



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  
**ΠΛΗΡΟΦΟΡΙΚΗΣ & ΔΙΚΤΥΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΤΕΧΝΙΚΩΝ ΕΞΙΣΟΡΡΟΠΗΣΗΣ**  
**ΦΟΡΤΙΟΥ ΣΕ ΥΠΟΛΟΓΙΣΤΙΚΑ ΝΕΦΗ**

Οικονόμου Αναστασία

A.M : 125

Επιβλέπουσα : Μαργαρίτη Σπυριδούλα

Άρτα, 2023

**COMPARATIVE STUDY OF LOAD BALANCING  
TECHNIQUES IN CLOUD COMPUTING**

**Εγκρίθηκε από τριμελή εξεταστική επιτροπή**

Άρτα, Ιούνιος 2023

## **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Επιβλέπουσα καθηγήτρια

Σπυριδούλα Μαργαρίτη,

ΕΔΙΠ, Α

2. Μέλος επιτροπής

Ελευθέριος Στεργίου,

ΔΕΠ, Αναπληρωτής καθηγητής

3. Μέλος επιτροπής

Λιάγκου Βασιλική,

ΔΕΠ, Επίκουρη καθηγήτρια

© Οικονόμου Αναστασία, 2023.

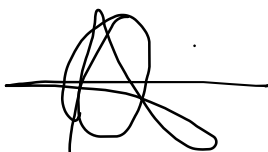
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## **Δήλωση μη λογοκλοπής**

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Οικονόμου Αναστασία

Υπογραφή

A handwritten signature in black ink, consisting of a series of loops and a horizontal line extending to the left.

## Ευχαριστίες

Την διπλωματική αυτή εργασία, εκπόνησα για το τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων. Με αυτόν τον τρόπο, μου δόθηκε η ευκαιρία να εμπλουτίσω τις γνώσεις μου και να διευρύνω τους πνευματικούς μου ορίζοντες, πάνω σε ένα αντικείμενο που είναι πραγματικά πολύ ενδιαφέρον. Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτριά μου κ. Μαργαρίτη Σπυριδούλα για την καθοδήγηση και τη βοήθεια που μου προσέφερε καθ' όλη τη διάρκεια της εκπόνησής της.

## Περίληψη

Το Cloud Computing είναι ένα ισχυρό μοντέλο που επιτρέπει στους χρήστες και τους οργανισμούς να αγοράζουν τις απαιτούμενες υπηρεσίες ανάλογα με τις ανάγκες τους. Προσφέρει πολλές υπηρεσίες, όπως αποθήκευση, πλατφόρμες για ανάπτυξη, εύκολη πρόσβαση σε υπηρεσίες Ιστού κ.λπ. Η Εξισορρόπηση Φορτίου είναι ένα κοινό ζήτημα στο Cloud που δυσκολεύει τη διατήρηση της απόδοσης των εφαρμογών. Οι πάροχοι Cloud αγωνίζονται να κατανείμουν ίσο φόρτο εργασίας μεταξύ των διακομιστών. Μια αποτελεσματική τεχνική Εξισορρόπησης Φορτίου θα πρέπει να βελτιστοποιεί και να εξασφαλίζει υψηλή ικανοποίηση των χρηστών χρησιμοποιώντας αποτελεσματικά τους πόρους των VM. Αυτή η εργασία παρουσιάζει μια ολοκληρωμένη ανασκόπηση διαφόρων τεχνικών Εξισορρόπησης Φορτίου όπως είναι η στατική, η δυναμική και η υβριδική. Πραγματοποιείται σύγκριση των αλγορίθμων και αναφέρονται τα χαρακτηριστικά, τα πλεονεκτήματα, καθώς και τα μειονεκτήματα αυτών. Τέλος, με τη χρήση του προσομοιωτή Cloud Analyst αλλά και μέσω των γραφημάτων που προκύπτουν από τις μετρήσεις, γίνεται ο εντοπισμός του πιο αποδοτικού αλγορίθμου.

Λέξεις κλειδιά: Εξισορρόπηση Φορτίου, Cloud Computing, τεχνικές, στατική, δυναμική, υβριδική, Cloud Analyst, προσομοίωση

## **Abstract**

Cloud Computing is a powerful model that allows users and organizations to purchase the required services according to their needs. It offers many services such as storage, platforms for development, easy access to web services, etc. Load balancing is a common issue in the Cloud that makes it difficult to maintain application performance. Cloud providers struggle to distribute workload equally across servers. An effective Load Balancing technique should optimize and ensure high user satisfaction by using VM resources efficiently. This paper presents a comprehensive review of various load balancing techniques such as static, dynamic and hybrid. Algorithms are compared and their characteristics, advantages, and disadvantages are mentioned. Finally, by using the Cloud Analyst simulator and through the graphs resulting from the measurements, the most efficient algorithm is identified.

Keywords: Load Balancing, Cloud Computing, techniques, static, dynamic, hybrid, Cloud Analyst, simulation



# Πίνακας Περιεχομένων

<b>Πίνακας Περιεχομένων</b> .....	9
Κεφάλαιο 1 <sup>ο</sup> : .....	11
1 Εισαγωγή .....	11
1.1 Πρόλογος.....	11
1.1.1 Cloud Computing .....	11
1.1.2 Εξισορρόπηση Φορτίου στο Cloud .....	12
1.2 Αντικείμενο διπλωματικής .....	14
1.2.1 Ορισμός του προβλήματος .....	14
1.2.2 Στόχοι εργασίας - Συνεισφορά .....	16
1.3 Δομή εργασίας.....	16
Κεφάλαιο 2 <sup>ο</sup> : .....	18
2 Θεωρητικό Υπόβαθρο .....	18
2.1 Cloud Computing .....	18
2.1.1 Ορισμός .....	18
2.1.2 Μοντέλο υπηρεσιών του Cloud Computing .....	18
2.1.3 Μοντέλα ανάπτυξης του Cloud Computing .....	20
2.1.4 Αρχιτεκτονική Cloud Computing .....	21
2.1.5 Εξισορρόπηση Φορτίου στο Cloud Computing .....	23
2.2 Εξισορρόπηση φορτίου .....	24
2.2.1 Αρχιτεκτονική Εξισορρόπησης Φορτίου.....	24
2.2.2 Μετρικές Εξισορρόπησης Φορτίου .....	26
2.2.3 Τύποι Εξισορροπητών βάσει διαμορφώσεων.....	28
2.2.4 Πλεονεκτήματα - Μειονεκτήματα Εξισορρόπησης Φορτίου .....	31
2.2.5 Προκλήσεις.....	32
Κεφάλαιο 3 <sup>ο</sup> : .....	34
3 Τεχνικές και Αλγόριθμοι Εξισορρόπησης.....	34
3.1 Εισαγωγή .....	34
3.2 Μοντέλο Εξισορρόπησης Φορτίου.....	35
3.3 Τεχνικές Εξισορρόπησης Φορτίου .....	36
3.3.1 Τεχνικές Στατικής Εξισορρόπησης Φορτίου .....	36
3.3.2 Τεχνικές Δυναμικής Εξισορρόπησης Φορτίου .....	37
3.3.3 Υβριδικές Τεχνικές Εξισορρόπησης Φορτίου .....	38
3.4 Αλγόριθμοι Εξισορρόπησης Φορτίου .....	39

3.4.1	Στατικοί Αλγόριθμοι.....	39
3.4.2	Δυναμικοί Αλγόριθμοι.....	41
3.4.3	Άλλοι Υφιστάμενοι Αλγόριθμοι.....	47
3.5	Πολιτικές Δυναμικού Αλγόριθμου Εξισορρόπησης Φορτίου .....	51
3.6	Σύγκριση Αλγορίθμων.....	53
	Κεφάλαιο 4°.....	60
4	Αξιολόγηση Απόδοσης.....	60
4.1	Το Λογισμικό Προσομοίωσης Cloud Analyst.....	61
4.1.1	Cloud Analyst Simulator .....	61
4.1.2	Χαρακτηριστικά του Cloud Analyst.....	62
4.2	Το περιβάλλον προσομοίωσης .....	65
	Κεφάλαιο 5°.....	67
5	Σενάρια προσομοίωσης .....	67
5.1	Σενάριο 1- Προσομοίωση με 1 DC & 6 UB στην ίδια περιοχή .....	67
5.1.1	Διαμόρφωση σεναρίου .....	67
5.1.2	Αποτελέσματα .....	69
5.2	Σενάριο 2- Προσομοίωση με 1 DC & 5 UB σε διαφορετικές περιοχές .....	74
5.2.1	Διαμόρφωση σεναρίου .....	74
5.2.2	Αποτελέσματα .....	76
5.3	Σενάριο 3- Προσομοίωση με 2 DC & 5 UB σε διαφορετικές περιοχές .....	81
5.3.1	Διαμόρφωση σεναρίου .....	81
5.3.2	Αποτελέσματα .....	82
5.4	Σενάριο 4 - Περίπτωση Μελέτης με τη χρήση έξι Data Center .....	86
5.4.1	Διαμόρφωση σεναρίου .....	86
5.4.2	Αποτελέσματα .....	86
5.5	Αποτελέσματα και Συγκριτική Ανάλυση .....	93
	Κεφάλαιο 6°.....	95
6	Σύνοψη .....	95
	<b>Βιβλιογραφία.....</b>	<b>96</b>

## Κεφάλαιο 1<sup>ο</sup> :

### 1 Εισαγωγή

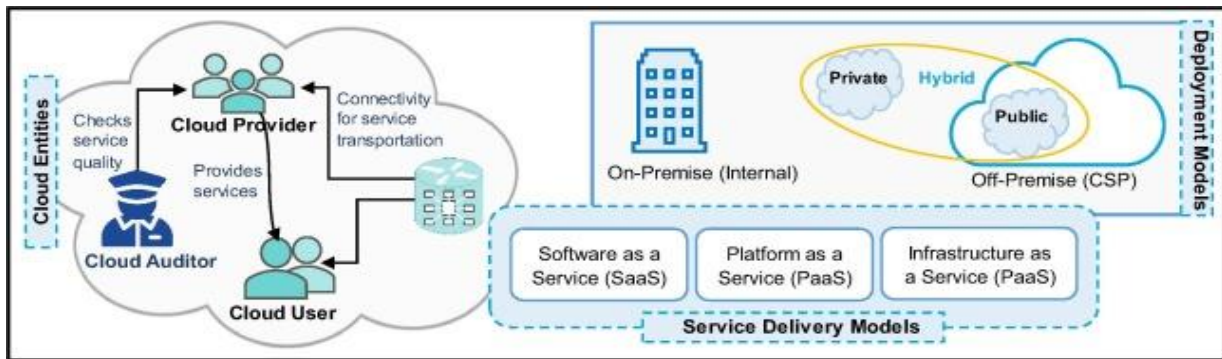
#### 1.1 Πρόλογος

##### 1.1.1 Cloud Computing

Με την ταχεία ανάπτυξη της τεχνολογίας πληροφορικής, το Cloud Computing έχει αναδειχθεί ως υποκατάστατο των παραδοσιακών τεχνολογιών για την παροχή υπηρεσιών σε πελάτες, ανά πάσα στιγμή και σε οποιαδήποτε τοποθεσία. Επιτρέπει στους χρήστες να έχουν πρόσβαση με υψηλές ταχύτητες σε μια ομάδα διαμορφώσιμων υπολογιστικών πόρων όπως είναι οι διακομιστές, ο αποθηκευτικός χώρος, το δίκτυο, καθώς και σε εφαρμογές. Αρκετές εταιρείες (π.χ. Amazon Web Services (AWS), Microsoft Azure, Google, IBM Cloud), που ονομάζονται επίσης πάροχοι Cloud Computing, προσφέρουν αυτές τις υπηρεσίες στους χρήστες.

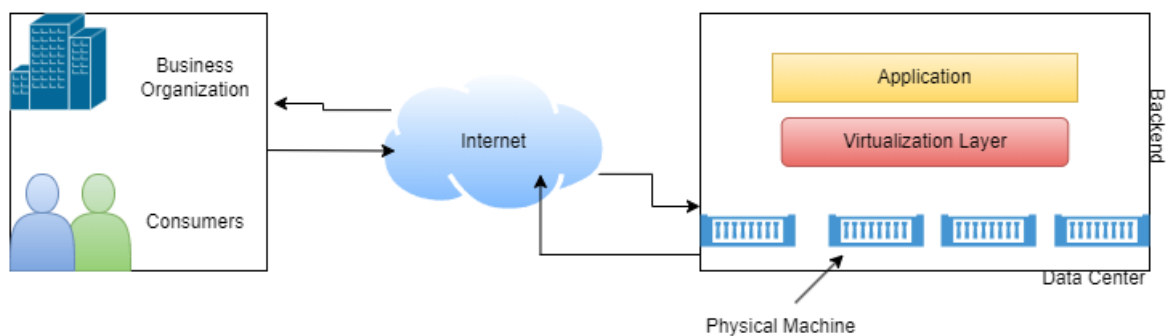
Το Cloud Computing αναδύεται ως ένα νέο μοντέλο ανάπτυξης πόρων (υλικού ή λογισμικού) αλλά και ως υπηρεσία προσβάσιμη μέσω δημόσιων ή ιδιωτικών δικτύων. Ωστόσο, λόγω της πολυπλοκότητάς του και της εμφάνισής του, δημιουργεί νέες προκλήσεις για τους παρόχους και τους τελικούς χρήστες. Επιπρόσθετα, είναι εννοιολογικά ένα καταναμημένο και ελαστικό σύστημα όπου οι πόροι θα διανέμονται μέσω του δικτύου. Οι πλήρεις πόροι του συστήματος πρέπει να συνεργάζονται για να ανταποκριθούν σε ένα αίτημα πελάτη που απαιτεί ενδοεπικοινωνία μεταξύ διαφόρων στοιχείων του συστήματος για να σχεδιαστεί ένα στοιχείο ή ένα υποσύνολο στοιχείων για την αντιμετώπιση του αιτήματος [1].

Η λειτουργία του Cloud Computing περιγράφεται συνοπτικά στην Εικόνα 1.1. Όλες οι οντότητες του Cloud συνεργάζονται μεταξύ τους. Για παράδειγμα, οι ελεγκτές Cloud λειτουργούν ως συντονιστές, διασφαλίζοντας ότι οι υπηρεσίες που προσφέρονται από τους παρόχους υπηρεσιών, είναι υψηλής ποιότητας και ακεραιότητας. Οι πάροχοι με τη σειρά τους, διασφαλίζουν ότι υπάρχει απρόσκοπτη σύνδεση για τη μεταφορά των υπηρεσιών σε πελάτες (χρήστες Cloud). Το κέντρο δεδομένων στο ιδιωτικό Cloud βρίσκεται μέσα στο δίκτυο του οργανισμού, ενώ αντίστοιχα το δημόσιο Cloud, εξαρτάται από τους Παρόχους Υπηρεσιών Cloud (Cloud Services Providers-CSP) και το υβριδικό μπορεί να βρίσκεται και στα δύο.



Εικόνα 1.1: Περίληψη λειτουργίας του Cloud Computing (πηγή: [2])

Σε ένα τυπικό περιβάλλον Cloud Computing, υπάρχουν δύο στοιχεία, το frontend side και το backend side. Το frontend βρίσκεται στην πλευρά του χρήστη όπου είναι προσβάσιμο μέσω συνδέσεων του διαδικτύου. Όσον αφορά την πλευρά του backend, ασχολείται με μοντέλα υπηρεσιών Cloud. Όπως φαίνεται στην Εικόνα 1.2, αποτελείται από ένα κέντρο δεδομένων στο οποίο αποθηκεύονται πολλαπλές φυσικές μηχανές



Εικόνα 1.2: Frontend & Backend side

(γνωστές ως διακομιστές). Τα επερχόμενα αιτήματα χρηστών λαμβάνονται από την εφαρμογή δυναμικά προγραμματισμένα και οι πόροι κατανέμονται στους πελάτες μέσω του επίπεδου Virtualization. Η τεχνική εικονικοποίησης χρησιμοποιείται για τη διαχείριση δυναμικών πόρων στο Cloud καθώς και για την εξισορρόπηση του φορτίου σε ολόκληρο το σύστημα. Αυτό το επίπεδο, είναι επίσης υπεύθυνο για τον προγραμματισμό και την αποτελεσματική κατανομή των πόρων [2], [3].

