PaaS, από εικονικά μηχανήματα σε περιέκτες λογισμικού. Οι λόγοι είναι πολλαπλοί, όπου εκτός από τη αμεσότητα στην έναρξη και παραμετροποίηση, και τη μεγάλη κλιμακοσιμότητα, οι περιέκτες λογισμικού επιτυγχάνουν πάνω από 90% απόδοση όσον αφορά τη κατανάλωση πόρων. Συνεπώς, χρειάζονται λιγότεροι πόροι για τον ίδιο φόρτο εργασίας. Η εταιρία Pantheon [98], η οποία προσφέρει υπηρεσίες PaaS για την δημιουργία και διαχείριση εφαρμογών ιστού, στηρίζει την υποδομή των υπηρεσιών σε περιέκτες λογισμικού, επιτυγχάνοντας από 1000 φορές αποδοτικότητα στη διαχείριση των πόρων της υποδομής υλισμικού¹⁷.

Η Google έχει επιτύχει ένα επίπεδο ωφέλιμης χρησιμοποίησης των υπολογιστικών της κέντρων στο 50-70%¹⁸, γεγονός το οποίο της επιτρέπει να χρειάζεται λιγότερους ενεργούς εξυπηρετητές για τις ίδιες απαιτήσεις σε φορτία εργασίας που παράγονται από τη πληθώρα υπηρεσιών νέφους που προσφέρει. Κάθε εβδομάδα δημιουργούνται περίπου 2 δις. περιέκτες για τις ανάγκες των υπηρεσιών (σύμφωνα με αναφορά του 2014)¹⁹.

Πέρα όμως από την αξιοποίηση της ανερχόμενης τεχνολογίας των περιεκτών λογισμικού, η διαχείριση της εικονικής υποδομής ως σύνολο, αποτελεί κύριο μέλημα πολλών παρόχων υπηρεσιών νέφους. Το πρόγραμμα Αποκατάστασης και Φιλοξενίας Πόρων (Azure Site Recovery – ASR) της Microsoft [99] αποτελεί μία ολοκληρωμένη λύση για τις επιμέρους ομάδες της εταιρίας, ώστε αρχικά να ελέγχουν τη χρησιμοποίηση των εξυπηρετητών και στη συνέχεια να τους κατηγοριοποιούν σε επίπεδα, ανάλογα με το προφίλ χρήσης τους. Τα αποτελέσματα της εν λόγω ανάλυσης μπορούν να οδηγήσουν στην ανακατανομή ενός εικονικού μηχανήματος ή τη παύση λειτουργίας ενός εξυπηρετητή. Ο τελικός στόχος του προγράμματος είναι η ενεργοποίηση μίας πιο αποδοτικής χρήσης της υποδομής ώστε να μειωθεί το συνολικό αποτύπωμα από κατά τη χρήση της.

Η πλατφόρμα Eco4Cloud [100], αποτελεί μία ολοκληρωμένη λύση που προσφέρει τις απαραίτητες διεπαφές προς τους διαχειριστές υποδομών νέφους, και προσφέρουν τις απαραίτητες λειτουργίες για τη διαμόρφωση της εικονικής υποδομής με στόχευση στην ενεργειακή αποδοτικότητα. Η υπηρεσία Κατανομής Φόρτου Εργασίας (E4C Workload

¹⁷ <https://pantheon.io/blog/how-software-containers-can-save-more-co2-electric-cars>

¹⁸ http://www.computerweekly.com/news/450296750/Datacentre-energy-consumption-habits-overlooked-by-campaigners?utm_campaign=cw_uk&utm_medium=social&utm_source=twitter&utm_content=1464024680

¹⁹ <https://opensource.googleblog.com/2014/06/>

Consolidation), λειτουργεί πάνω από το υπάρχον επίπεδο εικονικοποίησης και προσφέρει λειτουργικότητες βελτιστοποίησης κατανομής φόρτου. Η τεχνική βασίζεται στη χρήση στατιστικών και προσαρμοστικών αλγορίθμων με στόχο την αύξηση της πυκνότητας των εικονικών μηχανημάτων ανά εξυπηρετητή από 40-150% και αύξηση της χρησιμοποίησης των εξυπηρετητών κατά 90%²⁰. Επίσης, η υπηρεσία Smart Ballooning προφέρει ευφυή διαχείρισης μνήμης εικονικών μηχανημάτων ESXi [101], απελευθερώνοντας όση μνήμη βρίσκεται σε κατάσταση αεργίας και δεν χρησιμοποιείται, ώστε ο υπερεπόπτης ESX να την ανακαταναείμει σε άλλα εικονικά μηχανήματα.

Τέλος, αξία αναφοράς είναι πλατφόρμα λογισμικού, για τη δημιουργία νεφών IaaS, Openstack [9]. Η ανοικτού κώδικα πλατφόρμα Openstack βασίζεται σε μία συλλογή υπηρεσιών που κάθε μία καλύπτει συγκεκριμένες ανάγκες της υποδομής. Κάθε υπηρεσία αποτελεί και ένα ανεξάρτητο έργο το οποίο συντηρείται από κοινότητες χρηστών που συμβάλλουν στην ανάπτυξη τους.

Η υπηρεσία με την κωδική ονομασία Watcher [102], προσφέρει μία ευέλικτη και κλιμακώσιμη υπηρεσία βελτιστοποίησης πόρων για υποδομές νεφών Openstack. Η υπηρεσία καλύπτει συνολικά την διαδικασία της βελτιστοποίησης, αρχικά με την συλλογή δεδομένων χρήσης της υποδομής (εικονικής και πραγματικής), και στη συνέχεια με τη διαδικασία κατανομής και χρονοπρογραμματισμού και στρατηγικού σχεδιασμού (action planner). Ως εκ τούτου, καθίσταται κατάλληλη για τη επίτευξη στρατηγικών στόχων της υποδομής, με την αύξηση της ενεργειακής αποδοτικότητας να είναι ένας από αυτούς.

Η υπηρεσίας Watcher εκτός των αλγορίθμων βελτιστοποίησης που προσφέρει εξ' αρχής, παρέχει και τις κατάλληλες διεπαφές για την ανάπτυξη παραμετροποιήσιμων αλγορίθμων, φίλτρων και μετρικών από τους ίδιους τους χρήστες της υποδομής, με σκοπό την πλήρη κάλυψη των αναγκών κάθε περίπτωσης χρήσης. Τέλος, η υπηρεσία προσφέρεται σε δύο μορφές λειτουργίας: α) την λειτουργία συμβουλευτικού τύπου, όπου παράγει ένα στρατηγικό πλάνο διαχείρισης και κατανομής των πόρων προς τους διαχειριστές, οι οποίοι λαμβάνουν την απόφαση για εφαρμογή του ή μη, και β) την αυτόματη λειτουργία, όπου το πλάνο παράγεται και εφαρμόζεται απευθείας από την υπηρεσίας, παρακάμπτοντας τον ανθρώπινο παράγοντα.

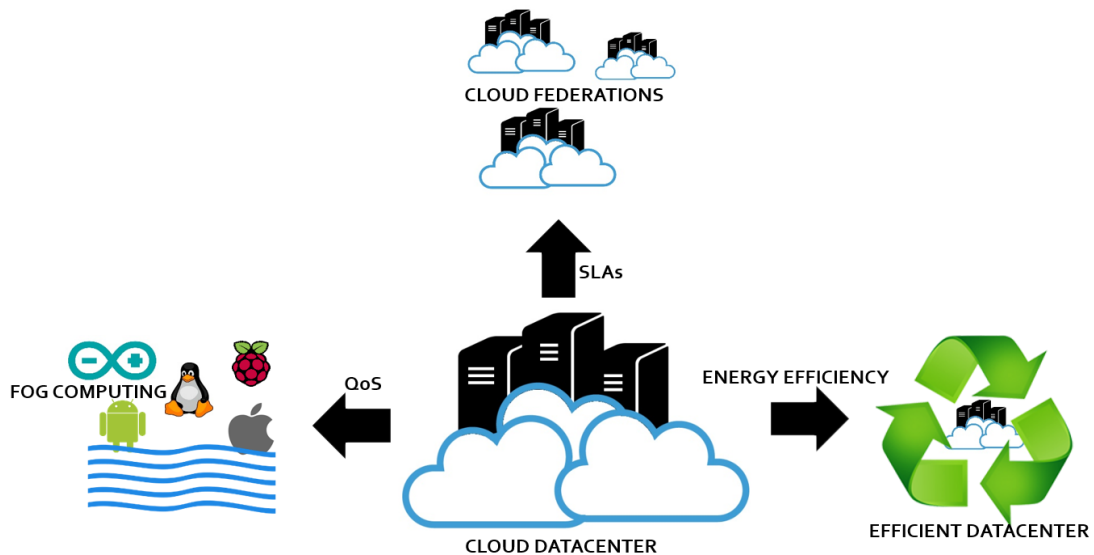
²⁰ <http://www.eco4cloud.com/wp-content/uploads/2016/05/E4C-Workload-Consolidation-Data-Sheet.pdf>

Κεφάλαιο 3

Αρχιτεκτονικές Ευφυών Συστημάτων Διαχείρισης Υποδομών Νεφελώδους Υπολογισμού

3.1 Περιγραφή και αρχιτεκτονική προσέγγιση

Η παρούσα διατριβή προτείνει αρχιτεκτονικές λύσεις που επιλύουν σημαντικά ζητήματα στη διαχείριση υποδομών που προσφέρουν υπηρεσίες νέφους. Κύριος σκοπός είναι η επέκταση των υπάρχουσών προσεγγίσεων και η επέκταση του υπολογιστικού κέντρου με τρόπο που να βελτιώνει τρεις άξονες που σχετίζονται με τη λειτουργία και τη προσφορά των υπηρεσιών. Η Εικόνα 15 παρουσιάζει τους τρεις αυτούς άξονες γύρω από τους οποίους αναπτύχθηκε η παρούσα μελέτη. Τα κύρια ζητούμενα που εξετάστηκαν ήταν α) η βελτιστοποίηση της παρεχόμενης ποιότητας της υπηρεσιών υπολογισμού προς πλατφόρμες και υπηρεσίες, κυρίως στα μοντέλα PaaS και SaaS, β) την τήρηση των συμφωνηθέντων SLA με τους χρήστες, και τέλος γ) την αύξηση της ενεργειακής αποδοτικότητας που προκύπτει συνολικά από τη χρήση κάθε είδους υπηρεσίας «νεφελώδους» υπολογισμού.



Εικόνα 15 Το υπολογιστικό κέντρο υπηρεσιών «νέφους» και οι τρεις άξονες επέκτασης

Κάθε ζητούμενο από τα τρία που αναφέρθηκαν ανωτέρω, εκφράζεται μέσα από πραγματικές περιπτώσεις χρήσης οι οποίες αναλύονται και εξετάζονται. Οι ανάγκες αυτές είναι α) δίκτυα και πλατφόρμες IoT, β) «ομοσπονδίες» υποδομών νέφους με τη δημιουργία συνεργατικών δικτύων αποτελούμενων από ανεξάρτητους παρόχους, και γ) η διαχείριση των πόρων ενός υπολογιστικού κέντρου μέσω ευφυών τεχνικών κατανομής και χρονοπρογραμματισμού για την εξοικονόμηση πόρων, ενέργειας και κόστους λειτουργίας. Η παρούσα διατριβή εξετάζει τη φύση των ζητημάτων αυτών και προτείνει επί μέρους προσεγγίσεις και τεχνικές που έχουν εφαρμογή στις συγκεκριμένες περιπτώσεις χρήσης, χωρίς όμως να περιορίζεται σε αυτές.

Η προτεινόμενη αρχιτεκτονική μπορεί να αναλυθεί σε τρεις επιμέρους. Κάθε προσέγγιση προτείνει βελτιώσεις σε υπάρχοντα μοντέλα. Συγκεκριμένα, όσον αφορά τη διαχείριση πόρων και υπηρεσιών, προτείνεται η επέκταση του μοντέλου του «νεφελώδους» υπολογισμού με τη χρησιμοποίηση του μοντέλου «ομιχλώδους» υπολογισμού με τη προσθήκη των κατάλληλων λειτουργιών και της κατάλληλης προτυποποίησης. Η λύση αυτή αξιολογείται σε υπηρεσίες IoT, που αποτελούν ένα χαρακτηριστικό παράδειγμα υπηρεσιών και εφαρμογών που έχουν ιδιαίτερα αυξημένες απαιτήσεις όσον αφορά τη ποιότητα της υπηρεσίας.

Σε δεύτερο στάδιο, προτείνεται αρχιτεκτονικό μοντέλο δημιουργίας «ομόσπονδων» δικτύων υπολογιστικών νεφών. Συγκεκριμένα, προτείνεται ένας καινοτόμος διαμεσολαβητής, στα πλαίσια μίας γενικής αρχιτεκτονικής «ομοσπονδίας» που επιτρέπει την διαλειτουργικότητα μεταξύ παρόχων υπηρεσιών νεφών. Η παρούσα προσέγγιση αξιολογεί την εν λόγω λειτουργία διαμεσολαβητή με βάση τη απόδοση των παρόχων υπηρεσιών σχετικά με τα συμβόλαια χρήσης (SLA) που να συνάπτουν με τους πελάτες.

Σε τρίτο και τελευταίο στάδιο, εξετάζεται η ενεργειακή αποδοτικότητα υπολογιστικών κέντρων που χρησιμοποιούνται από παρόχους υπηρεσιών υπολογισμού. Η χρήση των σύγχρονων υπολογιστικών κέντρων γίνεται με τρόπο μη αποδοτικό, κάτι που συνεπάγεται την αύξηση του κόστους λειτουργίας, την μείωση της απόδοσης των εξυπηρετητών που μπορούν να οδηγήσουν με τη σειρά τους σε μείωση της ποιότητας της υπηρεσίας και σε κακή απόδοση SLA. Η προτεινόμενη προσέγγιση προσπαθεί να αντιμετωπίσει το ζήτημα αυτό, παρουσιάζοντας μία πλατφόρμα διαχείρισης πόρων με τη ταυτόχρονη χρήση δύο τεχνολογιών εικονικοποίησης, των περιεκτών λογισμικού και των εικονικών μηχανημάτων υπερεπόπτη. Η χρήση των περιεκτών λογισμικού με ευφυή μέθοδο, μπορεί να προσφέρει ευελιξία στη παροχή πόρων για όλα τα μοντέλα υπηρεσιών

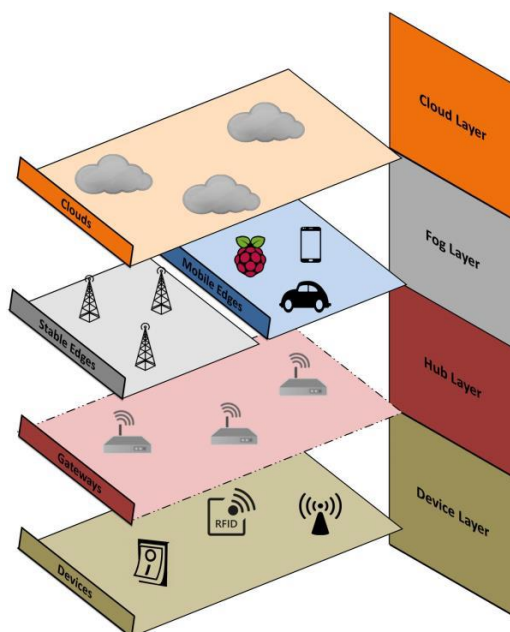
ώστε να εξοικονομούνται εξυπηρετητές και να μειώνεται συνολικά η συνολική κατανάλωση ενέργειας ενός υπολογιστικού κέντρου.

Στο κεφάλαιο αυτό περιγράφονται αναλυτικά οι προτεινόμενες λύσεις, η αρχιτεκτονική δομή τους και τα επιμέρους χαρακτηριστικά των λειτουργικών τμημάτων τους.

3.2 Συστήματα «νεφελώδους» και «ομίχλώδους» υπολογισμού με έμφαση στη ποιότητα υπηρεσίας

Η προτεινόμενη λύση «ομίχλώδους» υπολογισμού επικεντρώνεται στη δημιουργία δομών που θα επιτρέπουν, εκτός από τη παροχή πόρων για την άμεση επεξεργασία δεδομένων, τη επεκτασιμότητα των δικτύων αυτών έχοντας τη δυνατότητα να συνδεθούν με τρίτα γειτονικά ή ακόμα και με εξωτερικές συσκευές (π.χ. κινητά smartphones, έξυπνα αυτοκίνητα) λαμβάνοντας πάντα υπόψιν το χαρακτηριστικό της κινητικότητας και της αβεβαιότητας όσον αφορά τη παρουσία τους στο δίκτυο.

Για τη εύκολη επικοινωνία των δικτύων «ομίχλης» και λοιπών συσκευών, και σε αυτή τη περίπτωση χρησιμοποιείται η μέθοδος επικοινωνίας δημοσίευσης/συνδρομής (Pub/Sub) με το πρωτόκολλο MQTT [11]. Κάθε δίκτυο «ομίχλης» εκπροσωπείται αρχιτεκτονικά από μία συσκευή η οποία αναφέρεται και ως διαμεσολαβητής «ομίχλης» (Fog Broker). Το δίκτυο «ομίχλης» που βρίσκεται στη περιοχή του IoT δικτύου συσκευών ορίζεται ως σταθερό Fog (stable Fog). Το σταθερό δίκτυο «ομίχλης» δημιουργεί το κατάλληλο MQTT θέμα στο οποίο συνδέονται γειτονικά δίκτυα «ομίχλης» και οποιεσδήποτε κινητές συσκευές. Είναι σημαντικό να σημειωθεί ότι κάθε διαμεσολαβητής «ομίχλης» μπορεί να δημιουργεί αλλά και να εγγράφεται σε συνεργατικά θέματα, αναλαμβάνοντας δηλαδή το ρόλο και σταθερού αλλά και γειτονικού διαμεσολαβητή «ομίχλης». Η συνεργατική αρχιτεκτονική περιέχει τέσσερα επίπεδα: α) Επίπεδο Συσκευών, β) Επίπεδο δρομολόγησης, γ) Επίπεδο Ομίχλης και δ) Επίπεδο Νέφους. Η Εικόνα 16 παρουσιάζει μία σύνοψη της εν λόγω προσέγγισης.



Εικόνα 16 Προτεινόμενη αρχιτεκτονική της συνεργατικής πλατφόρμας ομίχλωσης υπολογισμού

3.2.1 Επίπεδο συσκευών

Αποτελεί το κατώτερο αρχιτεκτονικά επίπεδο και περιέχει τα δίκτυα IoT των έξυπνων συσκευών. Οι συσκευές αυτές τυπικά είναι αισθητήρες ή ενεργοποιητές (actuators) χαμηλής υπολογιστικής δυναμικότητας, οι οποίες είτε περιορίζονται στη μέτρηση και αποστολή δεδομένων ή λήψη σημάτων ελέγχου για τη διενέργεια λειτουργίας. Οι συσκευές αυτές θεωρείται ότι δεν έχουν τη δυνατότητα να εκτελέσουν οποιεσδήποτε πολύπλοκες λειτουργίες, και ουσιαστικά για αυτό το λόγο χρησιμοποιείται το επίπεδο «ομίχλης». Παραδείγματος χάρη, ένας ενεργοποιητής λαμβάνει μηνύματα από το επίπεδο «ομίχλης», μετά από ανάλυση των δεδομένων που έχουν αποσταλεί από τους αισθητήρες, ώστε να προβεί σε μία ενέργεια που θα ελέγχει συγκεκριμένες συσκευές του ελεγχόμενου περιβάλλοντος.

3.2.2 Επίπεδο δρομολόγησης

Το επίπεδο δρομολόγησης περιέχει τις πύλες (gateways) λογισμικού και διαδικτύου. Η λειτουργία τους εντοπίζεται στη συλλογή των δεδομένων από τους αισθητήρες παρέχοντας υποστήριξη και συνδεσιμότητα για πρωτόκολλα χαμηλού επιπέδου (π.χ. Bluetooth , ZigBee [103], Z-Wave [104]). Η πύλη περιέχει το κατάλληλο λογισμικό για τη δημιουργία και τον εμπλουτισμό των μηνυμάτων «ομίχλης» τα οποία εν συνεχεία θα αποσταλούν στο επίπεδο «ομίχλης» για την εκτέλεση των εργασιών και επεξεργασία των δεδομένων. Σε περίπτωση όπου μία αισθητήρια συσκευή διαθέτει τη δυνατότητα συνδεσιμότητας με πρωτόκολλα υψηλού επιπέδου (TCP/IP, HTTP) τότε

μπορούν να παρακάμψουν τις πύλες λογισμικού και να στείλουν τα δεδομένα τους απευθείας στο επίπεδο «ομίχλης».

3.2.3 Επίπεδο ομίχλης

Το επίπεδο «ομίχλης» περιέχει όλα τα δίκτυα «ομίχλης» που δραστηριοποιούνται συνεργατικά για τη παροχή υπολογιστικών υπηρεσιών. Για κάθε δίκτυο συσκευών ΙοΤ δημιουργείται και το αντίστοιχο MQTT θέμα στο οποίο μία συσκευή από το σταθερό δίκτυο «ομίχλης» αναλαμβάνει το ρόλο του ενδιάμεσου διαμεσολαβητής «ομίχλης», ο οποίος θα είναι υπεύθυνος για την λήψη των μηνυμάτων «ομίχλης» και την κατανομή των εργασιών στα τερματικά «ομίχλης». Κάθε διαμεσολαβητής «ομίχλης» διατηρεί μία λίστα με όλα τα εγγεγραμμένα δίκτυα και συσκευές στο θέμα, καθώς και τα επιμέρους τερματικά που βρίσκονται στα δίκτυα «ομίχλης» με τα χαρακτηριστικά τους και τις υπολογιστικές δυνατότητες τους. Ο διαμεσολαβητής «ομίχλης» εκτελεί τις παρακάτω λειτουργίες:

3.2.3.1 Κατανομή φόρτου εργασίας

Για την κατανομή των εργασιών που ορίζονται από κάθε μήνυμα «ομίχλης» στα αντίστοιχα τερματικά τα οποία είναι έμμεσα ή άμεσα εγγεγραμμένα στο θέμα, ο διαμεσολαβητής «ομίχλης» χρησιμοποιεί έναν απλοϊκό αλγόριθμο βαθμολόγησης για να κατατάξει τα διαθέσιμα τερματικά ανάλογα με τα χαρακτηριστικά (δυναμικά ή μη) τους κάθε στιγμή. Στη βιβλιογραφία αναφέρονται αρκετοί αλγόριθμοι για κατανομή πόρων και εργασιών, προτείνοντας λύσεις στο χαρακτηριστικό πρόβλημα βελτιστοποίησης BIN PACKING [81] [88]. Οι αλγόριθμοι αυτοί οργανώνονται σε κατηγορίες ανάλογα με τον τρόπο με τον οποίο ψάχνουν να βρουν την υπό-βέλτιστη λύση. Δύο χαρακτηριστικές ομάδες αλγορίθμων είναι οι άπληστης φύσης (greedy) και οι στοχαστικού τύπου. Και οι δύο κατηγορίες δεν είναι κατάλληλες για την περίπτωση ενός δικτύου «ομίχλης» καθώς η μεν κατηγορία greedy ακολουθεί μία λογική επιλογής του πρώτου διαθέσιμου που καλύπτει τις βασικές ανάγκες, ενώ οι στοχαστικού τύπου αλγόριθμοι επιτυγχάνουν καλύτερα αποτελέσματα επιτρέποντας με τυχαίο τρόπο γειτονικές λύσεις με στόχο τη μεγαλύτερη έρευνα του συνόλου λύσεων. Η πρώτη κατηγορία αλγορίθμων, παρότι αρκετά γρήγοροι στην εκτέλεση τους, δεν φημίζονται για τη ποιότητα των λύσεων που προσφέρουν, ενώ οι στοχαστικοί έχουν το μειονέκτημα της αργής κατάληξης λόγω των επιπλέον εκτελέσεων που χρειάζονται για την κατάλληλη έρευνα του πεδίου λύσεων. Για τις ανάγκες της αρχιτεκτονικής ο κατάλληλος αλγόριθμος χρειάζεται να καταλήγει σε μία πολύ καλή λύση όσο πιο άμεσα είναι δυνατόν. Επίσης, καθώς τα χαρακτηριστικά που εισάγονται στον αλγόριθμο δεν αφορούν μόνο τις υπολογιστικές δυνατότητες μία συσκευής, αλλά και κάποια χαρακτηριστικά τα οποία μπορούν να επηρεάσουν την

εκτέλεσης μίας εργασίας, ο αλγόριθμος θα πρέπει να αναθέτει σε κάθε τερματικό στο επίπεδο «ομίχλης» ένα αντιπροσωπευτικό βαθμό που να λαμβάνει όλα τα χαρακτηριστικά του τερματικού στο δίκτυο.

