



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

**Αποδοτικοί, κλιμακώσιμοι και δίκαιοι αλγόριθμοι
αντιστοίχισης για ανάθεση εικονικών μηχανών σε
υποδομή υπολογιστικού νέφους**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΝΙΚΟΛΑΟΣ ΚΟΡΑΣΙΔΗΣ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

**Αποδοτικοί, κλιμακώσιμοι και δίκαιοι αλγόριθμοι
αντιστοίχισης για ανάθεση εικονικών μηχανών σε
υποδομή υπολογιστικού νέφους**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΝΙΚΟΛΑΟΣ ΚΟΡΑΣΙΔΗΣ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20η Ιουλίου 2016.

.....
Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

.....
Δημήτριος Φωτάκης
Επίκουρος Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Γκούμας
Λέκτορας Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

.....
Νικόλαος Κορασίδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Νικόλαος Κορασίδης, 2016.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Η ανάθεση εικονικών μηχανών σε φυσικά μηχανήματα που απαρτίζουν υπολογιστικές δομές (σύννεφα) έχει καθοριστική επίδραση τόσο στο όφελος του παρόχου όσο και στην ικανοποίηση των αναγκών του πελάτη. Παρ' όλο που αυτές οι παράμετροι μπορούν να μεταβάλλονται με το χρόνο και τις περιστάσεις, οι τρέχουσες πρακτικές διαχείρισης των πόρων ενός datacenter δεν είναι ικανοποιητικές, είτε διότι υλοποιούν πάγιες και στατικές πολιτικές είτε διότι φροντίζουν μονομερώς για το συμφέρον του διαχειριστή του σύννεφου. Σε αυτή την εργασία προτείνουμε ένα ευέλικτο και δημοκρατικό μηχανισμό αντιστοίχισης εικονικών μηχανών σε φυσικούς πόρους, ο οποίος αποσκοπεί στο να ικανοποιεί ισορροπημένα και τις δύο πλευρές του προβλήματος, πελάτες και παρόχους. Προς τούτο, η ανάθεση εικονικών μηχανών ανάγεται στο Πρόβλημα του Ισότιμου Σταθερού Ταιριάσματος (ESMP), αφού μετασχηματίσουμε τις πολιτικές κάθε πλευράς σε λίστες προτίμησης. Έπειτα, αναπτύσσουμε μια αποδοτική και κλιμακώσιμη ευριστική ώστε να αντλήσουμε προσεγγιστικές λύσεις-ταιριάσματα σε εύλογο χρονικό διάστημα, δεδομένου ότι το ESMP είναι ένα NP-Hard πρόβλημα. Η πειραματική αποτίμηση της προσέγγισής μας δείχνει ότι, χρησιμοποιώντας όχι περισσότερη μνήμη από όση διαθέτει ένας συνήθης προσωπικός υπολογιστής και γνωρίζοντας μόλις το άνω 20% των λιστών προτίμησης των αντιπροσώπων, μπορούμε να επιλύσουμε αποδοτικά άνω του 85% των παραγόμενων στιγμιότυπων ESMP σε λιγότερο από $N\sqrt{N}$ γύρους ταιριάσματος.

Λέξεις κλειδιά

ισότιμος σταθερός γάμος, σταθερός γάμος με ατελείς λίστες, ανάθεση εικονικών μηχανών

Abstract

VM placement, be it in public or private clouds, has a decisive impact on the provider's interest and the customer's needs alike, both of which may vary over time and circumstances. However, current resource management practices are either statically bound to specific policies or unilaterally favor the needs of Cloud operators. In this work we argue for a flexible and democratic mechanism to map virtual to physical resources, trying to balance satisfaction on both sides of the involved stakeholders. To that end, VM placement is expressed as an Equitable Stable Matching Problem (ESMP), where each party's policy is translated to a preference list. A practical approximation for this NP-hard problem, modified accordingly to ensure efficiency and scalability, is applied to provide equitable matchings within a reasonable time frame. Our experimental evaluation shows that, requiring no more memory than what a high-end desktop PC provides and knowing no more than the top 20% of the agent's preference lists, our solution can efficiently resolve more than 90% of large-scale ESMP instances within $N\sqrt{N}$ rounds of matchmaking.

Key words

equitable stable marriage, stable marriage with incomplete lists, VM placement

Ευχαριστίες

Η παρούσα εργασία σηματοδοτεί την ολοκλήρωση της ακαδημαϊκής μου πορείας στο Εθνικό Μετσόβιο Πολυτεχνείο. Θέλω να ευχαριστήσω όσες και όσους συνάντησα σε αυτή τη δύσκολη και πλούσια περίοδο της ζωής μου, για το μικρό ή εκτενές διάστημα που ήταν συνοδοιπόροι μου.

Θα ήθελα καταρχάς να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Νεκτάριο Κοζύρη για την ευκαιρία να εκπονήσω αυτή τη διπλωματική εργασία καθώς και για τις αναρίθμητες πλευρές της αρχιτεκτονικής σύγχρονων υπολογιστών που με δίδαξαν τα μαθήματά του. Για τον ενθουσιασμό και την καθοδήγησή τους καθ' όλα τα στάδια της εργασίας, για την υπομονή και την εμπιστοσύνη που μου έδειξαν, θέλω να ευχαριστήσω την Κατερίνα Δόκα και τον Γιάννη Γιαννακόπουλο του CSLab, με τους οποίους συνέγραψα και υπέβαλα την πρώτη μου ακαδημαϊκή δημοσίευση. Θα ήθελα επίσης να ευχαριστήσω τον καθηγητή κ. Γεώργιο Γκούμα για τη μεθοδικότητα που μου καλλιέργησε και το σεβασμό στη δημόσια φύση του ακαδημαϊκού έργου που μου ενέπνευσε. Οι εμπειρίες που αποκόμισα από τη διαγωνιστική μου προσπάθεια στο μάθημά του τροφοδότησαν μεγάλο μέρος της παρούσας εργασίας. Τέλος, για τις πολύτιμες γνώσεις που μου χάρισαν οι διαλέξεις του, την αδιαπραγμάτευτη αγάπη και το ενδιαφέρον του σε δύσκολες εποχές αλλά και την οξυδερκή του κριτική εκεί που τη χρειαζόμουν, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον καθηγητή μου κ. Δημήτρη Φωτάκη.

Η πορεία μου στο ΕΜΠ σημαδεύτηκε σε μεγάλο βαθμό από τους συμφοιτητές μου, με αρκετούς από τους οποίους έκανα, ελπίζω, ανεξίτηλες φιλίες. Ευχαριστώ τον Παναγιώτη, το Μανόλη και τον Πάτρικ για τις πλείστες βραδιές που περάσαμε περπατώντας και φιλοσοφώντας. Είμαι ευγνώμων για το υπέροχο κλίμα που δημιουργούσαν στο εργαστήριο ο Άρης, ο Άγγελος και ο Μάριος. Θα ήθελα να μνημονεύσω ειδικά το Διονύση για τον επαγγελματισμό και τον αλτρουισμό που μου μετέδωσε, και τον Ορέστη με τον οποίο είχαμε ποικίλες γόνιμες συνεργασίες, ακαδημαϊκές και μη. Ευχαριστώ την Ειρήνη και το Δημήτρη, οι οποίοι ήταν εκεί όταν τους χρειάστηκα, την Κατερίνα, το Γιάννη και το Σταύρο Ζ., οι οποίοι με στήριξαν με όλη τους την αγάπη, και το Σταύρο Α. που εκτός από επιστήθιος φίλος φροντίζει να προπορεύεται ώστε να μην επαναπαύομαι.

Η αμέριστη αγάπη των γονιών μου ήταν και είναι κινητήριος δύναμη για μένα. Τους ευχαριστώ για το κουράγιο που μου έδωσαν με τα λόγια και τα έργα τους σε κρίσιμες στιγμές.

Είχα την τύχη κατά την είσοδό μου στο ΕΜΠ να συναντήσω δύο εξέχοντες καθηγητές, τους κκ. Στάθη Ζάχο και Νίκο Παπασπύρου, οι οποίοι με ενέταξαν από νωρίς στις ακαδημαϊκές τους οικογένειες και μου εμπιστεύθηκαν, ως βοηθό εργαστηρίου, πολλές τάξεις φοιτητών. Θα έχω πάντα μια ξεχωριστή θέση στην καρδιά μου για τους ακαδημαϊκούς μου γονείς και τα μικρά ακαδημαϊκά μου αδέρφια. Η καθημερινή επαφή μαζί τους με γέμισε όνειρα, αυτοεκτίμηση και ταπεινότητα.

Νικόλαος Κορασίδης,

Αθήνα, 20η Ιουλίου 2016

Περιεχόμενα

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Περιεχόμενα	11
Κατάλογος πινάκων	13
Κατάλογος σχημάτων	15
Κατάλογος αρκτικόλεξων	17
1. Εισαγωγή	19
1.1 Σκοπός και Οργάνωση της διπλωματικής	20
2. Επισκόπηση προβλημάτων	21
2.1 Πρόβλημα Σταθερού Γάμου	21
2.1.1 Παράδειγμα SMP	21
2.2 Πρόβλημα Ισότιμου Σταθερού Γάμου	22
2.3 Ανάθεση εικονικών μηχανών (VM)	24
3. Μοντελοποίηση	27
3.1 Συνθετικά στιγμιότυπα	27
3.2 Κατανομές πόλωσης	28
3.3 Αιτιολόγηση για τις UNI, CONT, INV	29
4. Ισότιμος Σταθερός Γάμος με Ατελείς Λίστες	31
4.1 Μια κρίσιμη μετρική	31
4.2 ESMA με οικονομία μνήμης	33
4.3 Απόδειξη σταθερότητας	33
4.4 Άλλες προσεγγίσεις	34
5. Πειραματική αξιολόγηση	35
5.1 Ορθότητα εκτέλεσης	35
5.2 Διατηρώντας υψηλή ποιότητα	37

5.3	Αποτυγχάνοντας ομαλά	37
6.	Υλοποίηση	39
6.1	Γλώσσα & Περιβάλλοντα	39
6.2	Επιλογή πομπών	39
6.3	Ντετερμινιστική παραγωγή (υπο)στιγμοτύπων	40
6.4	Συμπαγής αναπαράσταση των συναρτήσεων προτίμησης	41
7.	Συμπεράσματα και μελλοντικές προεκτάσεις	43
	Βιβλιογραφία	45

Κατάλογος πινάκων

5.1	Εκτελέσεις που διασφαλίζουν ικανοποιημένους πομπούς σε ανώμαλο τερματισμό .	38
-----	---	----

Κατάλογος σχημάτων

2.1	Στιγμιότυπο SMP & λύση (σταθερό ταιρίασμα) [5]	21
2.2	Τοποθέτηση VM σε VM slots [19]	25
3.1	Παραδείγματα κατανομών πόλωσης (UNI, CONT, INV)	29
4.1	Δυσaréσκεια και μέγιστη ενεργός δυσaréσκεια πομπών για στιγμιότυπα με πλήρεις λίστες	32
5.1	Ποσοστό επιτυχημένων εκτελέσεων	36
5.2	Ποιότητα ταιριάσματος (κόστη) για διάφορα κλάσματα λιστών προτίμησης	37
5.3	Ικανοποιημένοι πομποί VS ομαλά αποτυχημένες εκτελέσεις	38

Κατάλογος αρκτικόλεξων

Αρκτικόλεξο	Επεξήγηση
ESMA	Equitable Stable Matching Algorithm
ESMP	Equitable Stable Marriage Problem
QoS	Quality of Service
SMA	Stable Matching Algorithm
SMP	Stable Marriage Problem
VM	Virtual Machine
WRS	Weighted Reservoir Sampling

Κεφάλαιο 1

Εισαγωγή

Η *ανάθεση εικονικών μηχανών* είναι ένα θεμελιώδες πρόβλημα που αντιμετωπίζεται από παρόχους υπολογιστών υποδομών (σύννεφων) [3, 2, 4]. Η πολιτική με την οποία οι εικονικές μηχανές αντιστοιχίζονται σε φυσικά μηχανήματα (hosts) καθορίζει σε μεγάλο βαθμό την αξιοποίηση του data center, το λειτουργικό κόστος και την κατανάλωση ενέργειας [2] ενώ ταυτόχρονα επηρεάζει τις επιδόσεις των VMs, και άρα, την ικανοποίηση των πελατών της υποδομής. Τα πρώτα σχήματα ανάθεσης που προτάθηκαν ήταν στατικά και επιχειρούσαν να βελτιστοποιήσουν μόνο την αξιοποίηση των φυσικών πόρων. Από τότε τα σχήματα αυτά γίνονται ολοένα και πιο εκλεπτυσμένα και δημοκρατικά, λαμβάνοντας υπόψη και τις ιδιαίτερες ανάγκες των πελατών. Μάλιστα, ένα σχήμα που βελτιώνει την αξιοποίηση του datacenter παράλληλα με τις επιδόσεις των VMs μπορεί να ωφελεί τόσο τους πελάτες όσο και τους παρόχους. Για παράδειγμα, ένας χρήστης μπορεί να επωφελείται από την συνεγκατάσταση (collocation) δύο VMs με απαιτήσεις για δίκτυο ταχείας απόκρισης (low latency) και ο πάροχος μπορεί ταυτόχρονα να εξοικονομεί πόρους (π.χ. εύρος ζώνης) αναθέτοντας τις δύο εικονικές μηχανές στο ίδιο φυσικό μηχάνημα.

Στη γενική περίπτωση, βέβαια, τα συμφέροντα των δύο πλευρών μπορεί να αντιτίθενται, περιπλέκοντας τη διαδικασία ανάθεσης. Για το λόγο αυτό, η ανάθεση VM προσεγγίζεται συχνά από μια παιγνιοθεωρητική σκοπιά: κάθε πελάτης και πάροχος μοντελοποιείται ως παίκτης που προσπαθεί μέσα από πολιτικές να μεγιστοποιήσει το όφελος που αποκομίζει (βλ. [14, 23]). Στην περίπτωση αυτή η “λύση” του προβλήματος εκφράζεται ως ένα σχήμα εξατομικευμένης χρεώσης για κάθε πελάτη, ώστε να διασφαλίζεται κάποιος βαθμός δικαιοσύνης μεταξύ των πελατών, αλλά όχι απαραίτητα μεταξύ πελατών και παρόχων. Εναλλακτικά, το πρόβλημα ανάθεσης μοντελοποιείται ως πρόβλημα ταιριάσματος (matching) εικονικών μηχανών και θέσεων (slots) σε φυσικά μηχανήματα [22, 21, 18]. Αντί για περίπλοκες αντικειμενικές συναρτήσεις, οι πολιτικές κάθε παίκτη εκφράζονται με *λίστες προτίμησης* που διατάσσουν τα μέλη του ετέρου συνόλου ανάλογα με την εκτιμώμενη καταλληλότητα. Έτσι, το πρόβλημα ανάθεσης ανάγεται στο *Πρόβλημα Σταθερού Γάμου (Stable Marriage Problem – SMP)* [9] και στόχος είναι να βρεθεί ένα *σταθερό* ταιρίασμα μεταξύ των μελών των δύο συνόλων.