### 1.1.2 Εξισορρόπηση Φορτίου στο Cloud

Το Cloud Computing παρέχει διάφορες υπηρεσίες και πόρους στους χρήστες σύμφωνα με τις απαιτήσεις τους. Στους πόρους περιλαμβάνονται η αποθήκευση, η επεξεργαστική ισχύς των βάσεων δεδομένων, των δικτύων, των αναλυτικών

στοιχείων, της τεχνητής νοημοσύνης και των εφαρμογών λογισμικού. Ένα παράδειγμα είναι τα VMs με μνήμη, επεξεργαστική ικανότητα, κλπ. Οι πόροι του Cloud, στους οποίους κατανέμονται οι εργασίες των χρηστών, μπορεί να έχουν διαφορετικές παραμέτρους και χαρακτηριστικά. Επιπλέον, το πλήθος των αιτημάτων των χρηστών δεν είναι ομοιόμορφα κατανεμημένο στο χρόνο, αλλά ούτε και ως προς το είδος των πόρων - υπηρεσιών που αιτείται από το cloud.

Αυτό μπορεί να οδηγήσει σε σημεία συμφόρησης στο δίκτυο και σε μη ισορροπημένη χρέωση σε ένα κατανεμημένο σύστημα, όπου ορισμένοι πόροι θα υπερφορτώνονται ενώ άλλοι θα φορτώνονται ελαφρά ή και καθόλου. Για να λύσουμε αυτό το πρόβλημα, χρησιμοποιούμε αλγόριθμους Εξισορρόπησης Φορτίου (ΕΦ) για κατανεμημένα συστήματα, όμως αρκετοί από αυτούς δεν είναι πλήρως προσαρμοσμένοι στο σύστημα του Cloud γεγονός το οποίο οδηγεί στην ανάπτυξη νέων αλγορίθμων ή προσαρμογή των ήδη υπάρχοντων για κατανεμημένα συστήματα.

Καταστάσεις όπως η υποφόρτωση και η υπερφόρτωση προκαλούν διαφορετικές αστοχίες του συστήματος σχετικά με την κατανάλωση ενέργειας, τον χρόνο εκτέλεσης, και πολλά ακόμη τα οποία θα εξετάσουμε στα επόμενα κεφάλαια. Επομένως, απαιτείται ΕΦ για να ξεπεραστούν όλα τα αναφερόμενα προβλήματα. Μια κατάλληλη ΕΦ βοηθά στην ελαχιστοποίηση της κατανάλωσης πόρων, τη μεγιστοποίηση της κλιμακωσιμότητας (scalability), και την αποφυγή σημείων συμφόρησης στους κόμβους.

Συνεπώς, η ΕΦ ή αλλιώς Load Balancing είναι η διαδικασία ανακατανομής του φόρτου εργασίας σε ένα κατανεμημένο σύστημα όπως το Cloud Computing, διασφαλίζοντας ότι καμία υπολογιστική μηχανή δεν είναι υπερφορτωμένη, υποφορτωμένη ή σε αδράνεια. Θα πρέπει δηλαδή να κατανέμει ομοιόμορφα τον φόρτο εργασίας μεταξύ όλων των κόμβων. Επιπρόσθετα, συμβάλει στη δίκαιη κατανομή των υπολογιστικών πόρων για την επίτευξη υψηλού επιπέδου ικανοποίησης των χρηστών και της σωστής χρήσης αυτών. Ακόμη, προσπαθεί να επιταχύνει διάφορες περιορισμένες παραμέτρους όπως ο χρόνος απόκρισης, ο χρόνος εκτέλεσης και η σταθερότητα του συστήματος, βελτιώνοντας με αυτό το τρόπο την απόδοση του Cloud [4], [2].

## 1.2 Αντικείμενο διπλωματικής

### 1.2.1 Ορισμός του προβλήματος

Τα τελευταία χρόνια, η χρήση του Cloud Computing έχει αυξηθεί ραγδαία, κυρίως για τα πλεονεκτήματα και την ευελιξία που παρέχει στους χρήστες τους. Αυτό όμως έχει ως αποτέλεσμα να καταφθάνει στις υποδομές του Cloud τεράστιος αριθμός αιτημάτων και ταυτόχρονα να απαιτείται η μεταφορά και η ανταλλαγή πολύ μεγάλου όγκου δεδομένων μεταξύ ενός ή περισσοτέρων συστημάτων. Η επιτυχία και η αποτελεσματικότητα ενός τέτοιου συστήματος εξαρτάται σε πολύ μεγάλο βαθμό από την κατανομή των πόρων [3]. Η κατανομή πόρων επηρεάζει επίσης και την δυνατότητα κλιμάκωσης του συστήματος, την κατανάλωση ενέργειας, τη διαθεσιμότητα των υπηρεσιών και την ικανοποίηση των πελατών.

Όταν υποβάλλονται πολλαπλά αιτήματα από διαφορετικούς χρήστες σε διαφορετικές τοποθεσίες και για διαφορετικούς πόρους, η ανάθεση εργασιών πρέπει να γίνει ορθά προκειμένου να αυξηθεί η ικανοποίηση των χρηστών. Οι εργασίες με υψηλή προτεραιότητα πρέπει να εκτελούνται πρώτες, και παράλληλα πρέπει να λαμβάνεται υπόψη ο χρόνος απόκρισης. Λόγω της υψηλής ζήτησης για υπηρεσίες, θα πρέπει να εξεταστεί η ίση κατανομή του φόρτου εργασίας μεταξύ των πελατών και των διακομιστών για να αυξηθεί η αποτελεσματικότητα και η διαθεσιμότητα των πόρων. Η σωστή χρήση των πόρων οδηγεί σε βελτιστοποίηση διακομιστών στα κέντρα δεδομένων, οπότε θα πρέπει να αποφεύγονται καταστάσεις υπερφόρτωσης.

Η ΕΦ στο Cloud είναι μια διαδικασία που μπορεί να αφορά το διαμοιρασμό του φορτίου:

- a) στα κέντρα δεδομένων
- b) στις φυσικές ή εικονικές συσκευές (VMs)
- c) στην ανάθεση εργασιών (tasks) στους πόρους του συστήματος (π.χ. σε πυρήνες CPU)

Για παράδειγμα, εάν ένα VM<sup>1</sup> είναι υπερφορτωμένο με πάρα πολλά αιτήματα χρήστη, η εργασία πρέπει να μετεγκατασταθεί σε ένα άλλο VM με μικρότερο φορτίο,

---

<sup>1</sup> Το Virtual Machine (VM) είναι ένας υπολογιστικός πόρος που χρησιμοποιεί λογισμικό αντί για φυσικό υπολογιστή για την εκτέλεση προγραμμάτων και την ανάπτυξη εφαρμογών. Μία ή περισσότερες εικονικές μηχανές "επισκέπτης" λειτουργούν σε μια φυσική μηχανή "κεντρικού υπολογιστή". Κάθε εικονική μηχανή εκτελεί το δικό της λειτουργικό σύστημα και λειτουργεί ξεχωριστά από τα άλλα VM, ακόμα και όταν όλα εκτελούνται στον ίδιο κεντρικό υπολογιστή.

λαμβάνοντας πάντα υπόψη την προτεραιότητα των εργασιών. Αυτό είναι γνωστό ως μετεγκατάσταση εργασιών. Έτσι, η βασικότητα των προσομοιωτών προκύπτει για τη δοκιμή του Cloud σύμφωνα με την απαίτηση που μπορεί να καταναλώσει λιγότερο χρόνο, κόστος και ενέργεια. Ακολουθούν μερικά τυπικά προβλήματα κατά τη διάρκεια της κατανομής εργασιών και της εξισορρόπησης φορτίου [5].

- **Μοντέλα ελέγχου:** Για να επιτευχθεί το βέλτιστο αποτέλεσμα για την κατανομή εργασιών και την ΕΦ, απαιτείται μια κεντρική μονάδα ελέγχου για τη συλλογή των πληροφοριών κατάστασης του συνόλου του συστήματος σε πραγματικό χρόνο. Ωστόσο, αυτό είναι δύσκολο να εφαρμοστεί επειδή τα κατανεμημένα συστήματα είναι πάντα μεγάλα, δυναμικά και χωρίς μονάδες ελέγχου. Αντίθετα, η πλήρως αποκεντρωμένη προσέγγιση μπορεί να απαιτεί σχετικά υψηλό υπολογιστικό κόστος από τους κόμβους, οι οποίοι ενδέχεται να βάζουν υπολογιστικά βαριά φορτία σε μεγάλα συστήματα με αποτέλεσμα να καθιστούν δύσκολο τον έλεγχο της διαδικασίας κατανομής των εργασιών.
- **Βελτιστοποίηση πόρων:** Πολλά υπάρχοντα μοντέλα κατανομής εργασιών και ΕΦ, έχουν υλοποιηθεί με βάση τη βελτιστοποίηση της προσβασιμότητας στους απαιτούμενους πόρους. Σε αυτά τα μοντέλα η μέτρηση της προσβασιμότητας σε πόρους και ο τρόπος βελτιστοποίησης ενός δείκτη απόδοσης, αποτελούν βασικά προβλήματα.
- **Αξιοπιστία:** Σε ανοικτά κατανεμημένα συστήματα, μερικοί κόμβοι μπορεί να είναι αναξιόπιστοι. Έτσι, ένα βασικό πρόβλημα είναι η εξασφάλιση πρόσβασης για τους κόμβους, σε αξιόπιστους πόρους καθώς και η δυνατότητα μετάβασης σε άλλους πόρους, για την περίπτωση σφάλματος σε έναν κόμβο. Επομένως, θα πρέπει να σχεδιαστεί μια προσέγγιση για την εύρεση ενός κυρίαρχα αξιόπιστου τρόπου καταμερισμού των εργασιών και τον πόρων.
- **Δικτυακές Δομές:** Η επικοινωνία και η αλληλεπίδραση κόμβων περιορίζεται από τις δικτυακές ιδιαιτερότητες. Η μέτρηση της επίδρασης των δομών (π.χ η σύνδεση δύο κόμβων) του δικτύου σχετικά με την εκτέλεση των καθηκόντων είναι ένα πρόβλημα.

### 1.2.2 Στόχοι εργασίας - Συνεισφορά

Στόχοι της παρούσας διπλωματικής εργασίας είναι να μελετηθεί το ζήτημα της Εξισορρόπησης Φορτίου σε ένα σύστημα όπως το Cloud Computing σύμφωνα με την πρόσφατη βιβλιογραφία και να:

- ✓ εξερευνηθούν οι παράγοντες που προκαλούν το πρόβλημα της ανισορροπίας φορτίου.
- ✓ παρουσιαστούν οι τεχνικές ΕΦ
- ✓ καταγραφούν τα ισχυρά και τα αδύναμα στοιχεία τους
- ✓ συγκριθούν ως προς την απόδοση, τον μέσο όρο του χρόνου απόκρισης, τον μέσο όρο του χρόνου επεξεργασίας του κέντρου δεδομένων, την υπεραπόδοση, αλλά και ως προς το κόστος.
- ✓ συγκριθούν συγκεκριμένες τεχνικές εξισορρόπησης φορτίου με χρήση προσομοίωσης
- ✓ παρουσιαστούν τα αποτελέσματα των παραπάνω διαδικασιών

### 1.3 Δομή εργασίας

Η παρούσα εργασία οργανώνεται συνολικά σε πέντε κεφάλαια. Το πρώτο κεφάλαιο όπως φαίνεται παραπάνω αποτελείται από τον πρόλογο, το αντικείμενο αυτής της εργασίας και τη δομή της.

Το δεύτερο κεφάλαιο χωρίζεται σε δυο ενότητες, στο θεωρητικό υπόβαθρο του Cloud Computing και στην Εξισορρόπηση φορτίου. Στην ενότητα του Cloud γίνεται μια επισκόπηση από τη πλευρά της ΕΦ. Πιο συγκεκριμένα δίνεται ο ορισμός του Cloud, αναλύεται το μοντέλο υπηρεσιών και τα μοντέλα ανάπτυξης καθώς και η αρχιτεκτονική αυτού. Αντίστοιχα στην ενότητα της ΕΦ εξετάζονται η αρχιτεκτονική και οι μετρικές της ΕΦ. Επιπλέον, γίνεται μια αναλυτική περιγραφή του τρόπου λειτουργίας ενός εξισορροπητή φορτίου. Έπειτα, παρατίθενται οι υπάρχοντες τύποι εξισορροπητών. Κλείνοντας αυτό το κεφάλαιο, γίνεται μια εκτενής αναφορά όσον αφορά τα πλεονεκτήματα και τα μειονεκτήματα της εν λόγω τεχνολογίας.

Το τρίτο κεφάλαιο αφορά την ταξινόμηση των τεχνικών ΕΦ. Γίνεται ο διαχωρισμός αυτών σε τρεις μεγάλες κατηγορίες. Στατική, δυναμική και υβριδική. Έπειτα παρουσιάζονται και εξηγούνται οι λειτουργίες των πιο γνωστών αλγορίθμων



αλλά και μερικών άλλων υφιστάμενων. Τέλος, αναλύονται τα πλεονεκτήματα, οι περιορισμοί τους, αλλά και οι πολιτικές τους.

Το τέταρτο κεφάλαιο περιέχει το λογισμικό προσομοίωσης του Cloud Analyst, τα χαρακτηριστικά του καθώς και το περιβάλλον προσομοίωσης.

Το πέμπτο κεφάλαιο αποτελείται από τέσσερα σενάρια προσομοίωσης και τη συγκριτική ανάλυση αυτών με βάση τα αποτελέσματά που λάβαμε. Οι αλγόριθμοι που χρησιμοποιούνται είναι ο Round Robin, ο ESCE ,ο Throttled, ο Honey Bee και ο PSO. Αυτή η διπλωματική διερευνά την απόδοση αυτών των αλγορίθμων συγκριτικά με την αύξηση του αριθμού των χρηστών και τον αριθμό των κέντρων δεδομένων. Τέλος, στο έκτο και τελευταίο κεφάλαιο γίνεται μια σύνοψη.

## Κεφάλαιο 2° :

### 2 Θεωρητικό Υπόβαθρο

#### 2.1 Cloud Computing

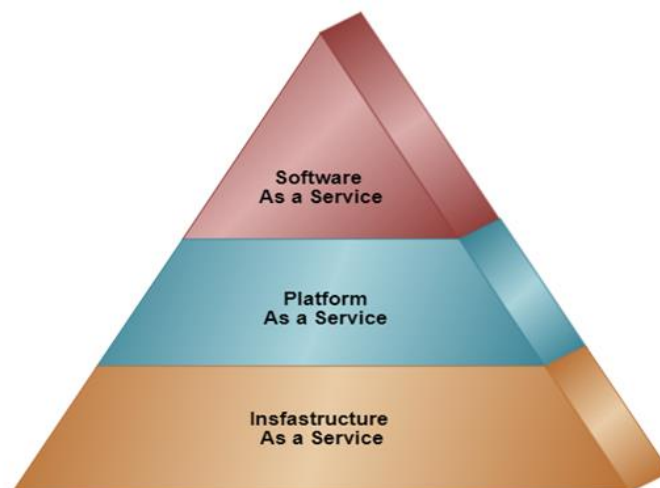
##### 2.1.1 Ορισμός

Σύμφωνα με το NIST [6] το Cloud Computing ορίζεται ως "ένα μοντέλο που επιτρέπει την πανταχού παρούσα, εύκολη και κατά παραγγελία πρόσβαση σε ένα κοινόχρηστο δίκτυο σύνολο διαμορφώσιμων υπολογιστικών πόρων (π.χ. δίκτυα, διακομιστές, αποθήκευση, εφαρμογές και υπηρεσίες) που μπορούν να παρέχονται και να αποδεσμεύονται ταχέως με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης με τον πάροχο υπηρεσιών."