Ο τελικός αλγόριθμος που χρησιμοποιήθηκε εκφράζεται ως εξής:

$$S_i = \sigma_U u_i + \sigma_L \frac{1}{l_i} + \sigma_B b_i \quad (1)$$

Τα χαρακτηριστικά u_i, l_i, b_i εκφράζουν τη ποσοστιαία χρησιμοποίηση πόρων, τη δικτυακή καθυστέρηση και τη στάθμη μπαταρίας (σε περίπτωση κινητών συσκευών) αντίστοιχα, για κάθε τερματικό i οποιαδήποτε στιγμή. Καθώς όμως τα χαρακτηριστικά των δικτύων μεταβάλλονται δυναμικά είναι αναγκαίο ο αλγόριθμος να αναθέτει βάρη σε κάθε χαρακτηριστικό ($\sigma_U, \sigma_L, \sigma_B$). Η επιλογή των εν λόγω βαρών δεν είναι στατική, το οποίο σημαίνει ότι τα βάρη μεταβάλλονται κάθε φορά που ο αλγόριθμος τρέχει. Ο δυναμικός υπολογισμός των βαρών γίνεται με βάση τη διασπορά των τιμών κάθε ενός χαρακτηριστικού για όλες τα κατάλληλα τερματικά. Για παράδειγμα, εάν η διασπορά των τιμών που αφορούν τη χρησιμοποίηση πόρων του συνόλου των τερματικών είναι μικρή αυτό σημαίνει ότι όλες οι συσκευές παρέχουν την ίδια διαθεσιμότητα σε πόρους περίπου. Συνεπώς, ο αλγόριθμος θα δώσει μεγαλύτερη βαρύτητα στα χαρακτηριστικά δικτυακής καθυστέρησης ή στάθμης μπαταρίας.

Η ανωτέρω λογική αναγνωρίζει το γεγονός ότι, παρότι σημαντική, η υπολογιστική ικανότητα δεν παίζει το μεγαλύτερο ρόλο για την επιλογή του καταλληλότερου τερματικού για την διεκπεραίωση μία εργασίας «ομίχλης», καθώς π.χ. μεγάλη δικτυακή καθυστέρηση μπορεί να επιφέρει χειρότερα αποτελέσματα όσον αφορά το χρόνο εκτέλεσης μίας εργασίας «ομίχλης».

3.2.4 Επίπεδο νέφους

Το επίπεδο νέφους αποτελεί το τελευταίο αρχιτεκτονικά επίπεδο της προτεινόμενης λύσης, και αφορά τη κεντρική πλατφόρμα IoT που αποστέλλονται τα δεδομένα για μακροχρόνια αποθήκευση και επεξεργασία. Σε αντίθεση με τις υπάρχουσες προσεγγίσεις που αναφέρονται στη βιβλιογραφία, ο προτεινόμενη λύση χρησιμοποιεί το επίπεδο νέφους και ως μία ακόμα οντότητα η οποία μπορεί να προσφέρει υπολογιστικά στη επεξεργασία των IoT δεδομένων.

Ο διαμεσολαβητής «ομίχλης» στέλνει απευθείας στην πλατφόρμα «νέφους» όποια εργασία χαρακτηρίζεται από αυξημένη πολυπλοκότητα ή περιέχει μεγάλο όγκο δεδομένων, ώστε να εξασφαλισθεί η όσο το δυνατόν πιο άμεση εκτέλεση της. Παρατηρούμε σε αυτή τη περίπτωση πως το επίπεδο νέφους μπορεί να λειτουργεί και

ως PaaS όταν χρησιμοποιείται από το επίπεδο «ομίχλης», και ως IaaS όταν απλά παρέχει τους πόρους τους για μακροχρόνια αποθήκευση των δεδομένων.

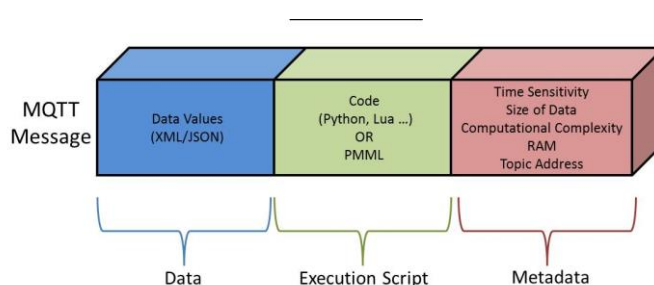
3.2.5 Πρότυπο μηνύματος «ομίχλης»

Η προτεινόμενη λύση επιτυγχάνει μέρος της δρομολόγησης εισάγοντας ένα επίπεδο προτυποποίησης όσον αφορά την ανταλλαγή μηνυμάτων. Η συγκεκριμένη ιδέα αφορά την ανταλλαγή μηνυμάτων μεταξύ του ελεγκτή στο επίπεδο των συσκευών και του δικτύου «ομίχλης». Στη προτεινόμενη αρχιτεκτονική η ανταλλαγή των μηνυμάτων στο επίπεδο «ομίχλης» γίνεται με ασύγχρονο τρόπο, χρησιμοποιώντας το πρωτόκολλο Pub/Sub MQTT. Όπως αναφέρεται και στην περιγραφή της αρχιτεκτονικής μία συσκευή του επιπέδου «ομίχλης» διατελεί ρόλο διαμεσολαβητή (Broker) και διαθέτει την απαραίτητη λογική για τη δρομολόγηση των εργασιών στα επιμέρους «ομίχλης» τερματικά. Ένα από τα καινοτόμα στοιχεία επίσης, είναι η υποστήριξη όχι μόνο γειτνιαζουσών δικτύων «ομίχλης», αλλά και κινητών συσκευών.

Η παροχή υποστήριξης για κινητές συσκευές, παρ' όλα αυτά, απαιτεί ιδιαίτερη αντιμετώπιση λόγω της αβεβαιότητας της διαθεσιμότητας τους, καθώς ανά πάσα στιγμή κάθε κινητή συσκευή μπορεί να βρεθεί εκτός δικτύου ή να απενεργοποιηθεί λόγω χαμηλής στάθμης μπαταρίας. Για το λόγο αυτό κατά την εγγραφή στο ανάλογο θέμα του MQTT οι κινητές συσκευές απολαμβάνουν διαφορετικής ποιότητας υπηρεσίας η οποία ενεργοποιείται από το δρομολογητή του θέματος (MQTT Broker). Τα χαρακτηριστικά του MQTT που επιτρέπουν τη σύναψη αυστηρότερων συνδέσεων ορίζονται κατά τη διαδικασία της εγγραφής της συσκευής στο θέμα και είναι τα εξής πεδία:

- **Πεδίο QoS**, το οποίο μπορεί να πάρει τιμές 0,1 και 2. Η προεπιλεγμένη τιμή για κάθε νέα σύνδεση είναι 0, και δεν εμπεριέχει κανένα είδους έλεγχο όσον αφορά για τη λήψη των μηνυμάτων από τις εγγεγραμμένες συσκευές μέσω επιβεβαιώσεων. Ο MQTT Broker αποστέλλει τα μηνύματα προς όλες τις εγγεγραμμένες οντότητες χωρίς να αναμένει καμία επιβεβαίωση. Οι συνδέσεις τύπου 0 ενδείκνυνται για συσκευές που μπορούν να επιτύχουν σταθερή και υψηλής ποιότητας σύνδεση με τον MQTT Broker. Αντίθετα, για πιο ασταθείς συνδέσεις, όπως π.χ. με κινητές συσκευές, συνιστάνται συνδέσεις τύπου 1. Σε αυτή τη περίπτωση ο MQTT Broker για κάθε μήνυμα αναμένει επιβεβαίωση από την εγγεγραμμένη οντότητα στέλνοντας ένα μήνυμα τύπου PUBLISH μέχρις ότου να λάβει ένα επιβεβαιωτικού χαρακτήρα μήνυμα PUBREC από τον εγγεγραμμένο συνδρομητή

- **Πεδίο *cleanSession***, το οποίο παίρνει τιμές TRUE/FALSE και σηματοδοτεί για το αν ο MQTT Broker χρειάζεται να διατηρήσει τα μηνύματα που προορίζεται για έναν εγγεγραμμένο συνδρομητή σε περίπτωση που αυτός για κάποιο χρονικό διάστημα δεν είναι διαθέσιμος
- **Πεδίο *keepAlive***, το οποίο σηματοδοτεί το ανώτατο χρονικό διάστημα που δύναται να μεσολαβήσει μεταξύ δύο μηνυμάτων PING τα οποία αποστέλλονται από τον συνδρομητή προς τον MQTT Broker για να επιβεβαιώσει τη διαθεσιμότητα του



Εικόνα 17 Πρότυπο μηνύματος «ομίχλης»

Με τη κατάλληλη χρήση των παραπάνω χαρακτηριστικών ο MQTT Broker είναι σε θέση να εντοπίσει σε πολύ μικρό χρονικό διάστημα για το αν μία συσκευή είναι διαθέσιμη ή όχι θέτοντας τα παραπάνω πεδία αναλόγως.

Για τις ανάγκες της αρχιτεκτονικής χρειάζεται η παραμετροποίηση ενός MQTT μηνύματος, καθώς η τυπική του μορφή δεν παρέχει τις απαραίτητες πληροφορίες που θα ικανοποιήσουν τις απαιτήσεις ενός συστήματος «ομίχλης». Για το λόγο αυτό προτείνεται ένα πρότυπο μηνύματος το οποίο θα παρέχει όλες τις απαραίτητες πληροφορίες στον διαμεσολαβητή «ομίχλης» ώστε να δρομολογήσει κατάλληλα κάθε μήνυμα. Το μήνυμα «ομίχλης» περιέχει τρία είδη πληροφορίας:

- τα κυρίως δεδομένα, όπως αυτά έχουν συλλεχθεί από το επίπεδο των συσκευών
- τη ρουτίνα που θα πρέπει να εκτελεστεί αναλύοντας τα δεδομένα. Εκφράζεται με τη μορφή πηγαίου κώδικα η οποία παρέχεται από τον κατασκευαστή και χρησιμοποιεί μία «ελαφριά» γλώσσα προγραμματισμού υψηλού επιπέδου όπως Python, Lua [105] ή Julia [106]. Επιπρόσθετα υπάρχει η δυνατότητα χρησιμοποίησης της γλώσσας σήμανσης PMML, η οποία βασίζεται στην XML και παρέχει μοντέλα προβλέψεων για αλγορίθμους μηχανικής μάθησης.
- τα μετά-δεδομένα, τα οποία σηματοδοτούν τις ανάγκες που πρέπει να ικανοποιεί οποιοδήποτε τερματικό «ομίχλης» ώστε να εκτελέσει το

κώδικα που περιέχεται στο μήνυμα «ομίχλης». Ορίζονται και αυτά από τον κατασκευαστή και περιέχουν τις εξής πληροφορίες: α) κρισιμότητα όσον αφορά το χρόνο εκτέλεσης, β) μέγεθος των δεδομένων, γ) πολυπλοκότητα του αλγορίθμου, δ) απαιτούμενη μνήμη RAM και στ) τη MQTT διεύθυνση του θέματος (topic address)

Τα μετά-δεδομένα του μηνύματος «ομίχλης» είναι εξόχως σημαντικά καθώς χρησιμοποιούνται από το αλγόριθμο ανάθεσης για την επιλογή του καταλληλότερου τερματικού «ομίχλης».

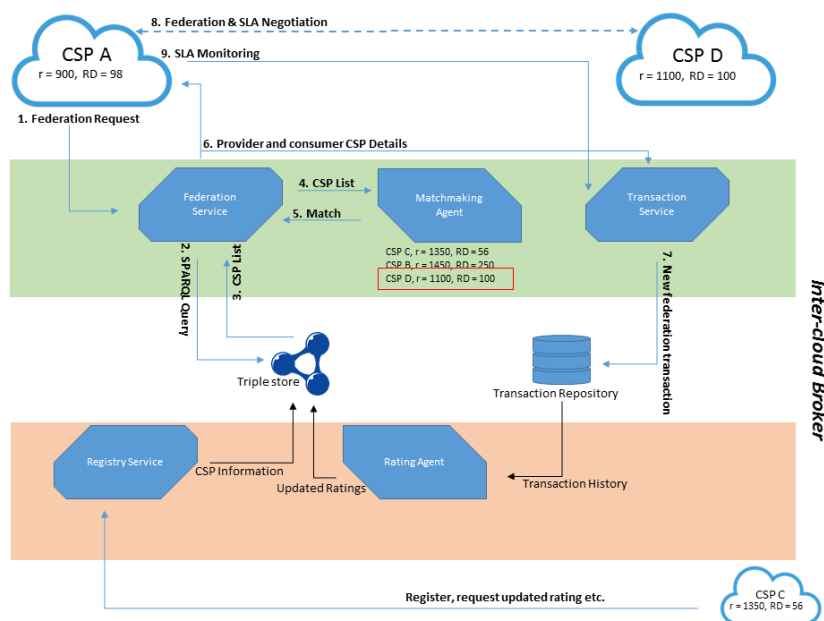
Σημαντικό, για την απόδοση της πλατφόρμας είναι το μήνυμα να περιέχει την απαραίτητη πληροφορία χωρίς να επιβαρύνει την λειτουργία στέλνοντας μεγάλο όγκο μετά-δεδομένων. Η Εικόνα A - 1 που παρουσιάζεται στο Παράρτημα A.2 δείχνει μερικές προσομοιώσεις χρόνων απόκρισης για πολλαπλά ταυτόχρονα αιτήματα μηνυμάτων «ομίχλης» (ένα παράδειγμα μηνύματος μεγέθους περίπου 35KB παρουσιάζεται στο Παράρτημα A.1).

3.3 Συστήματα «ομόσπονδων» δικτύων υποδομών υπολογιστικών νεφών με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA)

Η σημασία ενός μηχανισμού μεσάζοντος όπως παρουσιάστηκε και παραπάνω είναι αναγκαίος για την σωστή δημιουργία συνεργιών μεταξύ δύο ή περισσότερων υποδομών υπολογιστικού νέφους. Στόχος του διαμεσολαβητή «ομοσπονδίας» είναι α) είναι σωστή εξυπηρέτηση των αιτημάτων αιτούντων παρόχων υπηρεσιών «νέφους» (Cloud Service Providers – CSPs) για μίσθωση πόρων και β) συνολική διατήρησης της αξίας του δικτύου. Το δεύτερο σημείο προϋποθέτει μία προτροπή προς όλους τα μέλη του δικτύου να συνεργάζονται με γνώμονα τη τήρηση των SLA, δηλαδή οι πάροχοι να εξασφαλίζουν με τη λειτουργία τους ότι η υπηρεσία που παρέχουν προς ομόσπονδους CSPs τη καλύτερη δυνατή υπηρεσία. Η παρούσα αρχιτεκτονική πρόταση στοχεύει στην εισαγωγή ενός μηχανισμού που θα επιβραβεύει τους παρόχους ανάλογα με το πόσο επιτυγχάνουν στη τήρηση των SLAs, και θα τους βαθμολογεί ανάλογα. Επιπρόσθετα, ο εν λόγω μηχανισμός θα αξιολογεί και τη συχνότητα συμμετοχής CSP σε συνεργίες με άλλους CSPs. Πριν, προχωρήσουμε προς τη περιγραφή της αρχιτεκτονικής και των λειτουργιών τους, χρειάζεται να αναφέρουμε συγκεκριμένες συμβάσεις που χαρακτηρίζουν ιδιότητες ενός «διαδικτύου νεφών» και δικαιολογούν την προτεινόμενη λύση:

- Το μελετώμενο δίκτυο είναι εθελοντικής μορφής. Κάθε πάροχος συμμετέχει επιλέγοντας το πότε και πώς θα δημιουργήσει συνεργίες. Είναι σημαντικό να υπενθυμίσουμε ότι κάθε πάροχος αποτελεί μία εντελώς ανεξάρτητη οντότητα που ανήκει σε ένα οργανισμό. Επίσης, δεν υπάρχει κάποιος περιορισμός σχετικά με τη κατανομή των παρόχων γεωγραφικά.
- Οι συμμετέχοντες CSPs λειτουργούν κυρίως με το μοντέλο IaaS, προσφέρουν δηλαδή ως υπηρεσία τη χρησιμοποίηση πόρων, χωρίς αυτό να αποκλείει τη συμμετοχή παρόχων που προσφέρουν υπηρεσίες διαφορετικών μοντέλων. Σε κάθε περίπτωση κάθε πάροχος με την είσοδο του στο δίκτυο διαφημίζει τις διαθέσιμους προς μίσθωση πόρους στη μορφή εικονικών μηχανημάτων, μαζί με κάποια ποιοτικά χαρακτηριστικά όπως γεωγραφική κατανομή της υπολογιστικής κατανομής ή συντελεστής αντιγράφων σε περίπτωση υπηρεσιών αποθήκευσης δεδομένων (storage services).
- Η αρχιτεκτονική ακολουθεί μία κεντρικοποιημένη δομή, παρόμοια με αυτή που παρουσιάζεται κατά το πρότυπο IEEE P2302 [7]
- Κάθε στιγμή οι συμμετέχοντες CSPs χωρίζονται σε αιτούντες και παρόχους. Παρόλα αυτά οι όροι αυτοί είναι εναλλάξιμοι, δηλαδή κάθε CSP μπορεί να είναι και πάροχος και αιτούμενος
- Ο μεσάζων είναι υπεύθυνος για να εντοπίσει το κατάλληλο πάροχο για κάθε αίτημα, και ως συνέχεια να επικοινωνήσει τα χαρακτηριστικά επικοινωνίας προς τον αιτούντα CSP. Η επικοινωνία μεταξύ των δύο για τις τελικές διαπραγματεύσεις και την δικτυακή επικοινωνία, σύνδεση και τυχόν μεταφορά δεδομένων είναι ανεξάρτητη και αφορά τους εμπλεκόμενους της συναλλαγής.

Η προτεινόμενη αρχιτεκτονική μεσάζοντα περιγράφει τη λειτουργία ενός μηχανισμού διαμεσολαβητή «ομοσπονδίας». Η αρχιτεκτονική χαρακτηρίζεται από το συνδυασμό υπηρεσιοστρεφών και πολυπρακτορικών συστημάτων. Η αναλυτική περιγραφή των λειτουργιών παρουσιάζεται στην Εικόνα 18 και περιγράφεται αναλυτικά στα παρακάτω υπό-κεφάλαια.



Εικόνα 18 Αρχιτεκτονική διαμεσολαβητή ομοσπονδίας υποδομών υπολογιστικών νεφών

3.3.1 Μοντέλο διαχείρισης πληροφορίας

Για την αναπαράσταση όλων των δεδομένων που αφορούν τις οντότητες του δικτύου νεφών, έχουν προταθεί συγκεκριμένα οντολογικά μοντέλα [28, 70]. Καθώς τα αιτήματα σε ένα τέτοιο περιβάλλον τυπικά περιέχουν πολλούς περιορισμούς, η σημασιολογική αναπαράσταση των δεδομένων και η αποθήκευσή τους σε οντολογικές βάσεις δεδομένων (π.χ. αποθετήρια τριπλετών) μπορούν να δώσουν τη λύση για την αποτελεσματική επερώτηση τέτοιων αποθετηρίων.

3.3.2 Πρότυπο αιτήματος χρήσης υπηρεσίας

Το αίτημα ομοσπονδίας αποστέλλεται από τους παρόχους CSP τα οποία επιθυμούν να μισθώσουν υπολογιστικούς πόρους. Τα αιτήματα είναι προτυποποιημένα σε μορφή XML ή JSON και περιγράφουν τις απαιτήσεις του αιτούντος σε πόρους αλλά και περιορισμούς όπως π.χ. απαιτήσεις σε γεωγραφική κατανομή δεδομένων και πόρων, αρχιτεκτονική υπερεπόπτη, τεχνολογία αποθηκευτικών πόρων (SSD ή HDD) κ.α. Λόγω της οντολογικής αναπαράστασης των δεδομένων που περιγράφουν τα χαρακτηριστικά των διαθέσιμων παρόχων τα αιτήματα προς την ομοσπονδία μπορεί να στέλνονται απευθείας σε μορφή επερώτησης SPARQL [17]. Το υπό ανάπτυξη πρότυπο της IEEE P2302 παρουσιάζει ένα παράδειγμα αίτησης μέσω του πρωτοκόλλου XMPP (Παράρτημα A.4). Η παρούσα αρχιτεκτονική πρόταση υιοθετεί το εν λόγω πρότυπο.

3.3.3 Υπηρεσίας ομοσπονδίας

Η υπηρεσία ομοσπονδίας αποτελεί το σημείο επικοινωνίας των αιτούντων CSPs με το οντότητα του μεσάζοντος. Η υπηρεσία δέχεται τα προτυποποιημένα αιτήματα και εξάγει τις απαραίτητες πληροφορίες σχετικά με τα αιτήματα σε πόρους και τα λοιπά χαρακτηριστικά. Σε περίπτωση που το αίτημα έχει ληφθεί σε μορφή JSON/XML η υπηρεσία ομοσπονδίας αναλαμβάνει να δημιουργήσει την ανάλογη επερώτηση SPARQL που θα αποστείλει προς το αποθετήριο τριπλετών, ώστε να λάβει τη φιλτραρισμένη λίστα με τους έγκυρους υποψήφιους παρόχους ώστε να τους προωθήσει προς το πράκτορα αντιστοίχισης.

3.3.4 Πράκτορας αντιστοίχισης

Ο πράκτορας αντιστοίχισης αποτελεί μέρος του πολυπρακτορικού υποσυστήματος του διαμεσολαβητή «ομοσπονδίας». Αποτελεί έναν πράκτορα λογισμικού που εφαρμόζει τον προτεινόμενο αλγόριθμο ώστε να βαθμολογήσει και ταξινομήσει τους κατάλληλους παρόχους CSP με βάση α) τη απόδοση τους ιστορικά όσον αφορά προηγούμενες συναλλαγές και β) τη συχνότητα δραστηριότητας στο δίκτυο. Τέλος, διαλέγει το κατάλληλο πάροχο ώστε με βάση τις ανωτέρω δύο προδιαγραφές του αιτούντος CSP αυτή τη φορά, ώστε να προκύψει η κατάλληλη αντιστοίχιση. Οι λεπτομέρειες επικοινωνίας επιστρέφονται στην υπηρεσία ομοσπονδίας η οποία με τη σειρά τις αποστέλλει στον αιτούντα CSP. Τελικώς, ο πράκτορας αντιστοίχισης αποστέλλει τις λεπτομέρειες αιτούντος-παρόχου προς την Υπηρεσία Συναλλαγών.

3.3.5 Υπηρεσία συναλλαγών

Η Υπηρεσία Συναλλαγών αποτελεί το σημείο επικοινωνίας όλων των πληροφοριών που σχετίζονται με τη συναλλαγές μεταξύ CSP. Κάθε νέα συναλλαγή που εκτελείται αποθηκεύεται με μοναδικό κλειδί ευρετηρίου (ID) σε μία βάση το οποίο ονομάζεται Αποθετήριο Συναλλαγών. Κάθε πληροφορία που σχετίζεται με τη πρόοδο κάθε συναλλαγής αποστέλλεται στην Υπηρεσία Συναλλαγών η οποία ανανεώνει το αποθετήριο Συναλλαγών.

3.3.6 Πράκτορας κατάταξης

Ο Πράκτορας Λογισμικού Κατάταξης εφαρμόζει τον αλγόριθμο κατάταξης στους υποψήφιους παρόχους οι οποίοι πληρούν τα κριτήρια σύμφωνα με τις απαιτήσεις του αιτούντος. Το χαρακτηριστικό του πράκτορα λογισμικού και εν συνεχεία της προτεινόμενης λύσης είναι ότι εκτελεί τον αλγόριθμο κατάταξης ανεξάρτητα από τη λειτουργία της υπόλοιπης πλατφόρμας μεσάζοντος. Ο αλγόριθμος εκτελείται ανά τακτά χρονικά διαστήματα τα οποία αναφέρονται ως περίοδοι κατάταξης. Με την έναρξη και

το πέρας κάθε περιόδου κατάταξης ο αλγόριθμος, με βάση των συναλλαγών που έχουν πραγματοποιηθεί εντός της περιόδου, ανανεώνει τη βαθμολογία κάθε παρόχου που συμμετέχει στο ομόσπονδο δίκτυο.

3.3.7 Υπηρεσία ευρετηρίου

Η υπηρεσία καταλόγου αποτελεί το σημείο της πλατφόρμας του διαμεσολαβητή «ομοσπονδίας» όπου όλα τα δεδομένα σχετικά με τα χαρακτηριστικά των παρόχων. Καλείται από τους παρόχους κατά την εγγραφή τους όπου δημοσιεύουν στην Υπηρεσία Καταλόγου τα ποσοτικά και ποιοτικά χαρακτηριστικά τους. Επιπρόσθετα, η Υπηρεσία Καταλόγου καλείται για να προωθήσει τα ανανεωμένα χαρακτηριστικά κατάταξης προς τους παρόχους όταν αυτά ανανεώνονται.