Η έννοια της *σταθερότητας* είναι ιδιαίτερα σημαντική σε αυτή τη μοντελοποίηση, διότι εξασφαλίζει ότι δεν υπάρχει κανένα ζεύγος VM και VM slot που να είναι αμοιβαίως προτιμότερα το ένα για το άλλο από ότι για τις τρέχουσες αναθέσεις τους. Συνεπώς, ένα σταθερό ταιρίασμα εκφράζει κάποιου είδους βέλτιστη ανάθεση πόρων σε χρήστες. Μπορεί να αποδειχθεί ότι υπάρχει ένα σταθερό ταιρίασμα σε κάθε στιγμιότυπο του SMP και ότι μπορεί να βρεθεί με τον αλγόριθμο των Gale-Shapley [9]. Ωστόσο, ο αλγόριθμος αυτός αποτυγχάνει να αντιμετωπίσει ισότιμα τις δύο πλευρές. Αντιθέτως, λόγω του άκαμπτου τρόπου με τον οποίο μοιράζει τους ρόλους στα δύο μέρη,

παράγει ένα ταίριασμα που εγγενώς μεροληπτεί υπέρ μίας πλευράς. Επιπλέον, η εκτέλεση του αλγορίθμου απαιτεί μνήμη της τάξης του $O(N^2)$ όπου N είναι ο αριθμός των VM (και των VM slot), χαρακτηριστικά που τον κάνουν ακατάλληλο για τη διαχείριση πόρων σε σύγχρονα datacenter όπου εξυπηρετούνται κάθε στιγμή δεκάδες χιλιάδες εικονικές μηχανές.

1.1 Σκοπός και Οργάνωση της διπλωματικής

Προσπαθώντας να υπερβούμε τα εμπόδια της μεροληψίας και της τετραγωνικής διόγκωσης της απαιτούμενης μνήμης, προτείνουμε μια νέα προσέγγιση στο πρόβλημα της ανάθεσης VM, η οποία αναζητά δίκαιες λύσεις βασισμένη σε *ατελείς λίστες προτίμησης (incomplete preference lists)*. Συγκεκριμένα, μεταφράζουμε τις πολιτικές του παρόχου και των χρηστών σε λίστες προτίμησης και ύστερα εφαρμόζουμε την ευριστική του [12] ώστε να λάβουμε ένα αμερόληπτο ταίριασμα μεταξύ VM και VM slots. Για να μετριάσουμε τις απαιτήσεις σε μνήμη περιορίζουμε δραστικά τις λίστες σε K θέσεις ($K < N$) για κάθε πρόσωπο. Αυτή η επιλογή μεταβάλλει σημαντικά τις ιδότητες του προβλήματος, αφού, για παράδειγμα, κάθε πρόταση που προέρχεται από ένα χαμηλά προτιμώμενο πρόσωπο (με θέση μεγαλύτερη του K στη λίστα προτίμησης του δέκτη) απορρίπτεται αυτόματα. Στην παρούσα εργασία ερευνούμε πειραματικά τον αντίκτυπο αυτού του περιορισμού στην ορθότητα και την απόδοση της ευριστικής του [12] και δείχνουμε ότι η επιλογή μας είναι κατάλληλη για μεγάλα στιγμιότυπα του SMP, τα οποία προέρχονται από datacenter με πολλούς πελάτες και πολλούς διαθέσιμους πόρους. Αξιολογώντας διάφορες παραμέτρους που επηρεάζουν το πρόβλημα της ανάθεσης, προκύπτει ότι η προσέγγισή μας είναι ικανή να δώσει δίκαιες λύσεις σε στιγμιότυπα έως και 20000 VMs μέσα σε λίγα δευτερόλεπτα, χρησιμοποιώντας μόλις το 20% της μνήμης που απαιτείται στη συνήθη περίπτωση, χωρίς να επηρεάζεται η ορθότητα ή η ποιότητα της εξόδου του αλγορίθμου.

Η συμβολή μας μπορεί να περιληφθεί ως εξής:

- Μοντελοποιούμε το πρόβλημα ανάθεσης VM βάσει πολιτικών σαν ένα στιγμιότυπο *ισότιμου (ή ουδετερόφυλου) (sex-equal) SMP* και χρησιμοποιούμε έναν ευριστικό και προσεγγιστικό αλγόριθμο [12] για να αντλήσουμε μια δίκαιη λύση.
- Εισάγουμε μια παραλλαγή της παραπάνω προσέγγισης η οποία χρησιμοποιεί *ατελείς λίστες προτίμησης* ώστε να είναι κλιμακώσιμη σε μεγάλα στιγμιότυπα.
- Παρουσιάζουμε μια εκτενή πειραματική αξιολόγηση της μεθόδου μας υπό διάφορες περιστάσεις που ανακύπτουν σε σύγχρονα datacenter και δείχνουμε ότι είναι κατάλληλη για καταστάσεις όπου μεγάλες παρτίδες από VM πρέπει να δρομολογηθούν για εγκατάσταση σε hosts.

Η διάρθρωση της παρούσας έχει ως εξής: Στο κεφάλαιο 2 παρουσιάζονται τα κύρια προβλήματα που μας απασχόλησαν, το SMP και η ανάθεση εικονικών μηχανών. Στο κεφάλαιο 3 αναλύουμε την αναγωγή του δεύτερου προβλήματος στο πρώτο και ακολουθεί, στο κεφάλαιο 4, η πειραματική θεμελίωση της κεντρικής ιδέας για την κλιμακώσιμη επίλυση του στιγμιότυπου που προκύπτει. Η εκτενή πειραματική αξιολόγηση της προσέγγισης παρουσιάζεται στο κεφάλαιο 5 ενώ στο κεφάλαιο 6 παρατίθενται λεπτομέρειες της υλοποίησης. Τέλος, στο κεφάλαιο 7 καταγράφονται τα συμπεράσματά μας και οι προτάσεις για μελλοντική έρευνα και ανάπτυξη.

Κεφάλαιο 2

Επισκόπηση προβλημάτων

2.1 Πρόβλημα Σταθερού Γάμου

Ένα στιγμιότυπο I του προβλήματος σταθερού γάμου (Stable Marriage Problem – SMP) αποτελείται από N άνδρες και N γυναίκες, όπου κάθε πρόσωπο έχει μια λίστα προτίμησης που διατάσσει ολικά τα μέλη του άλλου φύλου. Εάν ένας άνδρας m προτιμά την w_1 από την w_2 , γράφουμε $w_1 \succ_m w_2$. Παρόμοιος συμβολισμός χρησιμοποιείται για τις προτιμήσεις των γυναικών.

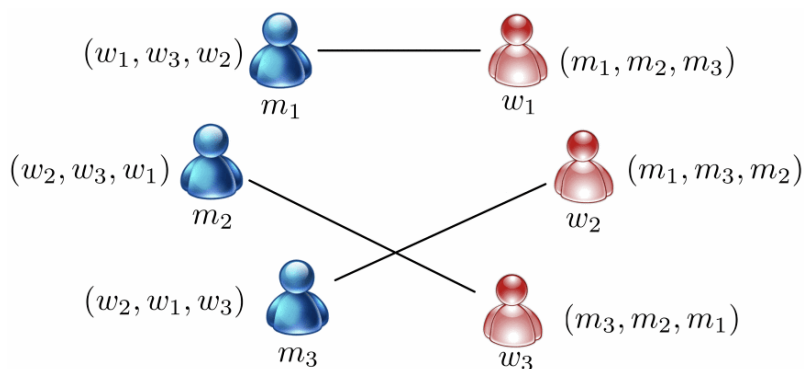
Ένα τέλειο ταίριασμα M στο I είναι ένα σύνολο από ζεύγη ανδρών-γυναικών του I , ώστε κάθε άνδρας και κάθε γυναίκα να εμφανίζονται ακριβώς μία φορά. Όταν ένα ζεύγος (m, w) ανήκει στο M , γράφουμε $M(m) = w$ και $M(w) = m$. Ένας άνδρας m και μια γυναίκα w αποτελούν κωλύον ζεύγος του M εάν καθένας τους προτιμά τον άλλο από τον τρέχον σύντροφό του, δηλαδή αν:

$$M(m) \neq w \wedge w \succ_m M(m) \wedge m \succ_w M(w)$$

Ένα ταίριασμα M καλείται *ασταθές* εάν υπάρχει κωλύον ζεύγος για το M , διαφορετικά καλείται *σταθερό*. Το SMP ζητά να βρεθεί για δεδομένο I ένα τέλειο και σταθερό ταίριασμα M .

2.1.1 Παράδειγμα SMP

Στην παρακάτω εικόνα φαίνεται ένα στιγμιότυπο SMP μαζί με ένα σταθερό ταίριασμα $M = \{(m_1, w_1), (m_2, w_3), (m_3, w_2)\}$. Οι τούπλες αναπαριστούν τις προτιμήσεις, από αριστερά προς τα δεξιά. Το ταίριασμα που προκύπτει αν εναλλάξουμε τις συντρόφους των m_2 και m_3 είναι *ασταθές*: Ο m_3 προτιμά την w_2 από τη w_3 και η w_2 προτιμά τον m_3 από τον m_2 , δηλαδή το (m_3, w_2) θα ήταν κωλύον ζεύγος για το $M' = \{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}$.



Σχήμα 2.1: Στιγμιότυπο SMP & λύση (σταθερό ταίριασμα) [5]

Ο καθιερωμένος αλγόριθμος για την επίλυση του SMP, όπως προτάθηκε από τους Gale and Shapley [9] (εφεξής SMA), εκτελεί μια σειρά επαναλήψεων-γύρων. Σε κάθε γύρο οι άντρες που δεν έχουν ταιρί προτείνουν στις γυναίκες και αυτές δέχονται ή απορρίπτουν τις προτάσεις, ανάλογα με τις προτιμήσεις τους. Ο αλγόριθμος τερματίζει μετά από το πολύ N^2 γύρους και βρίσκει ένα τέλειο και σταθερό ταιρίασμα. Ένα σοβαρό ελάττωμα του ταιριάσματος που παράγει ο εν λόγω αλγόριθμος είναι ότι εξασφαλίζει την υψηλότερη δυνατή ικανοποίηση για το σύνολο των ανδρών (proposer/male-optimal) και, ταυτοχρόνως, τη χαμηλότερη δυνατή ικανοποίηση για το σύνολο των γυναικών (acceptor/female-pessimal). Γνωρίζουμε, ωστόσο, ότι για ένα δεδομένο στιγμιότυπο I υπάρχουν πολλά διαφορετικά σταθερά ταιριάσματα, επομένως, έχει νόημα να αναζητήσουμε λύσεις που δεν είναι απλώς σταθερές αλλά ικανοποιούν και κάποιο άλλο ποιοτικό μέτρο, το οποίο συνήθως αντανακλά *έμφυλη δικαιοσύνη*, δηλαδή κάποιου είδους *αμεροληψία* του αλγορίθμου έναντι των δύο συνόλων.

2.2 Πρόβλημα Ισότητας Σταθερού Γάμου

Στη βιβλιογραφία έχουν προταθεί τρία κριτήρια (κόστη) τα οποία αποτιμούν την ποιότητα ενός ταιριάσματος M . Έστω ότι συμβολίζουμε με $pr_m(w)$ (αντίστοιχα $pr_w(m)$) τη θέση της γυναίκας w στη λίστα προτίμησης του άντρα m (αντίστοιχα, τη θέση του m στη λίστα της w). Η ποσότητα αυτή συνήθως είναι κάποιος ακέραιος στο διάστημα $[0, N)$, όπου N είναι το πλήθος των προσώπων σε κάθε πλευρά και χαμηλότερες τιμές αντιστοιχούν σε υψηλότερα επιθυμητούς συντρόφους. Με βάση αυτά, ορίζουμε, για δεδομένο M :

δυσαρέσκεια (regret cost)

$$r(M) = \max_{(m,w) \in M} \max\{pr_m(w), pr_w(m)\}$$

από κοινού κόστος (egalitarian cost)

$$c(M) = \sum_{(m,w) \in M} pr_m(w) + pr_w(m)$$

έμφυλο κόστος (sex-equality cost)

$$d(M) = \left| \sum_{(m,w) \in M} pr_m(w) - \sum_{(m,w) \in M} pr_w(m) \right|$$

Για τα πρώτα δύο κόστη έχουν βρεθεί αποδοτικοί αλγόριθμοι [13] που υπολογίζουν βέλτιστα ταιριάσματα. Όμως, όπως προκύπτει από τη βιβλιογραφία, μόνο ένα σταθερό ταιρίασμα που βελτιστοποιεί το έμφυλο κόστος εξασφαλίζει ουσιαστικά ίση αντιμετώπιση για τις δύο πλευρές. Η εύρεση ενός ταιριάσματος με χαμηλό $d(M)$ ονομάζεται *Πρόβλημα Ισότητας Σταθερού Γάμου (Equitable Stable Marriage Problem – ESMP)* και αποδεικνύεται ότι είναι σημαντικά πιο δύσκολο, για την ακρίβεια, είναι NP-hard σύμφωνα με το [17]. Έτσι, όλοι οι αλγόριθμοι που λύνουν το ESMP χρησιμοποιούν ευριστικές τεχνικές δύο ειδών: είτε παράγουν πολλά διαφορετικά ταιριάσματα και διαλέγουν αυτό με το χαμηλότερο $d(M)$ είτε επιχειρούν να κατασκευάσουν εξ αρχής ένα *ισότιμο ταιρία-*

σμα (*equitable matching*) φροντίζοντας τα δύο σύνολα να εναλλάσσουν τους ρόλους του πομπού (*proposer*) και του δέκτη (*acceptor*), σε αντίθεση με τον κλασσικό αλγόριθμο, όπου καθένα από τα δύο σύνολα παίζει μόνο έναν ρόλο.