##### 2.1.2 Μοντέλο υπηρεσιών του Cloud Computing

Το μοντέλο Cloud αποτελείται από τρία μοντέλα υπηρεσιών.

1. Υποδομή ως υπηρεσία (IaaS)
2. Πλατφόρμα ως υπηρεσία (PaaS)
3. Λογισμικό ως υπηρεσία (SaaS)



Εικόνα 2.1 Μοντέλα υπηρεσιών του Cloud

## 1. Υποδομή ως υπηρεσία (IaaS)

Το IaaS είναι επίσης γνωστό ως Hardware as a Service (HaaS). Αφαιρεί την πολυπλοκότητα της διαχείρισης των διακομιστών, του εύρους ζώνης και τις δικτυακές συνδέσεις από την εκάστοτε επιχείρηση. Η συγκεκριμένη υπηρεσία προσφέρει εικονικό Hardware το οποίο είναι επεκτάσιμο ανάλογα με τις ανάγκες της κάθε επιχείρησής. Το κύριο πλεονέκτημα της χρήσης του IaaS είναι ότι βοηθά τους χρήστες να αποφύγουν το κόστος και την πολυπλοκότητα της αγοράς και της διαχείρισης των φυσικών διακομιστών. Μερικά από τα πιο σημαντικά χαρακτηριστικά είναι ότι οι πόροι είναι διαθέσιμοι ως υπηρεσία, οι υπηρεσίες είναι εξαιρετικά επεκτάσιμες, είναι δυνατή η πρόσβαση που βασίζεται σε GUI<sup>2</sup> και API<sup>3</sup>, και τέλος είναι δυναμική και ευέλικτη.

## 2. Πλατφόρμα ως υπηρεσία (PaaS)

Το Cloud επιτρέπει στους πελάτες να χρησιμοποιούν τις υποδομές του ως πλατφόρμα, πάνω από την οποία αναπτύσσουν τις δικές του εφαρμογές αξιοποιώντας γλώσσες προγραμματισμού, βιβλιοθήκες, εργαλεία λογισμικού. Υποστηρίζεται από έναν πάροχο που αναπτύσσει σε εφαρμογές υποδομής που έχουν αποκτηθεί βάσει Cloud. Υπάρχουν τα ακόλουθα χαρακτηριστικά του PaaS :

- a. Προσβάσιμο σε διάφορους χρήστες μέσω της ίδιας εφαρμογής ανάπτυξης
- b. Ενσωματώνεται με υπηρεσίες web και βάσεις δεδομένων
- c. Βασίζεται στην τεχνολογία εικονικοποίησης, ώστε οι πόροι να μπορούν εύκολα να κλιμακωθούν προς τα πάνω ή προς τα κάτω ανάλογα με τις ανάγκες του οργανισμού
- d. Υποστήριξη πολλαπλών γλωσσών και πλαισίων
- e. Παρέχει τη δυνατότητα " Αυτόματης κλίμακας "

---

<sup>2</sup> Το GUI (Graphical User Interface), είναι μια μορφή διεπαφής χρήστη που επιτρέπει οι χρήστες να αλληλεπιδρούν με ηλεκτρονικές συσκευές μέσω γραφικών εικονιδίων και ενδείξεων ήχου, όπως η κύρια σημείωση, αντί για διεπαφή χρήστη που βασίζεται σε κείμενο, δακτυλογραφημένες ετικέτες εντολών ή πλοήγηση κειμένου.

<sup>3</sup> Ένα API, ή διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface), χρησιμοποιείται για τη μεταβίβαση δεδομένων μεταξύ εφαρμογών λογισμικού με έναν τυποποιημένο τρόπο. Πολλές υπηρεσίες προσφέρουν δημόσια API που επιτρέπουν σε οποιονδήποτε να στέλνει και να λαμβάνει περιεχόμενο από την υπηρεσία.

### 3. Λογισμικό ως υπηρεσία (SaaS)

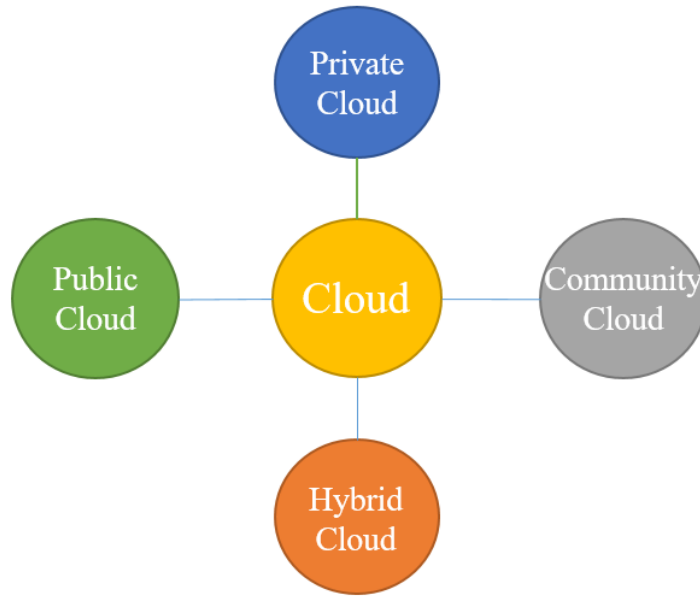
Το Cloud επιτρέπει στους πελάτες να χρησιμοποιούν εγκατεστημένες εφαρμογές που εκτελούνται στην υποδομή Cloud μέσω διαφορετικών προγραμμάτων ή διεπαφών από διαφορετικές συσκευές, έτσι ώστε ο χρήστης να έχει πρόσβαση σε βάσεις δεδομένων και εφαρμογές. Παραδείγματα εφαρμογών που παρέχονται είναι παιχνίδια και προγράμματα παραγωγικότητας όπως τα Έγγραφα Google και το Word Online. Οι εφαρμογές SaaS ενδέχεται να συνδέονται με εγκαταστάσεις αποθήκευσης Cloud ή φιλοξενίας αρχείων. Database as a Service (DaaS), Expert as a Service (EaaS), Storage as a Service (SaaS), Network as a Service (NaaS), Security as a Service (SECaaS), Communication as a Service (CaaS), Monitoring as a Service (MaaS) και Testing-as-a-service (TaaS) είναι μερικοί από τους άλλους πόρους που παρέχονται από το Cloud Computing [7], [8], [9]. Ακολουθούν ορισμένα χαρακτηριστικά του SaaS.

- a. Διαχείριση από κεντρική τοποθεσία
- b. Φιλοξενείται σε απομακρυσμένο διακομιστή
- c. Προσβάσιμο μέσω Διαδικτύου
- d. Οι ενημερώσεις υλικού και λογισμικού εφαρμόζονται αυτόματα
- e. Οι υπηρεσίες αγοράζονται βάσει πληρωμής ανά χρήση

#### 2.1.3 Μοντέλα ανάπτυξης του Cloud Computing

Το μοντέλο ανάπτυξης Cloud προσδιορίζει τον συγκεκριμένο τύπο περιβάλλοντος Cloud με βάση την ιδιοκτησία, την κλίμακα, την πρόσβαση και τη φύση και το σκοπό του. Υπάρχουν διάφορα μοντέλα ανάπτυξης που βασίζονται στην τοποθεσία και στο ποιος διαχειρίζεται την υποδομή. Ακολουθούν ορισμένοι σημαντικοί τύποι μοντέλων ανάπτυξης Cloud [8], [10]:

1. **Ιδιωτικό (Private Cloud):** Είναι μια υποδομή Cloud που είναι διαθέσιμη για χρήση μόνο από έναν οργανισμό
2. **Δημόσιο (Public Cloud):** Η δημόσια υποδομή Cloud είναι διαθέσιμη για χρήση από το ευρύ κοινό.
3. **Κοινοτικό (Community Cloud):** Η υποδομή του είναι διαθέσιμη για χρήση από μια συγκεκριμένη κοινότητα χρηστών Cloud.
4. **Υβριδικό (Hybrid Cloud):** Η υποδομή του αποτελείται από δύο ή περισσότερα διαφορετικά μοντέλα Cloud, τα οποία παραμένουν μια μοναδική οντότητα



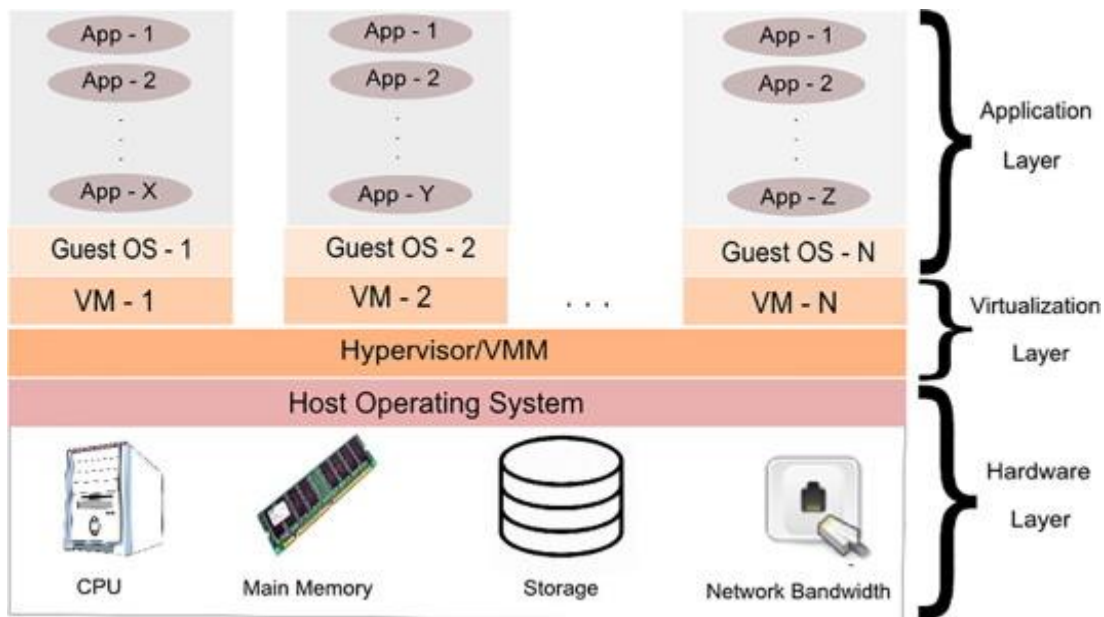
Εικόνα 2.2 Μοντέλα ανάπτυξης του Cloud

#### 2.1.4 Αρχιτεκτονική Cloud Computing

Σύμφωνα με τους Mishra κ.ά. [3] για το σχεδιασμό ενός συστήματος Cloud, πρέπει να λάβουμε υπόψη διάφορους παράγοντες όπως το κόστος, την πολυπλοκότητα, την ταχύτητα, την φορητότητα του συστήματος, καθώς και την ασφάλεια. Η αρχιτεκτονική του Cloud διαφέρει από την παραδοσιακή αρχιτεκτονική κατακεντρωμένου συστήματος στο ότι :

- i. είναι εξαιρετικά επεκτάσιμη
- ii. είναι μια αφηρημένη οντότητα και απευθύνεται σε διαφορετικά επίπεδα υπηρεσιών στον καταναλωτή cloud
- iii. το ελέγχουν οι οικονομίες κλίμακας
- iv. παρέχονται υπηρεσίες δυναμικής ζήτησης μέσω εικονικοποίησης

Ένα παράδειγμα αρχιτεκτονικής Cloud ενός μεμονωμένου κεντρικού υπολογιστή φαίνεται στην Εικόνα 2.3.



Εικόνα 2.3 A Single Host Architecture (πηγή: [3])

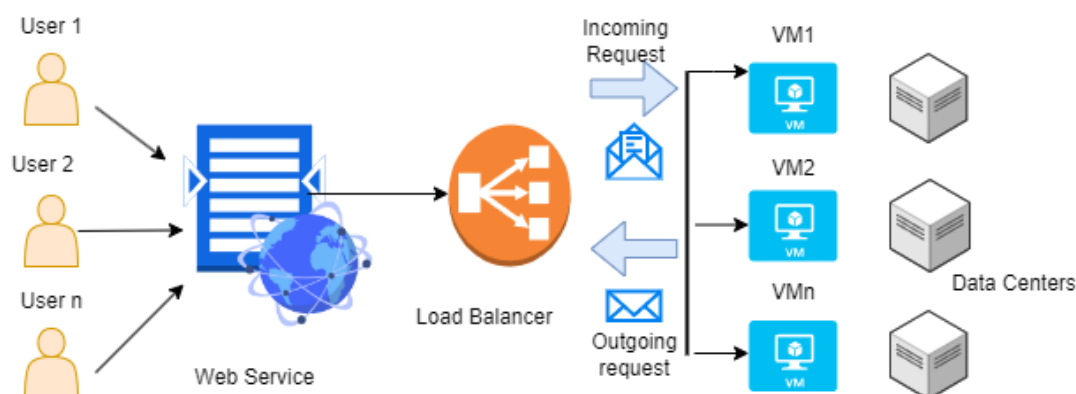
Το επίπεδο υλικού αποτελείται από πόρους υλικού όπως είναι ο επεξεργαστής, η κύρια μνήμη, η δευτερεύουσα αποθήκευση και το εύρος ζώνης δικτύου. Οι πόροι αυτοί είναι εικονικοποιημένοι. Το Virtual Machine Monitor (VMM) ή ο hypervisor<sup>4</sup> (όπως Xen, VMWare, UML, Denali, κ.λπ.) θα λειτουργήσει ως διεπαφή μεταξύ του επισκέπτη λειτουργικού συστήματος και των VM. Αυτό το VMM υποστηρίζει πολλαπλά λειτουργικά συστήματα για την εκτέλεση εφαρμογών σε μια ενιαία πλατφόρμα υλικού ταυτόχρονα. Ένας διαφορετικός αριθμός ετερογενών εφαρμογών εκτελείται σε κάθε επισκέπτη λειτουργικό σύστημα ή εικονική μηχανή που είναι η βασική μονάδα για την εκτέλεση μιας εφαρμογής ή ενός αιτήματος υπηρεσίας. Το  $\{VM_1, VM_2, \dots, VM_N\}$  είναι το σύνολο των εικονικών μηχανών που αναπτύσσονται στο σύστημα Cloud όπως φαίνεται στη παραπάνω εικόνα.

Ένα κέντρο δεδομένων Cloud αποτελείται από έναν πεπερασμένο αριθμό ετερογενών φυσικών κεντρικών υπολογιστών. Κάθε κεντρικός υπολογιστής αναγνωρίζεται από τον αριθμό αναγνώρισης του κεντρικού υπολογιστή, τις λίστες

<sup>4</sup> Ένας hypervisor (επίσης γνωστός ως virtual machine monitor, VMM, virtualizer) είναι ένας τύπος λογισμικού υπολογιστή, υλικολογισμικού ή υλικού που δημιουργεί και εκτελεί εικονικές μηχανές. Ένας υπολογιστής στον οποίο ένας hypervisor εκτελεί μία ή περισσότερες εικονικές μηχανές ονομάζεται *host machine* και κάθε εικονική μηχανή ονομάζεται *guest machine*. Ο hypervisor παρουσιάζει στα λειτουργικά συστήματα επισκέπτη μια εικονική λειτουργική πλατφόρμα και διαχειρίζεται την εκτέλεση των επισκεπτών λειτουργικών συστημάτων.

στοιχείων επεξεργασίας, την ταχύτητα επεξεργασίας ως προς τα εκατομμύρια εντολές ανά δευτερόλεπτο (MIPS), το μέγεθος της μνήμης, το εύρος ζώνης, κ.λπ. Κάθε κεντρικός υπολογιστής έχει πολλά εικονικά μηχανήματα.