3.3.8 Αλγόριθμος κατάταξης

Το χαρακτηριστικό της προτεινόμενης υποδομής μεσάζοντα εντοπίζεται στον αλγόριθμο που χρησιμοποιείται για τη βαθμολόγηση και κατάταξη των παρόχων σε ένα ομόσπονδο δίκτυο υποδομών υπολογιστικού νέφους. Η διατήρηση της ομαλής λειτουργίας του δικτύου, προκρίνει τη ανάγκη ενός μηχανισμού που θα προτρέπει τους συμμετέχοντες να λειτουργούν με τρόπο που θα εξασφαλίζει τη βελτιστοποίηση της χρηστικότητας του συνόλου του δικτύου. Για το λόγο αυτό η προτεινόμενη αρχιτεκτονική επιλέγει για τη κατάταξη και τη βαθμολόγηση τον αλγόριθμο Glicko, τον οποίο προσάρμοσε ανάλογα με τις ανάγκες του «διαδικτύου νεφών». Ο αλγόριθμος Glicko προτάθηκε από τον M. Glickman [107] για να παρέχει έναν αξιόλογο τρόπο βαθμολόγησης παικτών σκακιού. Είναι δηλαδή μία μέθοδος κατάταξης παικτών σε ανταγωνιστικά περιβάλλοντα όπου κάθε παίκτη συμμετέχει εθελοντικά. Το χαρακτηριστικό της εθελοντικής συμμετοχής προκρίνει την αναγκαιότητα, εκτός από ένα βαθμό με βάση την απόδοση κάθε παίκτη, να υπάρχει και ένα βαθμός ο οποίος θα προσδιορίζει τη συχνότητα της συμμετοχής του σε ανταγωνιστικά παιχνίδια. Η συχνότητα της συμμετοχής εκφράζει ταυτόχρονα και τη βεβαιότητα του συστήματος για τη πραγματική ικανότητα –δηλαδή απόδοση- κάθε συμμετέχοντα. Η ανανέωση των μετρικών του συστήματος για κάθε παίκτη γίνεται κατά την έναρξη και τη λήξη της περιόδου βαθμολόγησης, εντός της οποίας προσμετρούνται όλα τα παιχνίδια στα οποία συμμετείχε ένας παίκτης. Κατά τη λήξη της περιόδου βαθμολόγησης το σύστημα ανανεώνει τη βαθμολογία ενός παίκτη συγκεντρώνοντας όλα τα παιχνίδια στα οποία συμμετείχε. Το σύστημα επιβραβεύει ανάλογα με την ικανότητα του αντιπάλου, δηλαδή όσο μεγαλύτερη βαθμολογία κατέχει κάποιος αντίπαλος τόσο περισσότερο επιβραβεύεται ο παίκτης που σε ένα παιχνίδι θα καταφέρει να τον νικήσει και

αντίστροφα. Η προσαρμογή του εν λόγω συστήματος βαθμολόγησης έγινε με βάση τις παρακάτω παραδοχές:

- Το σύστημα Glicko προτάθηκε για να εξυπηρετήσει περιβάλλοντα ανταγωνιστικών παιγνίων. Παρ' όλα αυτά στη περίπτωση ομόσπονδων δικτύων η έννοια του ανταγωνισμού δεν υφίσταται. Και η δύο συμμετέχοντες σε μία ομοσπονδία υποδομών θέλουν ο πάροχος που μισθώνει τους πόρους του να πετύχει υψηλά επίπεδα τήρησης των SLA. Για αυτό το λόγο θεωρείται πως ο πάροχος CSP δεν ανταγωνίζεται το αιτούντα αλλά επί της ουσίας ανταγωνίζεται το αίτημα –όσον αφορά το επίπεδο δυσκολίας- με βάση το συμφωνηθέν SLA, και ως εκ τούτου ο προσαρμοσμένος αλγόριθμος βαθμολογεί τα αιτήματα με βάση τις απαιτήσεις που περιγράφουν
- Όπως αναφέρθηκε και παραπάνω το σύστημα Glicko λαμβάνει υπόψιν και τη συχνότητα συμμετοχής ενός παίχτη, η οποία βοηθάει στο υπολογισμό της διασποράς βαθμολόγησης (Rating Deviation – RD). Στη παρούσα περίπτωση και λόγω του ότι ως αντίπαλος σε κάθε συναλλαγή θεωρείται το αίτημα, ως RD αντιπάλου εφαρμόζεται το ελάχιστο που επιτρέπεται στο σύστημα
- Το σύστημα Glicko λαμβάνει υπόψιν τη πιθανότητα ισοπαλίας σε ένα παιχνίδι, παρόλα αυτά στη προσαρμοσμένη έκδοση του η έννοια της ισοπαλίας δεν λογίζεται

3.3.8.1 Περιγραφή λειτουργίας

Η συνοπτική περιγραφή του συστήματος Glicko και του συνόλου των προσαρμογών που έγιναν για τις ανάγκες της αρχιτεκτονικής μεσάζοντα, παρουσιάζεται σε αυτή την υποενότητα. Είναι σημαντικό σε αυτό το σημείο να παραθέσουμε τα δομικά στοιχεία του αλγορίθμου. Αρχικά, όπως αναφέρθηκε και ανωτέρω, το σύστημα εκτός από τη βαθμολογία υπολογίζει και τη συχνότητα συμμετοχής (rating deviation) του στο συνεργατικό περιβάλλον, η οποία εκφράζει τη βεβαιότητα του συστήματος για τη πραγματική ικανότητα του συμμετέχοντος. Για παράδειγμα, αν ένας παίκτης κατέχει βαθμολογία 1850 και RD ίση με 50 τότε το σύστημα μπορεί να λεχθεί ότι είναι κατά 95% βέβαιο ότι η πραγματική του ικανότητα βρίσκεται μεταξύ 1750 και 1950. Στη περίπτωση που εξετάζεται η διασπορά βαθμολογίας είναι σημαντική για την τελική αντιστοίχιση δύο παρόχων.

Όπως περιεγράφηκε και παραπάνω ο αλγόριθμος εκτελείται σύγχρονα, ανά τακτά χρονικά διαστήματα, είναι δηλαδή offline φύσης. Τα χρονικά αυτά διαστήματα ή

αλλιώς rating periods ορίζονται ως οι περίοδοι όπου κάθε παίκτης λαμβάνει μέρος σε έναν -μη αυστηρά- ορισμένο αριθμό παιχνιδιών. Στην αρχή κάθε περιόδου το σύστημα ανανεώνει τη τιμή RD για όλους τους συμμετέχοντες. Στο τέλος κάθε περιόδου το σύστημα ανανεώνει τη βαθμολογία και τη τιμή RD. Τα παρακάτω βήματα περιγράφουν συνοπτικά τη λειτουργία του συστήματος:

Βήμα 1

Κατά την έναρξη της περιόδου αξιολόγησης υπολογίζεται η τιμή RD για κάθε συμμετέχοντα, βασισμένη στη τιμή της προηγούμενης περιόδου:

- Εάν ο παίκτης συμμετέχει για πρώτη φορά η βαθμολογία του και διασπορά βαθμολογίας τίθενται στη μέγιστη τιμή (1500 και 350 αντίστοιχα)
- Σε αντίθετη περίπτωση, η παρακάτω εξίσωση χρησιμοποιείται για τον υπολογισμό της ανανεωμένη τιμής RD

$$RD = \min \left(\sqrt{RD_{old}^2 + c^2}, RD_{max} \right) \quad (2)$$

Η (1) εξασφαλίζει ότι η διασπορά βαθμολόγησης δεν μπορεί να υπερβεί μία ανώτερη τιμή κατωφλίου (RD_{max}).

Βήμα 2

Το βήμα 2 εκτελείται με το πέρας της περιόδου αξιολόγησης. Το σύστημα ανανεώνει τη βαθμολογία και διασπορά RD, με βάση το σύνολο των συμμετοχών του, των «αποτελεσμάτων», και των χαρακτηριστικών τους. Αναλυτικά:

$$r_{new} = r + \frac{q g}{\frac{1}{RD^2} + 1/d^2} \sum_{j=1}^m (s_j - E_j^{CSP}) \quad (3)$$

$$RD_{new} = \sqrt{(1/RD^2 + 1/d^2)^{-1}} \quad (4)$$

Όπου r_{new} αντιπροσωπεύει τη νέα βαθμολογία του συμμετέχοντος, s_j και E_j^{CSP} αντιπροσωπεύουν το πραγματικό και το αναμενόμενο αποτέλεσμα απέναντι σε έναν αντίπαλο j (όπου στη προκειμένη περίπτωση αντιστοιχεί σε ένα αίτημα ενός CSP). Οι ποσότητες q και g αποτελούν σταθερές ποσότητες ($q=0.0057565$), σε αντίθεση με την αρχική έκδοση του αλγορίθμου όπου η τιμή g εκφράζεται ως συνάρτηση της τιμής RD του αντιπάλου. Τέλος η ποσότητα d λαμβάνεται από τη παρακάτω συνάρτηση:

$$g = \frac{1}{\sqrt{1 + 3q^2(RD_{req})^2/\pi^2}} \quad (5)$$

Όπου ως RD_{req} ορίζεται η τιμή RD του αιτήματος, η οποία είναι σταθερή.

3.3.8.2 Βαθμολόγησης αιτήματος υπηρεσίας

Κάθε αίτημα για δημιουργία συνεργίας ενός CSP για ενοικίαση πόρων, βαθμολογείται από το σύστημα κατάταξης ώστε να παρέχει έναν ενιαίο τρόπο για τον υπολογισμό της βαθμολογίας κάθε CSP που προσφέρει υπηρεσίες και υπολογιστικούς πόρους στο ομόσπονδο δίκτυο. Για τον υπολογισμό μίας συμβατής βαθμολογίας τα απαιτούμενα χωρίζονται σε ποσοτικά και ποιοτικά. Ως ποσοτικά ορίζονται οι πόροι επί τους ουσίας που χρειάζεται ο αιτούμενος, ενώ ως ποιοτικά εκείνα τα επιπλέον χαρακτηριστικά υπηρεσίας όπως γεωγραφική θέση, χειρισμός δεδομένων, αρχιτεκτονική υλισμικού, τεχνολογία υπερεπόπτη κλπ.

Όπως αναφέρθηκε και παραπάνω κάθε πάροχος προσφέρει τους πόρους του σε μορφή προτυποποιημένων εικόνων (π.χ. EC2 instances [108]). Θεωρούμε ότι κάθε πάροχος προσφέρει S τύπους VM, με $n_1, n_2, n_3, \dots, n_s$ να εκφράζουν τον ακριβή αριθμό VM για κάθε τύπο. Αντίστοιχα, δημοσιεύει και τη κανονικοποιημένη τιμή για κάθε τύπο VM, $p_1, p_2, p_3, \dots, p_s$. Τέλος, θεωρούμε ότι ένας CSP δημοσιεύει αίτημα για $s_1, s_2, s_3, \dots, s_s$ εικονικά μηχανήματα, όπου s αντιστοιχεί στον ακριβή αριθμό VM για κάθε τύπο. Η βαθμολογία ενός αιτήματος προκύπτει ως εξής:

$$r_{req} = \min \left(\frac{r_{max}}{S} \sum_{j=1}^S \frac{s_j}{n_j} p_j + r_{max} \sum_{k=1}^R r_k w_k, r_{max} \right) \quad (6)$$

Όπου r_{max} αντιστοιχεί στη μέγιστη βαθμολογία την οποία μπορεί να πάρει ένα αίτημα ενός CSP. Το δεύτερο μέρος του υπολογισμού της βαθμολογίας αποτελεί μία προσέγγιση για τις ποιοτικές απαιτήσεις του αιτούντος CSP r_k ($k \in \{1,2,3, \dots, R\}$), και μπορούν να εκφράζονται με αριθμητικές τιμές (0 έως 1) ή λογικές τιμές (1 εάν υφίσταται η απαίτηση, 0 αντίθετα), ενώ w_k εκφράζει το βάρος για κάθε κατηγορία ποιοτικών χαρακτηριστικών.

3.3.8.3 Καταγραφή αποτελέσματος

Για το καθορισμό του αποτελέσματος μία συναλλαγής, το σύστημα χρειάζεται να κατέχει πληροφορίες σχετικά με την απόδοση του παρόχου CSP σχετικά με το συμφωνηθέν SLA. Αν και έχουν προταθεί διάφορες μέθοδοι για την αξιολόγηση και βαθμολόγηση παρόχων υπηρεσιών «νεφελώδους» υπολογισμού [72, 109, 110], δεν υπάρχει ακόμα κάποια προτυποποιημένη και ευρέως διαδεδομένη μέθοδος. Για αυτό το

λόγο, η μέτρηση της απόδοσης αναφέρεται στο σύστημα από τον CSP που λαμβάνει τις υπηρεσίες. Εάν ο πάροχος, είναι σε θέση να ικανοποιήσει όλες τις απαιτήσεις του SLA τότε η συναλλαγή λογίζεται ως «νίκη» υπέρ του παρόχου στο σύστημα, ενώ σε αντίθετη περίπτωση η συναλλαγή αποθηκεύεται ως ήττα κατά του.

3.3.8.4 Τελική επιλογή παρόχου

Όταν ο διαμεσολαβητής «ομοσπονδίας» καταλήξει στη λίστα παρόχων που ικανοποιούν τα αιτήματα του αιτούντος CSP, και τους κατατάξει ανάλογα με τη βαθμολογία τους, στη συνέχεια επιλέγει τον κατάλληλο. Η αντιστοίχιση δεν περιλαμβάνει απλά την επιλογή του παρόχου με την υψηλότερη βαθμολογία. Αντίθετα, το σύστημα προσπαθεί να προσδώσει μία έννοια δικαιοσύνης και επιλέγει με βάση τη βαθμολογία και τη τιμή του αιτούντος CSP. Έτσι, αν ο αιτούμενος CSP έχει βαθμολογία 1200 και RD 150, το σύστημα θα προσπαθήσει να επιλέξει έναν πάροχο που η βαθμολογία του είναι ίση με το χαμηλό κατώφλι του εύρους βαθμολογίας που προκύπτει από τη τιμή RD. Έτσι, για το παραπάνω παράδειγμα θα επιλέξει ένα πάροχο που η βαθμολογία του θα προσεγγίζει την τιμή 900.

Με αυτή τη προσέγγιση το σύστημα επιχειρεί, να προτρέψει κάθε συμμετέχοντα στο δίκτυο α) να αποδίδει καλύτερα τηρώντας σε μεγάλο βαθμό τα SLA, αλλά και β) να παρέχει όσο το δυνατόν περισσότερο. Έτσι, είναι δυνατόν να αποφευχθούν καταστάσεις εκμετάλλευσης, ιδιαίτερα πιθανές σε εθελοντικού τύπου περιβάλλοντα, όπου ένας CSP μπορεί να συμμετέχει ζητώντας μόνο.

3.4 Συστήματα διαχείρισης πόρων και υποδομών υπολογιστικών νεφών με έμφαση στην ενεργειακή αποδοτικότητα

Η αρχιτεκτονική προτείνει τη λειτουργία μίας πλατφόρμας η οποία μπορεί να λειτουργήσει σαν ένα ενδιάμεσο μηχανισμό, αναλαμβάνοντας ανά περίπτωση και το ρόλο ενός διαμεσολαβητή. Η πλατφόρμα χρησιμοποιεί τα προσφερόμενα APIs της υποδομής IaaS για να συγκεντρώνει τις απαραίτητες πληροφορίες σχετικά με την υποδομή. Είναι σημαντικό στο σημείο αυτό να παραθέσουμε τα είδη των πληροφοριών που θα χρησιμοποιεί η πλατφόρμα:

- Χαρακτηριστικά των φυσικών εξυπηρετητών που συνθέτουν το στρώμα IaaS που προσφέρει η υποδομή. Αυτά εμπεριέχουν τους πόρους που διαθέτει το σύστημα (CPU, RAM, αποθηκευτικός χώρος), την αρχιτεκτονική του υλισμικού, το λειτουργικό του σύστημα καθώς και το υποστηριζόμενο υπερεπόπτη (KVM, Xen, Docker κλπ.)

- Δεδομένα παρακολούθησης τα οποία περιέχουν δεδομένα σχετικά με τη λειτουργία των φυσικών και εικονικών μηχανημάτων. Αυτά συνήθως μετρούν τα ποσοστά χρησιμοποίησης πόρων, τα οποία είναι ιδιαίτερα χρήσιμα καθώς επηρεάζουν τις αποφάσεις της πλατφόρμας σχετικά με το αν κάποιο εικονικό μηχάνημα θα πρέπει να μετατραπεί από μία τεχνολογία υπερεπόπτη σε περιέκτη λογισμικού
- Δεδομένα σχετικά με εφαρμογές που εκτελούνται σε κάθε εικονικό μηχάνημα. Τα δεδομένα αφορούν τη φύση των εφαρμογών των χρηστών, αλλά και τη συμπεριφορά τους σχετικά το πόσους πόρους καταναλώνουν

Σύμφωνα με τα παραπάνω δεδομένα που συλλέγει η πλατφόρμα, τα αρχιτεκτονικά της μέρη συνθέτουν την ολοκληρωμένη λειτουργία της ακολουθώντας το υπηρεσιοστρεφές μοντέλο REST. Οι επιμέρους υπηρεσίες της πλατφόρμας είναι οι εξής:

3.4.1 Υπηρεσία πελάτη

Η υπηρεσία πελάτη λειτουργεί δεχόμενη αιτήματα χρηστών για τη δημιουργία των εικονικών μηχανημάτων. Τα αιτήματα αυτά ακολουθούν το πρότυπο REST και μπορεί να είναι μορφής JSON ή XML. Κύριο χαρακτηριστικό τους είναι ότι περιέχουν εκτός από τους αιτούμενους υπολογιστικούς πόρους και την απαίτηση σε ποιότητα υπηρεσίας ώστε να παραχθεί το αντίστοιχο SLA μεταξύ πελάτη και παρόχου. Επιπρόσθετα, και για τις ανάγκες της πλατφόρμας το αίτημα θα περιέχει και συγκεκριμένα μετά-δεδομένα τα οποία θα περιγράφουν το τύπο της εφαρμογής, αλλά και τα τις εντολές αρχικοποίησης και παραμετροποίησης του εικονικού περιβάλλοντος.

Η υπηρεσία πελάτη ανατρέχοντας το αίτημα, εξάγει τις απαραίτητες πληροφορίες και προχωρά στις κλήσεις των υπηρεσιών της υποκείμενης υποδομής IaaS ώστε να δημιουργήσει τα καινούργια εικονικά μηχανήματα και να τα αρχικοποιήσει σύμφωνα με το κώδικα αυτοματισμού (π.χ. Ansible). Τέλος, η υπηρεσία πελάτη επικοινωνεί προς το χρήστη τις λεπτομέρειες σύνδεσης προς το μηχάνημα (π.χ. χαρακτηριστικά ασφαλείας, διεύθυνση και πρωτόκολλο απομακρυσμένης σύνδεσης).

3.4.2 Υπηρεσία εποπτείας

Η υπηρεσία παρακολούθησης είναι υπεύθυνη για το περιοδικό έλεγχο των εικονικών μηχανημάτων των χρηστών και την αποθήκευση των εν λόγω δεδομένων παρακολούθησης για μελλοντική χρήση και ανάλυση από τις λοιπές υπηρεσίες της πλατφόρμας. Ως δεδομένα παρακολούθησης θεωρούνται τα εξής:

- *Χρησιμοποίηση* (utilization) των πόρων του συστήματος, όπου μετράται ο φόρτος επεξεργαστή, η δεσμευμένη μνήμη RAM και ο χρησιμοποιούμενος αποθηκευτικός χώρος. Κάθε τιμή εκφράζεται ποσοστιαία επί της τιμής της συνολικής διαθεσιμότητας κάθε κατηγορίας.
- *Μετρικές δικτύου*, όπου μετράται η συνολική κίνηση σε kB/s, καθώς και λοιπές πληροφορίες όπως χρησιμοποιούμενες δικτυακές πόρτες.
- *Δεδομένα* που σχετίζονται με τις εφαρμογές. Καθώς κάθε αίτημα χρήστη παρέχει πληροφορίες σχετικά με την εφαρμογή/ες που θα εκτελούνται στο συγκεκριμένο μηχάνημα, η υπηρεσία παρακολούθησης συγκεντρώνει δεδομένα και επ' αυτών

Οι σύγχρονες πλατφόρμες λογισμικού IaaS (π.χ. Openstack) παρέχουν εκ των προτέρων υπηρεσίες παρακολούθησης για κάθε εικονικό περιβάλλον καθώς επίσης και για την υπολογιστική υποδομή που χρησιμοποιείται. Κάθε λειτουργία της πλατφόρμας IaaS προσφέρεται με τη μορφή υπηρεσίας, κάτι που δίνει τη δυνατότητα εύκολης ανάπτυξης τρίτων εφαρμογών που μπορούν να προσθέσουν λειτουργικότητα. Επάνω σε αυτή τη φιλοσοφία της επεκτασιμότητας λειτουργεί και η προτεινόμενη αρχιτεκτονική εκμεταλλευόμενη αυτά τα χαρακτηριστικά για να παρέχει τις επιμέρους λειτουργίες της.

3.4.3 Υπηρεσία κατανομής πόρων

Η υπηρεσία κατανομής είναι υπεύθυνη για τη δυναμική διαμόρφωση της υπάρχουσας κατανομής με τέτοιο τρόπο ώστε να διασφαλίζονται τα εξής:

- Σταθερή διατήρηση των επιπέδων κάλυψης των συμφωνηθέντων SLA, που έχει συνάψει ο πάροχος IaaS με τους χρήστες.
- Βελτιστοποίηση της κατανομής των εικονικών μηχανημάτων με τέτοιο τρόπο ώστε να επιτυγχάνεται η καλύτερη και αποδοτικότερη λειτουργία της υποδομής υλισμικού.

Κύρια λειτουργία της υπηρεσίας είναι η επεξεργασία των δεδομένων παρακολούθησης ώστε να εντοπιστούν εικονικά περιβάλλοντα τύπου υπερεπόπτη τα οποία θα μπορούσαν να μεταφερθούν σε εξυπηρετητές που προσφέρουν περιέκτες λογισμικού. Για να συμβεί αυτό θα πρέπει να ισχύουν οι παρακάτω προϋποθέσεις:

- Το εικονικό μηχάνημα να είναι τεχνολογίας υπερεπόπτη, δηλαδή να έχει δημιουργηθεί με ένα εκ των Xen, KVM ή Hyper-V που υποστηρίζονται από τις σύγχρονες πλατφόρμες IaaS

- Να παρατηρείται σταθερή υπό-χρησιμοποίηση (under-utilization) των πόρων του εικονικού περιβάλλοντος. Συνήθως η τιμή κατωφλίου εντοπίζεται μεταξύ 10-40% των διαθέσιμων πυρήνων CPU ή μνήμης RAM. Το φαινόμενο της υπό-χρησιμοποίησης χρήζει ιδιαίτερης προσοχής, καθώς μειώνει την αποδοτικότητα της υποδομής καταναλώνοντας άσκοπα πόρους

Η φύση της εφαρμογής επιτρέπει τη μετάβαση σε περιέκτη λογισμικού (software container). Καθώς ένα κύριο χαρακτηριστικό των περιεκτών είναι ότι δεν προδεδμεύουν πόρους για τη δημιουργία εικονικού μηχανήματος, οι διαθέσιμοι πόροι του υπερκεείμενου συστήματος δεν είναι εξασφαλισμένοι. Έτσι, οι περιέκτες λογισμικού ενδείκνυνται για δικτυακές εφαρμογές και υπηρεσίες (web servers, web services) καθώς και για εφαρμογές όπου εντοπίζεται έντονη δικτυακή κίνηση, καθώς τον επιπλέον επίπεδο που προστίθεται από τον υπερεπόπτη μειώνει τη δικτυακή απόδοση των VMs [111].

Υπό τις προϋποθέσεις αυτές η υπηρεσία κατανομής αναλαμβάνει να επανεκκινήσει το μηχάνημα σε άλλον φυσικό εξυπηρετητή, αυτή τη φορά ως περιέκτη λογισμικού. Στο σημείο αυτό φαίνεται η χρησιμότητα των τεχνολογιών και προτύπων αρχικοποίησης (automation tools) τα οποία περιέχουν την απαραίτητα πληροφορία και εντολές που εκτελούνται σε ένα σύστημα κατά την εκκίνησή του. Έτσι, όλες οι εφαρμογές και οι εξαρτώμενες βιβλιοθήκες που λειτουργούν στο σύστημα μπορούν μέσω προτυποποιημένων εντολών να εγκατασταθούν στο νέο περιβάλλον.

Με αυτό τον τρόπο, επιτυγχάνεται α) η άμεση μετάβαση -από πλευράς χρήστη- από το ένα περιβάλλον στο άλλο, χωρίς να αντιληφθεί απώλεια της σύνδεσης (το εικονικό μηχάνημα διαγράφεται μόνο όταν ο περιέκτης αρχικοποιηθεί επιτυχώς) και β) η εξοικονόμηση πόρων που ενδεχομένως να χρειαζόντουσαν σε περιπτώσεις «ζωντανής» μεταφοράς (live migration) εικονικών μηχανημάτων [5].