Από τα δύο αυτά είδη το πρώτο είναι απλό αλλά αδικαιολόγητα χρονοβόρο για τις online απαιτήσεις της παρούσας εργασίας. Θα μπορούσαμε να αντιμετωπίσουμε το ζήτημα της απόδοσης παραλληλοποιώντας τις διάφορες εκτελέσεις, όμως αυτό θα είχε απαγορευτικά υψηλές ανάγκες σε μνήμη, πόρος που αποτελεί το πιο κρίσιμο εμπόδιο όπως θα δούμε σε επόμενη ενότητα. Έτσι, στρεφόμαστε σε ευριστικές λύσεις για το ESMP. Μια πολλά υποσχόμενη ευριστική έχει προταθεί στο [12]: ο *Αλγόριθμος Ισότιμου Σταθερού Ταιριάσματος (Equitable Stable Matching Algorithm – ESMA)* λειτουργεί εναλλάσσοντας με τυχαίο τρόπο τους ρόλους των πομπών και των δεκτών σε κάθε γύρο. Έχει τεκμηριωθεί πειραματικά ότι ο ESMA υπερέχει από άλλους αλγορίθμους για το ίδιο πρόβλημα (π.χ. [8]) στους εξής τομείς:

1. Τερματίζει σε όλα τα μεγάλα στιγμιότυπα που έχουν ελεγχθεί.
2. Παράγει ισότιμα ταιριάσματα υψηλής ποιότητας.
3. Έχει χρόνο εκτέλεσης ανταγωνιστικό με τον SMA, υπερτερώντας πολύ του ανταγωνισμού.

Ο ESMA χρησιμοποιεί μια απλή απεριοδική συνάρτηση σαν ψευδοτυχαίο δίκαιο νόμισμα με το οποίο αποφασίζει τα μέλη ποιου συνόλου χρήζονται πομποί σε κάθε γύρο (και τα μέλη του άλλου συνόλου δέκτες). Κάθε πομπός προτείνει στον πλέον προτιμώμενο δέκτη από τον οποίον δεν έχει απορριφθεί, ενώ οι δέκτες αξιολογούν τις προτάσεις και κρατούν αυτή που τους ικανοποιεί περισσότερο, απορρίπτοντας τις υπόλοιπες, όπως στον SMA. Όταν οι δέκτες ενός γύρου γίνουν πομποί σε κάποιον επόμενο γύρο, προτείνουν και αυτοί με τη σειρά τους στα πρόσωπα της αρεσκείας τους, εκτός εάν προτιμούν τον τρέχοντα σύντροφό τους από το δέκτη στον οποίο πρόκειται να προτείνουν. Αυτό υλοποιείται ως εξής: κάθε πρόσωπο διατηρεί δύο δείκτες στη λίστα προτίμησής του, έναν που δηλώνει τη θέση του τρέχοντος συντρόφου και έναν που δηλώνει τη θέση του τρέχοντος πιο επιθυμητού δέκτη. Καθώς συμβαίνουν διάφορες προτάσεις που οδηγούν σε γάμους και διαζύγια, οι δείκτες αυτοί αλλάζουν θέση μέσα στη λίστα κάθε προσώπου. Ο ESMA τερματίζει όταν όλα τα πρόσωπα είναι ικανοποιημένα, δηλαδή είναι ταιριασμένα με την πλέον προτιμώμενη επιλογή τους που όμως δεν τους έχει απορρίψει, πράγμα που γίνεται αντιληπτό επειδή οι δύο δείκτες συμπίπτουν.

Όπως αναλύεται στο [12], αυτό το σχήμα εναλλαγής ρόλων μπορεί να οδηγήσει σε επαναλήψεις κάποιων προτάσεων, αφού δύο μέλη από διαφορετικά σύνολα μπορεί να ταιριάξουν περισσότερες από τη μία φορές, ανάλογα με το ποιος κάνει την πρόταση. Τέτοια περιστατικά είναι αδύνατο να συμβούν στον SMA, όπου τόσο οι πομποί όσο και οι δέκτες διασχίζουν τις λίστες προτίμησης με *μονοτονικό* τρόπο: οι πομποί εκκινούν από τις αρχές της λίστας και αυξάνουν την ανοχή τους όσο κυλούν οι γύροι ενώ οι δέκτες κάνουν το αντίθετο. Δυστυχώς, τα μοτίβα του ESMA δεν είναι σχεδόν ποτέ μονοτονικά, επομένως δεν μπορούμε να δώσουμε μια απόδειξη τερματισμού. Μάλιστα, μπορούμε εύκολα να φτιάξουμε (μικρά) στιγμιότυπα όπου ο αλγόριθμος πέφτει σε ατέρμονα βρόχο εκτός αν η ανάθεση ρόλων στα σύνολα γίνεται με τυχαίο τρόπο. Η ψευδοτυχαία συνάρτηση που έχει προταθεί στο [12] φαίνεται να μην προκαλεί τέτοια προβλήματα αλλά να οδηγεί τον αλγόριθμο σε τερματισμό σε όλα τα ελεγμένα στιγμιότυπα. Αυτή η ιδιότητα, σε συνδυασμό

με τη χαμηλή πολυπλοκότητα κάθε γύρου ($\Theta(N)$), μας οδηγεί να επιλέξουμε τον ESMA ως το μηχανισμό που θα χρησιμοποιήσουμε για να αναθέσουμε εικονικές μηχανές σε hosts, θεωρώντας τα VMs των πελατών και τις διαθέσιμες υποδοχές (*slots*) των hosts του datacenter ως τα δύο σύνολα. Το τελικό ταίριασμα αποτελείται από ζεύγη VM και slots, υλοποιώντας έτσι μια ανάθεση.

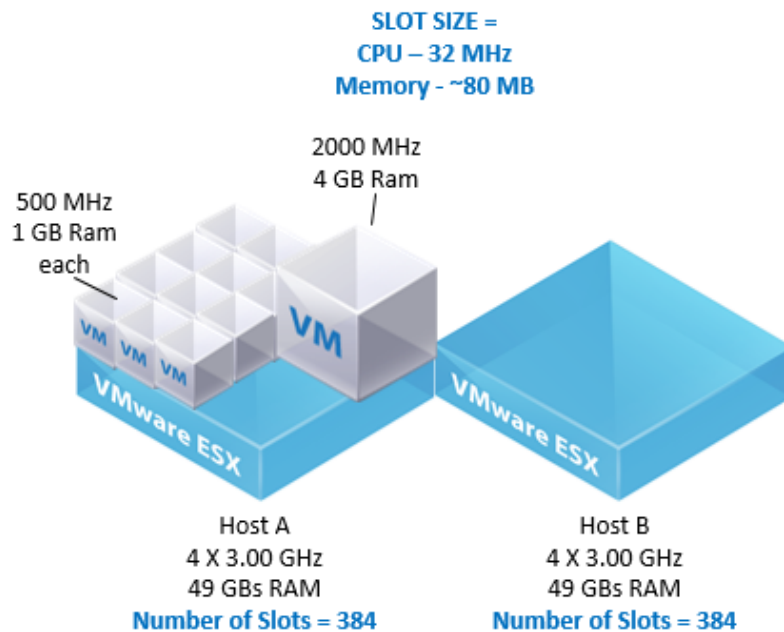
2.3 Ανάθεση εικονικών μηχανών (VM)

Η τοποθέτηση/ανάθεση εικονικών μηχανών σε πραγματικά μηχανήματα (hosts) είναι ένα πρόβλημα που έχει ερευνηθεί εκτενώς στη θεματολογία των υπολογιστικών νεφών. Μπορούμε να χωρίσουμε τις προτεινόμενες προσεγγίσεις σε δύο διακριτές κατηγορίες: σε αυτές που επιλύουν το πρόβλημα λαμβάνοντας υπόψη μόνο το όφελος του παρόχου και σε αυτές που επιχειρούν να πετύχουν ένα καλό συμβιβασμό μεταξύ του συμφέροντος του παρόχου και της ικανοποίησης των πολιτικών του πελάτη.

Στην πρώτη κατηγορία βρίσκουμε προσεγγίσεις που βελτιστοποιούν διάφορους δείκτες λειτουργίας του datacenter, όπως η αξιοποίηση των μηχανημάτων, το κόστος λειτουργίας, η κατανάλωση ενέργειας κ.ά. ενώ παράλληλα τηρούν τις προδιαγραφές της σύμβασης με τον πελάτη (Service Level Agreements – SLAs). Για παράδειγμα, το [3] εστιάζει στην εφάπαξ τοποθέτηση των εικονικών μηχανών σε ένα datacenter που δέχεται ροή από αιτήσεις εγκατάστασης (streaming deployment requests) ενώ το [2] προτείνει σχήματα ανάθεσης VM που στοχεύουν στην εξοικονόμηση ενέργειας και στη μείωση του λειτουργικού κόστους χρησιμοποιώντας πολιτικές μεταγκατάστασης (migration) που διατηρούν ανέπαφο το επίπεδο εξυπηρέτησης (Quality of Service – QoS). Στο [6] περιγράφεται μια μεθοδολογία κατηγοριοποίησης των VM που στοχεύει στη βελτιστοποίηση της αποδοτικότητας του datacenter μεγιστοποιώντας τις επιδόσεις των VMs. Παρόμοια, στο [20] οι συγγραφείς προτείνουν την αναγωγή του προβλήματος ανάθεσης σε στιγμιότυπο του διακριτού προβλήματος σακιδίου (discrete knapsack) και θέτουν στόχο να βελτιστοποιήσουν μια αντικειμενική συνάρτηση που ποσοτικοποιεί το λειτουργικό κόστος υπό τους περιορισμούς των SLAs. Η προσέγγιση αυτή επιλύει το παραγόμενο στιγμιότυπο χρησιμοποιώντας constraint programming. Ωστόσο, καμία από τις προηγούμενες μεθόδους δε λαμβάνει υπόψη πολιτικές του πελάτη, δηλαδή του χρήστη της εικονικής μηχανής.

Αντιθέτως, οι μεθοδολογίες της δεύτερης κατηγορίας θεωρούν ότι οι προτιμήσεις του πελάτη αφορούν και επηρεάζουν ουσιαστικά την ανάθεση VM. Μπορούμε και αυτές να τις διακρίνουμε σε δύο υποκατηγορίες ανάλογα με το θεωρητικό υπόβαθρο στο οποίο βασίζονται τη μοντελοποίηση των προτιμήσεων. Στην πρώτη υποκατηγορία συναντάμε παιγνιοθεωρητικές προσεγγίσεις που μοντελοποιούν τους πελάτες και τους παρόχους ως παίκτες με διακριτά και πιθανώς αντικρουόμενα συμφέροντα (εκπεφρασμένα ως πολυπαραγοντικές συναρτήσεις ωφέλειας). Σε αυτές, η επιτυχής επίλυση του προβλήματος αντιστοιχεί στην εύρεση ενός σημείου ισορροπίας του “παιχνιδιού”, η οποία εικάζεται ότι οδηγεί σε αμερόληπτη αντιμετώπιση όλων των παικτών. Συγκεκριμένα, οι χρήστες του datacenter ορίζουν τις παραμέτρους του φόρτου εργασίας που ζητούν να διεκπεραιωθούν και ο πάροχος ορίζει μια χρέωση που να είναι δίκαιη και αμοιβαίως αποδεκτή (συνεπώς επωφελής) τόσο για τον πελάτη όσο και για τον πάροχο. Ομοίως, οι Xu et al. στο [23] παρουσιάζουν μια τέτοια προσέγγιση που επιχειρεί να αντιμετωπίσει τίμια τους πελάτες ενός cluster ενώ παράλληλα αυξάνει την αξιοποίησή του.

Μια εναλλακτική γραμμή έρευνας επιλέγει να μοντελοποιεί το πρόβλημα της ανάθεσης VM



Σχήμα 2.2: Τοποθέτηση VM σε VM slots [19]

σαν Πρόβλημα Σταθερού Γάμου, όπου οι εικονικές μηχανές των χρηστών αντιστοιχούνται σε κατάλληλους hosts, με σεβασμό τόσο στις πολιτικές που ορίζει ο χρήστης όσο και στους περιορισμούς του datacenter. Η προσέγγιση αυτή εμπεριέχει μεθόδους μετασχηματισμού αφηρημένων πολιτικών κάθε οντότητας (εικονική μηχανή, host) σε συγκεκριμένες προτιμήσεις που μπορούν να περιγραφούν ως στιγμιότυπο SMP. Ένα τέτοιο σχήμα αναγωγής προτείνεται στο [22]: το σύστημα δέχεται πολιτικές των χρηστών και εκτελεί τον αλγόριθμο Gale-Shapley προκειμένου να βρει μια αντιστοίχιση VM και διαθέσιμων υπολογιστικών πόρων. Σε ανάλογο πνεύμα, το [18] χρησιμοποιεί τον ίδιο αλγόριθμο για να βρει μια αντιστοίχιση που να μειώνει την κατανάλωση ενέργειας. Οι προσεγγίσεις αυτές πάσχουν από το πάγιο ελάττωμα του αλγορίθμου αυτού: οι αντιστοιχίσεις είναι βέλτιστες για το σύνολο που λειτουργεί ως proposer (συνήθως ο πάροχος) και λιγότερο καλές για το άλλο σύνολο (συνήθως ο πελάτης).

Στη δική μας εργασία χρησιμοποιούμε έναν σχετικά καινούριο αλγόριθμο εύρεσης σταθερού ταιριάσματος (ESMA [12]), ο οποίος παράγει σταθερές και δίκαιες (αμερόληπτες) λύσεις. Στην παρούσα εργασία βελτιώνουμε τον ESMA με διάφορους τρόπους ώστε να εξακολουθεί να είναι κλιμακώσιμος παρά την τετραγωνική αύξηση των δεδομένων εισόδου όταν επιλύουμε μεγάλα στιγμιότυπα. Σημειώνουμε ότι ο ESMA παράγει μία μοναδική δίκαιη λύση αντίθετα με μεθόδους που παράγουν πολλά ταιριάσματα και διαλέγουν αυτό με τα καλύτερα χαρακτηριστικά, όπως στο [21]. Εφόσον, μάλιστα, έχει επιδόσεις αντίστοιχες του SMA, είναι ιδιαίτερα κατάλληλος για χρήση σε datacenter όπου οι αιτήσεις εγκατάστασης πρέπει να ικανοποιούνται σε πραγματικό χρόνο.