Ένα VM έχει επίσης τα ίδια χαρακτηριστικά όπως ένας κεντρικός υπολογιστής. Οι εργασίες φτάνουν από τους διαφορετικούς χρήστες στον κεντρικό εξισορροπητή φορτίου ή από τον σειριακό προγραμματιστή για τη χαρτογράφηση των πόρων του Cloud. Κάθε υπολογιστικός κόμβος (VM) πραγματοποιεί την εκτέλεση μιας μεμονωμένης εργασίας κάθε φορά. Όταν έρχεται ένα αίτημα, εκχωρείται σε ένα από τα VM από τον ΕΦ, εάν υπάρχουν επαρκείς πόροι για να ολοκληρωθεί εντός της προθεσμίας. Διαφορετικά, η εργασία θα περιμένει εάν οι Περιορισμοί Συμφωνιών Επιπέδου Υπηρεσιών SLA δίνουν άδειες. Μετά την ολοκλήρωση της εκτέλεσης της εργασίας, οι πόροι που χρησιμοποιούνται από την εργασία στο συγκεκριμένο VM απελευθερώνονται και μπορούν να χρησιμοποιηθούν για τη δημιουργία νέου VM, το οποίο θα χρειαστεί για την εξυπηρέτηση ενός άλλου αιτήματος.



Εικόνα 2.4 Βασική δομή ΕΦ

### 2.1.5 Εξισορρόπηση Φορτίου στο Cloud Computing

Το πρόγραμμα ΕΦ στέλνει ένα αίτημα από τον χρήστη στον πάροχο υποδομής. Το φορτίο πρέπει να σταθεροποιείται και να διαχειρίζεται από έναν πάροχο Cloud. Οι πάροχοι Cloud βασίζονται σε αυτοματοποιημένες υπηρεσίες ΕΦ, δίνοντας τη δυνατότητα στους χρήστες να επεκτείνουν τους πόρους της CPU ή της μνήμης τους σύμφωνα με τις αυξανόμενες απαιτήσεις. Αυτή η δυνατότητα δεν είναι υποχρεωτική και είναι προσαρμοσμένη στις συγκεκριμένες απαιτήσεις κάθε χρήστη. Κατά συνέπεια, οι εξισορροπητές φορτίου εξυπηρετούν δύο βασικούς σκοπούς. Τη διασφάλιση της διαθεσιμότητας πόρων Cloud και τη βελτίωση της απόδοσης. Το Cloud Computing

χρησιμοποιεί έναν δυναμικό αλγόριθμο που επιτρέπει στις οντότητες του να ανακοινώνουν την παρουσία τους σε διακομιστές παρουσίας και διευκολύνει την επικοινωνία μεταξύ των ενδιαφερομένων. Αυτή η λύση έχει ενσωματωθεί στο RFC3920 του Internet Engineering Task Force, γνωστό ως XMPP (Extensible Messaging and Presence Protocol) [11].

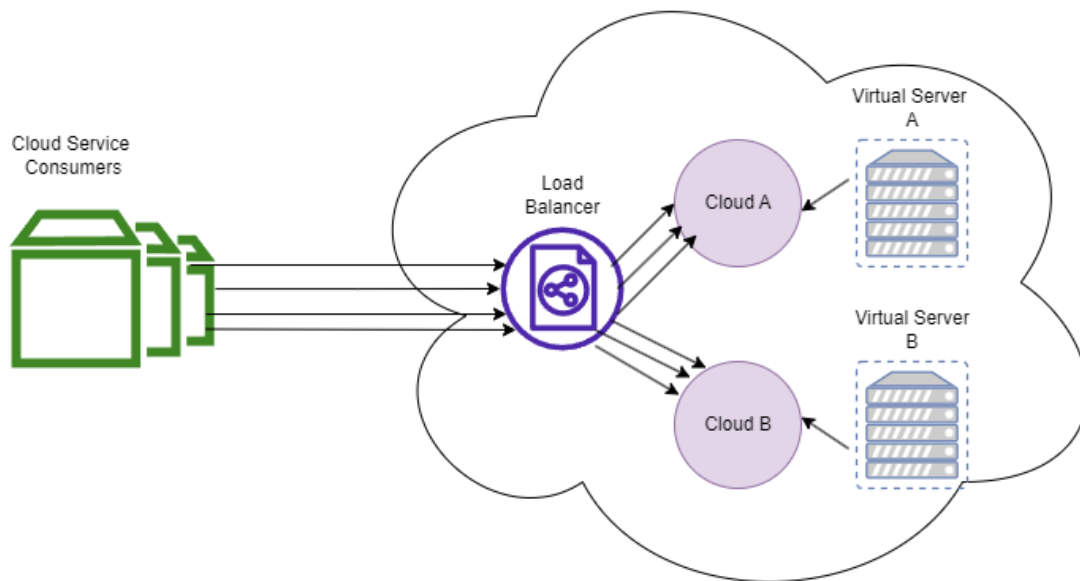
## 2.2 Εξισορρόπηση φορτίου

### 2.2.1 Αρχιτεκτονική Εξισορρόπησης Φορτίου

Οι Banisakher και Mohammed [12] στην πρόσφατη μελέτη τους, ανέλυσαν την αρχιτεκτονική κατανομής φόρτου εργασίας. Αυτή η αρχιτεκτονική επιτρέπει την βαθμιαία αύξηση (κλιμάκωση) των πόρων, ικανοποιώντας τα αιτήματα υπηρεσιών στο σύστημα Cloud. Οι πόροι κλιμακώνονται οριζόντια, πράγμα που σημαίνει ότι καθώς έρχονται περισσότερα αιτήματα, προστίθενται περισσότεροι πόροι στην ομάδα πόρων που ανταποκρίνονται στα αιτήματα. Η διαδικασία ΕΦ λειτουργεί γενικά ως εξής:

Ένας ή περισσότεροι καταναλωτές Cloud υποβάλλουν αιτήματα για μια υπηρεσία Cloud. Τα αιτήματα περνούν από έναν ΕΦ, ο οποίος στη συνέχεια καθορίζει τον τρόπο ομοιόμορφης κατανομής των αιτημάτων μεταξύ των διαθέσιμων πόρων. Στο πλαίσιο των αιτημάτων για μια υπηρεσία Cloud, ο πάροχος Cloud μπορεί να έχει πολλαπλές παρουσίες της υπηρεσίας σε πολλούς διακομιστές. Ο ΕΦ θα διασφάλιζε ομοιόμορφη κατανομή του υπολογιστικού φορτίου σε αυτούς τους διακομιστές. Η Εικόνα 2.5 αντιπροσωπεύει ένα παράδειγμα της αρχιτεκτονικής κατανομής φόρτου εργασίας.





Εικόνα 2.5 Κατανομή φόρτου εργασίας σε σύστημα Cloud

Εδώ, η υπηρεσία Cloud A και η υπηρεσία Cloud B είναι πανομοιότυποι πόροι πληροφορικής που τοποθετούνται παράλληλα και οριζόντια και έχουν πρόσβαση μέσω ενός ΕΦ, ο οποίος χρησιμοποιεί εσωτερικά αποτελεσματικούς αλγόριθμους για να αποφασίσει σε ποια υπηρεσία (A ή B) θα στείλει ένα αίτημα. Η υπηρεσία Cloud B, η οποία χαρτογραφείται εσωτερικά στη δεξιά μενή πόρων μέσω του εικονικού διακομιστή B, είναι στην πραγματικότητα ένα περιττό αντίγραφο της υπηρεσίας Cloud A και υλοποιείται στον φυσικό διακομιστή B. Ο ΕΦ είναι το πρώτο σημείο εισόδου για ένα αίτημα καταναλωτή υπηρεσίας Cloud και, μόλις λάβει το αίτημα, τον κατευθύνει είτε στους Φυσικούς Διακομιστές A είτε στον B. Ο ΕΦ φροντίζει να διασφαλίζει ομοιόμορφη κατανομή φόρτου εργασίας μέσω του υπολογισμού ενός φορτίου και μέσω της χρήσης αλγόριθμου εξισορρόπησης.

Η αρχιτεκτονική διανομής φόρτου εργασίας επιτρέπει έτσι στον πάροχο υπηρεσιών να επεκτείνει οριζόντια τις υπηρεσίες όπως χρειάζεται και να καλύψει αποτελεσματικά τις ανάγκες των πελατών. Διασφαλίζει επίσης ότι οι πόροι και η υποδομή πληροφορικής δεν υπερχρησιμοποιούνται (ή υποχρησιμοποιούνται), προσθέτει πλεονασμό και παρέχει τη δυνατότητα υψηλής διαθεσιμότητας.

Μια πραγματική περίπτωση χρήσης της αρχιτεκτονικής ΕΦ θα ήταν μια υπηρεσία Cloud που εξυπηρετεί πολλούς ιστότοπους ηλεκτρονικού εμπορίου σε ένα μοντέλο πολλαπλών μισθώσεων. Κάθε αίτημα για κάθε ιστότοπο ηλεκτρονικού

εμπορίου πρέπει να ανταποκρίνεται με διαθεσιμότητα, αποτελεσματικότητα και ασφαλή επεξεργασία. Το απολύτως απαραίτητο χαρακτηριστικό για μια διαδικτυακή υπηρεσία ηλεκτρονικού εμπορίου είναι η υψηλή διαθεσιμότητα. Έτσι λοιπόν, μια αρχιτεκτονική ΕΦ διασφαλίζει ότι κάθε υπηρεσία υποβάλλεται σε επεξεργασία χωρίς καθυστέρηση, διακοπές και άρνηση υπηρεσίας.

Το στοιχείο που κάνει την αρχιτεκτονική κατανομής φόρτου εργασίας διαφορετική από άλλες θεμελιώδεις αρχιτεκτονικές είναι ο ΕΦ. Ο εξισορροπητής αποτελείται ουσιαστικά από έναν αλγόριθμο που προσπαθεί να κατανείμει το υπολογιστικό φορτίο όσο το δυνατόν πιο ομοιόμορφα. Παρά το γεγονός ότι η ΕΦ βάσει Cloud είναι μια καλά καθορισμένη έννοια στο συγκεκριμένο περιβάλλον, η αποτελεσματική ΕΦ συνεχίζει να διαφεύγει από τους επαγγελματίες. Όπως θα μελετήσουμε στη συνέχεια, οι ερευνητές προσπαθούν συνεχώς να αναπτύξουν νέους ή βελτιωμένους αλγόριθμους για την κατανομή του φορτίου.

Εκτός από τον κύριο μηχανισμό ΕΦ, η αρχιτεκτονική μπορεί επίσης να περιλαμβάνει την παρακολούθηση ελέγχου, την παρακολούθηση χρήσης Cloud, τον hypervisor (υπερεπόπτη), τη λογική περίμετρο δικτύου, το σύμπλεγμα πόρων και τους μηχανισμούς αναπαραγωγής πόρων. Ένα εξαιρετικό παράδειγμα της αρχιτεκτονικής διανομής φόρτου εργασίας θα ήταν μια δημόσια υπηρεσία Cloud, όπως το Google Drive. Το Google Drive είναι μια υπηρεσία αποθήκευσης που προσφέρεται σε όλους τους χρήστες με λογαριασμό Google. Οι χρήστες μπορούν να ανεβάζουν, να κατεβάζουν και να χρησιμοποιούν αρχεία στο Google Drive. Με εκατοντάδες χιλιάδες χρήστες που χρησιμοποιούν τις υπηρεσίες της Google, η Google πρέπει να κατανείμει το φόρτο εργασίας μεταξύ των πολλών διακομιστών της, έτσι ώστε κανένας διακομιστής να υποχρησιμοποιείται περισσότερο ή να υπερχρησιμοποιείται από κάποιον άλλο. Αυτό διασφαλίζει την ομαλή λειτουργία των υπηρεσιών της Google .

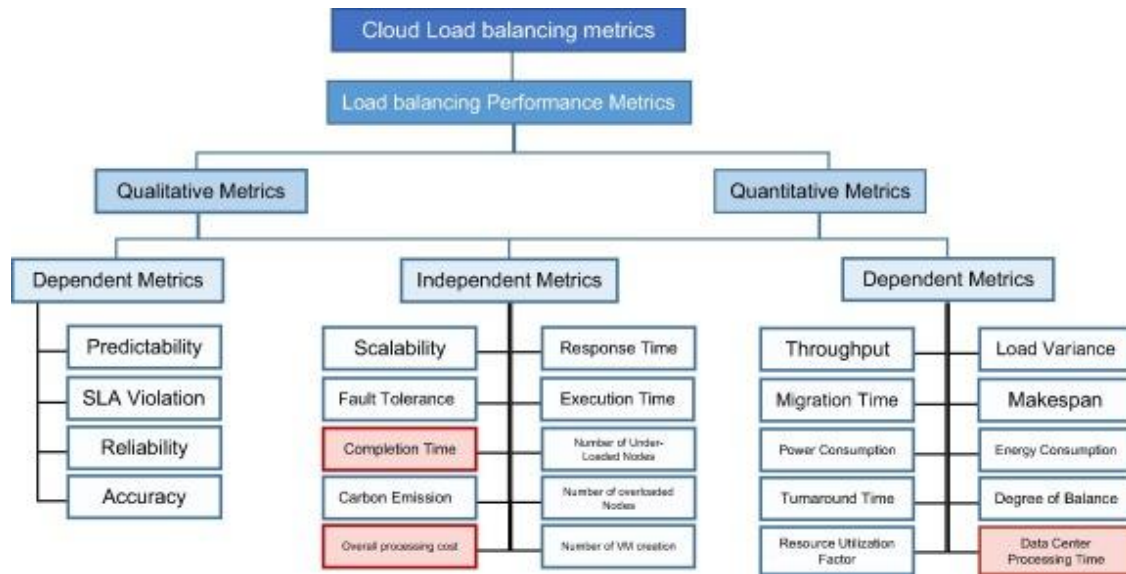
### 2.2.2 Μετρικές Εξισορρόπησης Φορτίου

Προκειμένου να επιτευχθεί καλύτερη χρήση πόρων και για να βελτιωθεί η απόδοση, ένας Εξισορροπητής φορτίου καθίσταται αναγκαίος για τη διασπορά του υπολογιστικού φορτίου στους διαθέσιμους πόρους. Αρκετές τεχνικές εξισορρόπησης φορτίου και διάφορες μετρικές (metrics) προτείνονται από ερευνητές για μεγαλύτερη ικανοποίηση των χρηστών και χρήση πόρων. Για να αυξηθεί η συνολική απόδοση του συστήματος, οι ερευνητές πρέπει να διασφαλίσουν ότι όλες οι παράμετροι πληρούνται

βέλτιστα. Αυτές έχουν εξεταστεί από τους Kumar P. και Kumar R. [13] και χωρίζονται σε δυο κατηγορίες. Ποσοτικές και ποιοτικές όπως συνοψίζονται στο ακόλουθο διάγραμμα στην Εικόνα 2.6.

- 1) **Απόδοση:** Η απόδοση μπορεί να εκφραστεί με διάφορους τρόπους, όπως για παράδειγμα με το πλήθος των διαδικασιών που υπολογίζονται στη μονάδα χρόνου. Η αποτελεσματικότητα του συστήματος πρέπει να επικυρωθεί μετά την εφαρμογή της τεχνικής σε σύγκριση με άλλες υπάρχουσες τεχνικές εξισορρόπησης φορτίου.
- 2) **Χρόνος απόκρισης:** Ο συνολικός χρόνος που απαιτείται για την ολοκλήρωση ενός υποβληθέντος αιτήματος στο σύστημα.
- 3) **Διακίνηση (Throughput):** Ο συνολικός αριθμός των υποβληθέντων εργασιών ή διεργασιών που ολοκληρώθηκαν σε μια μονάδα χρόνου σε ένα σύστημα. Όσο υψηλότερη είναι η τιμή του throughput, τόσο καλύτερη είναι η απόδοση του συστήματος.
- 4) **Επεκτασιμότητα:** Η ικανότητα του συστήματος να πραγματοποιεί ομοιόμορφη Εξισορρόπηση Φορτίου όταν αυξάνεται ο απαιτούμενος αριθμός κόμβων.
- 5) **Ανοχή ελαττωμάτων:** Η ικανότητα της τεχνικής ΕΦ να λειτουργεί ομοιόμορφα ακόμα και στη περίπτωση κατάρρευσης οποιουδήποτε συνδέσμου ή κόμβου.
- 6) **Χρόνος μετανάστευσης:** Ο χρόνος μετανάστευσης χρησιμοποιείται για να υπολογίσει το συνολικό χρόνο που απαιτείται για τη μετάδοση ενός αιτήματος ή εργασίας σε υποφορτωμένο μηχάνημα από υπερφορτωμένο μηχάνημα. Όσο χαμηλότερος είναι ο χρόνος μετεγκατάστασης, τόσο καλύτερη είναι η απόδοση του συστήματος Cloud.
- 7) **Χρήση πόρων:** Η χρήση πόρων αξιολογείται για να διασφαλιστεί ότι όλοι οι πόροι χρησιμοποιούνται σωστά στο σύστημα Cloud. Η μεγαλύτερη χρήση πόρων θα οδηγήσει στην ελαχιστοποίηση του συνολικού κόστους, στη μείωση της ενέργειας, των δαπανών.
- 8) **Βαθμός ανισορροπίας:** Περιγράφει τη διακύμανση μεταξύ των VM.