3.4.4 Υπηρεσία επιπέδου υποδομής

Η υπηρεσία υποδομής αποτελεί το σημείο επικοινωνίας της πλατφόρμας με την υποκείμενη υποδομή υπολογιστικού νέφους. Ο κύριος λόγος ύπαρξης της είναι η αποσύζευξη των λοιπών υπηρεσιών της πλατφόρμας από τις κλήσεις των APIs της υπολογιστικής υποδομής IaaS. Αυτό είναι αναγκαίο α) για την υποστήριξη πολλαπλών πλατφορμών IaaS οι οποίες παρέχουν διαφορετικό API και β) για την υποστήριξη διαφορετικών εκδόσεων των API της ίδιας πλατφόρμας. Κατ' αυτό τον τρόπο είναι

δυνατόν να παρέχεται υποστήριξη –και ως εκ τούτου να επιτυγχάνεται επεκτασιμότητα δίχως να χρειάζεται ενημέρωση όλων των υπηρεσιών της πλατφόρμας.

3.4.5 Επισημάνσεις σχετικά με τη προτεινόμενη λύση

Όπως αναφέρθηκε και στην εισαγωγή του παρόντος υπό-κεφαλαίου η παρούσα αρχιτεκτονική προτείνει την αυτοματοποιημένη διαχείριση υπολογιστικής υποδομής που προσφέρει υπηρεσίες IaaS και υποστηρίζει και τις δύο τεχνολογίες εικονικοποίησης. Η αποδοτική διαχείριση των πόρων υλισμικού αποτελεί κύριο στόχο κάθε παρόχου, αλλά η προσέγγιση των μεθόδων που παρατηρούνται στη βιβλιογραφία εξετάζουν, στη συντριπτική πλειοψηφία τους, τεχνολογίες υπερεπόπτη. Το γεγονός αυτό οδηγούσε σε λύσεις που παρουσίαζαν τα εξής μειονεκτήματα:

- Λόγω της ανάγκης προ-δέσμευσης των πόρων, πολλές λύσεις αντιμετωπίζουν το πρόβλημα ως μία παραλλαγή του BIN-PACKING, προβλήματος NP-Hard το οποίο απαιτεί πολύπλοκους αλγόριθμους για την εξαγωγή ικανοποιητικής λύσης. Τέτοιας μορφής αλγόριθμοι όπως οι γενετικοί, ant-colony και ευφυΐας σμήνους (swarm intelligence) [81, 89, 112] μπορεί να οδηγήσουν σε φαινόμενα αργής σύγκλισης
- Επίσης για την μεταφορά των εικονικών μηχανημάτων πολλές μέθοδοι εξετάζουν τη λειτουργία του live migration [5] η οποία παρ' όλα αυτά απαιτεί επιπλέον πόρους και προκαλεί μείωση της απόδοσης (performance degradation)

Η υιοθέτηση των περιεκτών λογισμικού ως εναλλακτικής όσον αφορά τη παροχή εικονικών συστημάτων αυξάνεται ολοένα και περισσότερο. Η ευελιξία που παρέχουν όσον αφορά τη κατανάλωση πόρων αλλά και τη εκκίνηση τους αποτελούν δραστικούς παράγοντες ώστε να προκριθούν ως κυρίαρχη λύση. Επιπρόσθετα, λόγω των εγγενών χαρακτηριστικών τους έχουν τη δυνατότητα να καταναλώνουν λιγότερους πόρους για τον ίδιο φόρτο εργασίας.

Παρ' όλα αυτά η πλήρης υιοθέτηση τους δεν ενδείκνυται για όλες τις περιπτώσεις χρήσεων. Η εγγυημένη διάθεση μέρους των πόρων του συστήματος ενδείκνυται για εφαρμογές που έχουν αυξημένες απαιτήσεις σε πόρους. Επιστημονικές εφαρμογές που εκτελούν εκτενείς και απαιτητικές προσομοιώσεις συνήθως καταλαμβάνουν μεγάλο μέρος των πόρων. Η λειτουργία τους σε περιέκτες λογισμικού είναι απαγορευτικές καθώς ο διαμοιρασμός των πόρων του υπερκείμενου συστήματος σε επιμέρους περιέκτες γίνεται εντελώς δυναμικά ανάλογα με τους διαθέσιμους πόρους αλλά και τις ανάγκες της κάθε θυρίδας. Επίσης, εφαρμογές που εξαρτώνται από την κατάσταση τους

κάθε στιγμή (μη-υπερσειστροφείς, προσομοιώσεις) δεν είναι δυνατόν να μεταφερθούν σε άλλο περιβάλλον καθώς η διακοπή τους θα σημάνει τη διαγραφή της πορείας της εφαρμογής μέχρι εκείνο το σημείο.

Η προτεινόμενη αρχιτεκτονική λοιπόν βασίζεται στην ανάγκη εξυπηρέτησης όλου του εύρους περιπτώσεων χρήσης, αναγνωρίζοντας την ανάγκη για την υποστήριξη και των δύο τεχνολογιών.

Κεφάλαιο 4

Αξιολόγηση αρχιτεκτονικών προτάσεων και πειραματική διαδικασία

4.1 Υποστηρικτικά εργαλεία για τη ανάπτυξη των προσομοιώσεων

Η ανάπτυξη των προσομοιώσεων βασίστηκε σε μία ομάδα εργαλείων τα οποία επεκτάθηκαν με σκοπό την ανάδειξη των χαρακτηριστικών των αρχιτεκτονικών προτάσεων. Συγκεκριμένα, η κύρια λύση που υποστήριξε τις προσομοιώσεις για τις πλατφόρμες «ομιχλώδους» υπολογισμού και διαχείρισης υπολογιστικών υποδομών υβριδικής εικονικοποίησης, ήταν η πλατφόρμα Cloudsim [113]. Για τις προσομοιώσεις της συνεργατικής πλατφόρμας «ομιχλώδους» υπολογισμού χρησιμοποιήθηκε και μία βιβλιοθήκη επεκτάσεων [114] που προσθέτει κάποιες απαραίτητες λειτουργίες στην πλατφόρμα Cloudsim. Για την αξιολόγηση του συστήματος βαθμολόγησης που χρησιμοποιείται από την πλατφόρμα ενδιαμέσου συνεργατικών δικτύων υποδομών υπολογιστικών νεφών, οι προσομοιώσεις βασίστηκαν σε μία υλοποίηση του συστήματος Glicko [107] η οποία αναπτύχθηκε στη γλώσσα προγραμματισμού Python.

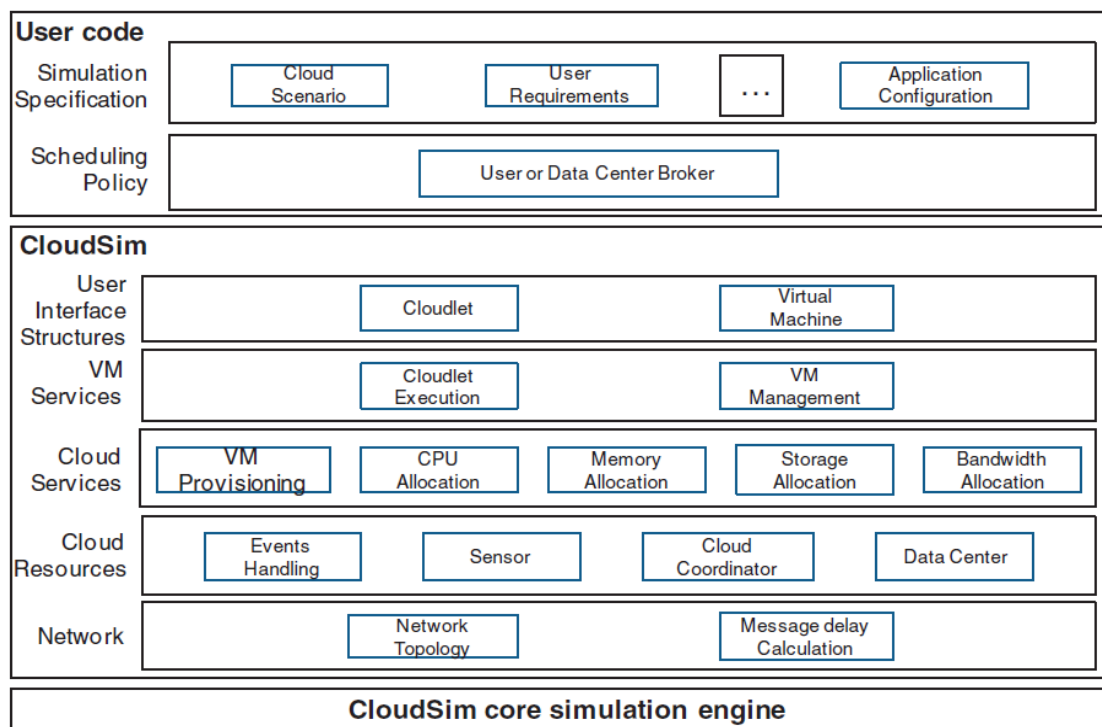
4.1.1 Πλατφόρμα προσομοιώσεων Cloudsim

Η πλατφόρμα Cloudsim αποτελεί μία βιβλιοθήκη ανεπτυγμένη στη γλώσσα προγραμματισμού Java, η οποία επιτρέπει τη διενέργεια προσομοιώσεων σεναρίων σε περιβάλλοντα υπολογιστικά νέφη. Η αρχιτεκτονική του οργανώνεται σε οντότητες που εκτελούνται παράγοντας και εκτελώντας γεγονότα τα οποία δημιουργούνται κατά τη διάρκεια της προσομοίωσης. Η κεντρική δομή της πλατφόρμας Cloudsim αποτελείται από μία ουρά, η οποία διατηρεί το σύνολο των γεγονότων τα οποία οργανώνονται με βάση τον εκτιμώμενο χρόνο εκτέλεσης τους. Κάθε γεγονός της πλατφόρμας Cloudsim περιέχει τα εξής πεδία:

- **Πεδίο αποστολέα**, το οποίο αφορά την οντότητα που παράγει το γεγονός ανά πάσα στιγμή κατά τη διάρκεια της εκτέλεσης της προσομοίωσης
- **Πεδίο παραλήπτη**, το οποίο αφορά την οντότητα που θα παραλάβει και θα αναλάβει την επεξεργασία του γεγονότος

- **Πεδίο τύπου εργασίας**, το οποίο περιγράφει το τύπο εργασίας η οποία πρέπει να εκτελεστεί, καθορίζοντας έτσι τη μέθοδο η οποία θα πρέπει να εκτελεστεί εντός μίας οντότητας. Η προσέγγιση αυτή δίνει τη δυνατότητα στα γεγονότα να επιτυγχάνουν υψηλά επίπεδα αφαιρετικότητας
- **Πεδίο δεδομένων**, το οποίο περιέχει τα κυρίως δεδομένα τα οποία θα επεξεργαστούν από τη παραλαμβάνουσα οντότητα
- **Πεδίο χρονοσφραγίδας**, το οποίο καταδεικνύει τον ακριβή χρόνο σε μονάδες προσομοιωτή όπου το γεγονός παραλήφθηκε από την οντότητα. Με βάση τη συγκεκριμένη χρονοσφραγίδα υπολογίζεται η χρονοσφραγίδα που θα ανατεθεί σε τυχόν επόμενα γεγονότα που θα παραχθούν από τη οντότητα

Οι οντότητες της πλατφόρμας μοντελοποιούν τα δομικά στοιχεία και λειτουργίες που εντοπίζονται σε ένα πραγματικό περιβάλλον υποδομής υπολογιστικού νέφους. Οι κύριες οντότητες της πλατφόρμας Cloudsim παρουσιάζονται στην Εικόνα 19 ενώ μερικές από αυτές περιγράφονται εκτενώς παρακάτω.



Εικόνα 19 Δομικές μονάδες της πλατφόρμας Cloudsim [113]

- **Κατάλογος οντοτήτων (Cloud Information Service)**. Αποτελεί επί της ουσίας μία στατική Java κλάση η οποία διατηρεί μία λίστα με όλες τις οντότητες του προσομοιωτή. Κάθε καινούργια οντότητα που δημιουργείται εγγράφεται στον εν λόγω κατάλογο. Η κύρια διαδικασία

προόδου της εκτέλεσης του Cloudsim, χρησιμοποιεί τον κατάλογο οντοτήτων για να ανατρέχει σε τυχόν οντότητες συγχρονισμού και σηματοδοσίας

- **Υπολογιστικό κέντρο (Datacenter).** Αποτελεί την οντότητα η οποία μοντελοποιεί τη δομή και τις υπηρεσίες IaaS ενός υπολογιστικού κέντρου. Περιλαμβάνει μία συστοιχία εξυπηρετητών οι οποίοι θα φιλοξενούν τα εικονικά περιβάλλοντα. Διατηρεί μία εσωτερική κλάση η οποία περιγράφει τα χαρακτηριστικά του, όπως τεχνολογία εικονικοποίησης και υπερεπόπτη, γεωγραφικά και δικτυακά χαρακτηριστικά, πολιτικές χρήσης και κατάλογο κοστολόγησης υπηρεσιών (χρήση πόρων/ώρα)
- **Διαμεσολαβητής νέφους (Cloud/Datacenter Broker).** Μοντελοποιεί την υπηρεσία διαμεσολαβητή στο επίπεδο IaaS, η οποία αναλαμβάνει να φέρει εις πέρας τα αιτήματα χρηστών όσον αφορά τις απαιτήσεις τους σε υπολογιστικούς πόρους. Οι πόροι αυτοί εκφράζονται βεβαίως σε μορφή εικονικών μηχανημάτων με συγκεκριμένα χαρακτηριστικά, όπως αριθμός και επεξεργαστική ισχύς πυρήνων, μνήμη RAM, αποθηκευτικός χώρος και διαθέσιμο εύρος ζώνης δικτύου. Ο διαμεσολαβητής αναλαμβάνει επίσης συγκεκριμένες αρμοδιότητες, όπως π.χ. η εύρεση του κατάλληλου φορέα που θα αναλάβει να εξυπηρετήσει το αίτημα πελάτη. Η οντότητα διαμεσολαβητή λειτουργεί επίσης και στο SaaS επίπεδο, όπου αναλαμβάνει να εκτελέσει αιτήματα που αφορούν υπολογιστικές εφαρμογές, διατηρώντας ανάλογες αρμοδιότητες με αυτές του IaaS επιπέδου
- **Εξυπηρετητής (Host).** Η οντότητα αυτή μοντελοποιεί τη λειτουργία της κύριας δομικής μονάδας ενός υπολογιστικού κέντρου, τον εξυπηρετητή. Ο εξυπηρετητής είναι υπεύθυνος, από πλευράς υλισμικού, να προσφέρει τους απαραίτητους πόρους για τη δημιουργία των εικονικών δομών
- **Εικονικό μηχανήμα (VM).** Αποτελεί από τη κύριες οντότητες της πλατφόρμας Cloudsim, και μοντελοποιεί τη λειτουργία ενός εικονικού μηχανήματος που δημιουργείται και φιλοξενείται από έναν εξυπηρετητή, και καταλαμβάνει ένα μέρος των πόρων του
- **Πάροχος πόρων (Provisioner).** Η οντότητα αυτή μοντελοποιεί τη πολιτική ανάθεσης πόρων των εξυπηρετητών προς τα εικονικά μηχανήματα. Ανάλογα με τη φύση των πόρων, υπάρχει και η αντίστοιχη υποκλάση που κληρονομεί τα κύρια πεδία και μεθόδους της κύριας

- **Πολιτική κατανομής εικονικών πόρων (VM allocation policy).** Μοντελοποιεί τη πολιτική κατανομής και ανάθεσης εικονικών μηχανημάτων στους εξυπηρετητές ενός υπολογιστικού κέντρου. Κάθε υπολογιστικό κέντρο μπορεί να έχει μόνο μία πολιτική. Η πολιτική κατανομής κατέχει ένα πολύ σημαντικό ρόλο στη λειτουργία του υπολογιστικού κέντρου, καθώς μπορεί να έχει άμεσο αντίκτυπο στην συνολική κατανάλωση ενέργειας αλλά και στην απόδοση και τη τήρηση των SLA. Η πλατφόρμα Cloudsim υλοποιεί εξ' αρχής διάφορες πολιτικές κατανομής, οι οποίες όμως επί της ουσίας οργανώνονται σε δύο βασικές κατηγορίες, τις α) στατικής κατανομής και β) τις πολιτικές δυναμικής κατανομής. Οι στατικοί κατανεμητές κατανέμουν τα εικονικά μηχανήματα με βάση μία λειτουργική απαίτηση του υπολογιστικού κέντρου και ολοκληρώνουν την εργασία τους. Αντίθετα, οι δυναμικοί κατανεμητές πραγματοποιούν μία αρχική κατανομή, και εν συνεχεία μέσω συνεχούς εποπτείας προχωρούν σε ανακατανομές (migrations), με σκοπό η λειτουργική απαίτηση του υπολογιστικού κέντρου να ικανοποιείται σε όλη τη διάρκεια της λειτουργίας του
- **Χρονοπρογραμματιστής εικονικών μηχανημάτων (VM Scheduler).** Η εν λόγω οντότητα εκφράζει την πολιτική διαμοιρασμού των πόρων του εξυπηρετητή. Δύο κύριες μορφές που προσφέρονται εξ' αρχής από τη βασική έκδοση του Cloudsim είναι η πολιτικές διαμοιρασμού time-shared και space-shared. Η πολιτική time-shared επιτρέπει τη απασχόληση μία επεξεργαστικής μονάδας από παραπάνω από ένα εικονικά μηχανήματα διαμοιράζοντας όμως τη διαθέσιμη επεξεργαστική ισχύ ανάλογα με τον αριθμό των εικονικών μηχανημάτων που τον μοιράζονται. Αντίθετα, η πολιτική space-shared επιτρέπει τη χρήση μίας επεξεργαστικής μονάδας μόνο από ένα εικονικό μηχανήματα ανά πάσα στιγμή. Κάθε πολιτική έχει διαφορετικό αντίκτυπο στην εκτέλεση τόσο των εικονικών μηχανημάτων, όσο και των εργασιών που εκτελούνται εντός αυτών
- **Εργασία νέφους (Cloudlet).** Μοντελοποιεί μία οποιαδήποτε μορφή φόρτου εργασίας (εφαρμογή/υπηρεσία) που ανατίθεται για εκτέλεση σε ένα εικονικό μηχανήματα. Κάθε εργασία διαθέτει συγκεκριμένα χαρακτηριστικά όπως το μέγεθος (σε αριθμό εντολών επεξεργαστή), το μέγεθος των δεδομένων που παράγει, τις απαιτήσεις σε μνήμη RAM και σε εύρος ζώνης δικτύου. Κάθε εργασία ανήκει σε ένα χρήστη και η

κατανομή της σε εικονικό μηχάνημα αναλαμβάνεται από τον διαμεσολαβητή νέφους

- **Χρονοπρογραμματιστής εργασιών νέφους (Cloudlet Scheduler).** Η λειτουργία του είναι αντίστοιχη με αυτή του χρονοπρογραμματιστή εικονικών μηχανημάτων ενός εξυπηρετητή. Κάθε εικονικό μηχάνημα διατηρεί μία έκδοση του χρονοπρογραμματιστή εργασιών, ο οποίος αντίστοιχα εφαρμόζει πολιτικές σχετικά με την ανάθεση πόρων του εικονικού μηχανήματος προς τις εργασίες νέφους

Οι παραπάνω βασικές οντότητες επεκτείνονται εντός της πλατφόρμας Cloudsim ώστε να προσφέρουν πιο συγκεκριμένες και στοχευμένες λειτουργίες ανάλογα με τις απαιτήσεις του χρήστη προσομοιώσεων. Για παράδειγμα, η πλατφόρμα προσφέρει πακέτα επεκτάσεων προσαρμοσμένων στη ενεργειακή αποδοτικότητα όπου οι οντότητες των υπολογιστικών κέντρων, εξυπηρετητών, εικονικών μηχανημάτων και πολιτικών κατανομής έχουν προσαρμοστεί στη εποπτεία της ενεργειακής κατανάλωσης. Τέλος, η πρόσφατη έκδοση του Cloudsim προσθέτει υποστήριξη για τη λειτουργία περιεκτών λογισμικού, οι οποίοι όμως λειτουργούν σαν ένα ενδιάμεσο επίπεδο μεταξύ των εικονικών μηχανημάτων και των εργασιών νέφους. Αξίζει να σημειωθεί ότι λόγω της άδειας χρήσης του προσομοιωτή, η οποία είναι ανοικτού κώδικα, διάφορες επεκτάσεις έχουν δημοσιευθεί [115] [116] [117] οι οποίες έχοντας ως βάση το API του Cloudsim προσφέρουν συγκεκριμένες βελτιωτικές λειτουργίες, όπως υποστήριξη γραφικού περιβάλλοντος σύνθεσης προσομοιώσεων ή υποστήριξη επιπρόσθετων πολιτικών κατανομής με τη χρήση μετα-ευριστικών αλγορίθμων κατανομής και βελτιστοποίησης.

4.1.2 Πλατφόρμα προσομοιώσεων CloudsimEx

Η πλατφόρμα προσομοιώσεων CloudsimEx [114] αποτελεί μία από τις κύριες επεκτάσεις της πλατφόρμας Cloudsim. Αντίστοιχα με την πλατφόρμα Cloudsim είναι λογισμικό ανοικτού κώδικα και κύριος στόχος της είναι η υποστήριξη και η μοντελοποίηση υπηρεσιών διαδικτύου. Αντίθετα με τη βασική μορφή εργασιών που υποστηρίζει το Cloudsim, οι εργασίες διαδικτύου (WebCloudlets) χαρακτηρίζονται από τη παραγωγή συνεχούς φόρτου εργασίας μεταβλητού μεγέθους. Κάθε εργασία ανατίθεται σε ένα εικονικό μηχάνημα από τον αντίστοιχο διαμεσολαβητή (WebBroker) και ανά τακτά χρονικά διαστήματα ανανεώνει τις απαιτήσεις του σε πόρους επεξεργαστή, μνήμης και αποθηκευτικού χώρου (I/O). Πέρα από τη υποστήριξη υπηρεσιών διαδικτύου η πλατφόρμα CloudsimEx προσφέρει και τις εξής βελτιώσεις:

- Αυτόματη αποθήκευση των αποτελεσμάτων σε προτυποποιημένα αρχεία CSV

- Αυτόματη ανάθεση μοναδικών χαρακτηριστικών (Id) στις οντότητες
- Μοντελοποίηση εργασιών δίσκου (I/O operations)
- Υποστήριξη πολλαπλών προσομοιώσεων παράλληλα
- Μοντελοποίηση δικτυακής καθυστέρησης
- Προσομοιώσεις MapReduce

Η πλατφόρμα CloudsimEx αποτέλεσε τη βάση για τη δημιουργία της πλατφόρμας προσομοιώσεων που υποστήριξαν την αξιολόγηση της συνεργατικής πλατφόρμας διαχείρισης εργασιών ομιχλώδους υπολογισμού.

4.2 Συστήματα «νεφελώδους» και «ομιχλώδους» υπολογισμού με έμφαση στη ποιότητα υπηρεσίας

4.2.1 Περιγραφή πειραματικής διαδικασίας

Για την αξιολόγηση της προτεινόμενης αρχιτεκτονικής λύσης διενεργήθηκαν μία σειρά από προσομοιώσεις, χρησιμοποιώντας τη πλατφόρμα Cloudsim και τη σειρά επεκτάσεων CloudsimEx η οποία παρείχε την απαραίτητη λειτουργικότητα για τη δυναμική άφιξη εργασιών. Η συνολική διαδικασία στόχευσε στη σύγκριση της συνεργατικής πλατφόρμας «ομιχλώδους» υπολογισμού έναντι του βασικού μοντέλου Fog, καθώς επίσης και έναντι του παραδοσιακού μοντέλου «νεφελώδους» υπολογισμού.

Ο βασικός λόγος για τη χρησιμοποίηση πλατφόρμας προσομοιώσεων ήταν η δημιουργία σεναρίων τα οποία α) θα ανταποκρίνονται σε πραγματικές συνθήκες και περιπτώσεις χρήσης και β) θα επιχειρούν να αξιολογήσουν την εν λόγω πλατφόρμα υπό απαιτητικές συνθήκες φόρτου εργασίας. Επίσης, δόθηκε η δυνατότητα για τη συλλογή μετρήσιμων αποτελεσμάτων, όπως ο εκτιμώμενος συνολικός και μέσος χρόνος εκτέλεσης εργασιών, η μέση χρησιμοποίηση πόρων κάθε τερματικού και το σύνολο των επιτυχώς και ανεπιτυχώς εκτελεσθέντων εργασιών.