Κεφάλαιο 3

Μοντελοποίηση

Προκειμένου να ανάγουμε ένα πρόβλημα τοποθέτησης VM σε στιγμιότυπο SMP, χρειάζεται να εξάγουμε λίστες προτίμησης από τις πολιτικές του πελάτη και του εξυπηρετητή. Αυτό συνήθως είναι μια επίπονη διαδικασία τα αποτελέσματα της οποίας εξαρτώνται στενά από τη μοντελοποίηση των δύο πλευρών. Για παράδειγμα, το [22] υποθέτει ότι οι χρήστες ορίζουν τις πολιτικές τους με τη μορφή κανόνων, χρησιμοποιώντας μετρικές (κόστη) όπως $1/\text{CPU}_{\text{freq}}$ οι οποίες χωρίζουν τους διαθέσιμους hosts σε ομάδες όπου όλα τα μέλη μιας ομάδας έχουν παρόμοια τιμή για τη δεδομένη μετρική, και η ομάδα με το χαμηλότερο κόστος προτιμάται. Οι μετρικές αυτές μπορούν να αφορούν πάγια χαρακτηριστικά του μηχανήματος (αρχιτεκτονική της CPU, μέγεθος μνήμης, χωρητικότητα δίσκου, ταχύτητα σύνδεσης, κ.ά.) ή δυναμικές παραμέτρους του περιβάλλοντος εκτέλεσης, (π.χ. απαίτηση για χαμηλό I/O contention). Με τη σύνθεση αρκετών τέτοιων μετρικών, οι ομάδες προτιμήσεων γίνονται όλο και πιο λεπτομερείς, ώστε, οριακά, να μπορούν να παρασταθούν ως διατεταγμένη λίστα. Αντίστοιχες μετρικές μπορούμε να ορίσουμε και για τον εξυπηρετητή, ο οποίος κατατάσσει τους πελάτες του αφενός αντιπαραβάλλοντας τις προδιαγραφές των υποβαλλόμενων έργων/VMs με την κατάσταση του datacenter (αξιοποίηση πόρων, κατανάλωση ενέργειας κ.ά.) κι αφετέρου το πάγιο καθεστώς του πελάτη (π.χ. αν έχει απλή ή προνομιούχα συνδρομή).

Πιο προχωρημένες τεχνικές τοποθέτησης (όπως στο [14]) απαιτούν από τον πελάτη να προσδιορίσει ένα *δυναμικό προφίλ* απαιτήσεων: ο χρήστης υποβάλλει ένα γράφο των ζητούμενων πόρων μαζί με ένα χρονοδιάγραμμα αξιοποίησής τους. Μια τέτοια πολιτική απαιτεί έναν αρκετά διαφορετικό αλγόριθμο εξαγωγής προτιμήσεων. Σε κάθε περίπτωση, θεωρούμε ότι ο πελάτης υποβάλλει τις απαιτήσεις του με ένα τρόπο που είναι εύληπτος για εκείνον και ο εξυπηρετητής αναλαμβάνει να τις μετασχηματίσει κατάλληλα ώστε να παραχθεί ένα στιγμιότυπο ταιριάσματος. Στην παρούσα εργασία εστιάζουμε κυρίως στην επίλυση του παραγόμενου προβλήματος και όχι στο μετασχηματισμό. Θεωρούμε ότι το σχήμα εξαγωγής ακολουθεί έναν απλό μηχανισμό σαν του [22] και μελετούμε πώς μεταβάλλεται η απόδοση της προσέγγισής μας όταν οι πολιτικές των δύο μερών οδηγούν σε προτιμήσεις με διάφορες στατιστικές ιδιότητες.

3.1 Συνθετικά στιγμιότυπα

Δε στάθηκε δυνατό να συλλέξουμε πραγματικά δεδομένα επαρκούς όγκου ώστε να αξιολογήσουμε την προσέγγισή μας. Για το λόγο αυτό χρησιμοποιήσαμε συνθετικά δεδομένα που προέκυψαν από τη δική μας μοντελοποίηση των πελατών και των εξυπηρετητών. Η προσέγγιση που δοκιμάσαμε αρχικά ήταν να χρησιμοποιήσουμε *Reservoir Sampling* ώστε (για κάθε έναν πελάτη ή εξυπηρετητή) να διαλέξουμε μια διάταξη K μελών του άλλου συνόλου από N με ομοιόμορφα

τυχαίο τρόπο. Αυτό είναι αρκετά εύκολο στην υλοποίηση και μοντελοποιεί σωστά την περίπτωση όπου όλα τα μέλη είναι περίπου της ίδιας ποιότητας και οι προτιμήσεις εξαρτώνται κυρίως από τα δυναμικά χαρακτηριστικά τόσο του VM όσο και του host.

Ωστόσο, σε ρεαλιστικές εφαρμογές ανακύπτουν διαφοροποιήσεις που αξιολογούνται σχεδόν ομόφωνα, όπως για παράδειγμα οι δυνατότητες του μηχανήματος ή το καθεστώς του πελάτη, όπως αναφέρθηκε. Οι παράμετροι αυτές μπορούν να έχουν διακριτή μεταβολή (απλοί VS προνομιακοί συνδρομητές) ή πρακτικά συνεχή μεταβολή (χωρητικότητα δίσκου). Για να συμπεριλάβουμε αυτή την παρατήρηση στη μοντελοποίησή μας, θεωρούμε ότι υπάρχει μια *συνάρτηση πόλωσης* η οποία αντιστοιχεί a priori κάθε μέλος του προβλήματος σε κάποια *πόλωση*, η οποία είναι μια θετική τιμή που σχετίζεται με την πιθανότητα το εκάστοτε μέλος να βρεθεί μεταξύ των υψηλών θέσεων προτίμησης στις λίστες των μελών του άλλου συνόλου.

Προκειμένου να παράξουμε στιγμιότυπα που να σέβονται τις πολώσεις των μελών, χρησιμοποιούμε *Weighted Random Sampling* (εφεξής WRS), όπως περιγράφεται στο [7]. Αν και η πόλωση καθαυτή δεν είναι πιθανότητα, ακολουθώντας την ερμηνεία *A-Res* του WRS έχουμε ότι: εάν το άθροισμα όλων των πολώσεων των μελών ενός συνόλου είναι B , τότε ένα μέλος με πόλωση b_i έχει πιθανότητα b_i/B να επιλεγεί σε ένα δείγμα, και άρα να είναι υψηλά επιθυμητό.

3.2 Κατανομές πόλωσης

Η εφαρμογή πόλωσης υλοποιείται αποδίδοντας διαφορετικές τιμές (βάρη) σε κάθε μέλος και ύστερα δειγματοληπτώντας τα βεβαρυμένα σύνολα μέσω WRS. Στην παρούσα εργασία μελετούμε τρεις διαφορετικές κατανομές βαρών: την *ομοιόμορφη κατανομή*, την *contested κατανομή* και την *φθίνουσα κατανομή*.

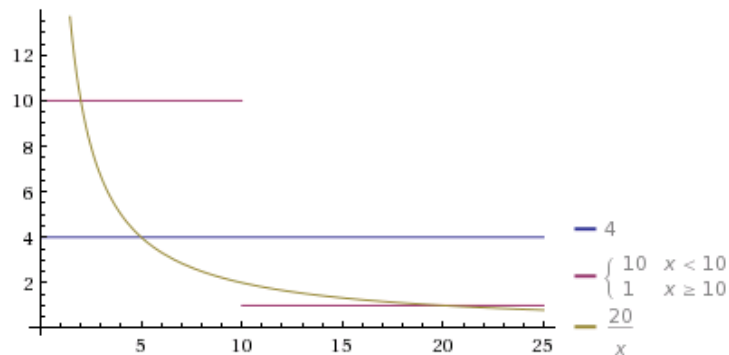
Στην ομοιόμορφη κατανομή (UNI), κάθε μέλος έχει το ίδιο, σταθερό βάρος. Επομένως, έχει την ίδια πιθανότητα με οποιοδήποτε άλλο μέλος να καταλάβει μια θέση στη λίστα προτίμησης ενός μέλους του άλλου συνόλου. Με τον τρόπο αυτό παράγονται λίστες που επιλέγονται ομοιόμορφα τυχαία από όλες τις K -μεταθέσεις των N στοιχείων.

Στην *contested κατανομή* (CONT), τα μέλη χωρίζονται σε δύο διακριτά σύνολα: ένα σύνολο περιλαμβάνει τα μέλη υψηλής ποιότητας που εκτιμάται ότι προτιμώνται από όλους, ενώ το δεύτερο σύνολο περιλαμβάνει όλα τα άλλα μέλη. Τα μέλη του δεύτερου συνόλου λαμβάνουν κάποιο σταθερό βάρος w ενώ τα μέλη του πρώτου συνόλου λαμβάνουν βάρος hw . Φυσικά, η παράμετρος h είναι σχετικά αυθαίρετη –στην παρούσα εργασία επιλέξαμε $h = 10$. Μια παράμετρος p καθορίζει τα σχετικά μεγέθη των συνόλων. Στην παρούσα εργασία εξετάζουμε τι συμβαίνει όταν το υψηλής ποιότητας σύνολο περιέχει περίπου 10% όλων των μελών ($p = 0.1$). Οι λίστες προτίμησης που παράγονται από αυτή την κατανομή βαρών έχουν αντίστοιχο προφίλ με των συνόλων: στις πρώτες θέσεις βρίσκονται (σε τυχαία σειρά) τα μέλη με υψηλή πόλωση και στις ακόλουθες θέσεις τα υπόλοιπα, επίσης με τυχαία σειρά.

Τέλος, στη *φθίνουσα κατανομή* (INV), κάθε μέλος λαμβάνει ένα βάρος που είναι μια *φθίνουσα* συνάρτηση του αναγνωριστικού του ($i < j \implies b_{a_i} > b_{a_j}$). Στα πειράματά μας χρησιμοποιήσαμε τη συνάρτηση $f_a(i) = (i + 1)^{-a}$, η οποία αντιστοιχεί φυσικούς αριθμούς σε θετικά βάρη. Μια φθίνουσα κατανομή παράγει δείγματα που έχουν μεγάλη συνάφεια μεταξύ τους: μέλη με υψηλά βάρη τείνουν να βρίσκονται περίπου στις ίδιες (υψηλές) θέσεις στις προτιμήσεις. Το φαινόμενο αυτό μπορεί να γίνει περισσότερο ή λιγότερο έντονο ανάλογα με τη συγκεκριμένη συνάρτηση

πόλωσης. Στην περίπτωση μας, όταν $a \rightarrow 0$ παίρνουμε πιο ομοιόμορφα βάρη ενώ όταν $a \gg 1$ παίρνουμε μια κατανομή βαρών σαν αυτή της contested. Η πειραματική αξιολόγηση έγινε για $a = 1.0$.

Εφόσον οι CONT και INV παράγουν λίστες προτίμησης που δεν είναι ομοιόμορφα επιλεγμένες από το σύνολο των K - μεταθέσεων, περιμένουμε ότι θα δημιουργούν πιο δύσκολα στιγμιότυπα, καθώς πολλά μέλη θα διαγωνίζονται για τις ίδιες προτιμήσεις. Σημειώνουμε, ωστόσο, ότι η καθαυτό παραγωγή των λιστών γίνεται από μια ψευδοτυχαία διαδικασία η οποία λαμβάνει υπόψη την πόλωση αλλά δεν καθορίζεται αποκλειστικά από αυτή. Ο ακριβής μηχανισμός παραγωγής στιγμιοτύπων αναλύεται στην ενότητα της υλοποίησης (6).



Σχήμα 3.1: Παραδείγματα κατανομών πόλωσης (UNI, CONT, INV)

3.3 Αιτιολόγηση για τις UNI, CONT, INV

Όπως αναφέρθηκε, ο σκοπός της πόλωσης είναι να μοντελοποιηθούν περιστάσεις που προκύπτουν στο πλαίσιο του προβλήματος ανάθεσης εικονικών μηχανών, και ειδικότερα τα πάγια χαρακτηριστικά των μελών. Δύο παράμετροι μας αφορούν κυρίως: η μεταβολή στον εξοπλισμό ενός datacenter με την πάροδο του χρόνου και η μεταβλητότητα στις απαιτήσεις των πελατών για απόδοση.

Ένα καινούριο datacenter συνήθως περιέχει μηχανήματα περίπου των ίδιων προδιαγραφών και άρα της ίδιας εκτιμώμενης απόδοσης, κατάσταση που μοντελοποιούμε με UNI κατανομή. Κάνοντας μια ελάσσουσα αναβάθμιση ή επέκταση υλικού, προκύπτει μια μικρή ομάδα από μηχανήματα πιο ισχυρά από το μέσο όρο, όπως σε μια CONT κατανομή. Μετά από πολλές συσσωρευμένες αναβαθμίσεις κι επεκτάσεις διαμορφώνεται με υλικό ποικίλων και σχετικά ομαλά μεταβαλλόμενων προδιαγραφών, σαν το προφίλ της INV κατανομής.

Αντίστοιχα, οι πελάτες του datacenter μπορούν να είναι διαφοροποιημένοι ως προς τα επίπεδα υπηρεσίας που απολαμβάνουν. Το datacenter μπορεί να παρέχει μόνο ένα επίπεδο QoS (μοντελοποιημένο ως UNI) ή πιθανώς μια οικονομική συνδρομή και μια προνομιούχα (CONT). Εναλλακτικά, μπορεί να παρέχει μια πιο εξειδικευμένη υπηρεσία η ποιότητα της οποίας να μεταβάλλεται σύμφωνα με μια παράμετρο μεγάλου εύρους, π.χ. τη διαθεσιμότητα της υπηρεσίας (“nines of uptime”), οπότε χρησιμοποιούμε μοντέλο INV για τους πελάτες.