9) **Makespan:** Χρησιμοποιείται για να αντιπροσωπεύει το συνολικό χρόνο ή το χρόνο ολοκλήρωσης που απαιτείται για την κατανομή πόρων στους χρήστες.



Εικόνα 2.6 Ταξινόμηση παραμέτρων ΕΦ (πηγή: [2])

### 2.2.3 Τύποι Εξισορροπητών βάσει διαμορφώσεων

#### A. Εξισορροπητές Φορτίου υλικού

Όπως υποδηλώνει το όνομα, πρόκειται για έναν φυσικό υλικό εξοπλισμό που χρησιμοποιείται για τη διανομή της κίνησης σε διάφορους διακομιστές. Πρόκειται δηλαδή για ολοκληρωμένα κυκλώματα ειδικών εφαρμογών (ASIC) προσαρμοσμένα για μια συγκεκριμένη χρήση. Αυτοί οι εξισορροπητές, αν και είναι σε θέση να χειριστούν έναν τεράστιο όγκο κίνησης, είναι αρκετά περιορισμένοι από άποψη ευελιξίας και είναι επίσης αρκετά υψηλοί σε τιμές [14], [15].

#### B. Εξισορρόπηση Φορτίου λογισμικού

Είναι οι εφαρμογές λογισμικού που πρέπει να εγκατασταθούν στο σύστημα και λειτουργούν παρόμοια με τους ΕΦ υλικού. Είναι δύο ειδών. Εμπορικοί και Ανοιχτού Κώδικα και αποτελούν μια οικονομικά αποδοτική εναλλακτική λύση σε σχέση με τα αντίστοιχα του υλικού.

#### C. Εικονικοί Εξισορροπητές Φορτίου

Αυτό το πρόγραμμα ΕΦ διαφέρει τόσο από το λογισμικό ΕΦ όσο και από το υλικό, καθώς είναι ο συνδυασμός του προγράμματος ενός ΕΦ υλικού που λειτουργεί

σε μια εικονική μηχανή. Μέσω της εικονικοποίησης, αυτό το είδος μιμείται την υποδομή που βασίζεται στο λογισμικό. Η εφαρμογή προγράμματος του εξοπλισμού υλικού εκτελείται σε μια εικονική μηχανή για να ανακατευθυνθεί η κίνηση ανάλογα. Όμως, τέτοιοι εξισορροπητές έχουν παρόμοιες προκλήσεις με τους φυσικούς εξισορροπητές, δηλαδή έλλειψη κεντρικής διαχείρισης, μικρότερη επεκτασιμότητα και πολύ περιορισμένη αυτοματοποίηση [15].

#### *2.2.3.1 Σύγκριση Εξισορροπητών Υλικού με Εξισορροπητές Φορτίου Λογισμικού*

Οι συσκευές ΕΦ υλικού απαιτούν αρχική επένδυση, διαμόρφωση και συνεχή συντήρηση. Ενδέχεται επίσης να μην χρησιμοποιηθούν σε πλήρη χωρητικότητα, ειδικά αν αγοραστεί ένας μόνο, με σκοπό να χειρίζεται τις αιχμές της κυκλοφορίας στο χρόνο αιχμής. Εάν ο όγκος της κίνησης αυξηθεί ξαφνικά πέρα από την τρέχουσα χωρητικότητά του, αυτό θα επηρεάσει τους χρήστες μέχρι να αγοραστεί και να ρυθμιστεί ένας άλλος ΕΦ.

Αντίθετα, οι ΕΦ που βασίζονται σε λογισμικό είναι πολύ πιο ευέλικτοι. Μπορούν να κλιμακωθούν προς τα πάνω ή προς τα κάτω εύκολα και είναι πιο συμβατά με τα σύγχρονα περιβάλλοντα Cloud. Επίσης κοστίζουν λιγότερο για τη ρύθμιση, τη διαχείριση και τη χρήση τους με την πάροδο του χρόνου.

#### *2.2.3.2 Παραδείγματα Λειτουργίας Εξισορροπητών*

Ενδεικτικά παραδείγματα ΕΦ είναι [16]:

**Τεχνική αποστολής αιτήματος άμεσης δρομολόγησης:** Ένας πραγματικός διακομιστής και μια συσκευή ΕΦ μοιράζονται την εικονική διεύθυνση IP. Σε αυτή τη τεχνική, ο εξισορροπητής παίρνει μια διεπαφή κατασκευασμένη με την εικονική διεύθυνση IP που δέχεται από πακέτα αιτημάτων και δρομολογεί απευθείας το πακέτο στους επιλεγμένους διακομιστές.

**Σύμπλεγμα ΕΦ βάσει διεκπεραιωτή (dispatcher):** Ένας διεκπεραιωτής κάνει έξυπνη ΕΦ χρησιμοποιώντας τη διαθεσιμότητα διακομιστή, το φόρτο εργασίας, τη δυνατότητα και άλλα κριτήρια που καθορίζονται από τον χρήστη για να ρυθμίσει πού θα σταλεί ένα αίτημα TCP/IP. Η μονάδα διεκπεραιωτή ενός ΕΦ μπορεί να διαχωρίσει τα αιτήματα HTTP σε διάφορους κόμβους σε ένα σύμπλεγμα. Ο διεκπεραιωτής μοιράζει το φορτίο σε πολλούς διακομιστές σε

ένα σύμπλεγμα, έτσι ώστε οι υπηρεσίες διαφόρων κόμβων να φαίνονται σαν μια εικονική υπηρεσία σε μια μόνο διεύθυνση IP. Οι καταναλωτές αλληλεπιδρούν σαν να ήταν ένας μόνος διακομιστής, χωρίς να έχουν πληροφορίες σχετικά με την υποδομή του back-end<sup>5</sup>.

**Linux Virtual Load Balancer:** Είναι μια βελτιωμένη λύση ΕΦ ανοιχτού κώδικα που χρησιμοποιείται για τη δημιουργία επεκτάσιμων και διαθέσιμων υπηρεσιών δικτύου όπως HTTP, POP3, FTP, SMTP, media και Voice Over Internet Protocol (VoIP). Είναι απλό και ισχυρό προϊόν κατασκευασμένο για ΕΦ. Το ίδιο το πρόγραμμα εξισορρόπησης φορτίου είναι το κύριο σημείο εισόδου των συστημάτων συμπλέγματος διακομιστών και μπορεί να εκτελέσει τον εικονικό διακομιστή πρωτοκόλλου Internet (IPVS), ο οποίος υλοποιεί την εξισορρόπηση φορτίου επιπέδου μεταφοράς στον πυρήνα του Linux.

### 2.2.3.3 Τρόπος λειτουργίας Εξισορροπητή Φορτίου Λογισμικού

Η ΕΦ στο Cloud ακολουθεί μια προσέγγιση που βασίζεται σε λογισμικό για τη διανομή της κυκλοφορίας δικτύου μεταξύ των πόρων, σε αντίθεση με την ΕΦ βάσει υλικού, η οποία είναι πιο κοινή στα κέντρα δεδομένων επιχείρησης. Ένα πρόγραμμα ΕΦ λαμβάνει την εισερχόμενη κίνηση και δρομολογεί αυτά τα αιτήματα σε ενεργούς στόχους με βάση μια διαμορφωμένη πολιτική. Μια υπηρεσία ΕΦ παρακολουθεί επίσης την υγεία των μεμονωμένων στόχων για να διασφαλίσει ότι αυτοί οι πόροι είναι πλήρως λειτουργικοί [17].

- **Ορισμός ονόματος IP ή DNS<sup>6</sup> για την ΕΦ:** Οι διαχειριστές ορίζουν μία διεύθυνση IP ή και όνομα DNS για μια δεδομένη εφαρμογή, εργασία ή ιστότοπο, στον οποίο θα υποβάλλονται όλα τα αιτήματα. Αυτή η διεύθυνση IP ή το όνομα DNS είναι ο διακομιστής ΕΦ.
- **Προσθήκη χώρου συγκέντρωσης υποστήριξης για ΕΦ:** Στη συνέχεια, ο διαχειριστής θα εισαγάγει στον διακομιστή εξισορρόπησης φορτίου τις διευθύνσεις IP όλων των πραγματικών διακομιστών που θα μοιράζονται το

---

<sup>5</sup> Το backend είναι το μέρος του συστήματος που περιέχει τον προγραμματισμό (π.χ. scripting), βάσεις δεδομένων και άλλα μέρη του συστήματος που περιέχουν πληροφορίες που ο χρήστης δεν χρειάζεται να δει ή, στις περισσότερες περιπτώσεις, τα μέρη που περιέχουν βασικές πληροφορίες.

<sup>6</sup> Domain Name System (DNS) είναι ένα ιεραρχικό σύστημα ονοματοδοσίας για δίκτυα υπολογιστών, που χρησιμοποιούν το πρωτόκολλο IP. Το σύστημα αυτό μπορεί και αντιστοιχίζει τα ονόματα των υπολογιστών υπηρεσίας σε αριθμητικές διευθύνσεις.

φόρτο εργασίας για μια δεδομένη εφαρμογή ή εργασία. Αυτή η ομάδα διαθέσιμων διακομιστών είναι προσβάσιμη μόνο εσωτερικά, μέσω του load balancer.

- **Ανάπτυξη ΕΦ:** Τέλος, το πρόγραμμα ΕΦ πρέπει να αναπτυχθεί είτε ως διακομιστής μεσολάβησης, ο οποίος βρίσκεται μεταξύ των διακομιστών της εφαρμογής και των χρηστών παγκοσμίως και δέχεται όλη την επισκεψιμότητα, είτε ως πύλη, η οποία εκχωρεί έναν χρήστη σε έναν διακομιστή μία φορά και αποχωρεί μόνος μετά από την αλληλεπίδραση.
- **Αιτήματα ανακατεύθυνσης:** Μόλις εγκατασταθεί το σύστημα ΕΦ, όλα τα αιτήματα προς την εφαρμογή έρχονται στον εξισορροπητή και ανακατευθύνονται σύμφωνα με τον προτιμώμενο αλγόριθμο του διαχειριστή.

#### 2.2.4 Πλεονεκτήματα - Μειονεκτήματα Εξισορρόπησης Φορτίου

Με βάση την μελέτη των Singh και Shukla τα πλεονεκτήματα ΕΦ τα οποία αναλύονται παρακάτω είναι τα εξής [7], [18]:

- **Εφαρμογές Υψηλής Απόδοσης:** Σε αντίθεση με τις συμβατικές τοπικές αντίστοιχές τους, οι προσεγγίσεις ΕΦ Cloud είναι λιγότερο δαπανηρές και πιο εύκολο να υιοθετηθούν. Οι εταιρείες είναι σε θέση να κάνουν τις εφαρμογές των πελατών τους να λειτουργούν πιο γρήγορα και να αποδίδουν καλύτερα από ό,τι σε πιθανώς φθηνότερες τιμές.
- **Βελτιωμένη επεκτασιμότητα:** Η εξισορρόπηση του Cloud βοηθά να διατηρείται η επισκεψιμότητα του ιστότοπου επεκτάσιμη και ευέλικτη. Μπορούμε απλώς να αντιστοιχίσουμε την αυξανόμενη επισκεψιμότητα των χρηστών με αποτελεσματικούς εξισορροπητές φορτίου και να τη διανείμουμε σε πολλούς διακομιστές ή συσκευές δικτύου. Για διαδικτυακούς ιστότοπους που έχουν να κάνουν με χιλιάδες επισκέπτες του διαδικτύου ανά δευτερόλεπτο, είναι πολύ σημαντικό. Τέτοιοι αποτελεσματικοί εξισορροπητές χρειάζονται για την κατανομή του φόρτου εργασίας κατά τις πωλήσεις ή άλλες προωθητικές υπηρεσίες.
- **Ικανότητα αντιμετώπισης αυξήσεων επισκεψιμότητας:** Κατά τη διάρκεια οποιασδήποτε ανακοίνωσης αποτελέσματος, ένας συνήθως λειτουργικός ιστότοπος του Πανεπιστημίου μπορεί να καταρρεύσει εντελώς. Αυτό οφείλεται

στη δυνατότητα λήψης αιτημάτων σε μεγάλη κλίμακα. Δεν θα χρειάζεται να ανησυχούν για την αύξηση της κυκλοφορίας εάν χρησιμοποιούν συσκευές ΕΦ Cloud. Όποιο και αν είναι το μέγεθος του αιτήματος, μπορεί να διανεμηθεί προσεκτικά μεταξύ πολλών διακομιστών για να αποφέρει τα καλύτερα αποτελέσματα στο συντομότερο χρονικό διάστημα.

**Συνέχεια με πλήρη ευελιξία:** Ο κύριος στόχος είναι να σώσουμε ή να υπερασπιστούμε έναν ιστότοπο από απότομες βλάβες με την ανάπτυξη ενός κατάλληλου εξισορροπητή. Ακόμη και αν ένας κόμβος αποτύχει, ο φόρτος εργασίας μπορεί να μετακινηθεί σε άλλον ενεργό κόμβο όταν διανέμεται μεταξύ πολλών διακομιστών ή μονάδων δικτύου.

Τα μειονεκτήματα ΕΦ τα οποία αναλύονται παρακάτω είναι τα εξής [22]:

Η ΕΦ είναι βασικά μια απαίτηση στο περιβάλλον Cloud, ωστόσο υπάρχουν μερικές προκλήσεις. Αν και η επεκτασιμότητα είναι ένα από τα πιο ελκυστικά πλεονεκτήματα τόσο του Cloud όσο και της ΕΦ, η επεκτασιμότητα είναι επίσης μια από τις πιο εμφανείς αδυναμίες της. Η επεκτασιμότητα παρεμποδίζεται για τους περισσότερους εξισορροπητές από έναν περιορισμένο αριθμό κόμβων για τη διανομή διεργασιών. Άλλες προκλήσεις περιλαμβάνουν την κατανάλωση ενέργειας, την παρακολούθηση της απόδοσης, τη διαχείριση QoS, τον προγραμματισμό πόρων και τη διαθεσιμότητα υπηρεσιών στο Cloud. Ορισμένες από αυτές θα αναλυθούν στη συνέχεια.

### 2.2.5 Προκλήσεις

Παρακάτω αναλύονται οι προκλήσεις ΕΦ [19]:

**Γεωγραφικά κατανεμημένοι κόμβοι:** Τα κέντρα δεδομένων βρίσκονται κυρίως σε γεωγραφικά απομονωμένες τοποθεσίες για μεγάλους παρόχους Cloud. Αυτά λειτουργούν ως ένας ενιαίος διακομιστής για την επίλυση των απαιτήσεων των χρηστών. Το να λειτουργούν ως ενιαία μηχανή είναι μια πιθανή πρόκληση, ιδιαίτερα για το περιβάλλον όπου αυτή η απόσταση είναι σημαντική. Υπάρχουν πολλά ζητήματα που σχετίζονται με την καθυστέρηση, όπως η ταχύτητα επικοινωνίας, η υποδομή δικτύου και πολλά άλλα παρόμοια. Εάν αυτός ο διαχωρισμός δεν είναι σε μεγάλη απόσταση, η καθυστέρηση δικτύου είναι ανεκτή. Άλλες προκλήσεις έρχονται ακόμη και σε γεωγραφικά



κοντινά κέντρα δεδομένων, όπως η επιλογή ενός κατάλληλου αλγορίθμου που λειτουργεί καλά για όλες τις εικονικές μηχανές.