Όπως αναφέρθηκε και στο υποκεφάλαιο 3.2.3.1 ο κατανεμητής φόρτου εργασίας εφαρμόζει μία συνάρτηση βαθμολόγησης (score function) για κάθε υποψήφιο τερματικό ή κινητή συσκευή που βρίσκεται σε κυρίως ή γειτονικό δίκτυο «ομίχλης». Ο αλγόριθμος βαθμολόγησης δίδεται από την ακόλουθη εξίσωση:

$$S_i = \sigma_U u_i + \sigma_L 1/l_i + \sigma_B b_i, \quad l_i > 0 \quad (7)$$

Τα χαρακτηριστικά u_i, l_i, b_i εκφράζουν τη χρησιμοποίηση επί τοις εκατό των επεξεργαστικών πόρων, τη δικτυακή καθυστέρηση και τη στάθμη μπαταρίας (σε περίπτωση κινητών συσκευών) αντίστοιχα, για κάθε τερματικό i οποιαδήποτε στιγμή. Καθώς όμως τα χαρακτηριστικά αυτά μεταβάλλονται δυναμικά είναι αναγκαίο ο

αλγόριθμος να αναθέτει βάρη σε κάθε χαρακτηριστικό ($\sigma_U, \sigma_L, \sigma_B$). Η επιλογή των εν λόγω βαρών δεν είναι στατική, το οποίο σημαίνει ότι τα βάρη μεταβάλλονται κάθε φορά που ο αλγόριθμος εκτελείται. Ο υπολογισμός κάθε βάρους γίνεται υπολογίζοντας τη διασπορά των τιμών όλων των υποψήφιων τερματικών για κάθε κατηγορία (χρησιμοποίηση, δικτυακή καθυστέρηση και στάθμη μπαταρίας). Με αυτό τον τρόπο ο αλγόριθμος προσπαθεί να προσαρμόσει την κατανομή των εργασιών ανάλογα με τις συνθήκες που επικρατούν στο εποπτευόμενο δίκτυο «ομίχλης» κάθε στιγμή. Για παράδειγμα, εάν η διασπορά των τιμών χρησιμοποίησης επεξεργαστικών πόρων είναι μικρή αυτό συνεπάγεται ότι όλα τα τερματικά έχουν την ίδια περίπου διαθεσιμότητα σε υπολογιστικούς πόρους, και κατ' αυτό τον τρόπο ο αλγόριθμος θα δώσει μεγαλύτερη βαρύτητα στα χαρακτηριστικά δικτύου κάθε τερματικού ή στη διαθέσιμη στάθμη μπαταρίας.

4.2.1.1 Σύνθεση και παραμετροποίηση προσομοιώσεων

Ο χρόνος εκτέλεσης κάθε σεναρίου, καθώς και κάθε εργασίας ξεχωριστά μετράται σε αόριστες μονάδες, χαρακτηριστικό που κληρονομείται από τη πλατφόρμα Cloudsim. Ένα επιπλέον χαρακτηριστικό που κληρονομείται από τη πλατφόρμα Cloudsim είναι η μέτρηση της επεξεργαστικής δυνατότητας κάθε τερματικού στη μονάδα MIPS (Million Instructions per Second). Η μονάδα μέτρησης MIPS αν και δεν αποτελεί την πιο χαρακτηριστική μέθοδο αποτύπωσης των δυνατοτήτων ενός επεξεργαστή, ιδιαίτερα όταν εξετάζονται σύγχρονες αρχιτεκτονικές επεξεργαστών με πολλαπλούς πυρήνες, επιτρέπει παρ' όλα αυτά την εύκολη μέτρηση χρόνου εκτέλεσης εργασιών, καθώς η πολυπλοκότητα αυτών μετράται επίσης σε μονάδες MI (Million Instructions σε σύνολο). Πέρα από τη επιλογή των προκαθορισμένων χαρακτηριστικών της πλατφόρμας Cloudsim, για κάθε σενάριο το οποίο θα περιγραφεί στη συνέχεια επιλέχθηκαν οι ακόλουθες παράμετροι:

- **Εργασίες (tasks).** Για κάθε σενάριο προσομοίωσης 5000 εργασίες χρονοπρογραμματίστηκαν για άφιξη σε τυχαία χρονικά σημεία. Ένα μικρό ποσοστό εργασιών (5%) χαρακτηρίστηκαν ως κρίσιμες όσον αφορά το χρόνο περάτωσης (time-critical). Ο εν λόγω χαρακτηρισμός οδήγησε τον κατανομητή εργασιών να επιλέξει τερματικό που ανήκει σε κάποιο δίκτυο «ομίχλης» για κρίσιμες εργασίες, ώστε να εξασφαλισθεί η ταχύτερη δυνατή εκτέλεση τους. Όλες οι τιμές (MI, RAM και μέγεθος δεδομένων που συνοδεύουν κάθε εργασία) επιλέγονται τυχαία από μία Gaussian κατανομή. Η πολυπλοκότητα κάθε εργασίας εκφράζεται σε MI (Million instructions) και επιλέχθηκε από ένα εύρος τιμών 4500 – 5500

MIs. Οι απαιτήσεις κάθε εργασίας σε μνήμη RAM επιλέχθηκε από ένα εύρος τιμών από 150 έως 350 MBs. Η διαθέσιμη μνήμη συστήματος δεν λαμβάνεται υπόψιν στον αλγόριθμο κατανομής, μιας που δεν θεωρείται ότι επηρεάζει την εκτέλεση της εργασίας (εφόσον υπάρχει η ελάχιστη απαιτούμενη διαθέσιμη στο σύστημα). Παρ' όλα αυτά οι απαιτήσεις μνήμης κάθε εργασίας λαμβάνονται υπόψιν κατά τη διάρκεια του αρχικού φιλτραρίσματος του συνόλου των διαθέσιμων τερματικών, καθώς αν κάποιο τερματικό δεν διαθέτει την απαραίτητη μνήμη RAM για την εκτέλεση κάθε εργασίας, τότε αυτό αποκλείεται από τη λίστα έγκυρων τερματικών. Τέλος, το μέγεθος των δεδομένων που συνοδεύουν κάθε εργασία επιλέχθηκε από ένα εύρος τιμών από 1 έως 5 MBs.

- **Επίπεδο νέφους (Σενάρια 1,2,3).** Για όλα τα σενάρια που εξετάζονται θεωρείται ένα επίπεδο νέφους το οποίο αποτελεί το σημείο κατάληξης όλων των δεδομένων που παράγονται από τις IoT συσκευές. Το επίπεδο νέφους μοντελοποιείται με τρόπο που να καταδεικνύει τους περιορισμούς που εισάγονται σε μία κεντρικοποιημένη πλατφόρμα, η οποία πρέπει να εξυπηρετεί πολλαπλές ροές δεδομένων από πολλαπλά δίκτυα IoT. Ως εκ τούτου θεωρούμε ότι οι διαθέσιμοι πόροι προσφέρονται μέσω ενός εικονικού μηχανήματος το οποίο διαθέτει ένα επεξεργαστική ισχύ 2660 MIPS (Intel Xeon 3075) [113], 4096 MB μνήμης RAM, και 10 Mb/s διαθέσιμου δικτυακού εύρους ζώνης. Η δικτυακή καθυστέρηση θεωρείται στα 16 ms, μοντελοποιώντας μία περιοχή εύρους 300 με 500 km.
- **Επίπεδο ομίχλης (Σενάρια 2,3).** Κάθε δίκτυο «ομίχλης» (σταθερό ή γειτονικό) περιέχει τρία τερματικά με επεξεργαστική ισχύ 1256 (ARM Cortex v5), 1536 (ARM v7) και 847 (ARM11 family) [118] MIPS αντίστοιχα, προσομοιώνοντας διάφορες διάσημες συσκευές όπως οι εκδόσεις RaspberryPi 1 και 2 [119]. Κάθε τερματικό διαθέτει 1024 MB μνήμης. Με τρόπο αυτό αναδεικνύεται ο κύριος ρόλος του μοντέλου ομιχλώδους υπολογισμού, χρησιμοποιώντας συσκευές χαμηλής επεξεργαστικής δυνατότητας για να δημιουργήσει ένα βοηθητικό επίπεδο επεξεργασίας μεταξύ του επιπέδου νέφους και των δικτύων παραγωγής δεδομένων. Το διαθέσιμο δικτυακό εύρος ζώνης επιλέχθηκε από ένα εύρος τιμών από 2 έως 6 ms που αντιστοιχεί σε περιοχή κάλυψης έως 10 km.

- **Κινητές συσκευές (Σενάριο 3).** Στο 3^ο σενάριο μοντελοποιήθηκαν δύο συσκευές επεξεργαστικής δυνατότητας 2500 MIPS (Qualcomm Scorpion) [118], 1024 MB διαθέσιμης μνήμης RAM, 20 Mb/s δικτυακού εύρους ζώνης με δικτυακή καθυστέρηση από 6 έως 10 ms.

Για κάθε εργασία που φθάνει στον διαμεσολαβητή «ομίχλης», ο αλγόριθμος κατανομής δημιουργεί μία λίστα έγκυρων τερματικών τα οποία διαθέτουν την ελάχιστη απαιτούμενη μνήμη RAM και την ελάχιστη επιτρεπτή στάθμη μπαταρίας (στη περίπτωση κινητών συσκευών). Εν συνεχεία, εφαρμόζει τη συνάρτηση βαθμολόγησης για κάθε τερματικό και επιλέγει εκείνο που συγκεντρώνει την υψηλότερη βαθμολογία.

Για την εισαγωγή δυναμικών καταστάσεων στις προσομοιώσεις εισήχθησαν επιπρόσθετες παράμετροι. Συγκεκριμένα, επιτράπηκε μία μικρή πιθανότητα (<0.05) για κινητές συσκευές να βρεθούν εκτός σύνδεσης κάτι που θα έχει ως αποτέλεσμα την αποτυχία εκτέλεσης μία εργασίας και θα αναγκάσει τον καταναμητή να την ανακαταλείψει σε νέο τερματικό. Επίσης, η στάθμη μπαταρίας για κάθε κινητή συσκευή μειώνεται με σταθερό ρυθμό (0.001%) σε κάθε επανάληψη του αλγορίθμου. Το χαρακτηριστικό αυτό είναι ιδιαίτερης σημασίας καθώς όταν η στάθμη μπαταρίας μίας συσκευής πέφτει κάτω από το 20% τότε αυτή αποκλείεται από τον αλγόριθμο κατανομής. Οι παραπάνω παράμετροι επιχειρούν να εισάγουν την έννοια της τυχαιότητας στη συμπεριφορά των κινητών συσκευών, και να μας δώσουν μία εικόνα για τον τρόπο με τον οποίο αντιδρά η προτεινόμενη πλατφόρμα σε τέτοιες καταστάσεις.

Η απόδοση της πλατφόρμας εκτιμήθηκε μέσω οκτώ μετρήσεων για όλα τα σενάρια. Αυτά παρουσιάζονται στον παρακάτω πίνακα:

Πίνακας 2 Περιγραφή μετρήσεων προσομοιώσεων

1	Μέσος χρόνος εκτέλεσης εργασίας	Περιγράφει την κατά μέσο όρο χρονική διάρκεια, σε μονάδες χρόνου Cloudsim, που απαιτήθηκε για την επιτυχή εκτέλεση κάθε εργασίας
2	Μέση καθυστέρηση εκτέλεσης	Περιγράφει την κατά μέσο όρο χρονική καθυστέρηση, επίσης σε μονάδες χρόνου Cloudsim, που μετρήθηκε από τη στιγμή άφιξης κάθε εργασίας μέχρι τη στιγμή έναρξης εκτέλεσης της

3	Συνολικός χρόνος περάτωσης	Περιγράφει το συνολικό χρόνο που απαιτήθηκε για την επιτυχή εκτέλεση και των 5000 εργασιών
4	Αριθμός χρονικά κρίσιμων εργασιών	Περιγράφει τον ακριβή αριθμό των χαρακτηρισμένων χρονικά κρίσιμων εργασιών για κάθε σενάριο προσομοίωσης
5	Μέσος χρόνος εκτέλεσης χρονικά κρίσιμων εργασιών	Παρόμοια με τη μέτρηση 1 του πίνακα, όμως για το σύνολο μόνο των χρονικά κρίσιμων εργασιών
6	Αριθμός εργασιών που προωθήθηκαν στο επίπεδο νέφους	Περιγράφει τον ακριβή αριθμό των εργασιών που προωθήθηκαν στο επίπεδο νέφους
7	Αριθμός εργασιών που ανατέθηκαν σε κινητές συσκευές	Περιγράφει τον ακριβή αριθμό εργασιών τις οποίες ο αλγόριθμος κατανομής προώθησε σε κινητές συσκευές
8	Αριθμός αποτυχημένων και επανυποβληθέντων εργασιών	Περιγράφει τον ακριβή αριθμό εργασιών που απέτυχαν για οποιοδήποτε λόγο και επανυποβλήθηκαν σε μεταγενέστερη χρονική στιγμή για κατανομή σε κάποιο τερματικό

Ο Πίνακας 2 περιγράφει ένα σύνολο μετρήσεων οι οποίες συνδυαστικά μπορούν να δώσουν μία ξεκάθαρη εικόνα σχετικά με την απόδοση της πλατφόρμας σε σύγκριση με υπάρχουσες προσεγγίσεις. Ιδιαίτερο ενδιαφέρον παρουσιάζει η μέτρηση 5 η οποία σε συνδυασμό με τη μέτρηση 2 μπορεί να παρουσιάσει το πόσο αποτελεσματικά δουλεύει ο αλγόριθμος κατανομής και το πώς διαχειρίζεται κρίσιμες εργασίες. Επίσης, η μέτρηση 8 σε συνδυασμό με τις μετρήσεις 1 και 3 μας δίνουν μία εικόνα σχετικά με την ικανότητα διαχείρισης καταστάσεων όπου υπάρχουν κινητές συσκευές ή υπάρχει υπερβολικά αυξημένος φόρτος.

4.2.2 Παρουσίαση αποτελεσμάτων πειραματικής διαδικασίας

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, οι προσομοιώσεις οργανώθηκαν σε τρία σενάρια τα οποία επιχειρήσαν να συγκρίνουν την προτεινόμενη αρχιτεκτονική έναντι σε υπάρχουσες προσεγγίσεις. Στο 1ο σενάριο εξετάζεται η

λειτουργία της πλατφόρμας IoT με την ύπαρξη μόνο του επιπέδου νέφους, δίχως να περιλαμβάνεται οποιαδήποτε λειτουργία «ομιχλώδους» πλατφόρμας. Στο 2^ο σενάριο εξετάζεται η λειτουργία της IoT πλατφόρμας, με τη χρησιμοποίηση ενός επιπρόσθετου ενδιάμεσου επιπέδου «ομίχλης» αυτή τη φορά. Τέλος, το 3^ο σενάριο χωρίζεται σε δύο ξεχωριστά υπό-σενάρια τα οποία εξετάζουν την προτεινόμενη αρχιτεκτονική. Το 1^ο υπό-σενάριο παρουσιάζει την λειτουργία της εν λόγω πλατφόρμας με τη συνεργασία του σταθερού δικτύου «ομίχλης» με δύο γειτονικά δίκτυα. Το 2^ο υπό-σενάριο παρουσιάζει την ίδια ακριβώς προσέγγιση με τη προσθήκη δύο κινητών συσκευών αυτή τη φορά. Αξίζει να σημειωθεί πως το επίπεδο «νέφους» περιλαμβάνεται σε όλα τα σενάρια. Αντίθετα με τις υπάρχουσες προσεγγίσεις «ομιχλώδους» υπολογισμού, η προτεινόμενη αρχιτεκτονική συμπεριλαμβάνει το επίπεδο νέφους όχι ως τελικό αποθετήριο για την μακροπρόθεσμη επεξεργασία και ανάλυση των παραγόμενων δεδομένων, αλλά και ως ένα ακόμα τερματικό το οποίο μπορεί να συνεισφέρει στη άμεση εκτέλεση εργασιών IoT εάν και εφόσον αυτό κριθεί απαραίτητο.

Τα αποτελέσματα των εκτελέσεων του προσομοιωτή παρουσιάζονται στον Πίνακα 3:

Πίνακας 3 Παρουσίαση αποτελεσμάτων προσομοιώσεων²¹

		Μέσος χρόνος εκτέλεσης εργασιών	Μέση χρον. καθυστέρηση	Χρόνος περάτωσης	Αρ. κρίσιμων εργασιών	Μέση χρον. καθυστέρηση κρίσιμων εργασιών	Αρ. εργασιών προς επίπεδο «νέφους»	Αρ. εργασιών προς κινητές συσκευές	Αρ. αποτυχημένων εργασιών
Σενάριο 1	1	18,22	2,07	9323	249	14,82	-	-	-
	2	12,82	0,37	6770	248	10,88	108 (2,16%)	-	-
Σενάριο 3	1	3,44	0,1	5407	278	2,78	105 (2,1%)	-	-
	2	3,35	0,145	5470	237	2.85	110 (2,19%)	334 (6,68%)	21

²¹ Τα ονόματα των μετρήσεων του Πίνακα 3 αντιστοιχούν κατά σειρά στη περιγραφή των μετρήσεων του Πίνακα 2

4.2.2.1 Σενάριο 1

Το 1ο σενάριο παρουσιάζει τα αποτελέσματα προσομοίωσης μίας κεντρικής δομής η οποία αντιστοιχεί στο επίπεδο «νέφους». Το σενάριο αυτό προσομοιώνει επίσης ένα χαρακτηριστικό περιβάλλον όπου η πλατφόρμα υποδοχής και ανάλυσης δεδομένων δεν διαθέτει τους απαραίτητους υπολογιστικούς πόρους ούτε το απαραίτητο δικτυακό εύρος ζώνης ώστε να φέρει εις πέρας το φόρτο εργασίας από ενδεχόμενες πολλαπλές ροές δεδομένων. Έτσι, στο συγκεκριμένο σενάριο, για τη ροή δεδομένων που εξετάζεται (5000 εργασίες), παρατηρείται συμφόρηση των εργασιών. Αυτό φαίνεται κυρίως από τον μέσο χρόνο εκτέλεσης κάθε εργασίας (18,22 μονάδες χρόνου), γεγονός που εξηγείται και από τη μέση καθυστέρηση (2,07 μονάδες χρόνου για κάθε εργασία). Η έλλειψη διαθέσιμων υπολογιστικών και δικτυακών πόρων δημιουργεί μία ουρά από εργασίες οι οποίες αναμένουν την εκτέλεση τους, και εφόσον η εκτέλεση αυτή εκκινήσει καθυστερεί σημαντικά λόγω του διαμοιρασμού των πόρων με άλλες εργασίες (ακολουθείται time-sharing μοντέλο). Αντίστοιχες χαμηλής απόδοσης μετρήσεις παρατηρούνται και για τις χρονικά κρίσιμες εργασίες (14,82 μέσος χρόνος εκτέλεσης), μιας και ελλείπει συγκεκριμένης μεθόδου κατανομής, εκτελούνται με τον ίδιο ακριβώς τρόπο όπως οι υπόλοιπες. Όλα τα παραπάνω έχουν ως αποτέλεσμα τον μεγάλο χρόνο περάτωσης της προσομοίωσης (9323 μονάδες χρόνου).

4.2.2.2 Σενάριο 2

Στο σενάριο 2 παρουσιάζεται η παραδοσιακή προσέγγιση «ομιχλώδους» αρχιτεκτονικής. Ανάμεσα στο δίκτυο συσκευών IoT και στο τελικό επίπεδο νέφους εισέρχεται το επίπεδο «ομίχλης», όπου αναλαμβάνει την άμεση επεξεργασία των δεδομένων. Στο σενάριο αυτό μόνο οι εργασίες αυξημένης πολυπλοκότητας με ιδιαίτερες απαιτήσεις σε επεξεργαστική ισχύ αποστέλλονται απευθείας στο επίπεδο «νέφους». Οι υπόλοιπες εκτελούνται στο επίπεδο «ομίχλης» και τα αποτελέσματα τους αποστέλλονται πίσω στο δίκτυο συσκευών, αλλά και στο επίπεδο νέφους για αποθήκευση και μεταγενέστερη επεξεργασία.

Όπως ξεκάθαρα παρατηρείται, η λειτουργία της προσομοιούμενης IoT πλατφόρμας βελτιώνεται αισθητά. Παρατηρείται σημαντικά χαμηλότερος μέσος χρόνος εκτέλεσης (12,82) καθώς επίσης και σημαντικά χαμηλότερη μέση καθυστέρηση (0,37). Αντίστοιχα χαμηλότερος είναι και ο συνολικός χρόνος περάτωσης (6770). Δεν παρατηρήθηκε αξιοσημείωτη διαφορά στο μέσο χρόνο εκτέλεσης κρίσιμων χρονικά εργασιών, γεγονός το οποίο ήταν αναμενόμενο καθώς και σε αυτό το σενάριο δεν χρησιμοποιήθηκε κάποια στρατηγική κατανομής. Περίπου το 2% των εργασιών προωθήθηκε προς το επίπεδο νέφους. Η βελτιωμένη απόδοση οφείλεται βεβαίως στην

λειτουργία του ενδιάμεσου επιπέδου «ομίχλης» καθώς παρείχε τρία τερματικά και έδινε τη δυνατότητα εκτέλεσης πολλαπλών εργασιών παράλληλα. Επίσης, πιθανά χρονοβόρες εργασίες προωθήθηκαν απευθείας στη κεντρική πλατφόρμα νέφους ώστε να μην δεσμεύουν για μεγάλο χρονικό διάστημα τα τερματικά του επιπέδου «ομίχλης» λόγω της αυξημένης πολυπλοκότητας τους.

4.2.2.3 Σενάριο 3

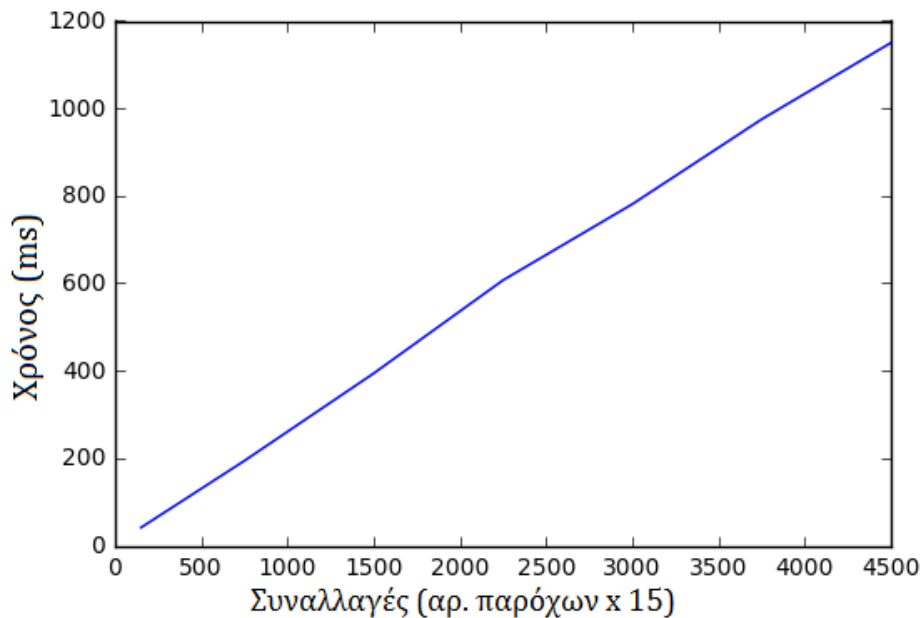
Στο τρίτο σενάριο παρουσιάζεται η λειτουργία της προτεινόμενης αρχιτεκτονικής προσέγγισης «ομιχλώδους» υπολογισμού. Αρχικά στο πρώτο υπό-σενάριο εισάγονται δύο συνεργατικά δίκτυα «ομίχλης» με το ίδιο αριθμό διαθέσιμων τερματικών και τις ίδιες υπολογιστικές δυνατότητες. Σε αυτή τη περίπτωση εξετάζεται και η λειτουργία του προτεινόμενου αλγορίθμου κατανομής εργασιών. Παρατηρείται ότι η προτεινόμενη προσέγγιση αποδίδει πολύ καλύτερα σε όλες τις μετρήσεις χρόνων. Ο μέσος χρόνος εκτέλεσης είναι πολύ χαμηλότερος (3,44) και η μέση καθυστέρηση κινείται σε αμελητέα επίπεδα (0,1). Ο μέσος χρόνος εκτέλεσης χρονικά κρίσιμων διεργασιών είναι και αυτός ιδιαίτερα ικανοποιητικός (2,78), ενώ και ο συνολικός χρόνος προσομοίωσης μειώνεται κατά περίπου 20% (5407).

Ιδιαίτερο ενδιαφέρον παρουσιάζει και το υπό-σενάριο 3.2, καθώς αυτή τη φορά μετρήθηκε η απόδοση της προτεινόμενης αρχιτεκτονικής με τη προσθήκη δύο κινητών συσκευών. Η ύπαρξη κινητών συσκευών είχε ως αποτέλεσμα την αποτυχία ενός αριθμού εργασιών που είχαν ανατεθεί στις εν λόγω συσκευές, καθώς κατά την εκτέλεση τους οι συσκευές αυτές βρέθηκαν εκτός σύνδεσης ή έπαυσαν να λειτουργούν λόγω χαμηλής στάθμης μπαταρίας. Παρότι ένα ποσοστό των εργασιών απέτυχε (0,4%) και χρειάστηκε να ανακατανεμηθεί, παρατηρήθηκε ανεπαίσθητη μείωση της απόδοσης ως προς τον χρόνο της πλατφόρμας (3,35, 0,145 και 2,85 αντίστοιχα), γεγονός υποδεικνύει τη χρησιμότητα των κινητών συσκευών εφόσον υπάρχει μία ευέλικτη λειτουργία κατανομής φόρτου εξασφαλίζοντας κάθε στιγμή ότι κάθε εργασία θα κατανέμεται στο κατάλληλο τερματικό με σκοπό τη άμεση ολοκλήρωση της εκτέλεσης της.