Από τους εννέα συνδυασμούς κατανομών για πελάτες και παρόχους ελέγξαμε πειραματικά τους τρεις συνδυασμούς που έχουν ίδια πόλωση σε κάθε πλευρά. Αυτό έγινε για απλότητα αλλά

συμβαδίζει με τη διαίσθηση που έχουμε για το πλαίσιο του προβλήματος: Ένα καινούριο datacenter μικρής κλίμακας μάλλον απευθύνεται σε πελάτες χωρίς εξειδικευμένες ανάγκες. Αντίθετα, κάθε μεγάλο datacenter με ευρεία και ποικίλη πελατεία είναι μάλλον αρκετά παλιό ώστε να έχει συσσωρεύσει εξοπλισμό διαφόρων ειδών και απόδοσης.

Κεφάλαιο 4

Ισότιμος Σταθερός Γάμος με Ατελείς Λίστες

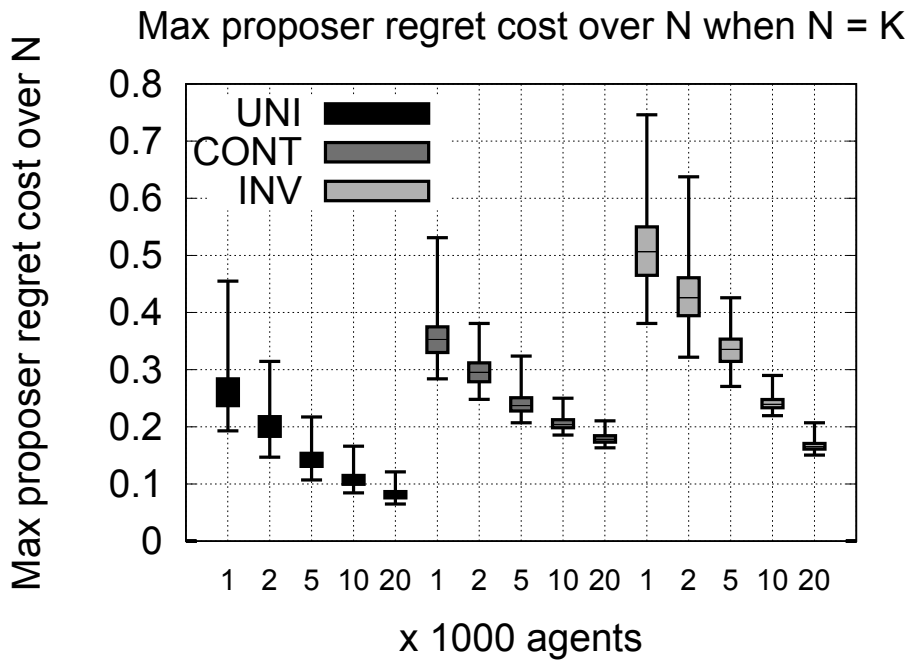
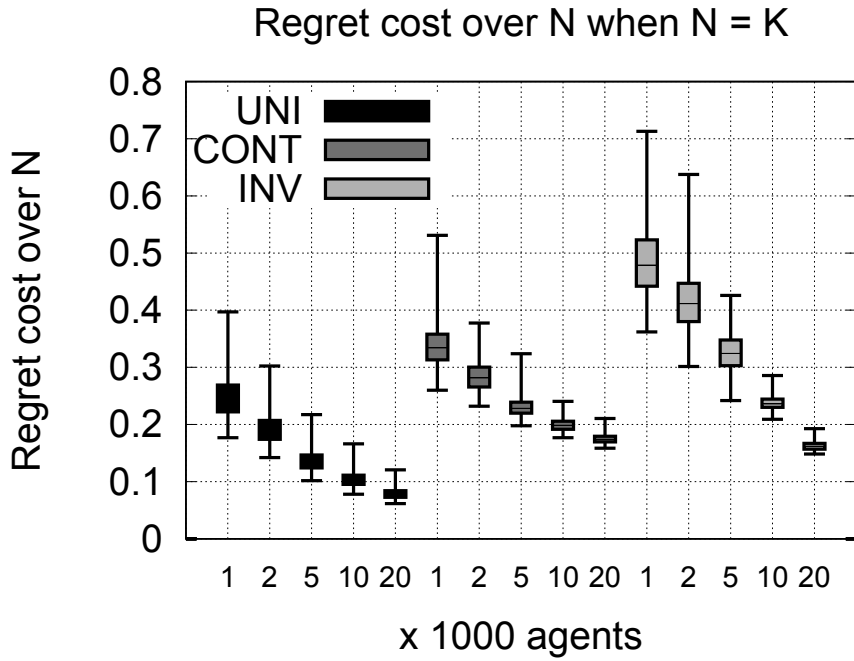
Σύμφωνα με τη διατύπωση του προβλήματος, η επιτυχής ανάθεση εικονικών μηχανών σε hosts ανάγεται στην επίλυση μεγάλων στιγμιοτύπων του ESMP. Δυστυχώς, ο όγκος των δεδομένων που απαιτούνται για την περιγραφή ενός στιγμιότυπου αυξάνεται τετραγωνικά με το μέγεθος των συνόλων: ένα στιγμιότυπο ESMP με $N = 20000$ απαιτεί περισσότερο από 6 GB χώρου. Η επιτυχής επίλυση ακόμη μεγαλύτερων στιγμιοτύπων απαιτεί πόρους που συνήθως είναι διαθέσιμοι μόνο σε high-end μηχανήματα. Συνεπώς, ο πρωταρχικός μας στόχος είναι να βρούμε μεθόδους ώστε να μειώσουμε δραστικά τη μνήμη που απαιτεί μια εκτέλεση του ESMA.

4.1 Μια κρίσιμη μετρική

Τρέχοντας τον ESMA πάνω από τυχαία στιγμιότυπα με τουλάχιστον 5000 μέλη, παρατηρούμε ότι βρίσκουμε λύσεις με χαμηλή δυσαρέσκεια, παρ' όλο που ο ESMA δεν κάνει καμία ρητή προσπάθεια να βελτιστοποιήσει αυτό το κόστος. Μάλιστα, επιλύοντας ακόμη μεγαλύτερα στιγμιότυπα προκύπτει ένα ενδιαφέρον μοτίβο: η δυσαρέσκεια ως κλάσμα του N μειώνεται καθώς το N αυξάνει. Με απλά λόγια αυτό σημαίνει ότι καθώς αυξάνονται οι διαθέσιμοι σύντροφοι για κάθε πρόσωπο, ο ESMA γίνεται ολοένα και πιο αποτελεσματικός στο να ταιριάζει πρόσωπα σε συντρόφους που είναι υψηλά επιθυμητά.

Η χαμηλή δυσαρέσκεια πιθανόν είναι ένα καλοδεχούμενο μεν, τυχαίο δε χαρακτηριστικό του τελικού ταιριάσματος, ωστόσο τα στατιστικά από τις εκτελέσεις αποκαλύπτουν μια πιο συστηματική αιτία. Πράγματι, σε κάθε επιτυχή εκτέλεση του ESMA όλα τα πρόσωπα προτείνουν αποκλειστικά στις κορυφαίες προτιμήσεις τους σε όλους τους γύρους. Καταγράφοντας την ενεργό “δυσαρέσκεια των πομπών” από τον πρώτο ως και τον τελευταίο γύρο, παρατηρούμε ότι το μέγιστό της είναι μόλις λίγο μεγαλύτερο από την τελική δυσαρέσκεια (η τελευταία υπολογισμένη πάνω από όλα τα πρόσωπα). Συνεπώς, αν και ένα πρόσωπο μπορεί –ως δέκτης– να δεχτεί προτάσεις από ένα πομπό που είναι χαμηλά στη λίστα αρεσκείας του, κανένα πρόσωπο –ως πομπός– δεν προτείνει ποτέ σε ένα μη υψηλά επιθυμητό δέκτη. Τα ακόλουθα διαγράμματα box-n-whisker είναι ενδεικτικά.

Από τα διαγράμματα προκύπτει άμεσα ότι η συμπεριφορά των πομπών εξασφαλίζει χαμηλή δυσαρέσκεια. Ωστόσο, μπορεί να δειχθεί ότι και οι δέκτες ενεργούν προς την ίδια κατεύθυνση. Ας θεωρήσουμε ένα δέκτη που είναι ταιριασμένος με ένα σχετικά ανεπιθύμητο πομπό. Το ταιριασμα αυτό είναι μάλλον ικανοποιητικό για τον πομπό αλλά όχι και για το δέκτη. Σύμφωνα με την ευριστική του ESMA, για να είναι ένα πρόσωπο a ικανοποιημένο με το σύντρόφό του b , ο b θα πρέπει να βρίσκεται ψηλότερα στη λίστα προτιμήσεων του a από το σημείο ανοχής. Το σημείο αυτό κα-



Σχήμα 4.1: Δυσaréσκεια και μέγιστη ενεργός δυσaréσκεια πομπών για στιγμιότυπα με πλήρεις λίστες

θορίζεται από τα πρόσωπα που έχουν απορρίψει τον a και γίνεται ολοένα πιο ελαστικό όσο ο a μένει χωρίς ταίρι. Όμως, όπως είπαμε, ένας πομπός στον ESMA προτείνει μόνο σε (και απορρίπτεται μόνο από) υψηλά προτιμώμενους δέκτες, επομένως περιμένουμε όλα τα πρόσωπα να έχουν σχετικά αυστηρά σημεία ανοχής. Έτσι, για να είναι ένας δέκτης ικανοποιημένος όταν δέχεται μια πρόταση, ο πομπός θα πρέπει να είναι της υψηλής αρεσκείας του. Ένα μη-ικανοποιητικό ταίριασμα δεν μπορεί να είναι μόνιμο. Επομένως, όταν όλα τα πρόσωπα είναι ικανοποιημένα, είναι επειδή όλα είναι ταιριασμένα σε ένα υψηλά προτιμώμενο σύντροφο, εξ ου και η χαμηλή δυσαρέσκεια.

4.2 ESMA με οικονομία μνήμης

Διαισθητικά, εφόσον κανένα πρόσωπο δεν φτιάχνει σταθερό δεσμό με ένα σύντροφο της απαρεσκείας του, ένα μεγάλο κλάσμα των λιστών προτίμησης είναι ουσιαστικά άχρηστο για την κατάστρωση της λύσης. Πρόκειται για το τμήμα των λιστών που απαριθμεί τα αναγνωριστικά των μνηστήρων με χαμηλά σκορ καθώς και τις καταχωρίσεις στους χάρτες που αντιστοιχούν τα σκορ σε αναγνωριστικά (βλ. κεφάλαιο 6). Όπως φαίνεται από τα γραφήματα αυτό το άχρηστο τμήμα εξαρτάται σημαντικά από την συνάρτηση πόλωσης αλλά ξεπερνά το 70% για $N \geq 10000$. Ο κύριος σκοπός της υπόλοιπης εργασίας είναι να διαπιστωθεί αν και σε τι βαθμό είναι δυνατόν να λύσουμε επιτυχώς στιγμιότυπα του ESMP όπου οι λίστες προτίμησης έχουν περιοριστεί στο “ενεργό” τους κομμάτι, αυτό που κωδικοποιεί τους υψηλά προτιμώμενους συντρόφους. Τα αποτελέσματά μας, όπως περιγράφονται στο επόμενο κεφάλαιο, δείχνουν ότι αυτό είναι εφικτό στη μεγάλη πλειοψηφία των στιγμιότυπων, χωρίς μάλιστα να θυσιάζεται η ποιότητα της λύσης.

4.3 Απόδειξη σταθερότητας

Μπορούμε να αποδείξουμε ότι όποτε ο ESMA τερματίζει επιτυχώς, το ταίριασμα που προκύπτει είναι σταθερό, παρά τις ατελείς λίστες. Η απόδειξη δεν είναι μια τετριμμένη επανάληψη της απόδειξης ότι το SMA με ατελείς λίστες είναι σταθερό, καθώς στην περίπτωσή μας δεν έχουμε ένδεια δεδομένων αλλά έχουμε εσκεμμένα αγνοήσει ένα μέρος της διαθέσιμης πληροφορίας. Συγκεκριμένα, θέλουμε να δείξουμε ότι το ταίριασμα που παράγεται από τις ατελείς λίστες είναι σταθερό ακόμη και εάν τα πρόσωπα γνωρίζουν τις πλήρεις λίστες. Έστω ένα στιγμιότυπο με N πρόσωπα σε κάθε σύνολο, όπου κάθε πρόσωπο γνωρίζει τις K πρώτες προτιμήσεις του. Διακρίνουμε δύο περιπτώσεις:

Εάν $K = N$, μπορούμε να αποκλείσουμε την ύπαρξη κωλύοντος ζεύγους επικαλούμενοι την απόδειξη σταθερότητας του αρχικού ESMA, όπως αναλύεται στο [12]: όταν όλα τα πρόσωπα είναι ικανοποιημένα, το προκύπτον ταίριασμα είναι σταθερό και κανένα πρόσωπο δεν έχει λόγο να περιμένει ότι θα γίνει αποδεκτό από κάποιο σύντροφο που προτιμά περισσότερο από τον τρέχοντα. Παρατηρούμε ότι το ίδιο ισχύει και για $K < N$, αφού αφενός ο ESMA δεν τερματίζει έως ότου ικανοποιηθούν όλα τα πρόσωπα και αφετέρου δουλεύουμε με λίστες που δεν έχουν ενδιάμεσα κενά.

Στην περίπτωση που $K < N$, μπορούμε να δείξουμε ότι κανένα επιπλέον κωλύον ζεύγος δεν προκύπτει από τη στιγμή που αποκαλύπτονται οι πλήρεις λίστες, με εις άτοπον απαγωγή. Ας υποθέσουμε ότι έχουμε παράξει ένα ταίριασμα M με ατελείς λίστες και έστω ότι ένα κωλύον ζεύγος (m_i, w_j) σχηματίζεται μόνον μετά την αποκάλυψη. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι ο

m_i προτιμά την w_j από την τρέχουσα σύντροφό του w_{m_i} . Εφόσον το κωλύον ζεύγος δεν υφίστατο κατά τον τερματισμό του ESMA, είτε ο m_i βρίσκεται στο περικομμένο τμήμα της λίστας της w_j είτε η w_j βρίσκεται στο περικομμένο τμήμα της λίστας του m_i . Στην πρώτη περίπτωση, η w_j δεν μπορεί να προτιμά τον m_i από τον τρέχον σύντροφό της, άτοπο. Στη δεύτερη περίπτωση, εφόσον η w_{m_i} βρίσκεται χαμηλότερα από την w_j στη λίστα του m_i , συμπεραίνουμε ότι και οι δύο περιέχονται στο περικομμένο τμήμα της λίστας του m_i , άτοπο, εφόσον ο m_i ταίριαξε με σύντροφο στο ενεργό κομμάτι τις λίστας. Άρα, το ζεύγος (m_i, w_j) δεν είναι κωλύον.