**Μεμονωμένο σημείο αποτυχίας:** Επί του παρόντος, οι διαθέσιμοι αλγόριθμοι ΕΦ χρησιμοποιούν ως επί το πλείστον ένα μόνο μηχάνημα για τον έλεγχο της ΕΦ άλλων εικονικών μηχανών. Ειδικότερα, οι περισσότερες εικονικές μηχανές στο περιβάλλον Cloud λειτουργούν με κατανεμημένο τρόπο. Ως εκ τούτου, υπάρχει ανάγκη για κατανεμημένους αλγόριθμους εξισορρόπησης φορτίου, έτσι ώστε η αστοχία μιας μόνο μηχανής να μην προκαλεί την αστοχία όλων των υπολογιστικών πόρων.

**Μετεγκατάσταση εικονικής μηχανής:** Μπορούν να διαμορφωθούν περισσότερες από μία εικονικές μηχανές σε ένα φυσικό μηχάνημα ή διακομιστή. Αυτές οι εικονικές μηχανές λειτουργούν ως διαφορετικά συστήματα. Σε περίπτωση που οι υποκείμενες φυσικές μηχανές υπερφορτωθούν, οι εικονικές μηχανές πρέπει να μετεγκατασταθούν από μια μηχανή σε άλλες φυσικές μηχανές. Η ΕΦ για τέτοιου είδους μετεγκαταστάσεις VM είναι μια πρόκληση που πρέπει να αντιμετωπιστεί.

**Ετερογενείς κόμβοι:** Οι πρώτοι ερευνητές ΕΦ ανέπτυξαν αλγόριθμους για ομοιογενείς κόμβους. Καθώς οι απαιτήσεις των χρηστών είναι πολύπλοκες, υπάρχει μια ιδέα ετερογενών κόμβων σε περιβάλλον Cloud. Νέες τεχνικές ΕΦ είναι απαραίτητες για να ικανοποιηθούν οι απαιτήσεις των ετερογενών κόμβων.

**Επεκτασιμότητα Εξισορρόπησης Φορτίου:** Η επεκτασιμότητα αναφέρεται στην αύξηση ή τη μείωση των υπολογιστικών πόρων των χρηστών ανάλογα με τη ζήτησή τους. Η επεκτασιμότητα και η διαθεσιμότητα κατ' απαίτηση είναι βασικά χαρακτηριστικά του Cloud. Για την επίτευξη επεκτασιμότητας είναι υποχρεωτική ύπαρξη κατάλληλου αλγορίθμου.

**Πολυπλοκότητα αλγορίθμων:** Οι αλγόριθμοι ΕΦ θα πρέπει να είναι όσο το δυνατόν πιο αποτελεσματικοί και απλοί, προκειμένου να ξεπεραστούν οι επιβαρύνσεις στις υπηρεσίες Cloud.

## Κεφάλαιο 3<sup>ο</sup> :

### 3 Τεχνικές και Αλγόριθμοι Εξισορρόπησης

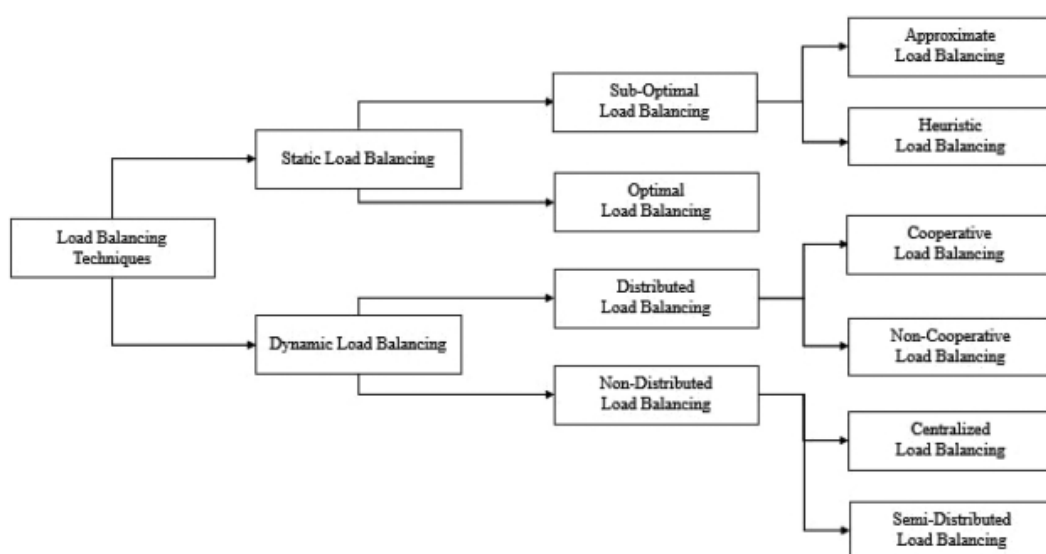
#### 3.1 Εισαγωγή

Οι τεχνικές ΕΦ ταξινομούνται κυρίως με βάση την κατάσταση του συστήματος και την έναρξη της διαδικασίας. Οι προσεγγίσεις ΕΦ που βασίζονται στην κατάσταση του συστήματος ταξινομούνται περαιτέρω σε στατικές και δυναμικές κατηγορίες, όπως φαίνεται στην Εικόνα 3.1. Με βάση την έναρξη της διαδικασίας, ταξινομούνται ως εκκίνηση αποστολέα, εκκίνηση παραλήπτη και συμμετρική. Αναλυτικές πληροφορίες για την κάθε μια κατηγορία δίνονται παρακάτω [13].

**Εκκίνηση αποστολέα:** Σε αυτή την τεχνική, εάν ένας κόμβος είναι υπερφορτωμένος, αναζητά άλλους κόμβους που είναι ελαφρά φορτωμένοι για να μοιραστεί το φόρτο εργασίας. Ο αποστολέας ξεκινά τη διαδικασία εύρεσης των υποφορτωμένων κόμβων όταν οι κόμβοι είναι υπό συμφόρηση.

**Εκκίνηση παραλήπτη:** Σε αυτήν την τεχνική, ο δέκτης ή οι κόμβοι με ελαφρά φόρτωση αναζητούν κόμβους με μεγάλο φορτίο για να μοιραστούν το φόρτο εργασίας.

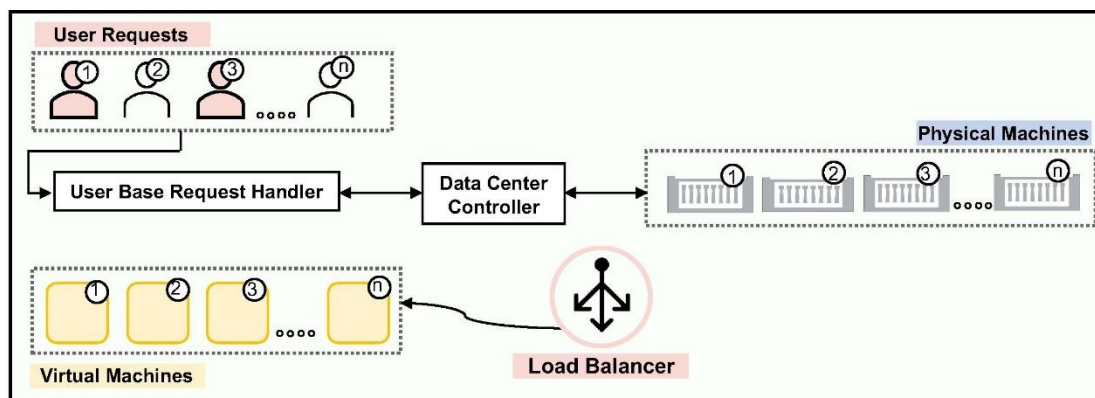
**Συμμετρική:** Στη συγκεκριμένη τεχνική, συνδυάζονται τόσο οι τεχνικές διεργασίας που εκκινούνται από τον αποστολέα όσο και οι τεχνικές διεργασίας που εκκινούνται από τον παραλήπτη.



Εικόνα 3.1 Ταξινόμηση στρατηγικών ΕΦ με βάση την κατάσταση του συστήματος (πηγή: [13])

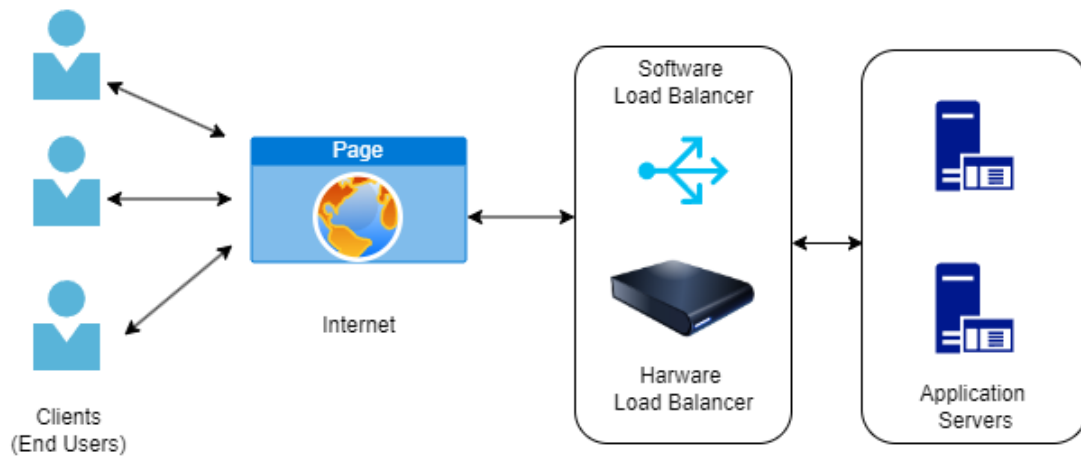
### 3.2 Μοντέλο Εξισορρόπησης Φορτίου

Ένα αφαιρετικό μοντέλο που αναπαριστά τη δομή και τη λειτουργία ενός ΕΦ στο Cloud Computing παρουσιάζεται στην Εικόνα 3.2. Οι χρήστες υποβάλλουν τα αιτήματά τους προς το Cloud, τα οποία αφού αναλυθούν μεταβιβάζονται σε ένα κέντρο δεδομένων με βάση την εφαρμοζόμενη πολιτική και τα κριτήρια που πληρούνται. Για την ισοκατανομή του φορτίου στους διακομιστές (VM) χρησιμοποιείται ο ΕΦ. Επιπλέον, μπορεί να υπάρχει άνιση κατανομή φορτίου σε πολλούς παράγοντες, ένας από αυτούς είναι ο προγραμματισμός εργασιών. Χωρίς τον κατάλληλο προγραμματισμό εργασιών, οι πόροι δεν θα χρησιμοποιηθούν αποτελεσματικά. Τέλος, η ΕΦ πραγματοποιείται στη πλευρά του server του Cloud.



Εικόνα 3.2 Αναθεωρημένο μοντέλο Εξισορρόπησης Φορτίου (πηγή: [2])

Στο Cloud, η ΕΦ έχει δύο σημαντικούς στόχους. Αυτοί είναι η κατανομή πόρων και ο προγραμματισμός εργασιών. Αρχικά, γίνεται η αντιστοίχιση εργασιών στα VMs έτσι ώστε να μην υπάρχει υπερφορτωμένος κόμβος ή κενός κόμβος ή κόμβος με μικρότερο φόρτο εργασίας. Στη συνέχεια, εφαρμόζονται τεχνικές προγραμματισμού εργασιών για τη βελτίωση της διαδικασίας ολοκλήρωσης των εργασιών με βάση τις απαιτήσεις του χρήστη για την εκπλήρωση της καθορισμένης προθεσμίας [2].



Εικόνα 3.3 Διάγραμμα Εξισορρόπησης Φορτίου (πηγή: [20])

### 3.3 Τεχνικές Εξισορρόπησης Φορτίου

#### 3.3.1 Τεχνικές Στατικής Εξισορρόπησης Φορτίου

Οι τεχνικές στατικής ΕΦ δεν χρειάζονται γνώση της τρέχουσας κατάστασης ενός συστήματος. Διατηρούν εκ των προτέρων μόνο γνώση των πόρων του συστήματος, όπως ο χρόνος εκτέλεσης, η μνήμη, η χωρητικότητα αποθήκευσης και η επεξεργαστική ισχύς των κόμβων. Η εξισορρόπηση στατικού φορτίου δεν επιτρέπει την κατανομή πόρων κατά το χρόνο εκτέλεσης. Αυτές οι τεχνικές είναι εύκολο να εφαρμοστούν και να εκτελεστούν, αλλά είναι χρήσιμες για μικρότερα συστήματα ή δίκτυα με μικρότερο αριθμό πόρων. Επειδή δεν λαμβάνουν υπόψη την τρέχουσα κατάσταση του συστήματος, αυτές οι τεχνικές δεν είναι χρήσιμες για υπολογιστικά συστήματα που εκτελούν κατανεμημένους υπολογισμούς. Επίσης, δεν επιτρέπουν την ανίχνευση του συνδεδεμένου υπολογιστή διακομιστή κατά το χρόνο εκτέλεσης, που οδηγεί σε άνιση κατανομή των πόρων. Ορισμένες από τις πιο γνωστές τεχνικές στατικής εξισορρόπησης φορτίου είναι:

1. Round Robin
2. Min-Min
3. Max-Min

Μια τεχνική στατικής ΕΦ διατηρεί όλες τις πληροφορίες που σχετίζονται με την εργασία διαθέσιμες πριν από την ελαχιστοποίηση του χρόνου αναμονής των

εργασιών. Παίρνει τη λίστα όλων των εισερχόμενων εργασιών και καθορίζει τον χρόνο εκτέλεσής τους. Επιπλέον, μια εργασία με ελάχιστο χρόνο εκτέλεσης θα εκτελεστεί πρώτη και η εργασία με τον μέγιστο χρόνο εκτέλεσης θα υποβληθεί σε επεξεργασία τελευταία. Πρώτα θα εκτελεστούν οι πιο σύντομες εργασίες. Ωστόσο, το σύστημα μπορεί να αντιμετωπίσει “ασυτία” για ορισμένες από τις εργασίες, καθώς απαιτούν περισσότερο χρόνο επεξεργασίας επειδή δεν βρίσκονται στην ουρά για εκτέλεση.

Η στατική ΕΦ βάσει εργασιών επιτρέπει στις εργασίες να εκτελούνται κυκλικά. Κάθε εργασία εκτελείται για ένα χρονικό διάστημα και μπαίνει ξανά στην ουρά για να εκτελέσει μια άλλη διαδικασία. Ακολουθείται μια τεχνική ΕΦ έως ότου όλες οι διεργασίες ολοκληρώσουν τις εργασίες τους. Αυτή η τεχνική είναι χρήσιμη για διακομιστές ιστού όπου όλα τα αιτήματα που φτάνουν είναι του ίδιου τύπου. Ωστόσο, δεν είναι καλό για το περιβάλλον Cloud, όπου μια διαδικασία φτάνει με διαφορετική διαμόρφωση.