4.3 Συστήματα «ομόσπονδων» δίκτυων υποδομών υπολογιστικών νεφών με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA)

Για την αξιολόγηση του προτεινόμενου συστήματος χρησιμοποιήθηκε η βιβλιοθήκη υλοποίησης του συστήματος Glicko [107], η οποία αναπτύχθηκε με τη γλώσσα προγραμματισμού Python. Η χαμηλή πολυπλοκότητα και το γεγονός ότι εκτελείται περιοδικά και ανεξάρτητα, δεν επιβαρύνει τη λειτουργία των υπόλοιπων υπηρεσιών και πρακτόρων της πλατφόρμας ενδιαμέσου. Το χαρακτηριστικό αυτό είναι

ιδιαίτερος επιθυμητό στις περιπτώσεις συγκεντρωτικών αρχιτεκτονικών. Για την παρουσίαση της απόδοσης της διαδικασίας ενημέρωσης των βαθμολογιών, εξετάστηκαν μερικά τυχαία σενάρια με διαφορετικούς αριθμούς παρόχων και συναλλαγών και μετρήθηκε ο συνολικός χρόνος ενημέρωσης του παραμετροποιημένου συστήματος Glicko. Οι επιλογές των ποσοτήτων r_{max} , RD_{max} και c έγιναν με βάση τις προτάσεις που παρουσιάζονται στη περιγραφή του συστήματος Glicko. Στη συνέχεια, οι προσομοιώσεις κλιμακώθηκαν με βάση τον αριθμό των παρόχων και θεωρώντας ότι κάθε πάροχος συμμετείχε σε 15 συναλλαγές περίπου. Στην παρούσα διαδικασία προσομοιώσεων δεν συμπεριλήφθηκε η διαδικασία βαθμολόγησης του αιτήματος υπηρεσίας καθώς και αυτή είναι ανεξάρτητη και δεν επηρεάζει τη κύρια λειτουργία του συστήματος βαθμολόγησης. Η Εικόνα 20 δείχνει τη γραμμική φύση της διαδικασίας όσον αφορά τη κλιμακωσιμότητα της.



Εικόνα 20 Απόκριση της διαδικασίας βαθμολόγησης έναντι σε διαφορετικά σενάρια αριθμού παρόχων και συναλλαγών

4.4 Συστήματα διαχείρισης πόρων και υποδομών υπολογιστικών νεφών με έμφαση στην ενεργειακή αποδοτικότητα

4.4.1 Περιγραφή πειραματικής διαδικασίας

Η προτεινόμενη αρχιτεκτονικής υβριδικής εικονικοποίησης αξιολογείται μέσω προγράμματος προσομοιωτή βασισμένου στη πλατφόρμα Cloudsim. Ο προσομοιωτής επεκτείνει τη λειτουργία του Cloudsim, ώστε να προσφέρει την υποστήριξη για περιέκτες λογισμικού οι οποίοι λειτουργούν απευθείας στο εξυπηρετητή. Η έκδοση 4.0 του

Cloudsim προσφέρει λειτουργικότητα για περιέκτες λογισμικού, οι οποίοι λειτουργούν μέσα σε εικονικά μηχανήματα. Η λειτουργία περιεκτών λογισμικού απευθείας στον «φυσικό» εξυπηρετητή ήταν αναγκαία ώστε να πραγματοποιηθεί μία άμεση μελέτη των διαφορών των δύο τεχνολογιών και το πόσο επιβαρύνουν το υπερκείμενο λειτουργικό σύστημα του εξυπηρετητή. Επιχειρήθηκε η προσομοίωση σεναρίων που καλύπτουν διάφορα μεγέθη υπολογιστικών κέντρων ώστε να αξιολογηθεί η κλιμακωσιμότητα της προτεινόμενης πλατφόρμας. Η προτεινόμενη προσέγγιση αξιολογήθηκε σε σύγκριση με δύο παραδοσιακές μεθόδους κατανομής εικονικών μηχανημάτων τεχνολογίας υπερεπόπτη σε υπολογιστικά κέντρα παροχής υπηρεσιών IaaS. Η πρώτη μέθοδος καταναίμει τα εικονικά μηχανήματα σε εξυπηρετητές ακολουθώντας μία στρατηγική βέλτιστης προσπάθειας, κατανέμοντας κάθε εικονικό μηχάνημα στον πρώτο διαθέσιμο εξυπηρετητή που έχει τους ελάχιστους απαραίτητους πόρους. Εν συνεχεία, μέσω περιοδικής εποπτείας, ανακατανέμει εικονικά μηχανήματα από εξυπηρετητές που υποχρησιμοποιούνται σε άλλους εξυπηρετητές ώστε να επιτύχει καλύτερη συνολική χρησιμοποίηση (utilization) της υπολογιστικής υποδομής και να εξοικονομήσει περισσότερη ενέργεια [5]. Η δεύτερη μέθοδο δεν εφαρμόζει κάποια διαδικασία μεταφοράς εικονικών μηχανημάτων (migration), αλλά αντ' αυτού κατανέμει τα μηχανήματα ακολουθώντας την ίδια στρατηγική βέλτιστης προσπάθειας με την προηγούμενη μέθοδο.

Αντίθετα, η προτεινόμενη «υβριδική» μέθοδος μοντελοποιεί ένα υπολογιστικό κέντρο όπου ένα μικρό μέρος των εξυπηρετητών υποστηρίζει εικονικοποίηση τεχνολογίας περιέκτη. Σε όλα τα σενάρια αξιολόγησης κάθε εικονικό μηχάνημα χαρακτηρίζεται ανάλογα με το τύπο της εφαρμογής την οποία θα φιλοξενήσει σύμφωνα με το αίτημα του τελικού χρήστη. Με αυτό τον τρόπο χρησιμοποιούνται τέσσερις τύποι λειτουργίας:

- **hpc**, Το οποίο υποδηλώνει ότι το εικονικό μηχάνημα θα χρησιμοποιηθεί για τη λειτουργία επιστημονικής εφαρμογής. Τέτοιου είδους εφαρμογές (high performance computing) συνήθως απαιτούν πολλαπλούς και υψηλών προδιαγραφών πόρους. Επιπρόσθετα, η πορεία της εφαρμογής δεν είναι δυνατό να διακοπεί και να επανεκκινήσει από το ίδιο σημείο χωρίς να χαθεί η πρόοδος της. Για αυτούς τους λόγους τέτοιου είδους εφαρμογές είναι απαραίτητο να λειτουργούν στα πλαίσια εικονικών μηχανημάτων υπερεπόπτη, ώστε να εγγυάται η διαθεσιμότητα των αιτούμενων πόρων

- **service**, Το πεδίο αυτό υποδηλώνει ότι το εικονικό μηχάνημα θα φιλοξενήσει υπηρεσίες λογισμικού. Οι διαδικτυακές υπηρεσίες χαρακτηρίζονται από την έλλειψη οποιασδήποτε κατάστασης (stateless), γεγονός που τις καθιστά ευέλικτες στο τρόπο λειτουργίας τους. Οι υπηρεσίες λογισμικού μπορούν να διακόπτουν και να συνεχίζουν τη λειτουργία τους χωρίς να απαιτείται κάποια παραμετροποίηση. Τα χαρακτηριστικά τους αυτά τις καθιστούν ιδανικές για λειτουργία στα πλαίσια περιεκτών λογισμικού
- **agent**, το οποίο υποδηλώνει ότι το εικονικό περιβάλλον θα φιλοξενεί ένα μέρος ενός κατανεμημένου πολυπρακτορικού συστήματος. Οι πράκτορες λογισμικού αποτελούν «σπονδυλωτά» τμήματα μία κατανεμημένης πλατφόρμας, οι οποίοι συνεργάζονται για την επίτευξη ενός κοινού στόχου. Παρόμοια, με τις υπηρεσίες λογισμικού οι πράκτορες λογισμικού χαρακτηρίζονται επίσης από την απουσία κατάστασης λειτουργίας (stateless). Συνεπώς, και αυτού του τύπου οι εφαρμογές ενδείκνυνται για λειτουργία εντός ενός περιέκτη λογισμικού
- **generic**, το οποίο υποδηλώνει ότι το εικονικό μηχάνημα προορίζεται για γενική χρήση (π.χ. αποθήκευση δεδομένων). Και σε αυτή τη περίπτωση το η λειτουργία μπορεί να επιτευχθεί εντός των πλαισίων ενός περιέκτη

Για την επίτευξη καλύτερης αξιοπιστίας στα πλαίσια της πειραματικής διαδικασίας θεωρήθηκε ότι σε όλα τα σενάρια δεν χρησιμοποιήθηκε καμία τεχνική άμεσης μεταφοράς (live migration) εικονικών μηχανημάτων τεχνολογίας υπερεπόπτη. Σε όλα τα σενάρια θεωρήθηκε ότι η μεταφορά εικονικών μηχανημάτων οποιαδήποτε τεχνολογίας εικονικοποίησης, γίνεται με τη βοήθεια εργαλείων αυτοματισμού λειτουργικών συστημάτων (π.χ. Ansible [6]). Με αυτό τον τρόπο τα δεδομένα, οι βιβλιοθήκες και το συνοδευτικό λογισμικό του λειτουργικού συστήματος αποθηκεύονται σε ειδικά προτυποποιημένα αρχεία τα οποία χρησιμοποιούνται για να εγκαταστήσουν και να διαμορφώσουν το νέο λειτουργικό σύστημα ώστε να προκύψει ένα παρόμοιο περιβάλλον και ώστε να επανεκκινηθεί η εφαρμογή του τελικού χρήστη.

4.4.1.1 Σύνθεση και παραμετροποίηση προσομοιώσεων

Η πλήρης παραμετροποίηση και περιγραφή των σεναρίων των προσομοιώσεων παρουσιάζονται στον Πίνακα 44, όπου παρουσιάζονται ο ακριβής αριθμός των εικονικών μηχανημάτων και εξυπηρετητών για κάθε μέγεθος υπολογιστικού κέντρου. Στο «υβριδικό» σενάριο ο συνολικός αριθμός δίνεται ο συνολικός αριθμός εξυπηρετητών στη μορφή *Αριθμός εξυπηρετητών υπερεπόπτη + Αριθμός εξυπηρετητών περιεκτών* για να

διαχωριστούν οι ομάδες των εξυπηρετητών ανάλογα με τη τεχνολογία εικονικοποίησης που υποστηρίζουν.

Πίνακας 4 Διαμόρφωση σεναρίων προσομοιώσεων πλατφόρμας υβριδικής εικονικοποίησης

Μέγεθος υπολογιστικού κέντρου	Αριθμός εξυπηρετητών		Αριθμός VM
	Τεχνολογία εικονικοποίησης		
	Τεχνολογία υπερεπόπτη	Υβριδική προσέγγιση	
Μικρό	5	4+1	30
Μεσαίο	30	27+3	150
Μεγάλο	150	130+20	1000

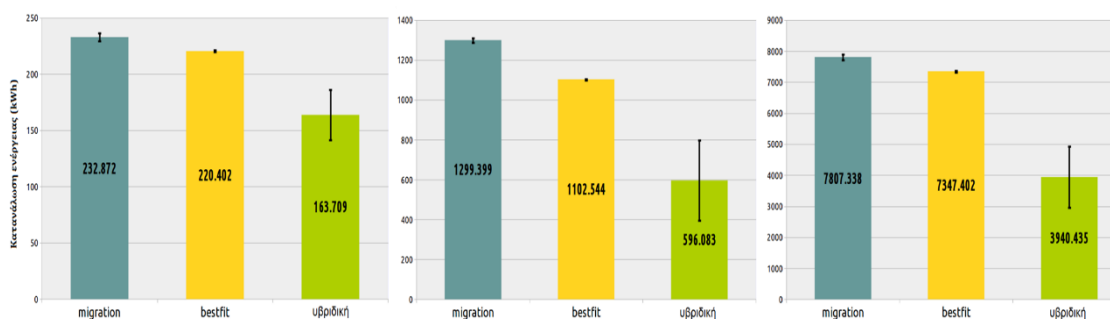
Για κάθε στρατηγική εικονικοποίησης και μέγεθος υπολογιστικού κέντρου διενεργήθηκαν 10 προσομοιώσεων καταλήγοντας σε ένα συνολικό αριθμό 90 ξεχωριστών προσομοιώσεων. Η συνολικός χρόνος για κάθε εκτέλεση του προσομοιωτή αντιστοιχεί σε ένα μήνα σε μονάδες χρόνου Cloudsim. Για κάθε εκτέλεση μετράται η συνολική κατανάλωση ενέργειας, συνολική χρησιμοποίηση (utilization) του υπολογιστικού κέντρου και η παραβίαση SLA. Το υπολογιστικό κέντρο που μοντελοποιείται είναι ομοιογενές σε όλα τα σενάρια και περιέχει εξυπηρετητές που εξομοιώνουν τη λειτουργία των μοντέλων HP ProLiant ML110 G4 με 2 μονάδες επεξεργασίας. Και σε αυτή τη περίπτωση χρησιμοποιείται η μονάδα μέτρησης MIPS (Million instructions per second) για τη μέτρηση της υπολογιστικής δυνατότητας επεξεργαστών στη πλατφόρμα Cloudsim. Κάθε εξυπηρετητής διαθέτει 2 μονάδες επεξεργασίας CPU (2x2500 MIPS), 128 GB RAM, 1 TB αποθηκευτικού χώρου και 100Gb/s διαθέσιμο εύρος ζώνης. Το μοντέλο κατανάλωσης ισχύος βασίστηκε στις μετρήσεις SPECpower_ssj2008 [78], συμπεριλαμβάνεται εκ των προτέρων στη πλατφόρμα Cloudsim και υιοθετήθηκε στις επεκτάσεις που δημιουργήθηκαν για τις ανάγκες των προσομοιώσεων.

Κάθε εικονικό μηχάνημα τεχνολογίας υπερεπόπτη αιτήθηκε μίας μονάδας επεξεργαστή χωρητικότητας 500 MIPS, 512 MB RAM, 1 GB αποθηκευτικού και 100 Mb/s δικτυακού εύρους ζώνης. Σε όλα τα σενάρια το 20% του αριθμού των εικονικών μηχανημάτων χαρακτηρίστηκαν ως «hrc», γεγονός που τα κατέστησε μη διαθέσιμα για μετατροπή από εικονικά μηχανήματα υπερεπόπτη σε εικονικά περιβάλλοντα τύπου περιέκτη λογισμικού. Τα εικονικά περιβάλλοντα μοιράζονταν τους διαθέσιμους επεξεργαστικούς πόρους του εξυπηρετητή ακολουθώντας το μοντέλο time-shared.

Η πλατφόρμα Cloudsim προσομοίωσε σταθερό φόρτο εργασίας για τα εικονικά μηχανήματα βασισμένο σε ένα στοχαστικό μοντέλο. Ο φόρτος εργασίας κατανάλωνε κυρίως πόρους επεξεργαστή. Για τη διαφοροποίηση της απόδοσης μεταξύ των δύο τεχνολογιών εισάχθηκε ένα ποσοστό της τάξης του 1% ως επιβάρυνση φόρτου εργασίας στα εικονικά μηχανήματα τεχνολογίας υπερεπόπτη. Με βάση τα αποτελέσματα από τις εργασίες [79] και [120] επάνω σε εκτενή και απαιτητικά φορτία εργασιών, προκύπτει ότι και οι δύο τεχνολογίες έχουν ωριμάσει αρκετά και οι διαφορές στην απόδοση, αν και προφανείς, είναι πολύ μικρές. Η βασική διαφορά παραμένει η εξομοίωση εικονικού υλισμικού που πραγματοποιεί ο υπερεπόπτης μεταξύ εικονικού και «φυσικού» μηχανήματος.

4.4.2 Παρουσίαση αποτελεσμάτων πειραματικής διαδικασίας

Ο τρόπος με τον οποίο πραγματοποιήθηκαν οι προσομοιώσεις σεναρίων της προτεινόμενης αρχιτεκτονικής αντιστοιχεί φυσικά στο πραγματικό τρόπο λειτουργίας των περιεκτών λογισμικού. Στην Εικόνα 21 παρουσιάζεται η συνολική κατανάλωση ενέργειας για όλες τις στρατηγικές και μεγέθη υπολογιστικών κέντρων στη προσομοιούμενη διάρκεια ενός μήνα. Επιπρόσθετα, στον Πίνακα 5 παρουσιάζονται χαρακτηριστικά αποτελέσματα μίας εκτέλεσης του προσομοιωτή, σχετικά με την απόδοση SLA και την μέση χρησιμοποίηση πόρων του υπολογιστικού κέντρου.



Εικόνα 21 Συνολική κατανάλωση ενέργειας ανάλογα με την εφαρμοσμένη στρατηγική κατανομής

Τα αποτελέσματα που συλλέχθηκαν δείχνουν ότι σε σύγκριση με τις δύο παραδοσιακές στρατηγικές κατανομής, η «υβριδική» προσέγγιση παρουσιάζει 38% καλύτερη ενεργειακή αποδοτικότητα για μικρά υπολογιστικά κέντρα, ενώ οι διαφορές αυτές είναι μεγαλύτερες για υπολογιστικά κέντρα μεσαίου και μεγάλου μεγέθους. Όσον αφορά τη τήρηση των SLA οι στρατηγικές bestfit και hybrid δεν επιβάλλουν κάποια παραβίαση, ενώ η πρώτη προσέγγιση επιβάλλει μία παραβίαση που κυμαίνεται σε επίπεδα από 1 έως 3% του συνολικού χρόνου. Αυτό ήταν αναμενόμενο, καθώς οι παραβιάσεις SLA παρατηρήθηκαν όταν χρειάστηκε να μεταφερθούν εικονικά μηχανήματα τύπου «hrc», όπου δεν ήταν δυνατόν να γίνει μεταφορά σε περιέκτη και ως

εκ τούτου έπρεπε να γίνει άμεση μεταφορά (live migration) εν ώρα λειτουργίας, κάτι που με βάση μοντέλα που παρουσιάζονται στην εργασία [5] επιβάρυναν περίπου κατά 10% τη λειτουργία του εικονικού μηχανήματος για συνολική διάρκεια ανάλογη με το μέγεθος της εικόνας (image/snapshot) του εικονικού μηχανήματος που χρειάστηκε ώστε να μεταφερθεί από έναν εξυπηρετητή σε έναν άλλο. Η στρατηγική bestfit δεν χρησιμοποίησε κανενός είδους μεταφορές, ενώ η υβριδική χρησιμοποίησε μεταφορές με τη βοήθεια εργαλείων αυτοματισμού και μόνο για εικονικά μηχανήματα τα οποία δεν είναι χαρακτηρισμένα ως «hrc».

Η «υβριδική» στρατηγική και η συνολική αρχιτεκτονική της πλατφόρμας παρουσιάζει τα πλεονεκτήματα των σύγχρονων τεχνολογιών εικονικοποίησης και διαμόρφωσης. Από τη μία πλευρά, τα εργαλεία αυτοματισμού όπως Ansible [6], Puppet [121] και Cloud-init [97] παρέχουν τα χαρακτηριστικά για την αυτόματη διαμόρφωση των λειτουργικών συστημάτων των εικονικών μηχανημάτων μπορούν να προτυποποιηθούν και να συνοψιστούν σε ευέλικτα, διαχειρίσιμα και ανταλλάξιμα αρχεία. Με αυτό τον τρόπο σε πολλές περιπτώσεις μπορεί να καταστήσει μη αναγκαία τη μεταφορά ενός μεγάλου εικονικού μηχανήματος μεταξύ εξυπηρετητών. Όλα τα παραπάνω, σε συνδυασμό με την ανερχόμενη τεχνολογία περιεκτών λογισμικού (π.χ. Docker [35]) παρέχουν τις προδιαγραφές για μία περισσότερο ευέλικτη και αποδοτικότερη διαχείριση πόρων σε υπολογιστικά κέντρα παροχής υπηρεσιών νέφους όλων των επιπέδων (IaaS, PaaS ή SaaS).

Πίνακας 5 Παρουσίαση αποτελεσμάτων προσομοιώσεων σχετικά με την κατανάλωση ενέργειας, παραβίασης SLA και μέσο αριθμό ενεργών εξυπηρετητών κατά τη διάρκεια μίας εκτέλεσης του προσομοιωτή

Μέγεθος υπολογιστικού κέντρου	Συνολική κατανάλωση ενέργειας (kWh)		
	Παραβίαση SLA (%)		
	Μέσος αριθμός ενεργών εξυπηρετητών (%)		
	<i>Migration</i>	<i>BestFit</i>	<i>Υβριδική</i>
Μικρό	231,8	219,75	146,84
	6,34	-	-
	40	60	40
Μεσαίο	1308,73	1105,28	662,31
	2,72	-	-
	66	50	29
Μεγάλο	7852,81	7340,03	3768,64
	2,29	-	-
	76	67	33

Η προτεινόμενη πλατφόρμα, με βάση τα αποτελέσματα των προσομοιώσεων αποδίδει καλύτερα για δύο λόγους. Αρχικά, οι περιέκτες λογισμικού δεν επιβαρύνουν το υπερκείμενο λειτουργικό σύστημα, καθώς επί της ουσίας η εφαρμογή που φιλοξενούν

εμφανίζεται ως μία ξεχωριστή διεργασία. Το γεγονός αυτό, εγείρει συγκεκριμένα ζητήματα ασφαλείας, αλλά από άποψη αποδοτικότητας οι περιέκτες στη παρούσα στιγμή αποτελούν τη βέλτιστη επιλογή. Η μεταφορά εικονικών μηχανημάτων τεχνολογίας υπερεπόπτη σε περιέκτες λογισμικού οδήγησε σε αποδοτικότερη κατανάλωση πόρων του εξυπηρετητή, κάτι το οποίο σημαίνει ότι λιγότερη ισχύς χρειάστηκε για τον ίδιο φόρτο εργασίας.

Επιπρόσθετα, ένα από τα κύρια χαρακτηριστικά των περιεκτών λογισμικού είναι η μη ανάγκη κράτησης πόρων εξ' αρχής καθώς επί της ουσίας περικλείουν μία και μόνο εφαρμογή. Ως εκ' τούτου η κατανάλωση πόρων εξαρτάται αποκλειστικά από τη λειτουργία της εφαρμογής. Συνεπώς, δίνεται η δυνατότητα φιλοξενίας περισσότερων περιεκτών εξυπηρετητών σε σχέση με εικονικά μηχανήματα τεχνολογίας υπερεπόπτη, κάτι που οδηγεί στη χρησιμοποίηση συνολικά μικρότερου αριθμού εξυπηρετητών. Όπως δείχνουν και τα αποτελέσματα των προσομοιώσεων κατά την εκτέλεση των σεναρίων «υβριδικής» στρατηγικής, προς το τέλος του χρόνου προσομοίωσης μόνο τα εικονικά μηχανήματα τύπου «hrc» παρέμειναν στην ίδια κατάσταση χωρίς να διακοπεί η λειτουργία της περικλειόμενης εφαρμογής.

Κεφάλαιο 5

Συμπεράσματα και προτάσεις για μελλοντική έρευνα

Η παρούσα διατριβή παρουσιάζει μία ολοκληρωμένη πρόταση αρχιτεκτονικών και συστημάτων οι οποίες αντιμετωπίζουν σύγχρονα και κρίσιμα ζητήματα, που προκύπτουν από τη διαχείριση υποδομών παρόχων υπηρεσιών «νέφους». Οι αρχιτεκτονικές προτάσεις επικεντρώθηκαν στη δημιουργία νέων και βελτίωση υπάρχουσών δομών, ώστε οι προσφερόμενες υπηρεσίες να επιτυγχάνουν υψηλής ποιότητας υπηρεσία, αλλά και οι πάροχοι να είναι σε θέση να διατηρούν υψηλή απόδοση SLA, παραμένοντας ανταγωνιστικοί. Επιπρόσθετα, θεωρήθηκε εξίσου κρίσιμη και άμεσου ενδιαφέροντος, η αποδοτική διαχείριση της λειτουργίας της υποδομής με σκοπό την πιο αποδοτική και οικονομική χρήση της. Εκτός από τον άμεσο αντίκτυπο που θα παρατηρηθεί στην ποιότητα των υπηρεσιών, μια αποδοτική διαχείριση, όσον αφορά τη κατανάλωση ενέργειας, συνολικά μπορεί να συμβάλει σε μία πιο φιλική προς το περιβάλλον προσέγγιση μίας αγοράς που επεκτείνεται συνεχώς, και προβλέπεται να αποτελέσει έναν από τους κύριους τομείς κατανάλωσης ενέργειας.