Εφόσον κατά τον τερματισμό όλα τα πρόσωπα είναι ικανοποιημένα, δεν υπάρχουν κωλύοντα ζεύγη, ούτε όμως δημιουργούνται μετά την αποκάλυψη των πλήρων λιστών. Άρα, η επίλυση του ESMP με ατελείς λίστες παράγει σταθερά ταιριάσματα στις περιπτώσεις που τερματίζει.

4.4 Άλλες προσεγγίσεις

Εκτός από τα πειραματικά δεδομένα, κύρια πηγή έμπνευσης για την παραπάνω προσέγγιση αποτελεί η δουλειά των Iwama et al. [15], οι οποίοι μελετούν το πρόβλημα της εύρεσης ενός ισότιμου ταιριάσματος που ταυτόχρονα ελαχιστοποιεί το από κοινού κόστος. Συγκεκριμένα, αναζητούν ένα M με ελάχιστο $c(M)$ και $d(M)$ που να είναι μικρότερο κατ' απόλυτη τιμή από e φορές το χειρότερο δυνατό, δηλαδή αυτό που παράγεται από ένα proposer-optimal ταίριασμα. Δείχνουν ότι το πρόβλημα αυτό είναι NP-Hard και δίνουν έναν $2 - (e - \delta) / (2 + 3e)$ -προσεγγιστικό αλγόριθμο (ως προς το $d(M)$) με χρόνο εκτέλεσης $\mathcal{O}\left(N^{4 + \frac{1+\epsilon}{\delta}}\right)$, για $0 < \delta < e$.

Αν και αποτελεί σημαντικό θεωρητικό επίτευγμα, ο αλγόριθμος αυτός έχει απαγορευτικά υψηλή πολυπλοκότητα για το παρόν πρόβλημα. Συγκριτικά, ο τροποποιημένος ESMA έχει πολυπλοκότητα $\mathcal{O}(rN)$, όπου r το πλήθος των γύρων που λαμβάνουν χώρα έως ότου βρεθεί ταίριασμα ή τερματιστεί η εκτέλεση με ανώμαλο τρόπο. Όπως θα δούμε στην επόμενη ενότητα, ο ESMA με ατελείς λίστες βρίσκει ταιριάσματα εξαιρετικής ποιότητας ως προς όλα τα κόστη σε όλα τα στιγμιότυπα όπου τερματίζει και απαιτεί $r < N\sqrt{N}$ για να επιλύσει τη μεγάλη πλειοψηφία των στιγμιότυπων. Συνεπώς, παρά τις θεωρητικές του ελλείψεις, θεωρούμε ότι είναι μια πολύ ανταγωνιστική προσέγγιση, η οποία χρήζει περαιτέρω μελέτης.

Κεφάλαιο 5

Πειραματική αξιολόγηση

Με οδηγό την ανάλυση που προηγήθηκε, στην παρούσα ενότητα ερευνούμε πειραματικά τι ποσό των δεδομένων μπορούμε να αγνοήσουμε και ωστόσο να έχουμε επιτυχή ταιριάσματα μέσω του ESMA. Η testsuite μας αποτελείται από στιγμιότυπα ESMP τα οποία ποικίλουν με βάση τις εξής παραμέτρους: πλήθος προσώπων ανά σύνολο (N), κλάσμα της πλήρους λίστας που είναι διαθέσιμο (K/N) και μοντελοποίηση πόλωσης. Για κάθε συνδυασμό παραμέτρων από τους παρακάτω τρέξαμε 500 εκτελέσεις, κάθεμιά με διαφορετική είσοδο:

- $N \in \{1000, 2000, 5000, 10000, 20000\}$
- $K/N \in \{0.1, 0.2, 0.5, 1.0\}$
- ομοιόμορφη πόλωση, contested πόλωση για $p = 0.1$, φθίνουσα πόλωση για $a = 1.0$

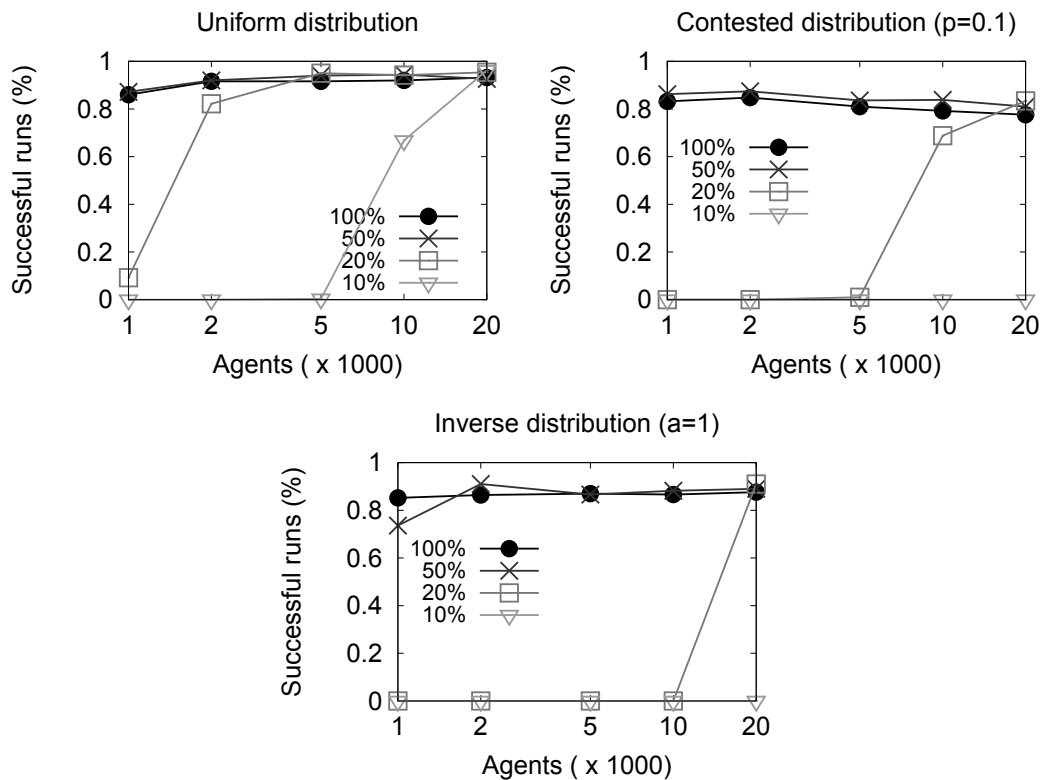
5.1 Ορθότητα εκτέλεσης

Θεωρούμε μια εκτέλεση *ορθή/επιτυχής* εάν τερματίζει ομαλά μετά από το πολύ $N\sqrt{N}$ γύρους ταιριάσματος, διαφορετικά η εκτέλεση κρίνεται *αποτυχημένη*. Στα ακόλουθα γραφήματα παρουσιάζεται το ποσοστό των επιτυχών εκτελέσεων για κάθε συνδυασμό των παραμέτρων. Η αιτιολόγηση του συγκεκριμένου ορίου γύρων γίνεται σε επόμενη ενότητα, μαζί με μια διερεύνηση των αποτυχημένων εκτελέσεων. Παρατηρούμε ότι για μικρό κλάσμα K/N και οποιαδήποτε πόλωση, το ποσοστό των επιτυχών εκτελέσεων αυξάνεται ελαφρώς καθώς το N μεγαλώνει, πράγμα που σημαίνει ότι η προσέγγισή μας είναι πιο κατάλληλη για μεγάλα στιγμιότυπα ESMP παρά για μικρά, όπου το κέρδος είναι ασήμαντο.

Έξετάζοντας τα αποτελέσματα ως προς το K/N προκύπτει ένα πολύ ενδιαφέρον εύρημα: το ποσοστό επιτυχίας αρχικά αυξάνεται καθώς το ενεργό μήκος των λιστών μειώνεται αλλά στη συνέχεια μειώνεται ταχύτατα ή μηδενίζεται άπαξ και οι λίστες γίνουν ανεπαρκώς μικρές. Φαίνεται να υπάρχει ένα *sweet spot* για το κατάλληλο μέγεθος της λίστας, το οποίο εξαρτάται από το N και την πόλωση, για παράδειγμα είναι κάπου μεταξύ 20% και 50% για την contested κατανομή όταν $N = 10000$. Ερμηνεύουμε αυτό το αποτέλεσμα ως εξής: Όπως αναλύθηκε προηγουμένως, κανένας δέκτης δεν ταιριάζει σταθερά με έναν σύντροφο της απαρεσκείας του. Μειώνοντας το μέγεθος της λίστας προτίμησης (του δέκτη) προκαλεί την αυτόματη αποτυχία τέτοιων ασταθών ταιριασμάτων, αφού ο πομπός απορρίπτεται από την πρώτη στιγμή. Έτσι, ο πομπός αναγκάζεται να συνεχίσει τις προτάσεις σε άλλους δέκτες αντί να παραμένει ανενεργός. Αυτό μειώνει το πλήθος των γύρων που απαιτούνται για ένα πλήρες ταιρίασμα, άρα περισσότερες εκτελέσεις επιτυγχάνουν εντός

του δεδομένου ορίου γύρων. Φυσικά, όταν οι λίστες έχουν μικρύνει υπερβολικά, δεν υφίστανται πια αρκετές επιλογές ώστε να ταιριάζουν όλα τα πρόσωπα, και ο ESMA αποτυγχάνει.

Άλλα πειράματα, που έχουν παραλειφθεί για λόγους συντομίας κατέδειξαν ότι το φαινόμενο του “sweet spot” μειώνεται ή αναστρέφεται ολοσχερώς αν επιτρέψουμε ένα πολύ μεγάλο όριο γύρων, της τάξης του $\mathcal{O}(N^2)$. Συγκεκριμένα, όλα τα στιγμιότυπα με πλήρεις λίστες τερματίζουν αν τους επιτραπεί να τρέχουν για μη περιορισμένο αριθμό γύρων ενώ αντιθέτως μερικά στιγμιότυπα με ατελείς λίστες φαίνεται να μην τερματίζουν ποτέ. Το φαινόμενο αυτό είναι αναμενόμενο σύμφωνα με όσα γνωρίζουμε από τη βιβλιογραφία για SMP με ατελείς λίστες καθώς και την εμπειρία του [12].



Σχήμα 5.1: Ποσοστό επιτυχημένων εκτελέσεων

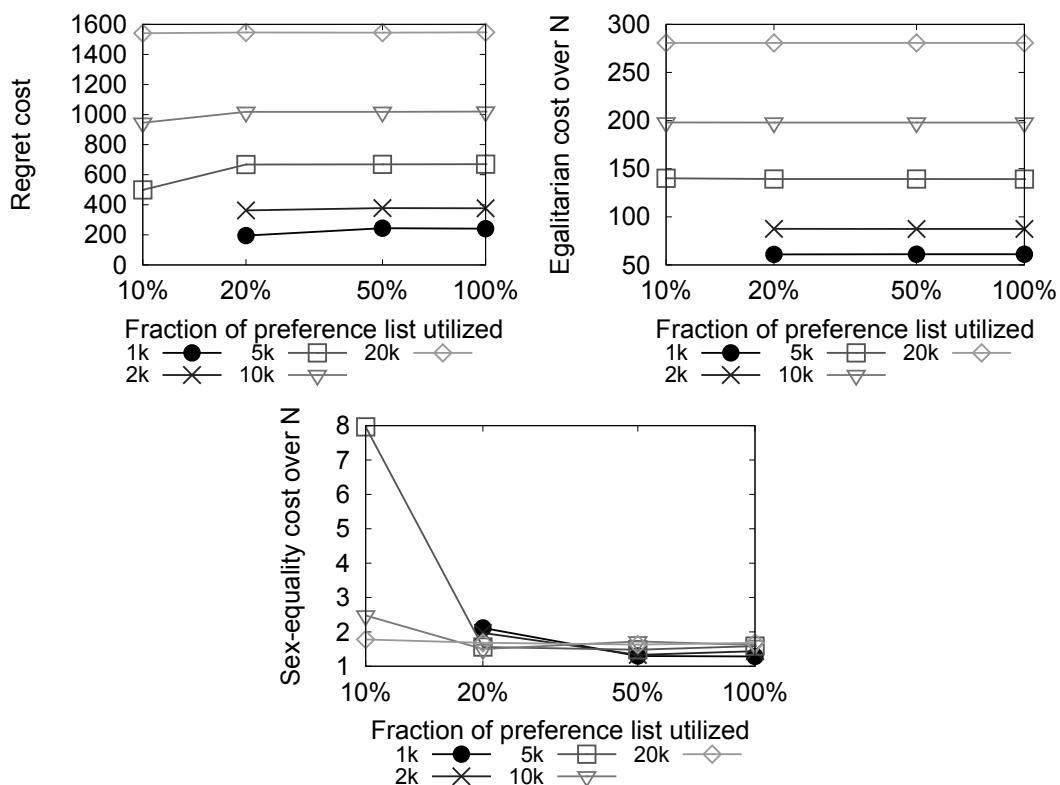
Όπως αναμένεται, η ομοιόμορφη πόλωση παράγει τα ευκολότερα στιγμιότυπα, αντίθετα η contested και η φθίνουσα πόλωση είναι πιο απαιτητικές σε μνήμη. Για $N = 20000$ και χρησιμοποιώντας μόνο το 20% των δεδομένων, μπορούμε να λύσουμε πάνω από 90% των στιγμιότυπων ομοιόμορφης και φθίνουσας πόλωσης και πάνω από το 80% των στιγμιότυπων με contested πόλωση, χρησιμοποιώντας όχι περισσότερο από 610 MB μνήμης και το πολύ 2.8 εκατομμύρια γύρους ταιριάσματος. Αυτό δείχνει τη δύναμη της προσέγγισής μας.