Όταν πρόκειται για κατανομή εργασιών, μια μέθοδος ΕΦ είναι στατική όπως είδαμε και προηγουμένως εάν αγνοεί την κατάσταση του συστήματος. Η κατάσταση του συστήματος περιλαμβάνει ενέργειες όπως ο βαθμός φόρτωσης διαφόρων επεξεργαστών (και περιστασιακά ακόμη και υπερχειλίσης). Αντίθετα, καθορίζονται εκ των προτέρων υποθέσεις για ολόκληρο το σύστημα, όπως οι περίοδοι άφιξης και οι απαιτήσεις εισερχόμενων πόρων. Οι αριθμοί, η ισχύς και οι ταχύτητες επικοινωνίας των CPU είναι επίσης γνωστοί. Ως εκ τούτου, η εξισορρόπηση στατικού φορτίου προορίζεται να συνδυάσει ένα δεδομένο σύνολο φόρτων εργασίας με διαθέσιμους επεξεργαστές, έτσι ώστε μια δεδομένη λειτουργία απόδοσης να ελαχιστοποιείται.

Τα χαρακτηριστικά των στατικών αλγορίθμων είναι τα εξής:

- ✓ Αποφασίζουν με βάση έναν σταθερό κανόνα, για παράδειγμα, φορτίο εισόδου
- ✓ Δεν είναι ευέλικτοι
- ✓ Χρειάζονται προηγούμενη γνώση για το σύστημα

### 3.3.2 Τεχνικές Δυναμικής Εξισορρόπησης Φορτίου

Οι δυναμικοί αλγόριθμοι λαμβάνουν υπόψη το τρέχον φορτίο κάθε μονάδας υπολογιστή (γνωστό και ως κόμβος) στο σύστημα, σε αντίθεση με τις τεχνικές στατικής

κατανομής φορτίου. Επομένως, οι εργασίες μπορούν να μεταφερθούν δυναμικά από έναν υπερφορτωμένο κόμβο σε έναν υποφορτωμένο κόμβο για να υποβληθούν σε επεξεργασία πιο γρήγορα. Αν και είναι πολύ πιο δύσκολο να κατασκευαστούν αυτοί οι αλγόριθμοι, μπορούν να αποφέρουν εξαιρετικά αποτελέσματα, ειδικά εάν οι χρόνοι εκτέλεσης ποικίλλουν σημαντικά μεταξύ των εργασιών. Σχεδόν όλοι οι δυναμικοί αλγόριθμοι ακολουθούν τέσσερα βήματα [21]:

1. *Παρακολούθηση φορτίου*: Σε αυτό το βήμα, παρακολουθείται το φορτίο και η κατάσταση των πόρων
2. *Συγχρονισμός*: Σε αυτό το βήμα, ανταλλάσσονται οι πληροφορίες φορτίου και κατάστασης.
3. *Κριτήρια εξισορρόπησης*: Είναι απαραίτητο να υπολογιστεί μια νέα κατανομή εργασίας και στη συνέχεια να ληφθούν αποφάσεις εξισορρόπησης φορτίου με βάση αυτόν τον νέο υπολογισμό.
4. *Task Migration*: Σε αυτό το βήμα, πραγματοποιείται η πραγματική μετακίνηση των δεδομένων. Όταν το σύστημα αποφασίσει να μεταφέρει μια εργασία ή μια διαδικασία, αυτό το βήμα θα εκτελεστεί.

Μερικά παραδείγματα δυναμικών τεχνικών είναι τα εξής :

- ✓ Honey Bee
- ✓ Throttled
- ✓ ESCE

Τα χαρακτηριστικά των δυναμικών αλγορίθμων είναι τα εξής:

- ✓ Αποφασίζουν με βάση την τρέχουσα κατάσταση του συστήματος
- ✓ Είναι ευέλικτοι
- ✓ Βελτιώνουν την απόδοση του συστήματος

### 3.3.3 Υβριδικές Τεχνικές Εξισορρόπησης Φορτίου

Οι υβριδικές τεχνικές ΕΦ αναπτύσσονται για να ξεπεραστούν τα μειονεκτήματα των στατικών και δυναμικών τεχνικών διατηρώντας τα χαρακτηριστικά και τα πλεονεκτήματά τους. Οι υβριδικές τεχνικές ελαχιστοποιούν τον χρόνο απόκρισης και παρέχουν αποτελεσματική χρήση των πόρων. Ένα υβριδικό σχήμα

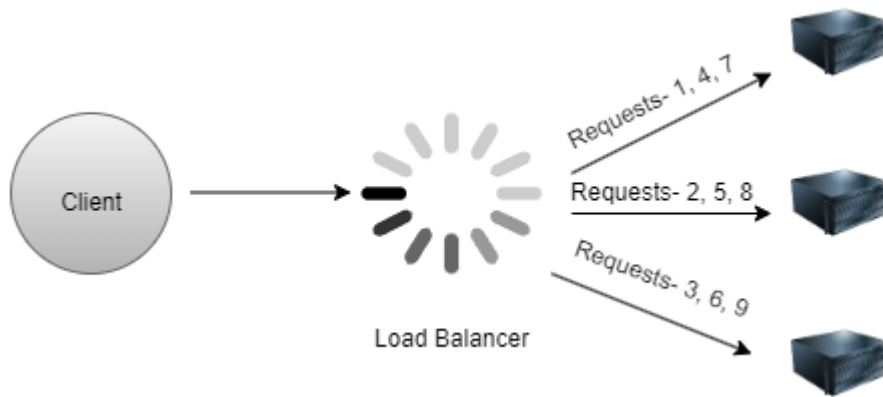
εξισορρόπησης φορτίου που αποτελείται από προγραμματισμό κατ' απαίτηση, εργασία Querying and Migrating (QMT) και Staged Task Migration (STM) βοηθά στην αποτελεσματικότερη διαχείριση του φορτίου στους κόμβους. Κάθε φορά που ένας κόμβος ανιχνεύει μεγάλο φορτίο, το QMT θα εξισορροπεί το φορτίο μεταφέροντας τον τελευταίο εισερχόμενο κόμβο σε έναν άλλο κόμβο με χαμηλό φορτίο. Τόσο το QMT όσο και το SMT λειτουργούν καλά για εξαρτημένες και ανεξάρτητες εργασίες, αλλά υποφέρουν από υψηλό χρόνο μετάδοσης και προγραμματισμού που μπορεί να ελαχιστοποιηθεί στο μέλλον [13].

### 3.4 Αλγόριθμοι Εξισορρόπησης Φορτίου

#### 3.4.1 Στατικοί Αλγόριθμοι

Οι στατικοί αλγόριθμοι χωρίζονται σε δύο κατηγορίες, βέλτιστους και υποβέλτιστους. Σε βέλτιστους αλγόριθμους, ο ελεγκτής κέντρου δεδομένων καθορίζει πληροφορίες σχετικά με τις εργασίες και τους πόρους και ο ΕΦ μπορεί να κάνει μια βέλτιστη κατανομή σε εύλογο χρόνο. Εάν ο ΕΦ δεν μπορούσε να υπολογίσει μια βέλτιστη απόφαση για οποιονδήποτε λόγο, υπολογίζεται μια υποβέλτιστη κατανομή. Σε έναν κατά προσέγγιση μηχανισμό, ο αλγόριθμος ΕΦ τερματίζεται μετά την εύρεση μιας καλής λύσης, δηλαδή, δεν ψάχνει ολόκληρο το χώρο λύσης. Μετά από αυτό, η λύση αξιολογείται από μια αντικειμενική λειτουργία. Με έναν ευρετικό τρόπο, οι αλγόριθμοι ΕΦ κάνουν λογικές υποθέσεις σχετικά με τις εργασίες και τους πόρους. Με αυτόν τον τρόπο, αυτοί οι αλγόριθμοι λαμβάνουν πιο προσαρμοστικές αποφάσεις που δεν περιορίζονται από τις υποθέσεις. Οι αλγόριθμοι σε μια στρατηγική που ξεκινά από τον αποστολέα λαμβάνουν αποφάσεις σχετικά με την άφιξη ή τη δημιουργία εργασιών, ενώ οι αλγόριθμοι σε μια στρατηγική που ξεκινά από τον δέκτη λαμβάνουν αποφάσεις ΕΦ σχετικά με την αναχώρηση των τελικών εργασιών. Σε μια συμμετρική στρατηγική, είτε ο αποστολέας είτε ο δέκτης λαμβάνει αποφάσεις εξισορρόπησης φορτίων [21].

**Round-Robin [23]:** Είναι ένας από τους πιο συχνά χρησιμοποιούμενους αλγόριθμους για την ΕΦ. Είναι πολύ απλός. Τα αιτήματα από διάφορους πελάτες διανέμονται στον διακομιστή εφαρμογών εκ περιτροπής. Ας υποθέσουμε ότι έχουμε τρεις διακομιστές εφαρμογών. Στη συνέχεια, ο πρώτος διακομιστής εφαρμογών θα ζητηθεί από τον πρώτο πελάτη, ο δεύτερος διακομιστής εφαρμογής θα ζητηθεί από τον δεύτερο πελάτη και ο τρίτος διακομιστής εφαρμογής θα ζητηθεί από τον τρίτο πελάτη.



Εικόνα 3.4 Παράδειγμα Round Robin

**Weighted Round Robin [24]:** Αυτός ο αλγόριθμος βασίζεται σε έναν απλό αλγόριθμο ΕΦ Round-Robin . Οι διακομιστές βαθμολογούνται με βάση τον σχετικό αριθμό αιτημάτων που μπορεί να επεξεργαστεί ο καθένας. Ο διαχειριστής εκχωρεί ένα βάρος σε κάθε διακομιστή εφαρμογών με βάση τα κριτήρια της επιλογής του που υποδεικνύει τη σχετική ικανότητα διαχείρισης κίνησης κάθε διακομιστή στο σύστημα. Όσοι έχουν υψηλότερες ικανότητες δέχονται και περισσότερα αιτήματα. Είναι καταλληλότερο για τη διάδοση των εισερχόμενων αιτημάτων πελατών σε ένα σύνολο διακομιστών που έχουν διαφορετικές δυνατότητες ή διαθέσιμους πόρους. Έτσι, για παράδειγμα, εάν ο διακομιστής εφαρμογών #1 είναι δύο φορές πιο ισχυρός από τον διακομιστή εφαρμογών #2 (και τον διακομιστή εφαρμογών #3), ο διακομιστής εφαρμογών #1 λαμβάνει μεγαλύτερο βάρος και οι διακομιστές εφαρμογών #2 και #3 έχουν χαμηλότερο βάρος. Εάν υπάρχουν πέντε (5) διαδοχικά αιτήματα πελάτη, τα δύο πρώτα (2) πηγαίνουν στον διακομιστή εφαρμογών #1, το τρίτο (3) πηγαίνει στον διακομιστή εφαρμογής #2, το τέταρτο (4) στον διακομιστή εφαρμογής #3. Το πέμπτο (5) αίτημα θα πήγαινε στη συνέχεια στον διακομιστή εφαρμογής #1 και ούτω καθεξής.

**Αλγόριθμος Min-Min [22]:** Σε αυτόν τον αλγόριθμο υπολογίζεται πρώτα ο ελάχιστος χρόνος ολοκλήρωσης όλων των κόμβων. Επιλέγεται μια εργασία με ελάχιστο χρόνο ολοκλήρωσης και εκχωρείται στον αντίστοιχο κόμβο. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να ανατεθούν όλες οι εργασίες στους κόμβους.

**Αλγόριθμος Max-Min [22]:** Λειτουργεί σε αντίθετη στρατηγική σε σύγκριση με το Min-Min όπου μια εργασία με μέγιστο χρόνο ολοκλήρωσης λαμβάνεται



πρώτα υπόψη. Σε αυτόν τον αλγόριθμο υπολογίζεται πρώτα ο αναμενόμενος χρόνος ολοκλήρωσης όλων των εργασιών και επιλέγεται μια εργασία που απαιτεί μέγιστο χρόνο για την ολοκλήρωση.

**Οπορτουλιστικός αλγόριθμος εξισορρόπησης φορτίων [22]:** Αυτός ο αλγόριθμος δεν υπολογίζει το χρόνο εκτέλεσης και το τρέχον φορτίο του κόμβου. Αναθέτει τις εργασίες τυχαία στους κόμβους. Η επεξεργασία της εργασίας διαρκεί πολύ επειδή δεν υπολογίζει το χρόνο εκτέλεσης του κόμβου.

**Active Monitoring Load Balancing [22]:** Ο αλγόριθμος AMLB διατηρεί έναν πίνακα πληροφοριών για κάθε εικονική μηχανή (VM), συμπεριλαμβανομένων των τύπων αιτημάτων που της έχουν εκχωρηθεί επί του παρόντος. Όταν φθάνει ένα νέο αίτημα, το σύστημα ελέγχει ποια εικονική μηχανή έχει το χαμηλότερο φόρτο εργασίας και εκχωρεί το αίτημα σε αυτό το VM. Εάν φθάσει ένα αίτημα για την αντικατάσταση μιας εικονικής μηχανής, το σύστημα προσδιορίζει την εικονική μηχανή με το χαμηλότερο φόρτο εργασίας και την εκχωρεί ως αντικατάσταση. Στη συνέχεια, το πρόγραμμα εξισορρόπησης φορτίου παρέχει το αναγνωριστικό του εκχωρημένου VM στον Ελεγκτή του Κέντρου Πληροφοριών, ο οποίος στέλνει το αίτημα σε αυτό το VM και ειδοποιεί το Active VM Load Balancer για τη νέα κατανομή. Ωστόσο, το σύστημα λαμβάνει υπόψη μόνο το τρέχον φορτίο ενός VM και όχι την επεξεργαστική του ισχύ όταν εκχωρεί ένα νέο αίτημα. Επομένως, ο χρόνος αναμονής για ορισμένες εργασίες μπορεί να αυξηθεί, γεγονός που παραβιάζει την απαίτηση ποιότητας υπηρεσίας (QoS).

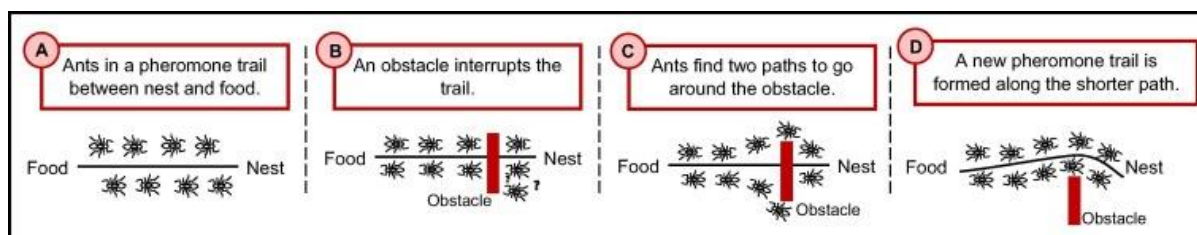
### 3.4.2 Δυναμικοί Αλγόριθμοι

Οι δυναμικοί αλγόριθμοι χωρίζονται σε δύο κατηγορίες, κατανεμημένοι και μη κατανεμημένοι αλγόριθμοι. Στην κατανεμημένη προσέγγιση, όλοι οι κόμβοι εκτελούν τον αλγόριθμο δυναμικής ΕΦ στο σύστημα και το έργο της ΕΦ μοιράζεται μεταξύ τους. Οι αλληλεπιδράσεις των κόμβων του συστήματος λαμβάνουν δύο μορφές, συνεταιριστική και μη συνεταιριστική. Στη μορφή συνεταιρισμού, οι κόμβοι συνεργάζονται για να επιτύχουν έναν κοινό στόχο, για παράδειγμα, τη μείωση του χρόνου απόκρισης όλων των εργασιών. Στην μη συνεταιριστική μορφή, κάθε κόμβος λειτουργεί ανεξάρτητα για να επιτύχει έναν τοπικό στόχο, για παράδειγμα, να μειώσει το χρόνο απόκρισης μιας τοπικής εργασίας.