Τα συμπεράσματα που εξήχθησαν από τη μελέτη των ζητημάτων παρουσιάζεται σε αυτό το κεφάλαιο. Επίσης, σε συνέχεια των συμπερασμάτων, κρίθηκε απολύτως χρήσιμο να παρουσιαστούν ουσιαστικές προτάσεις για μελλοντική έρευνα, με βάση τα αποτελέσματα και έχοντας ως γνώμονα τις σύγχρονες τάσεις και απαιτήσεις των συναφών τεχνολογικών και ερευνητικών πεδίων.

5.1 Συστήματα «νεφελώδους» και «ομιχλώδους» υπολογισμού με έμφαση στη ποιότητα υπηρεσίας

Η αποτελεσματική διαχείριση πόρων και δεδομένων στο πεδίο των υπολογιστικών νεφών, αλλά και στο ταχεία εξελισσόμενο πεδίο των υπηρεσιών «ομίχλης», θα αποτελεί συνεχώς ένα κρίσιμο ζητούμενο, του οποίου η επίτευξη θα είναι καταλυτικής σημασίας για την ενεργοποίηση όλο και περισσότερων μοντέλων, εφαρμογών και υπηρεσιών που θα λειτουργούν σε υπολογιστικά νέφη. Οι πλατφόρμες Διαδικτύου των Αντικειμένων (IoT) αποτελούν ένα χαρακτηριστικό παράδειγμα εφαρμογών που αναπτύσσεται σε εξάρτηση με τη πρόοδο των υπολογιστικών «νεφών», και η οι λειτουργία τους είναι άμεσα συνδεδεμένη με τη λειτουργία του υπολογιστικού «νέφους» ως μοντέλο. Η εμφάνιση ενός νέου μοντέλου υπηρεσίας, το οποίο θα λειτουργεί

ως ένα ενδιάμεσο υποστηρικτικό επίπεδο, ανάμεσα στο «νέφος» και στην πηγή των δεδομένων, καταδεικνύει την ανάγκη για την εξασφάλιση της βέλτιστης δυνατής λειτουργίας των εφαρμογών «νέφους».

Ως μέρος της παρούσας διατριβής, προτάθηκε μία αρχιτεκτονική πλατφόρμας «ομιχλώδους» υπολογισμού, ως άμεση επέκταση των υπηρεσιών μία πλατφόρμας που λειτουργεί σε υπολογιστικό «νέφος». Η προτεινόμενη λύση εκμεταλλεύεται τις δυνατότητες που προσφέρει το μοντέλο «ομιχλώδους» υπολογισμού, και το επεκτείνει. Σε αυτό το πλαίσιο, προτάθηκε η εισαγωγή διαχείρισης και κατανομής εργασιών και μοντελοποιήθηκε πλήρως ένας σχετικός μηχανισμός μαζί με έναν ευφυή αλγόριθμο που εξασφαλίζει τη γρήγορη κατανομή των σχετικών εργασιών σε τερματικά «ομίχλης». Επίσης, για την υποστήριξη διαλειτουργικότητας με γειτονικά δίκτυα, προτάθηκε αρχιτεκτονική δημοσίευσης/συνδρομής (Pub/Sub), με τη χρήση του πρωτοκόλλου MQTT, ώστε να επιτρέπεται η άμεση και αποδοτική εγγραφή και σύνδεση σε συνεργατικές δομές «ομίχλης», αλλά και η αξιόπιστη και άμεση ανταλλαγή μηνυμάτων.

Η συνειδητοποίηση των δυνατοτήτων των τεχνολογιών αιχμής, αλλά και των δυνατοτήτων των τεχνολογιών διαδικτύου, αποτέλεσε έμπνευση για την διαμόρφωση της λύσης ώστε να αποκτήσει περισσότερη «εξωστρέφεια», λαμβάνοντας υπόψιν τις σύγχρονες ανάγκες για εκτεταμένη και ολοκληρωμένη διασύνδεση. Συνεπώς, η προτεινόμενη λύση εκτός από τη διασύνδεση ενός ή περισσότερων δικτύων «ομίχλης» με ένα άλλο, δίνουν τη δυνατότητα για υποστήριξη και κινητών συσκευών, λαμβάνοντας πάντα υπόψιν τα ιδιαίτερα χαρακτηριστικά τους.

Η αξιολόγηση της αρχιτεκτονικής πραγματοποιήθηκε με τη βοήθεια ενός εξελιγμένου και επεκτάσιμου προσομοιωτή, ο οποίος αναπτύχθηκε ακριβώς για τις ανάγκες της εξεταζόμενης πλατφόρμας. Από τα αποτελέσματα, συμπεραίνεται η χρησιμότητα ευφυών μηχανισμών διαχείρισης, ακόμα και στο επίπεδο «ομίχλης». Αντίθετα, με τις υπάρχουσες προσεγγίσεις, η παρούσα αρχιτεκτονική δεν αποκλείει το επίπεδο «νέφους» από τη διαδικασία της βραχυπρόθεσμης εκτέλεσης και εξυπηρέτησης του φόρτου εργασιών. Τα αποτελέσματα των προσομοιώσεων, υποστηρίζουν αυτή τη χρησιμότητα. Επίσης, με την ύπαρξη του κατάλληλου χαρακτηρισμού των εργασιών, ανάλογα με την πηγή των δεδομένων, επιτρέπει την κατάλληλη προώθηση των εργασιών χωρίς να επιβαρύνει τη λειτουργία του επιπέδου «ομίχλης», με χρονοβόρους μηχανισμούς αποφάσεων.

Η αξιοποίηση κινήτρων για τη συμμετοχή συσκευών που λειτουργούν στο εγγύς περιβάλλον με τη πηγή των δεδομένων, είναι εξέχουσας σημασίας για τη εξέλιξη του μοντέλου του «ομιχλώδους» υπολογισμού. Ενώ σε συγκεκριμένες περιπτώσεις τα

πλεονεκτήματα της συμμετοχής προσωπικών και κινητών συσκευών στην όλη διαδικασία είναι εμφανή, σε κάποιες άλλες όμως αυτή η συμμετοχή δεν είναι εγγυημένη. Χαρακτηριστικό παράδειγμα αποτελούν τα συστήματα απομακρυσμένης εποπτείας περιβάλλοντος (π.χ. έγκαιρη ανίχνευση πυρκαγιάς σε δασικές περιοχές [122] ή εποπτεία περιβαντολογικών συνθηκών σε αρχαιολογικές υπηρεσίες [123]), όπου η χρήση κινητών συσκευών απλών πολιτών θα μπορούσαν χρησιμοποιηθούν εφόσον ο χρήστης το επιθυμεί. Ειδικά σε αυτές τις περιπτώσεις, συγκεκριμένες τακτικές αμοιβαίας χρήσης θα μπορούσαν να χρησιμοποιηθούν, όπου ο χρήστης θα είχε συγκεκριμένα οφέλη (π.χ. εκπτώσεις, καλύτερο εύρος ζώνης) από την εθελοντική προσφορά πόρων σε δίκτυα «ομίχλης».

Μία ακόμη ενδιαφέρουσα πρόταση για μελλοντική έρευνα, είναι η υποστήριξη ομοσπονδιακών συσκευών [124], αλλά και οι αλγόριθμοι μηχανικής μάθησης (π.χ. Q-learning) για το δυναμικό καθορισμό των βαρών της συνάρτησης βαθμολόγησης (1). Όσον αφορά την αξιολόγηση, η συνεχής εξέλιξη και επέκταση προσομοιωτών, όπως του Cloudsim [113], ώστε να συμπεριληφθούν περισσότερα τεχνικά χαρακτηριστικά μίας αρχιτεκτονικής «ομίχλης», είναι αναγκαία και ο υπάρχων προσομοιωτής βρίσκεται υπό συνεχή ανάπτυξη και υποστήριξη. Τέλος, εντοπίζεται η ανάγκη για την υποστήριξη απλών και μη-πολύπλοκων μηχανισμών κρυπτογράφησης και ελέγχου πρόσβασης, ως ένα στρώμα ασφαλείας, κάτι που αποτελεί επιπλέον στόχο για μελλοντική έρευνα.

5.2 Συστήματα «ομόσπονδων» δικτύων υποδομών υπολογιστικών νεφών με έμφαση στα συμβόλαια χρήσης υπηρεσίας (SLA)

Η ανάπτυξη αρχιτεκτονικών για την ενεργοποίηση διαλειτουργικότητας μεταξύ υπολογιστικών υποδομών «νέφους», είναι μία πρωτοβουλία που εκδηλώθηκε ως μία λύση που θα επιτρέπει σε παρόχους (επιχειρήσεις και οργανισμούς) να μπορούν να διατηρούν ένα υψηλό βαθμό τήρησης των συμβολαίων χρήσης που έχουν συνάψει με τους πελάτες τους. Η κύρια έκφραση τέτοιων αρχιτεκτονικών εντοπίζεται συνήθως ως «ομοσπονδία» νεφών, και ένα κοινό χαρακτηριστικό τους είναι η ύπαρξη δομών που σε ένα ιεραρχικό μοντέλο, αναλαμβάνουν τη διαμεσολάβηση ώστε να βρίσκουν τον κατάλληλο πάροχο που θα ικανοποιήσει τα απαιτούμενα του αντίστοιχου παρόχου-πελάτη.

Η οντότητα του διαμεσολαβητή και συνολικά η λειτουργία διαχείρισης «ομοσπονδιών» νέφους είναι το επίκεντρο της αρχιτεκτονικής πρότασης της παρούσας διατριβής. Στα πλαίσια της μελέτης και της έρευνας εξετάστηκε ένα σύστημα βαθμολόγησης παρόχων υπηρεσιών «νέφους». Η βαθμολόγηση ως έννοια επιχειρεί να

προσδώσει μία έννοια δικαιοσύνης στο δίκτυο, του οποίου κύριο χαρακτηριστικό είναι η εθελοντικού χαρακτήρα συμμετοχή παρόχων, ώστε να χρησιμοποιούν από κοινού τις υπηρεσίες τους και να μισθώνουν υπολογιστικούς πόρους της υποδομής.

Το κύριο χαρακτηριστικό της «ομοσπονδίας» είναι ότι κάθε πάροχος παραμένει μία ανεξάρτητη οντότητα και συνεχίζει να λειτουργεί ως τέτοια καθ' όλη τη διάρκεια της συμμετοχής της στο δίκτυο. Για την αποφυγή οπορτουνιστικών συμπεριφορών, ο διαμεσολαβητής πρέπει να εξασφαλίζει ότι η κάθε αιτούμενος πάροχος θα πρέπει να λαμβάνει τη κατάλληλη υπηρεσία, ανάλογα όμως και με την δικιά του συμπεριφορά και απόδοση σε αυτήν τη ενιαία αγορά. Σε αυτή τη λογική, η προτεινόμενη αρχιτεκτονική χρησιμοποιεί ένα σύστημα βαθμολόγησης που έχει έναν χαρακτήρα επιβράβευσης/ποινής.

Η βαθμολόγηση γίνεται, σε πρώτο στάδιο, με βάση την απόδοση ενός παρόχου σε σχέση με το SLA που συμφώνησε με έναν πάροχο/πελάτη. Σε δεύτερο στάδιο όμως, η βαθμολόγηση εξαρτάται και από τη συχνότητα προσφοράς στο δίκτυο. Με αυτό τον τρόπο εξασφαλίζεται η μέγιστη δικαιοσύνη στο συνεργατικό δίκτυο. Τέλος, η βαθμολογία κάθε παρόχου χρησιμοποιείται ως κριτήριο για την αντιστοίχιση του με ένα πάροχο αντίστοιχης βαθμολογίας στην διαδικασία απόφασης. Έτσι, δίνεται το κατάλληλο κίνητρο στους παρόχους για συνεχή παροχή της βέλτιστης υπηρεσίας, και για την συχνότερη δυνατή συμμετοχή.

Η συγκεκριμένη μελέτη ανοίγει βεβαίως το δρόμο για περαιτέρω έρευνα και πειραματισμό σε ένα πεδίο όπου δεν είναι ιδιαίτερα ώριμο. Ιδιαίτερο ενδιαφέρον θα παρουσίαζε, η μελέτη του εν λόγω μηχανισμού διαμεσολάβησης σε μη ιεραρχικές «ομοσπονδίες», όπου οι πάροχοι θα επικοινωνούσαν μέσω μίας δομής peer-to-peer, και όπου ανεξάρτητες υπηρεσίες διαμεσολάβησης θα πρέπει να λειτουργούν σε κάθε πάροχο ξεχωριστά. Επίσης, ένα κυρίαρχο ζήτημα προκύπτει και από την ανάγκη για μηχανισμούς εμπιστοσύνης και ασφάλειας, τόσο των επικοινωνιών μεταξύ των παρόχων, όσο και των μεταδιδόμενων δεδομένων (π.χ. μεταφορά δεδομένων χρηστών ενός παρόχου σε έναν άλλο). Η επέκταση της προτεινόμενης αρχιτεκτονικής και η εισαγωγή λειτουργιών ασφαλείας [125], ως κριτήρια διαμόρφωσης της βαθμολογίας και τελικής επιλογής στη διαδικασία αντιστοίχισης, αποτελεί ένα αρκετά ενδιαφέρον πεδίο για μελλοντική έρευνα.

5.3 Συστήματα διαχείρισης πόρων και υποδομών υπολογιστικών νεφών με έμφαση στην ενεργειακή αποδοτικότητα

Μία τυπική υποδομή υπολογιστικού «νέφους» προσφέρει υπηρεσίες μέσω μίας «δεξαμενής» πόρων που δημιουργείται από τη συλλογή υλισμικού του υπολογιστικού κέντρου. Ένα τυπικό υπολογιστικό κέντρο, το οποίο χρησιμοποιείται για την ανάπτυξη υποδομών «νέφους», περιέχει εκατοντάδες εξυπηρετητές, ενώ μεγάλοι πάροχοι όπως η Amazon και η Google, διατηρούν πολλαπλά υπολογιστικά κέντρα ανά την υφήλιο, με χιλιάδες εξυπηρετητές να βρίσκονται σε συνεχή λειτουργία για την υποστήριξη του συνόλου των προσφερόμενων υπηρεσιών.

Η διαχείριση των τεράστιων αυτών υποδομών, αποκτά όλο και μεγαλύτερη σημασία. Η λειτουργία τους επιφέρει τεράστιο κόστος στους παρόχους, ενώ στο σύνολο τους τα υπολογιστικά κέντρα καταναλώνουν το 2% της παραγόμενης ενέργειας παγκοσμίως²². Οι προσπάθειες για μία πιο αποδοτική διαχείριση των υπολογιστικών κέντρων, επικεντρώνονται συνήθως σε καινοτόμες τεχνολογίες υλισμικού (π.χ. καινούργιες αρχιτεκτονικές επεξεργαστών με χαμηλότερη απαίτηση σε ισχύ, καινοτόμες αρχιτεκτονικές διαμόρφωσης των χώρων φιλοξενίας των εξυπηρετητών [126], και εξελιγμένα συστήματα ψύξης [127]).

Οι πάροχοι υπηρεσιών «νέφους» εξετάζουν επίσης και την αποδοτική διαχείριση του επιπέδου των εικονικών πόρων, το πώς δηλαδή, θα οργανώσουν το σύνολο των εικονικών περιβαλλόντων με τέτοιο τρόπο, ούτως ώστε να θέτουν σε λειτουργία όσον το δυνατόν λιγότερους εξυπηρετητές.

Στα πλαίσια της παρούσας διατριβής, προτάθηκε μία ολοκληρωμένη πλατφόρμα διαχείρισης πόρων σε υπολογιστικές υποδομές υβριδικής εικονικοποίησης, υποδομών δηλαδή που προσφέρουν και τους δύο τύπους εικονικών συστημάτων, τα εικονικά μηχανήματα υπερεπόπτη και τους περιέκτες λογισμικού. Με την χρήση μηχανισμών εποπτείας, αλλά και με την πληροφορία που αντλείται από προτυποποιημένα μηνύματα των χρηστών, η πλατφόρμα κατηγοριοποιεί τα υπάρχοντα εικονικά μηχανήματα ανάλογα με το τύπο χρήσης τους. Όταν κρίνεται εφικτό, ο μηχανισμός κατανομής μετατρέπει ένα εικονικό μηχάνημα τύπου υπερεπόπτη σε περιέκτη λογισμικού. Λόγω της οικονομικής φύσης της λειτουργίας των περιεκτών, είναι δυνατόν να επιτευχθεί αποδοτικότερη χρήση των εξυπηρετητών, λόγω της χαμηλότερης επιβάρυνσης και της απουσίας ανάγκης για την εκ των προτέρων δέσμευση πόρων. Η πλατφόρμα

²² <http://www.economist.com/technology-quarterly/2016-03-12/after-moores-law>

προτείνεται να λειτουργεί σε συνεργασία με υπάρχουσες πλατφόρμες ανάπτυξης υπολογιστικών υποδομών νέφους (π.χ. καλώντας τις υπηρεσίες nova [128], neutron [129] και glance [130] μίας υποδομής Openstack [9], δίνοντας έτσι τη δυνατότητα να λειτουργήσει ως ένα τμήμα επέκτασης υπάρχουσών πλατφορμών υποδομών «νέφους».

Η αξιολόγηση της πλατφόρμας έγινε με τη βοήθεια προσομοιώσεων, οι οποίες βασίστηκαν στην επέκταση της πλατφόρμας προσομοιώσεων Cloudsim [113]. Συγκεκριμένα, η βασική πλατφόρμα επεκτάθηκε για την υποστήριξη περιεκτών λογισμικού ως οντοτήτων που λειτουργούν απευθείας σε εξυπηρετητές. Τα αποτελέσματα των προσομοιώσεων, συγκριτικά με δύο στρατηγικές ευφυούς κατανομής εικονικών μηχανημάτων σε περιπτώσεις εικονικοποίησης βασισμένης σε υπερεπόπτη, έδειξαν τη χρησιμότητα των περιεκτών λογισμικού, όπου με τη κατάλληλη χρήση, μπορούν να μειώσουν σημαντικά την κατανάλωση ενέργειας ενός υπολογιστικού κέντρου.

Η εν λόγω αρχιτεκτονική πρόταση, σχεδιάζεται να επεκταθεί για την ενσωμάτωση δυναμικών και στοχευμένων μηχανισμών εποπτείας, με σκοπό μία πιο έγκυρη και αποδοτική λήψη αποφάσεων με βάση τη ανάλυση ιστορικών δεδομένων με βάση τη κατανάλωση πόρων επιμέρους διεργασιών των εικονικών μηχανημάτων που λειτουργούν στην υποδομή. Επιπρόσθετα, η ανάλυση χρήσης πόρων της εφαρμογής του χρήστη θα μπορούσε να οδηγήσει στη περαιτέρω παραμετροποίηση του περιέκτη που επιλέγεται να τη φιλοξενήσει, ώστε να καταναλώνει όσους πόρους πραγματικά απαιτεί (π.χ. χρήση της λειτουργικότητας Cgroups για τον ορισμό ανώτατου ορίου χρόνου CPU που θα δεσμεύσει ένα περιέκτης). Μία ακόμα προσέγγιση που προτείνεται [100] [94] είναι η λειτουργία των περιεκτών λογισμικού ως ένα επίπεδο που θα λειτουργεί επάνω σε εικονικά μηχανήματα. Η λύση αυτή εφαρμόζεται ήδη [93] και ενεργοποιεί το λεγόμενο μοντέλο CaaS (Containers as a Service). Ως μελλοντική έρευνα, προτείνεται η μελέτη της προσέγγισης αυτής ώστε να διερευνηθεί το κατά πόσο συνεισφέρει στην μείωση του κόστους λειτουργίας, με τη χρήση βέβαια των κατάλληλων μηχανισμών κατανομής. Τέλος, η βελτίωση των επεκτάσεων του προσομοιωτή ώστε να υποστηρίζονται σύνθετα μοντέλα φόρτου εργασίας, θα βοηθήσουν στην εξαγωγή αποτελεσμάτων και συμπερασμάτων ακόμα μεγαλύτερης ακρίβειας.

Βιβλιογραφία

- [1] Dean, J. and S. Ghemawat, *MapReduce: Simplified data processing on large clusters*. Communications of the ACM, 2008. 51(1): p. 107-113.
- [2] Apache Hadoop. <http://hadoop.apache.org/>, accessed March 2017
- [3] Vera Home Control. <http://getvera.com/>, accessed March 2017
- [4] Piraghaj, S.F., et al. *A Framework and Algorithm for Energy Efficient Container Consolidation in Cloud Data Centers*. in *Proceedings - 2015 IEEE International Conference on Data Science and Data Intensive Systems; 8th IEEE International Conference Cyber, Physical and Social Computing; 11th IEEE International Conference on Green Computing and Communications and 8th IEEE International Conference on Internet of Things, DSDIS/CPSCoM/GreenCom/iThings 2015*. 2016.
- [5] Beloglazov, A. and R. Buyya, *Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers*. Concurrency Computation Practice and Experience, 2012. 24(13): p. 1397-1420.
- [6] Ansible. <https://www.ansible.com/>, accessed March 2017
- [7] Standard for Intercloud Interoperability and Federation (SIIF), IEEE Std. P2302, 2012
- [8] Yun, M. and B. Yuxin. *Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid*. in *2010 International Conference on Advances in Energy Engineering, ICAEE 2010*. 2010.
- [9] Openstack <https://www.openstack.org/>, accessed 21 March 2017
- [10] Keystone - Openstack Identity Service. <https://docs.openstack.org/developer/keystone/>, accessed March 2017
- [11] MQTT Connectivity Protocol. <http://mqtt.org/>, accessed March 2017
- [12] J. Davies, R.S., and P. Warren, *Semantic Web technologies: Trends and Research in Ontologybased Systems*. 2006: John Wiley & Sons,.
- [13] Berners-Lee, T., J. Hendler, and O. Lassila, *The semantic web*. Scientific American, 2001. 284(5): p. 34-43.
- [14] World Wide Web Consortium (W3C), "Resource Description Framework (RDF)". <https://www.w3.org/RDF/>, accessed 21/03/2017
- [15] World Wide Web Consortium (W3C), "RDF Schema 1.1.". <https://www.w3.org/TR/rdf-schema>, accessed 21/03/2017
- [16] World Wide Web Consortium (W3C), "OWL Web Ontology Language Overview.". <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, accessed 21/03/2017
- [17] World Wide Web Consortium (W3C), "SPARQL 1.1 Query Language.". <http://www.w3.org/TR/sparql11-query>, accessed 21/03/2017
- [18] A. Seaborne, G.M., C. Bizer, J. Breslin, S. Das, I. Davis, S. Harris, K. Idehen, O. Corby, K. Kjernsmo, et al., *SPARQL/Update: A Language for Updating RDF Graphs*. W3C Member Submission,, 2008. 15.
- [19] World Wide Web Consortium (W3C), "SPARQL 1.1 Update.". <http://www.w3.org/TR/sparql11-update/>, accessed March 2017

- [20] World Wide Web Consortium (W3C), "OWL 2 Web Ontology Language Document Overview (Second Edition)". <http://www.w3.org/TR/owl2-overview/>, accessed March 2017
- [21] Shadbolt, N., W. Hall, and T. Berners-Lee, *The semantic web revisited*. IEEE Intelligent Systems, 2006. 21(3): p. 96-101.
- [22] D. T. Meridou, M.-E.C.P., I. S. Venieris, C. Z. Patrikakis, P. Vrba, O. Harcuba, C. Marin, Y. Shepilov, D. Kazanskaia, N. Rodriguez, and P. Leitão, *Semantics – Architectures and Tools*, in *Adaptive Production Planning and Scheduling: The ARUM approach based on MAS and SOA technologies*, A. Schirrmann, Editor. 2016, Springer.
- [23] World Wide Web Consortium (W3C), "OWL 2 Web Ontology Language Profiles (Second Edition)". <https://www.w3.org/TR/owl2-profiles/>, accessed March 2017
- [24] mOSAIC Project (FP7 - ICT). <http://www.mosaic-cloud.eu/>, accessed March 2017
- [25] Open Cloud Computing Interface (OCCI). <http://occi-wg.org/>, accessed March 2017
- [26] Distributed Management Task Force - DMTF. <https://www.dmtf.org/>, accessed March 2017
- [27] National Institute of Standards and Technology - NIST. <https://www.nist.gov/>, accessed March 2017
- [28] Moscato, F., et al. *An analysis of mOSAIC ontology for cloud resources annotation*. in *2011 Federated Conference on Computer Science and Information Systems, FedCSIS 2011*. 2011.
- [29] KVM. https://www.linux-kvm.org/page/Main_Page, accessed March 2017
- [30] The Xen Project. <https://www.xenproject.org/>, accessed March 2017
- [31] Hyper-V. <https://www.microsoft.com/en-us/cloud-platform/server-virtualization>, accessed March 2017
- [32] Morabito, R., J. Kjällman, and M. Komu. *Hypervisors vs. lightweight virtualization: A performance comparison*. in *Proceedings - 2015 IEEE International Conference on Cloud Engineering, IC2E 2015*. 2015.
- [33] KVM and Docker LXC Benchmarking with OpenStack. <http://bodenr.blogspot.gr/2014/05/kvm-and-docker-lxc-benchmarking-with.html>, accessed November 2016
- [34] Analysis of Docker Security. <http://arxiv.org/abs/1501.02967>, accessed November 2016
- [35] Docker. <https://www.docker.com/>, accessed March 2017
- [36] LXD <https://linuxcontainers.org/lxd/>, accessed March 2017
- [37] Kubernetes. <https://kubernetes.io/>, accessed March 2017
- [38] Mell, P. and T. Grance, *The NIST definition of cloud computing*, in *Cloud Computing and Government: Background, Benefits, Risks*. 2011. p. 171-173.
- [39] Amazon Elastic Compute Cloud - EC2. <https://aws.amazon.com/ec2/>, accessed March 2017
- [40] Google AppEngine. <https://cloud.google.com/appengine/>, accessed
- [41] Office 365. <https://products.office.com/el-gr/business/compare-office-365-for-business-plans>, accessed March 2017