Ως προς την απόδοση της εκτέλεσης, ο αλγόριθμός μας χρειάζεται περίπου 15 δευτερόλεπτα για να επιλύσει στιγμιότυπα όπου $N = K = 20000$ ενώ για τον ίδιο αριθμό μελών αλλά $K = 2000$ (μικρότερο κατά μία τάξη μεγέθους), ο αλγόριθμος χρειάζεται περίπου 6.5 δευτερόλεπτα (διάμεσες τιμές). Αμφότερα τα πειράματα εκτελέστηκαν σε μηχάνημα με λειτουργικό Linux και πυρήνα Xeon X5650 CPU @ 2.67GHz με 24 νήματα επεξεργασίας. Η παρατηρούμενη επιτάχυνση αποδίδεται στο γεγονός ότι μικρότερες λίστες οδηγούν σε ταχύτερη προσπέλαση στις δομές που περιγράφουν τις

προτιμήσεις του κάθε προσώπου, όπως αναλύεται και στην ενότητα υλοποίησης.

5.2 Διατηρώντας υψηλή ποιότητα

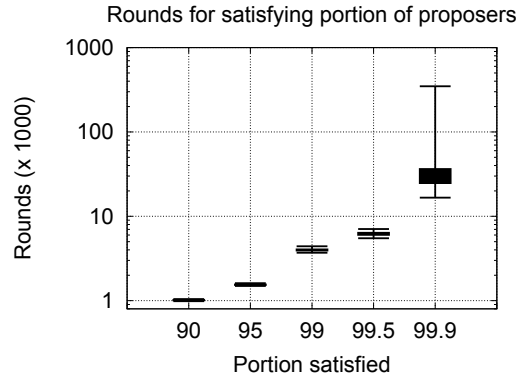
Η αυξημένη απόδοση συνήθως συνοδεύεται από μειωμένη ποιότητα. Στη δική μας περίπτωση, η περιορισμένη πληροφορία που έχει στη διάθεσή του ο αλγόριθμος είναι αναμενόμενο να οδηγεί σε ταιριάσματα χαμηλότερης ποιότητας. Ευτυχώς, το φαινόμενο αυτό φαίνεται να είναι αρκετά ήπιο, όπως φαίνεται στα γραφήματα που ακολουθούν. Η δυσαρέσκεια και το από κοινού κόστος μένουν πρακτικά ανεπηρέαστα. Το έμφυλο κόστος επηρεάζεται αρνητικά (αυξάνεται) ειδικά όταν το N είναι μικρό και δεν χρησιμοποιείται ομοιόμορφη πόλωση, ωστόσο η διαφορά είναι αρκετά μικρή. Τα παραπάνω αφορούν μέσες τιμές στο σύνολο 500 εκτελέσεων. Παρατηρήσαμε ότι η τυπική απόκλιση του έμφυλου κόστους από τη μέση τιμή αυξάνεται όσο οι λίστες περιορίζονται, όμως το φαινόμενο είναι εντονότερο για μικρό παρά για μεγάλο N . Όπως και στην προηγούμενη ενότητα, σημειώνουμε ότι η προσέγγισή μας φαίνεται να είναι καταλληλότερη για μεγάλα στιγμιότυπα ESMP.



Σχήμα 5.2: Ποιότητα ταιριάσματος (κόστη) για διάφορα κλάσματα λιστών προτίμησής

5.3 Αποτυγχάνοντας ομαλά

Όπως φαίνεται στα γραφήματα της πρώτης ενότητας, καμία εκτέλεση δεν ολοκληρώνεται επιτυχώς όταν οι λίστες έχουν περιοριστεί υπέρ το δέον. Αυτό μειώνει κάπως την χρηστικότητα της προσέγγισής μας, αφού δεν είναι πάντα εφικτό να προσδιοριστεί a priori ένα επαρκές K για κάθε



Σχήμα 5.3: Ικανοποιημένοι πομποί VS ομαλά αποτυχημένες εκτελέσεις

στιγμιότυπο (αν και μπορούν να γίνουν συντηρητικές προσεγγίσεις). Διορθώνουμε αυτό το μειονέκτημα δείχνοντας ότι ο ESMA αποτυγχάνει ομαλά (fails gracefully). Στις περιπτώσεις που είναι ανέφικτος ένας τερματισμός με πλήρες ταίριασμα, συνήθως επιτυγχάνει να αντιστοιχίσει έναν ικανοποιητικό σύντροφο στο συντριπτικό ποσοστό των προσώπων και των δύο συνόλων. Αυτό συμβαίνει ακόμα και κάτω από ακραίους περιορισμούς: εκτελέσαμε ESMA σε 500 στιγμιότυπα με $N = 20000$ και $K = 1000$ (δραστικά περιορισμένες λίστες) και καταγράψαμε πόσοι γύροι παρήλθαν έως ότου ένα επαρκώς μεγάλο ποσοστό r από πομπούς ήταν ικανοποιημένο με τους συντρόφους τους για $r \in \{90\%, 95\%, 99\%, 99.5\%, 99.9\%, 99.95\%, 99.99\%\}$. Τα αποτελέσματά μας ακολουθούν.

Percentile matched	90	95	99	99.5	99.9	99.95	99.99
Gracefully failed runs	500	500	500	500	420	2	0

Πίνακας 5.1: Εκτελέσεις που διασφαλίζουν ικανοποιημένους πομπούς σε ανώμαλο τερματισμό

Όπως φαίνεται, αν και όλες οι εκτελέσεις είναι αποτυχημένες, στη συντριπτική τους πλειοψηφία προσεγγίζουν ένα πλήρες ταίριασμα σε ένα εντυπωσιακά μικρό αριθμό γύρων: 99.9% των πομπών αντιστοιχίζονται με σύντροφο της αρεσκείας τους εντός 30000 γύρων, περίπου 1% του ανωτάτου επιτρεπτού ορίου για $N = 20000$. Δεδομένου ότι οι εκτελέσεις έγιναν με πολύ μικρές λίστες προτίμησης είναι λογικό να υποθέσουμε ότι παρόμοιο ποσοστό από δέκτες ήταν αντίστοιχα ικανοποιημένοι κατά τον (ανώμαλο) τερματισμό. Αναφορικά με το πρόβλημα ανάθεσης, το παρόν εύρημα δηλώνει ότι μετά από μικρό αριθμό γύρων μπορούμε να έχουμε ένα μερικό πλάνο που αναθέτει VMs σε κατάλληλους hosts και αφήνει ικανοποιημένους τους περισσότερους πελάτες. Οι λίγοι πόροι και πελάτες που δεν ταίριαξαν σωστά μπορούν να αντιμετωπιστούν ad hoc.

Τα παραπάνω φωτίζουν και το λόγο που επελέγη το σχετικά αυθαίρετο όριο των $N\sqrt{N}$ γύρων. Για μεγάλα στιγμιότυπα, οι N^2 γύροι ταιριάσματος που απαιτεί ο SMA στη χειρότερη περίπτωση είναι απαγορευτικά πολλοί. Από την άλλη, φαίνεται ότι $\mathcal{O}(N)$ γύροι επαρκούν ώστε ο ESMA να καταλήξει σε ένα σχεδόν πλήρες ταίριασμα. Επιλέξαμε να τρέξουμε τα πειράματά μας γύρω από μια ενδιάμεση τιμή, ώστε τα αποτελέσματα να μπορούν να είναι αξιόπιστη βάση για εξαγωγή συμπερασμάτων και αφετέρου να ολοκληρώνονται σε λελογισμένο χρόνο. Εξυπακούεται ότι ανάλογα με το φόρτο εργασίας και ισορροπώντας τις απαιτήσεις για ακρίβεια και ταχύτητα, ένας πάροχος υπολογιστικής υποδομής μπορεί να θέσει το δικό του όριο στον αλγόριθμό μας.

Κεφάλαιο 6

Υλοποίηση

Η υλοποίηση του κυρίου αλγορίθμου και των βοηθητικών προγραμμάτων έγινε με γνώμονα την ικανοποίηση των εξής απαιτήσεων: Πρώτον, να ελαχιστοποιείται η κατανάλωση μνήμης, δεύτερον να επιτυγχάνεται καλή εκμετάλλευση του υλικού και τρίτον να είναι εύκολη η αναπαραγωγή των αποτελεσμάτων. Προς τούτο, αξιοποιήθηκε η εμπειρία που αποκτήθηκε στο [12], μαζί με τις απαραίτητες αλλαγές και προσθήκες, οι οποίες ακολουθούν.

6.1 Γλώσσα & Περιβάλλοντα

Η αρχική υλοποίηση Java μεταγράφηκε σε C++11, διότι η τελευταία επιτρέπει στον προγραμματιστή ακριβέστερο έλεγχο της μνήμης. Αναλυτικότερα:

- Τα αντικείμενα κατασκευάζονται και καταστρέφονται σε γνωστά σημεία. Απαλασσόμαστε από το κόστος του αυτομάτου συλλέκτη σκουπιδιών, ο οποίος θα ήταν ούτως η άλλως περιττός κατά το μεγαλύτερο μέρος της εκτέλεσης.
- Μπορεί να προσδιοριστεί επακριβώς η διάταξη των πεδίων μιας κλάσης ώστε να αμβλύνεται το φαινόμενο του false sharing όταν το ίδιο αντικείμενο προσπελάζεται από πολλαπλά νήματα.
- Μπορούμε να πετύχουμε καλύτερη αξιοποίηση της κρυφής μνήμης μειώνοντας το χώρο που δεσμεύει ένα αντικείμενο, ορίζοντας προσεκτικά τους τύπους των πεδίων ώστε να εξυπηρετούν το απαιτούμενο εύρος χωρίς σπατάλες (βλ. τους διάφορους τύπους ακεραίων στην κεφαλίδα `cstdint`).

Τα παραπάνω είναι κρίσιμα για την απόδοση του αλγορίθμου. Μερικά ακόμη σημαντικά πλεονεκτήματα της C++11 είναι η εκτεταμένη και φορητή βιβλιοθήκη παραγωγής τυχαίων αριθμών, καθώς και το γεγονός ότι συνδυάζεται χωρίς κόπο με το περιβάλλον πολυνηματικής επεξεργασίας OPENMP. Πράγματι, μεγάλο μέρος του κώδικα ξαναγράφηκε με στόχο να είναι παραλληλοποιήσιμο, ώστε η παραγωγή, εισαγωγή και επίλυση ενός στιγμιότυπου να αξιοποιεί τις δυνατότητες ενός πολυπύρηνου υπολογιστή.

6.2 Επιλογή πομπών

Χρησιμοποιούμε την ψευδοτυχαία επιλογή πομπών του [12] με μια επιπλέον συνθήκη: εάν μετά το πέρας ενός γύρου όλα τα πρόσωπα ενός συνόλου είναι ικανοποιημένα με τους συντρόφους τους,

τότε στον αμέσως ακόλουθο γύρο χρήζονται πομποί τα πρόσωπα του άλλου συνόλου. Αυτό διασφαλίζει ότι σε κάθε γύρο υπάρχει τουλάχιστον ένας ανικανοποίητος πομπός —ο οποίος κάνει μια πρόταση— εκτός, ενδεχομένως, από τον τελευταίο γύρο. Έτσι, αποφεύγουμε την ύπαρξη αδρανών γύρων, φαινόμενο αρκετά πιθανό σε μεγάλα στιγμιότυπα, όπου πολύ λίγα πρόσωπα παραμένουν ανικανοποίητα κατά τους τελευταίους γύρους. Παρά τα οφέλη αυτής της ευριστικής, παραμένει ασαφές κατά πόσον αυτή η βελτιστοποίηση ενδέχεται να οδηγήσει την επιλογή πομπών σε επαναλαμβανόμενο κύκλο, δυσχεραίνοντας την ολοκλήρωση του ταιριάσματος.

6.3 Ντετερμινιστική παραγωγή (υπο)στιγμιότυπων

Ένα στιγμιότυπο ESMP παράγεται ντετερμινιστικά με βάση την πεντάδα (N, n, K, b, s, h) :

N είναι το πλήθος των προσώπων σε κάθε σύνολο

n είναι το μέγεθος του reservoir

K είναι το μέγεθος της λίστας προτιμήσεων

b είναι μια συνάρτηση που αντιστοιχεί πρόσωπα σε θετικά βάρη (αριθμοί κινητής υποδιαστολής)

s είναι μια αρχική ρίζα για το γεννήτορα ψευδοτυχαίων αριθμών (ακέραιος 64 bit)

h είναι μια συνάρτηση συμπίεσης, που παίρνει ως είσοδο δύο ακεραίους 32 bit και δίνει ως αποτέλεσμα έναν ακέραιο επίσης 32 bit (βλ. και [16])

Ως γεννήτορα ψευδοτυχαίων αριθμών χρησιμοποιούμε ένα `std::uniform_int_distribution` με κατάλληλα όρια, το οποίο τροφοδοτείται από μια γεννήτρια ψευδοτυχαιότητας τύπου Mersenne Twister (`std::mt19937_64`, εφεξής MT64). Αμφότερα περιέχονται στην στάνταρ βιβλιοθήκη της C++11, διασφαλίζοντας αφενός τη φορητότητα της υλοποίησης αλλά και καλής ποιότητας ψευδοτυχαία έξοδο.

Η καθαυτό παραγωγή μιας λίστας προτίμησης γίνεται ως εξής: Πρώτα εισάγουμε στην h τη ρίζα s μαζί με το μοναδικό αναγνωριστικό ενός προσώπου και παράγουμε μια μοναδική ρίζα s' . Με αυτή εκκινούμε μια γεννήτρια MT64. Δίνουμε τη γεννήτρια και τα (N, n, b) στον αλγόριθμο WRS και αυτός επιστρέφει ένα δείγμα n μνηστήρων για το δεδομένο πρόσωπο, καθέναν μαζί με το τυχαίο βάρος με το οποίο εισήχθη στο δείγμα. Επιλέγουμε τους K βαρύτερους μνηστήρες και τους τοποθετούμε τα μοναδικά αναγνωριστικά τους στη λίστα προτίμησης του προσώπου, κατά φθίνουσα σειρά βάρους. Κάθε τέτοια λίστα μπορεί να παραχθεί ανεξάρτητα (παράλληλα) για κάθε πρόσωπο.

Έστω δύο στιγμιότυπα που έχουν παραχθεί από την ίδια τετράδα (N, n, b, s, h) αλλά διαφέρουν στο πλήθος των επιτρεπτών προτιμήσεων: $K_1 < K_2$. Η διαδικασία δειγματοληψίας είναι ανεξάρτητη από το K , άρα θα παράξει το ίδιο δείγμα n μνηστήρων για κάποιο δεδομένο πρόσωπο. Λόγω του τρόπου που τοποθετούνται οι μνηστήρες στη λίστα προτίμησης, η λίστα του προσώπου στο στιγμιότυπο όπου $K = K_1$ θα συμπίπτει με τα K_1 πρώτα στοιχεία της λίστας του αντίστοιχου προσώπου στο στιγμιότυπο όπου $K = K_2$. Με τον τρόπο αυτό, μπορούμε να παράγουμε στιγμιότυπα που είναι γνήσια υποσύνολα άλλων, ούτως ώστε να μελετήσουμε με ακρίβεια τα αποτελέσματα που έχει ο περιορισμός της διαθέσιμης πληροφορίας στο επιθυμητό ποσοστό.

6.4 Συμπαγής αναπαράσταση των συναρτήσεων προτίμησης

Για κάθε πρόσωπο a θεωρούμε ότι γνωρίζουμε μια συνάρτηση f_a η οποία αντιστοιχεί βαθμό προτίμησης (σκορ) στους μνηστήρες του a . Κάθε σκορ είναι ένας ακέραιος στο διάστημα $[0, K]$ και κάθε μνηστήρας επίσης αναπαρίσταται με έναν ακέραιο στο διάστημα $[0, N]$. Επομένως, η συνάρτηση f_a μπορεί να υλοποιηθεί με έναν πίνακα K θέσεων, χρησιμοποιώντας K ακεραίους με μήκος τουλάχιστον $\log_2 N$ bit. Η συνάρτηση f_a χρησιμοποιείται κάθε φορά που ο a λειτουργεί ως πομπός.

Η κατάσταση είναι πιο περίπλοκη όταν ο a λειτουργεί ως δέκτης, διότι ενδέχεται ο a να δεχθεί πρόταση από μνηστήρα που δεν βρίσκεται στις πρώτες K προτιμήσεις του. Ένας τέτοιος μνηστήρας ονομάζεται a -άκυρος (διαφορετικά, a -έγκυρος). Μόνο ένας a -έγκυρος μνηστήρας αντιστοιχίζεται σε κάποιο (μοναδικό) ακέραιο στο διάστημα $[0, K]$. Θεωρούμε, συνεπώς, ότι για τον a γνωρίζουμε επίσης μια συνάρτηση g_a η οποία αντιστοιχεί μνηστήρες σε σκορ ή στην ειδική τιμή \perp . Τα αναγνωριστικά των a -έγκυρων μνηστήρων αποτελούν αραιό —συνήθως αδόμητο— υποσύνολο του $[0, N]$, επομένως η συνάρτηση g_a δεν είναι αποδοτικό να υλοποιηθεί με πίνακα, αφού αυτό θα είχε χωρική πολυπλοκότητα $\mathcal{O}(N \log_2 N)$.

Ένας εναλλακτικός τρόπος αναπαράστασης είναι μέσω *χάρτη* (*map, dictionary*): Αποθηκεύουμε τα ζεύγη (s, k) που αποτελούν τη λίστα προτιμήσεων του a , όπου s είναι το αναγνωριστικό ενός μνηστήρα και k το σκορ του. Όταν ο a δέχεται πρόταση από μνηστήρα b , εξετάζει αν ο b συμπίπτει με το πρώτο μέλος κάποιου ζεύγους. Αν ναι, ανακτά το k και κρίνει την πρόταση ανάλογα με την τρέχουσα κατάσταση. Αν όχι (δηλ. αν $g_a(b) = \perp$), τότε ο a απορρίπτει το αίτημα χωρίς περαιτέρω επεξεργασία. Ένας χάρτης συνήθως υλοποιείται ως *πίνακας κατακερματισμού*. Μια τέτοια λύση έχει πλεονέκτημα ότι χρησιμοποιεί μόλις $\mathcal{O}(K \log_2 N)$ χώρο αλλά υπολογίζει την $g_a(b)$ σε χρόνο $\mathcal{O}(1)$. Δοκιμάσαμε τους ενσωματωμένους πίνακες κατακερματισμού της C++11 (`std::unordered_map`) και παρατηρήσαμε ότι αν και εξαιρετικά ταχείς ήταν αρκετά απαιτητικοί σε μνήμη (μεγάλη σταθερά), καταρρίπτοντας το σκοπό τους.

Στο σημείο αυτό παρατηρούμε ότι το σύνολο των ζευγών (s, k) αποτελεί ένα *στατικό χάρτη*, αφού δεν υπάρχουν προσθαφαιρέσεις κατά τη διάρκεια της εκτέλεσης. Υπάρχει εκτεταμένη βιβλιογραφία σχετικά με την υλοποίηση χωρικά και χρονικά αποδοτικών στατικών χαρτών. Ωστόσο, επειδή η αποθηκευόμενη πληροφορία ανά πρόσωπο είναι σχετικά μικρή (το πολύ 10000 ακέραιοι), οι προαναφερθείσες υλοποιήσεις κρίνονται δυσανάλογα περίπλοκες σε σχέση με το όφελος που προσφέρουν. Δοκιμάσαμε μόνο μία υλοποίηση στατικού χάρτη, η οποία είναι μια παραλλαγή του `boost::flat_map`. Το σκεπτικό της υλοποίησης περιγράφεται στο [1]: Ταξινομούμε τα ζεύγη ως προς τα κλειδιά και τα τοποθετούμε σε έναν πίνακα. Για να ανακτήσουμε την τιμή που αντιστοιχεί σε ένα κλειδί s , εκτελούμε δυαδική αναζήτηση στον πίνακα. Αν το κλειδί βρεθεί, επιτρέφεται η τιμή που το συνοδεύει αλλιώς \perp . Για κλειδιά και τιμές ακεραίους, η υλοποίηση αυτή χρειάζεται χώρο $2K \log_2 N$ bit και χρόνο $\mathcal{O}(\log_2 K)$ ανά αναζήτηση. Όταν το K είναι σχετικά μικρό, η πτώση της ταχύτητας είναι αρκετά μικρή, ωστόσο το κέρδος σε χώρο σημαντικό.

Στα πειράματα επιλέξαμε να χρησιμοποιήσουμε τη 2η επιλογή, την οποία επαυξήσαμε με μια δομή *σύνοψης*: Αφού ταξινομήσουμε τα K ζεύγη, επιλέγουμε ένα δείγμα \sqrt{K} κλειδιών που βρίσκονται σε ίσες περίπου αποστάσεις και τα τοποθετούμε σε έναν βοηθητικό πίνακα με τη σειρά. Κάθε φορά που πρέπει να αναζητήσουμε ένα κλειδί, χρησιμοποιούμε τον βοηθητικό πίνακα για να προσδιορίσουμε το υποδιάστημα όπου ενδέχεται να βρίσκεται το κλειδί στον πλήρη πίνακα.

Επειδή ο βοηθητικός πίνακας έχει πολύ μικρό μέγεθος, χωράει στην κρυφή μνήμη κι έτσι επιτρέπει να γίνουν περίπου τα μισά βήματα της δυαδικής αναζήτησης χωρίς πρόσβαση στην κύρια μνήμη, βελτιώνοντας την απόδοση.

Κεφάλαιο 7

Συμπεράσματα και μελλοντικές προεκτάσεις

Σε αυτή την εργασία μελετήσαμε το πρόβλημα της ανάθεσης εικονικών μηχανών σε hosts, το οποίο ανακύπτει σε όλα τα σύγχρονα datacenter. Μοντελοποιήσαμε το πρόβλημα ως στιγμιότυπο του Προβλήματος Σταθερού Γάμου και προτείναμε μια μεθοδολογία ισότιμης μεταχείρισης των πολιτικών των πελατών και του παρόχου. Χρησιμοποιήσαμε μια ευριστική ώστε να βρίσκουμε γρήγορα σταθερά και δίκαια ταιριάσματα μεταξύ VMs και VM slots στα μηχανήματα της υποδομής, με τρόπο που να σέβεται τις προτιμήσεις των δύο μερών. Προκειμένου να κάνουμε τη λύση μας κλιμακώσιμη για ταυτόχρονη ανάθεση πολλών δεκάδων VMs, αναπτύξαμε μια βελτιστοποίηση της αρχικής ευριστικής ώστε να λειτουργεί με ατελείς λίστες προτιμήσεων.

Μέσα από μια εκτεταμένη πειραματική αξιολόγηση της μεθόδου μας, δείξαμε ότι η προσέγγισή μας είναι ικανή να βρει δίκαια ταιριάσματα για στιγμιότυπα έως $N = 20000$ VMs και VM slots, μια τάξη μεγέθους πάνω από παρόμοιες προσπάθειες. Η λύση μας χρησιμοποιεί ένα κλάσμα του χώρου που απαιτείται συνήθως και τερματίζει εντός $N\sqrt{N}$ γύρων για τη μεγάλη πλειοψηφία των στιγμιότυπων, ενώ η ποιότητα των λύσεων παραμένει σταθερά υψηλή, τόσο ως προς την αμεροληψία όσο και ως προς τη γενικότερη ικανοποίηση που επιτυγχάνει. Δείξαμε, επίσης, ότι όσο μεγαλύτερα είναι τα στιγμιότυπα τόσο μικρότερο είναι το ενεργό ποσοστό των λιστών που χρειαζόμαστε για επιτυχή τερματισμό, συνεπώς η προσέγγισή μας είναι κατάλληλη κυρίως για μεγάλα datacenter.

Τέλος, παρατηρήσαμε ότι στις περιπτώσεις που η λύση μας αποτυγχάνει να υπολογίσει ένα σωστό ταιρίασμα, εντούτοις παράγει ένα μερικό ταιρίασμα που είναι αρκετά κοντά στο επιθυμητό, δηλαδή έχει αντιστοιχίσει τη συντριπτική πλειοψηφία των πόρων σε VMs. Εφόσον η ανάθεση εικονικών μηχανών δεν είναι πρόβλημα που απαιτεί τέλειες λύσεις, κρίνουμε ότι η προσέγγισή μας είναι επαρκής σε όλες τις περιπτώσεις.

Φαίνεται πως η παρούσα προσέγγιση είναι αρκετά γόνιμη για το συγκεκριμένο πρόβλημα και θα θέλαμε να δούμε τις εξής προεκτάσεις στο μέλλον:

- Εφαρμογή σε ακόμη μεγαλύτερα ή δυσκολότερα στιγμιότυπα, ώστε να αποτιμηθεί ποσοτικά η σχέση που συνδέει το επαρκές / με το N και το είδος της πόλωσης.
- Συλλογή ή προσομοίωση ρεαλιστικών workload, ανάθεσή τους μέσω ESMA και εξαγωγή στατιστικών αποτελεσμάτων αναφορικά με την ικανοποίηση των χρηστών και του παρόχου.
- Περαιτέρω έρευνα και θεωρητική θεμελίωση των ευριστικών που οδηγούν “σχεδόν πάντα” σε ισότιμο ταιρίασμα εντός $o(N^2)$ γύρων.

Βιβλιογραφία

- [1] M. Austern. Why you shouldn't use set (and what you should use instead). <http://lafstern.org/matt/col1.pdf>.
- [2] A. Beloglazov and R. Buyya. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing*, pages 826–831. IEEE Computer Society, 2010.
- [3] N. M. Calcavecchia, O. Biran, E. Hadad, and Y. Moatti. Vm placement strategies for cloud scenarios. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 852–859. IEEE, 2012.
- [4] S. Chaisiri, B.-S. Lee, and D. Niyato. Optimal virtual machine placement across multiple cloud providers. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, pages 103–110. IEEE, 2009.
- [5] Y. Cui. Solving the stable marriage problem with erlang. <https://www.airpair.com/erlang/posts/solving-the-stable-marriage-problem-with-erlang>.
- [6] C. Delimitrou and C. Kozyrakis. Quasar: resource-efficient and QoS-aware cluster management. In *ACM SIGPLAN Notices*, volume 49, pages 127–144. ACM, 2014.
- [7] P. S. Efraimidis and P. G. Spirakis. Weighted random sampling with a reservoir. *Information Processing Letters*, 97(5):181–185, 2006.
- [8] P. Everaere, M. Morge, and G. Picard. Minimal concession strategy for reaching fair, optimal and stable marriages. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1319–1320. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [9] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- [10] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Local search algorithms on the stable marriage problem: Experimental studies. In *ECAI*, volume 10, pages 1085–1086, 2010.
- [11] M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Procedural fairness in stable marriage problems. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1209–1210. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

- [12] I. Giannakopoulos, P. Karras, D. Tsumakos, K. Doka, and N. Koziris. An Equitable Solution to the Stable Marriage Problem. In *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pages 989–996. IEEE, 2015.
- [13] D. Gusfield. Three fast algorithms for four problems in stable marriage. *SIAM Journal on Computing*, 16(1):111–128, 1987.
- [14] V. Ishakian, R. Sweha, A. Bestavros, and J. Appavoo. Dynamic pricing for efficient workload collocation. 2011.
- [15] K. Iwama, S. Miyazaki, and H. Yanagisawa. Approximation algorithms for the sex-equal stable marriage problem. *ACM Trans. Algorithms*, 7(1):2:1–2:17, Dec. 2010.
- [16] B. Jenkins. 4-byte integer hashing. <http://burtleburtle.net/bob/hash/integer.html>.
- [17] A. Kato. Complexity of the sex-equal stable marriage problem. *Japan Journal of Industrial and Applied Mathematics*, 10(1):1–19, 1993.
- [18] A. Kella and G. Belalem. VM Live Migration Algorithm Based on Stable Matching Model to Improve Energy Consumption and Quality of Service. In *CLOSER*, pages 118–128, 2014.
- [19] E. Shanks. Understanding vmware slot sizes. <http://theithollow.com/2013/02/05/slotsize/>.
- [20] H. N. Van, F. D. Tran, and J.-M. Menaud. Sla-aware virtual resource management for cloud infrastructures. In *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on*, volume 1, pages 357–362. IEEE, 2009.
- [21] H. Xu and B. Li. Egalitarian stable matching for VM migration in cloud computing. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 631–636. IEEE, 2011.
- [22] H. Xu and B. Li. Anchor: A versatile and efficient framework for resource management in the cloud. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1066–1076, 2013.
- [23] X. Xu and H. Yu. A game theory approach to fair and efficient resource allocation in cloud computing. *Mathematical Problems in Engineering*, 2014.