- [42] Cloud Standards Consumer Council - CSCC. <http://www.cloud-council.org/>, accessed March 2017
- [43] Cloud Standards Customer Council, *Practical Guide to Cloud Service Agreements Version 2.0 2015*, <http://www.cloud-council.org/deliverables/CSCC-Practical-Guide-to-Cloud-Service-Agreements.pdf>
- [44] Amazon EC2 SLA. <https://aws.amazon.com/ec2/sla/>, accessed March 2017
- [45] Baset, S.A., *Cloud Service Level Agreement*, in *Encyclopedia of Cloud Computing*, S.M.a.I. Bojanova, Editor. 2016, John Wiley & Sons.
- [46] SPEC Open Systems Group. <https://www.spec.org/osg/>, accessed March 2017
- [47] Standard Performance Evaluation Corporation - SPEC. <https://www.spec.org/>, accessed November 2016
- [48] NIST, *Cloud Computing Service Metrics Description, 2015*, <https://www.nist.gov/sites/default/files/documents/itl/cloud/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf>
- [49] Bonomi, F., et al. *Fog computing and its role in the internet of things*. in *MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*. 2012.
- [50] Xu, Y., V. Mahendran, and S. Radhakrishnan. *Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery*. in *2016 8th International Conference on Communication Systems and Networks, COMSNETS 2016*. 2016.
- [51] Aazam, M. and E.N. Huh. *Dynamic resource provisioning through Fog micro datacenter*. in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2015*. 2015.
- [52] Prazeres, C. and M. Serrano. *SOFT-IoT: Self-organizing FOG of things*. in *Proceedings - IEEE 30th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2016*. 2016.
- [53] Lasi, H., et al., *Industry 4.0*. Business and Information Systems Engineering, 2014. 6(4): p. 239-242.
- [54] Hermann, M., T. Pentek, and B. Otto. *Design principles for industrie 4.0 scenarios*. in *Proceedings of the Annual Hawaii International Conference on System Sciences*. 2016.
- [55] European Factories of the Future Research Association <http://www.effra.eu/>, accessed March 2017
- [56] Möller, D.P.F., et al. *Cyber-physical smart traffic light system*. in *2015 IEEE International Conference on Electro/Information Technology (EIT)*. 2015.
- [57] Yi, S., C. Li, and Q. Li. *A survey of fog computing: Concepts, applications and issues*. in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. 2015.
- [58] Yi, S., et al. *Fog computing: Platform and applications*. in *Proceedings - 3rd Workshop on Hot Topics in Web Systems and Technologies, HotWeb 2015*. 2016.
- [59] Google Glass. <https://www.google.com/glass/start/>, accessed March 2017
- [60] Sony SmartEyeGlass. <https://developer.sony.com/devices/mobile-accessories/smarteyeglass/>, accessed March 2017
- [61] Microsoft Hololens. <https://www.microsoft.com/microsoft-hololens/en-us>, accessed March 2017

- [62] Ha, K., et al. *Towards wearable cognitive assistance*. in *MobiSys 2014 - Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. 2014.
- [63] Microsoft Azure. <https://azure.microsoft.com/en-us/>, accessed March 2017
- [64] Google Cloud. <https://cloud.google.com/>, accessed March 2017
- [65] XMPP. <https://xmpp.org/>, accessed March 2017
- [66] XEP-0244. <https://xmpp.org/extensions/xep-0244.html>, accessed March 2017
- [67] XMPP Web Services for Java (XWS4J). <http://xws4j.sourceforge.net/>, accessed March 2017
- [68] Buyya, R., R. Ranjan, and R.N. Calheiros, *InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2010. p. 13-31.
- [69] Grozev, N. and R. Buyya, *Inter-Cloud architectures and application brokering: Taxonomy and survey*. *Software - Practice and Experience*, 2014. 44(3): p. 369-390.
- [70] Bernstein, D. and D. Vij. *Intercloud directory and exchange protocol detail using XMPP and RDF*. in *Proceedings - 2010 6th World Congress on Services, Services-1 2010*. 2010.
- [71] Garg, S.K., S. Versteeg, and R. Buyya. *SMICloud: A framework for comparing and ranking cloud services*. in *Proceedings - 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011*. 2011.
- [72] Rehman, Z.U., O.K. Hussain, and F.K. Hussain. *Multi-Criteria IaaS service selection based on QoS history*. in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*. 2013.
- [73] OpenNebula. <https://opennebula.org/>, accessed March 2017
- [74] Amazon Web Services - AWS. <https://aws.amazon.com/>, accessed March 2017
- [75] Azure Active Directory - ADS. <https://azure.microsoft.com/en-us/services/active-directory/>, accessed March 2017
- [76] Kubernetes Federation. <https://kubernetes.io/docs/concepts/cluster-administration/federation/>, accessed March 2017
- [77] BonFIRE Project. <http://www.bonfire-project.eu/>, accessed October 2016
- [78] SPECpower_ssj® 2008. https://www.spec.org/power_ssj2008/, accessed October 2016
- [79] Felter, W., et al. *An updated performance comparison of virtual machines and Linux containers*. in *ISPASS 2015 - IEEE International Symposium on Performance Analysis of Systems and Software*. 2015.
- [80] Setzer, T. and A. Stage. *Decision support for virtual machine reassignments in enterprise data centers*. in *2010 IEEE/IFIP Network Operations and Management Symposium Workshops, NOMS 2010*. 2010.
- [81] Feller, E., L. Rilling, and C. Morin. *Energy-aware ant colony based workload placement in clouds*. in *Proceedings - 2011 12th IEEE/ACM International Conference on Grid Computing, Grid 2011*. 2011.

- [82] Dorigo, M., G. Di Caro, and L.M. Gambardella, *Ant algorithms for discrete optimization*. Artificial Life, 1999. 5(2): p. 137-172.
- [83] Liu, X.F., et al. *Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach*. in *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*. 2014.
- [84] Pedemonte, M., S. Nesmachnow, and H. Cancela, *A survey on parallel ant colony optimization*. Applied Soft Computing Journal, 2011. 11(8): p. 5181-5197.
- [85] Holland, J.H., *Genetic algorithms*. Scientific American, 1992. 267(1): p. 66-72.
- [86] Mohamadi, N., *Application of Genetic Algorithm for the Bin Packing Problem with a New Representation Scheme*. Mathematical Sciences Quarterly Journal, 2010.
- [87] Falkenauer, E., *Genetic Algorithms and Grouping Problems*. 1998: John Wiley & Sons, Inc. 238.
- [88] Nguyen, Q.H., et al., *A genetic algorithm for power-aware virtual machine allocation in private cloud*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2013. p. 183-191.
- [89] Joseph, C.T., K. Chandrasekaran, and R. Cyriac. *A novel family genetic approach for virtual machine allocation*. in *Procedia Computer Science*. 2015.
- [90] Dong, Y.S., G.C. Xu, and X.D. Fu, *A distributed parallel genetic algorithm of placement strategy for virtual machines deployment on cloud platform*. Scientific World Journal, 2014. 2014.
- [91] Apache Mesos. <http://mesos.apache.org/>, accessed March 2017
- [92] CoreOS. <https://coreos.com/>, accessed March 2017
- [93] Openstack Magnum. <https://wiki.openstack.org/wiki/Magnum>, accessed March 2017
- [94] Piraghaj, S.F., et al., *ContainerCloudSim: An environment for modeling and simulation of containers in cloud data centers*. Software - Practice and Experience, 2017. 47(4): p. 505-521.
- [95] Bui, D.M., et al., *Energy efficiency for cloud computing system based on predictive optimization*. Journal of Parallel and Distributed Computing, 2017. 102: p. 103-114.
- [96] Liu, B., Y. Lin, and Y. Chen. *Quantitative workload analysis and prediction using Google cluster traces*. in *Proceedings - IEEE INFOCOM*. 2016.
- [97] cloud-init. <https://cloud-init.io/>, accessed March 2017
- [98] Pantheon.io. <https://pantheon.io>, accessed March 2017
- [99] Microsoft Azure Site Recovery <https://azure.microsoft.com/en-us/services/site-recovery/>, accessed March 2017
- [100] eco4cloud. <http://www.eco4cloud.com/>, accessed March 2017
- [101] ESXi. <http://www.vmware.com/products/esxi-and-esx.html>, accessed March 2017
- [102] Openstack Watcher. <https://wiki.openstack.org/wiki/Watcher>, accessed March 2017
- [103] ZigBee Protocol. <http://www.zigbee.org/>, accessed March 2017

-
- [104] Z-Wave Standard. <http://www.z-wave.com/>, accessed March 2017
- [105] The Lua Programming Language. <https://www.lua.org/>, accessed January 2017
- [106] The Julia Programming Language. <https://julialang.org/>, accessed January 2017
- [107] M. E. Glickman, *The Glicko system*, <http://www.glicko.net/glicko/glicko.pdf>
- [108] Amazon EC2 Instance Types. <https://aws.amazon.com/ec2/instance-types/>, accessed March 2017
- [109] Li, A., et al. *CloudCmp: Comparing public cloud providers*. in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*. 2010.
- [110] Unuvar, M., et al., *Selecting Optimum Cloud Availability Zones by Learning User Satisfaction Levels*. *IEEE Transactions on Services Computing*, 2015. 8(2): p. 199-211.
- [111] Kozhirbayev, Z. and R.O. Sinnott, *A performance comparison of container-based technologies for the Cloud*. *Future Generation Computer Systems*, 2017. 68: p. 175-182.
- [112] Rana, M., S. Bilgaiyan, and U. Kar. *A study on load balancing in cloud computing environment using evolutionary and swarm based algorithms*. in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. 2014.
- [113] Cloudsim Framework. <http://www.cloudbus.org/cloudsim/>, accessed September 2017
- [114] CloudsimEX Project. <https://github.com/Cloudslab/CloudSimEx>, accessed September 2017
- [115] RealCloudSim. <https://sourceforge.net/projects/realcloudsim/>, accessed March 2017
- [116] DynamicCloudSim. <https://code.google.com/archive/p/dynamiccloudsim/>, accessed March 2017
- [117] Cloud2Sim. <https://sourceforge.net/projects/cloud2sim/>, accessed March 2017
- [118] Dhrystone 2 MIPS Benchmarks. <https://en.wikipedia.org/wiki/Dhrystone>, accessed September 2016
- [119] Raspberry Pi. <https://www.raspberrypi.org/>, accessed September 2016
- [120] Estrada, Z.J., et al. *A performance evaluation of sequence alignment software in virtualized environments*. in *Proceedings - 14th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2014*. 2014.
- [121] Puppet. <https://puppet.com/>, accessed March 2017
- [122] Yu, L., N. Wang, and X. Meng. *Real-time forest fire detection with wireless sensor networks*. in *Proceedings - 2005 International Conference on Wireless Communications, Networking and Mobile Computing, WCNM 2005*. 2005.
- [123] STORM project, H2020, DRS-11-2015 - Disaster Resilience & Climate Change topic 3: Mitigating the impacts of climate change and natural hazards on cultural heritage sites, structures and artefact,. <http://www.storm-project.eu/>, accessed September 2016
- [124] Chen, M., et al., *On the computation offloading at ad hoc cloudlet: Architecture and service modes*. *IEEE Communications Magazine*, 2015. 53(6): p. 18-24.

- [125] Dhole, A., M.V. Thomas, and K. Chandrasekaran. *An efficient trust-based Game-Theoretic approach for Cloud Federation Formation*. in *ICACCS 2016 - 3rd International Conference on Advanced Computing and Communication Systems: Bringing to the Table, Futuristic Technologies from Around the Globe*. 2016.
- [126] Li, D., et al., *Revisiting the Design of Mega Data Centers: Considering Heterogeneity Among Containers*. *IEEE/ACM Transactions on Networking*, 2014. 22(5): p. 1503-1515.
- [127] Chen, S., Y. Hu, and L. Peng. *Optimization of Electricity and Server Maintenance Costs in Hybrid Cooling Data Centers*. in *2013 IEEE Sixth International Conference on Cloud Computing*. 2013.
- [128] Openstack Compute API. <https://developer.openstack.org/api-ref/compute/>, accessed March 2017
- [129] Openstack Networking API. <https://docs.openstack.org/mitaka/networking-guide/>, accessed March 2017
- [130] Openstack Image API. <https://docs.openstack.org/developer/glance/>, accessed March 2017

Παράρτημα

A.1 Παράδειγμα μηνύματος «ομίχλης»

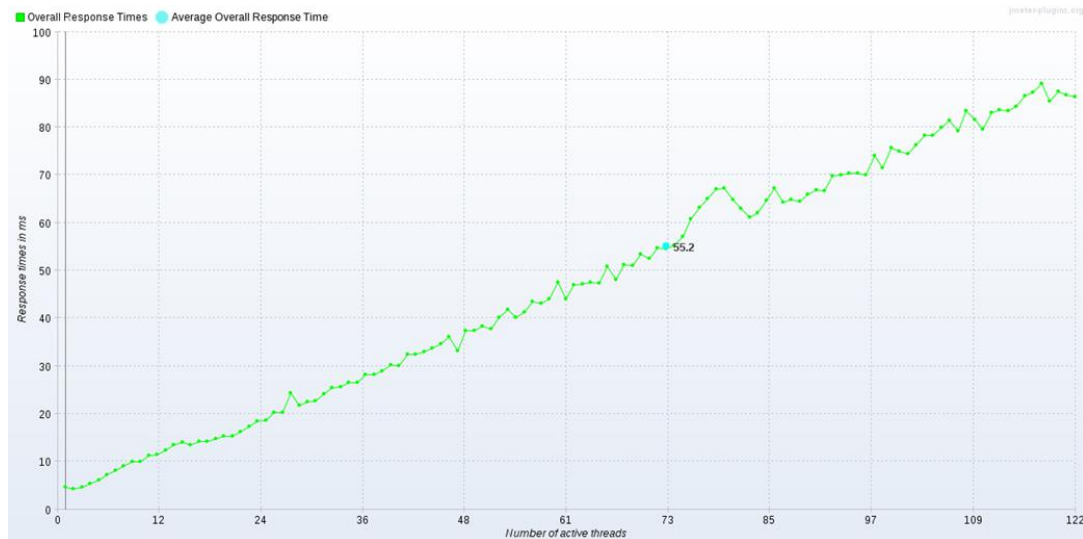
```
{
  "data": [
    {
      "date": "2016-09-05",
      "x1": 14,
      "mean1": 15,
      "variance1": 3,
      "logId": 1330991999001,
      "time": "21:00:00"
    },
    {
      "date": "2016-09-05",
      "x2": 13.5,
      "mean2": 14,
      "variance2": 1,
      "logId": 1330991999002,
      "time": "21:00:00"
    },
    {
      "date": "2016-09-05",
      "x3": 140,
      "mean3": 100,
      "variance3": 2,
      "logId": 1330991999003,
      "time": "21:00:00"
    },
    {
      "date": "2016-09-05",
      "x4": 83.5,
      "mean4": 80,
      "variance4": 2,
      "logId": 1330991999004,
      "time": "21:00:00"
    }
  ],
  "execution_script": {
    "code": [
      "import json",
      "import numpy as np",
      "from scipy.stats import norm",
      "epsilon = 5.2739369041654677e-20",
      "with open('event.json') as data_file:",
      "    data = json.load(data_file)",
      "X =
np.asarray([data['data'][0]['x1'],data['data'][1]['x2'],data['data'][2]['x3'],data['data'][3][
'x4']])",
      "mu =
np.asarray([data['data'][0]['mean1'],data['data'][1]['mean2'],data['data'][2]['mean3'],data['d
ata'][3]['mean4']])",
      "sigma2 =
np.asarray([data['data'][0]['variance1'],data['data'][1]['variance2'],data['data'][2]['varianc
e3'],data['data'][3]['variance4']])",
      "if np.prod(norm.pdf(data['X'],data['mu'],data['sigma2'])) < epsilon:",
      "    return True",
      "return False"
    ],
    "type": "py"
  },
  "metadata": [
    {
      "time_latency": 1200
    },
    {
      "data_size": 784
    },
    {
      "computational_power": "O(1)"
    }
  ]
}
```

```

    },
    {
      "ram":256
    },
    {
      "topic_address":"http://www.icbnet.gr"
    }
  ]
}

```

A.2 Προσομοιώσεις αποστολής μηνύματος «ομίχλης»



Εικόνα_Παρ Α - 1 Χρόνοι απόκρισης της υπηρεσίας ανάλογα με τα ταυτόχρονα αιτήματα «ομίχλης»

A.3 Αλγόριθμος βαθμολόγησης του διαμεσολαβητή «ομοσπονδίας»

Αλγόριθμος 1: Διαδικασία ανανέωσης βαθμολογίας παρόχων CSP

```

Input: c, q, RDreq //σταθερές τιμές
Input: getCSPList() // λίστα με τους εγγεγραμμένους παρόχους
Input: getCurrentRatings() //τρέχοντες βαθμολογίες των παρόχων
Input: getPreRatingRDs() //λίστα με τις πιο πρόσφατες τιμές διασποράς βαθμολόγησης (RD) των παρόχων
Data: getTransactionHistory() //ιστορικό συναλλαγών της τελευταίας περιόδου βαθμολόγησης
Output: updatedRatings // ανανεωμένος πίνακας βαθμολογιών
1 for csp in getCSPList() {
2   currentRating = getCurrentRatings(csp);
3   currentRD = getPreRatingRDs(csp);
4   transactionList = getTransactionHistory(csp);
5   if transactionList not empty(){
6     list requestList;
7     list requestRatingList;
8     list outcomeList;
9     for transaction in transactionList {
10      requestList.add(transaction.getId());
11      requestRating = calculateRequestRating(csp, transaction);
12      requestRatingList.add(requestRating);
13      outcomeList.add(transaction.getOutcome());
14      newRating, newRD = calculateNewRating(currentRating, currentRD, requestList,
15      requestRatingList, outcomeList);
16      updatedRatings.add(csp, [newRating, newRD]);
17    }
18  }
19  else {
20    csp.didNotCompete(true);
21    updatedRatings.add(csp, [currentRating, currentRD]);
22  }
}

```

23 saveRatings(updatedRatings);

A.4 Παράδειγμα αιτήματος «ομοσπονδίας»

Το παρακάτω μήνυμα αρχείο αποτελεί ένα πρότυπο μηνύματος αίτησης «ομοσπονδίας» ενός παρόχου-καταναλωτή προς τον Exchange, ώστε να βρει τον κατάλληλο πάροχο που να ικανοποιεί τις προδιαγραφές. Το πρότυπο, αλλά και η συνολική αρχιτεκτονική προδιαγράφεται μέσω του εν ενεργεία προτύπου P2302 της IEEE²³.

```
<?xml version="1.0" encoding="UTF-8"?>
<iq type="set" from="user@cloud1.org" to="service.intercloudexchg.com"
id="cloud1_id1">
  <command xmlns="http://jabber.org/protocol/commands" node="constraint_catal
og_resources" action="execute">
    <iodata xmlns="urn:xmpp:tmp:io - data" type="input">
      <in>
        <constraints xmlns="http://www.csp/resOntology">
          <constraint>
            <attribute>availabilityQuantity</attribute>
            <value>99.999</value>
          </constraint>
          <constraint>
            <attribute>replicationFactor</attribute>
            <value>5</value>
          </constraint>
          <constraint>
            <attribute>tierCountries</attribute>
            <value>JAPAN</value>
          </constraint>
          <constraint>
            <attribute>StorageReplicationMethod</attribute>
            <value>AMQP</value>
          </constraint>
          <constraint>
            <attribute>InterCloudStorageAccess</attribute>
            <value>NFS</value>
          </constraint>
        </constraints>
      </in>
    </iodata>
  </command>
</iq>
```

²³ http://www.intercloudtestbed.org/uploads/2/1/3/9/21396364/intercloud_p2302_draft_0.2.pdf

Δημοσιεύσεις

Διεθνή περιοδικά

- A. Kapsalis, P. Kasnesis, C. Z. Patrikakis, I. S. Venieris, D. I. Kaklamani, "A Cooperative Fog Approach for Effective Workload Balancing", *IEEE Cloud Computing Magazine Connecting Fog and Cloud Computing* [accepted for publication]
- A. Kapsalis, I. S. Venieris, D. I. Kaklamani, "A Cloud Provider Rating System for Voluntary Inter-cloud Federations", *IEEE Cloud Computing Magazine Multicloud* [under review]
- D. Meridou, A. Kapsalis, M.-E. Papadopoulou, M. Karamanis, C. Patrikakis, I. S. Venieris and D. I. Kaklamani. "An Ontology-based Smart Production Management System", *IEEE IT Professional Smart Systems, November-December, 2015*, pp. 36-46
- D. Meridou, M.-E. Papadopoulou, P. Kasnesis, C. Patrikakis, G. Lamprinakos, A. Kapsalis, I. S. Venieris and D. I. Kaklamani. "The Health Avatar: Privacy-aware monitoring and management of health and lifelog data", *IEEE IT Professional Wearable Computing, September - October, 2015*, pp. 20-27
- A. Kapsalis, D. I. Kaklamani and D. Kazanskaia. "Living Labs: The Way to Successful Introduction of Innovations in Production Management ", *Cutter IT Journal*, Vol. 28, April, 2015, pp. 15-20
- J. Fursse, G. Lamprinakos, K. A. Papadopoulos, R. Jones, M. Clarke, N. Jones, A. Kapsalis, D. I. Kaklamani, S. Patsilnakos, I. S. Venieris and S. Thomson. "Presenting evaluation results from the usage of the inCASA Remote Healthcare Monitoring Platform", *International Journal of Integrated Care*, Vol. 13. 2013
- A. Kapsalis, G. Lamprinakos, K. A. Papadopoulos, D. I. Kaklamani and I. S. Venieris. "The inCASA project: improving the quality of life and social care for the ageing population", *International Journal of Integrated Care*, Vol. 12. 2012

Κεφάλαια σε βιβλία

- D. T. Meridou, M.-E. Ch. Papadopoulou, A. P. Kapsalis, P. Kasnesis, C. Z. Patrikakis, I. S. Venieris, D.-T. I. Kaklamani. "Improving Quality of Life with the Internet of Everything". *Springer*

Πρακτικά Συνεδρίων

- A. Kapsalis, P. Kasnesis, P. C. Theofanopoulos, P. Gkonis, C. Lavranos, D. I. Kaklamani, I. S. Venieris and G. Kyriacou. "A Cloud Platform for Classification and Resource Management of Complex Electromagnetic Problems". *In Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015)*. 2015. pp. 388 – 393
- A. Kapsalis, P. Gkonis, C. L. Zekios, D. I. Kaklamani, I. S. Venieris and G. A. Kyriacou. "An Intelligent Platform for Effective Management of Time-Consuming Electromagnetic Simulation Problems". *PIERS Proceedings*. 2015. pp. 2199 – 2203
- D. T. Meridou, A. P. Kapsalis, P. Kasnesis, C. Z. Patrikakis, I. S. Venieris and D. I. Kaklamani. "An Event-driven Health Service Bus". *In Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare -*

- "Transforming healthcare through innovations in mobile and wireless technologies"* (MOBIHEALTH 2015). October, 14-16, 2015, London, United Kingdom. EAI 2015
- A. Kapsalis, P.-F. Bakalos, C. Patrikakis, I. S. Venieris and D. Kazanskaia. "ARUM Living Lab: A walk through the future of production management". *XVII International Conference on Complex Systems: Control and Modeling Problems (CSCMP 2015)*. 2015
 - P. Gkonis, A. Kapsalis, K. Zekios, D. I. Kaklamani, I. S. Venieris, M. Chrysomallis and G. Kyriakou. "On the Performance of Spatial Multiplexing in MIMO-WCDMA Networks with Principal Component Analysis at the Reception". *In the Proceedings of the 9th European Conference on Antennas and Propagation (EuCAP 2015)*. 2015
 - P. Gkonis, A. Kapsalis, C. Zekios, D. I. Kaklamani, I. S. Venieris, M. Chryssomallis and G. Kyriacou. "Performance Evaluation of MIMO-WCDMA Networks Employing Principal Component Analysis At the Reception". *Proceedings of the 10th Conference on Antennas & Propagation (LAPC)*. 2014
 - G. Lamprinakos, A. Mousas, A. Kapsalis, D. I. Kaklamani, I. S. Venieris, A. Boufis, P. Karmiris and S. Mantzouratos. "Using FHIR to develop a healthcare mobile application". *Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth 2014)*. 2014