Οι μη κατανεμημένοι αλγόριθμοι χωρίζονται σε δύο κατηγορίες, κεντρικοποιημένοι και ημικατανεμημένοι. Στην κεντρική μορφή, ένας ενιαίος κόμβος που ονομάζεται κεντρικός κόμβος εκτελεί τους αλγόριθμους ΕΦ και είναι πλήρως υπεύθυνος για την εξισορρόπηση φορτίου. Οι άλλοι κόμβοι αλληλεπιδρούν με τον κεντρικό κόμβο. Στην ημικατανεμημένη προσέγγιση, οι κόμβοι του συστήματος χωρίζονται σε συστάδες και κάθε συστάδα είναι κεντρικής μορφής. Οι κεντρικοί κόμβοι των συστάδων επιτυγχάνουν ΕΦ του συστήματος. Μερικά παραδείγματα δυναμικών αλγορίθμων είναι τα εξής:

**Dynamic Round Robin:** Είναι παρόμοιος με τον αλγόριθμο Weighted Round Robin, ωστόσο, ένα βάρος εκχωρείται για κάθε διακομιστή δυναμικά, με βάση τα δεδομένα σε πραγματικό χρόνο του τρέχοντος φορτίου και χωρητικότητας του διακομιστή. Τα βάρη αλλάζουν συνεχώς.

**Ant Colony Optimization technique (Τεχνική βελτιστοποίησης αποικίας μυρμηγκιών) [25]:** Αυτός ο αλγόριθμος βασίζεται στη φύση και τις κοινωνίες των πραγματικών μυρμηγκιών που σχηματίζουν ένα δίκτυο για αναζήτηση



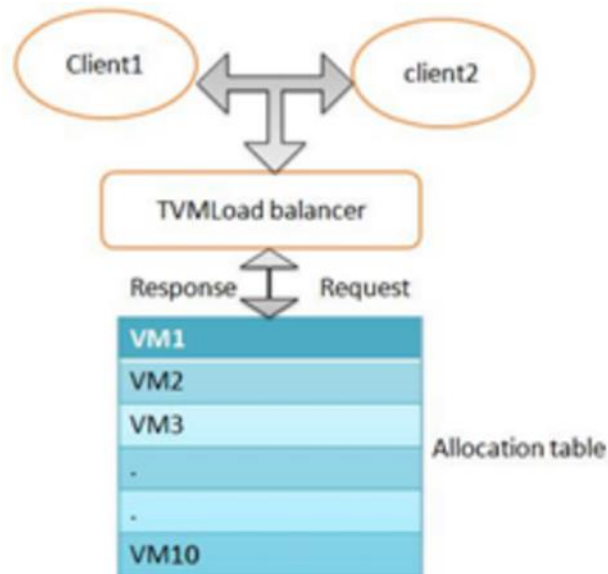
Εικόνα 3.5 Τεχνική βελτιστοποίησης αποικίας μυρμηγκιών (πηγή: [2])

τροφής. Σε αυτόν τον αλγόριθμο, όταν εκκινείται ένα αίτημα, το μυρμήγκι ξεκινά την κίνησή του προς τα εμπρός και επισκέπτεται τους κόμβους έναν προς έναν και ελέγχοντας εάν ένας κόμβος είναι υπερφορτωμένος ή υποφορτωμένος και καταγράφει τα δεδομένα. Εάν το μυρμήγκι βρει έναν υπερφορτωμένο κόμβο, αρχίζει να κινείται προς τα πίσω στον προηγούμενο κόμβο υποφορτωμένο κόμβο για να μοιραστεί δεδομένα.

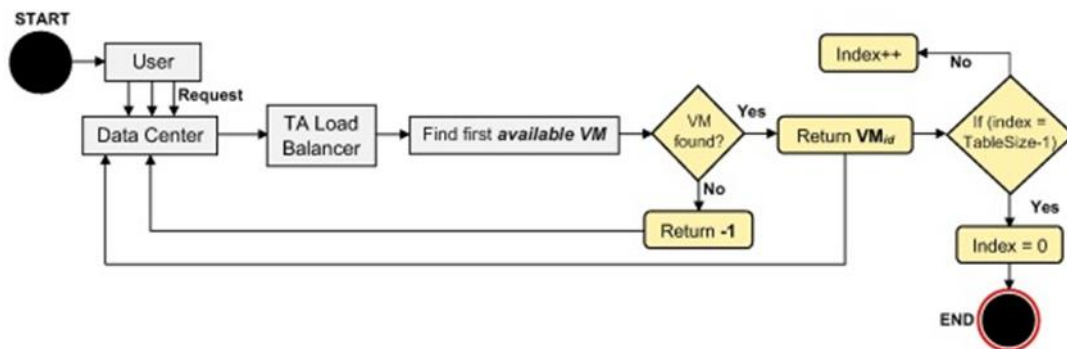
**Honey Bee Foraging Algorithm (Αλγόριθμος τροφοδοσίας μελισσών) [25]:** Είναι μια αποκεντρωμένη μέθοδος ΕΦ εμπνευσμένη από τη φύση και έχει σχεδιαστεί με βάση τη συμπεριφορά των μελισσών. Αυτός ο αλγόριθμος βοηθά στην εξισορρόπηση του φορτίου στους ετερογενείς κόμβους του Cloud. Σε αυτόν τον αλγόριθμο υπολογίζεται το πρώτο τρέχον φορτίο των κόμβων και στη συνέχεια αποφασίζεται εάν ο κόμβος είναι υπερφορτωμένος,

υποφορτωμένος ή ισορροπημένος. Μια εργασία από τον κόμβο με βαριά φόρτωση αφαιρείται και λαμβάνοντας υπόψη την προτεραιότητά του ανατίθεται σε έναν ελαφρά φορτισμένο κόμβο.

**Throttled Load Balancer [26]:** Σύμφωνα με τους Alkhatib κ.ά., η εκτέλεση του αλγορίθμου δημιουργεί ένα πίνακα που περιλαμβάνει τις εικονικές μηχανές VMs καθώς και την υπάρχουσα κατάσταση τους (διαθέσιμο/απασχολημένο). Εάν μια συγκεκριμένη εργασία εκχωρηθεί σε μια εικονική μηχανή, υποβάλλεται αίτημα στη μονάδα ελέγχου εντός του κέντρου δεδομένων, η οποία θα αναζητήσει την ιδανική εικονική μηχανή σε σχέση με τις ικανότητές της για να επιτύχουν την απαιτούμενη εργασία. Στην Εικόνα 3.6 παρουσιάζεται ο τρόπος λειτουργίας του Throttled Load Balancer. Αντίθετα, η διαδικασία αναζήτησης των ιδανικών εικονικών μηχανών πραγματοποιείται πάντα από την αρχή του πίνακα κάθε φορά. Επομένως, ορισμένα VM δεν χρησιμοποιούνται. Μετά τη λήψη του επόμενου αιτήματος, το VM στο ευρετήριο δίπλα στο VM που έχει ήδη εκχωρηθεί επιλέγεται με βάση την κατάσταση του VM. Επιπλέον, παρουσιάστηκε ένας αποτελεσματικός αλγόριθμος Throttled, ο οποίος αποτελούνταν από 3 αλγόριθμους, δηλαδή τον αλγόριθμο Throttled, τον ESCE (Equally Spread Current Execution algorithm) και τον Round Robin. Αυτός ο αλγόριθμος θεωρείται ως μια πρόοδος του κύριου αλγορίθμου Throttled, όπου ο βελτιωμένος αλγόριθμος χρησιμοποίησε τη δομή δεδομένων για να διατηρήσει πληροφορίες σχετικά με τα VM. Αντίθετα, το ευρετήριο Hash Map χρησιμοποιείται για την αναζήτηση VM και εκχωρημένες εργασίες όπου λειτουργεί με ταχύτερο ρυθμό σε σύγκριση με τον αλγόριθμο Throttled. Επιπλέον, η προσέγγιση Divide-and-Conquer and Throttled algorithm (DCBT) θεωρείται η υβριδική προσέγγιση επειδή δίνει προτίμηση στη σειρά κατά την κατανομή του VM. Αυτό γίνεται για να επιτευχθεί η υψηλότερη χρήση πόρων μειώνοντας τον συνολικό χρόνο που απαιτείται για την εκτέλεση της εργασίας. Επιπρόσθετα, αυτός ο αλγόριθμος επιδιώκει να ενημερώσει τον πίνακα διαιρώντας τις ανεξάρτητες εργασίες εξίσου μεταξύ όλων των VM. Ως εκ τούτου, λειτουργεί πιο γρήγορα από τον αλγόριθμο Throttled.



Εικόνα 3.6. Throttled Load Balancer (πηγή: [26])



Εικόνα 3.7 Process of Throttled Algorithm (πηγή: [2])

**Ενεργός αλγόριθμος ομαδοποίησης [25]:** Είναι μια βελτιωμένη έκδοση της τυχαίας δειγματοληψίας. Χρησιμοποιεί την έννοια της ομαδοποίησης. Η κύρια αρχή αυτού του αλγορίθμου είναι η ομαδοποίηση παρόμοιων κόμβων μαζί και η εργασία με βάση τους ομαδοποιημένους κόμβους. Ένας κόμβος που ξεκινά τη διαδικασία και επιλέγει μια άλλη διαδικασία ονομάζεται κόμβος δημιουργίας αντιστοίχισης. Ένας κόμβος δημιουργίας αντιστοίχισης συνδέει έναν αρχικό κόμβο με τον γείτονα και αυτός αποσυνδέεται. Αυτή η διαδικασία γίνεται επαναληπτικά για να εξισορροπηθεί το φορτίο.

**First Come First Serve [26]:** Αυτός ο αλγόριθμος λειτουργεί κατανέμοντας νέες εργασίες σε πόρους που έχουν το λιγότερο χρόνο αναμονής (δηλαδή πόρους που έχουν τον μικρότερο αριθμό εργασιών). Εδώ, υπάρχει διαδοχική εκτέλεση των εργασιών, με την εργασία να ξεκινά από την πρώτη εργασία μέχρι την ολοκλήρωσή της, μετά την οποία εκτελείται η επόμενη εργασία από την ουρά. Αυτός ο αλγόριθμος χρησιμοποιείται καθώς κατανέμει εργασίες στις εικονικές μηχανές χωρίς να λαμβάνει υπόψη τις προδιαγραφές των διαθέσιμων εικονικών μηχανών και επίσης τον χρόνο που θα χρειαστούν οι εργασίες στην ουρά κατά την κατανομή νέων εργασιών.

**Biased Random Sampling (Μεροληπτικός αλγόριθμος τυχαίας δειγματοληψίας):** Πρόκειται για έναν δυναμικό αλγόριθμο ΕΦ. Χρησιμοποιεί τυχαία δειγματοληψία του τομέα του συστήματος για να επιτύχει αυτό-οργάνωση, εξισορροπώντας έτσι το φορτίο σε όλους τους κόμβους του συστήματος. Σε αυτόν τον αλγόριθμο, κατασκευάζεται ένα εικονικό γράφημα με τη συνδεσιμότητα κάθε κόμβου που αντιπροσωπεύει το φορτίο στον διακομιστή. Κάθε κόμβος αναπαρίσταται ως κορυφή σε ένα κατευθυνόμενο γράφημα και κάθε σε βαθμό αντιπροσωπεύει τους ελεύθερους πόρους αυτού του κόμβου. Κάθε φορά που ένας πελάτης στέλνει ένα αίτημα στον ΕΦ, ο ΕΦ εκχωρεί την εργασία στον κόμβο που έχει τουλάχιστον έναν βαθμό. Μόλις μια εργασία εκχωρηθεί στον κόμβο, ο βαθμός αυτού του κόμβου μειώνεται κατά ένα. Αφού η δουλειά ολοκληρωθεί, ο κόμβος δημιουργεί ένα εισερχόμενο άκρο και αυξάνει τον βαθμό κατά ένα. Η προσθήκη και η διαγραφή διεργασιών γίνεται με τη διαδικασία της τυχαίας δειγματοληψίας. Κάθε διεργασία χαρακτηρίζεται από μια παράμετρο γνωστή ως τιμή κατωφλίου, η οποία υποδεικνύει το μέγιστο μήκος βαδίσματος. Ο περίπατος ορίζεται ως η διέλευση από τον έναν κόμβο στον άλλο μέχρι να βρεθεί ο προορισμός. Σε κάθε βήμα της διαδρομής, ο γειτονικός κόμβος του τρέχοντος κόμβου επιλέγεται ως ο επόμενος κόμβος. Σε αυτόν τον αλγόριθμο, μόλις λάβει το αίτημα από τον ΕΦ, θα επέλεγε έναν κόμβο τυχαία και θα συγκρίνει το τρέχον μήκος διαδρομής με την τιμή κατωφλίου. Εάν το τρέχον μήκος βαδίσματος είναι ίσο ή μεγαλύτερο από την τιμή κατωφλίου, η εργασία εκτελείται σε αυτόν τον κόμβο.

Διαφορετικά, το μήκος βόλτας της εργασίας αυξάνεται και ένας άλλος γειτονικός κόμβος επιλέγεται τυχαία. Η απόδοση υποβαθμίζεται καθώς ο αριθμός των διακομιστών αυξάνεται λόγω πρόσθετων επιβαρύνσεων για τον υπολογισμό του μήκους της διαδρομής.

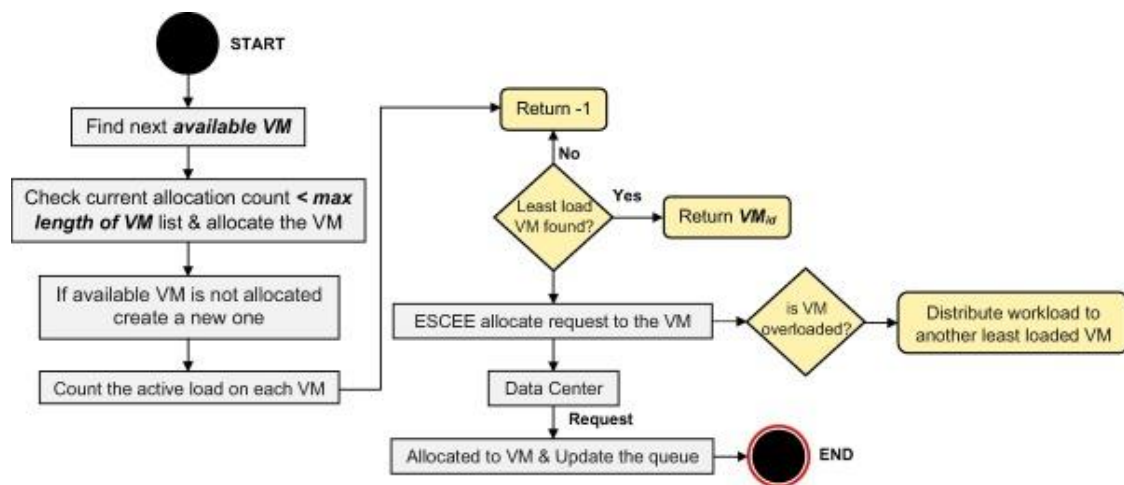
**Least Connection Load Balancing method [24]:** Η ΕΦ ελάχιστης σύνδεσης είναι ένας δυναμικός αλγόριθμος όπου τα αιτήματα πελάτη διανέμονται στον διακομιστή εφαρμογής με τον λιγότερο αριθμό ενεργών συνδέσεων τη στιγμή που λαμβάνεται το αίτημα πελάτη. Σε περιπτώσεις όπου οι διακομιστές εφαρμογών έχουν παρόμοιες προδιαγραφές, ένας διακομιστής μπορεί να υπερφορτωθεί λόγω των συνδέσεων μεγαλύτερης διάρκειας. Αυτός ο αλγόριθμος λαμβάνει υπόψη το ενεργό φορτίο σύνδεσης. Αυτή η τεχνική είναι κατάλληλη για εισερχόμενα αιτήματα που έχουν διαφορετικούς χρόνους σύνδεσης και ένα σύνολο διακομιστών που είναι σχετικά παρόμοιοι από την άποψη της επεξεργαστικής ισχύος και των διαθέσιμων πόρων.

**Weighted Least Connections [24]:** Η βαθμολογία του διακομιστή βασίζεται πλήρως στην ικανότητα επεξεργασίας. Η σχετική χωρητικότητα διακομιστών και ο αριθμός των ενεργών συνδέσεων είναι παράγοντες σύμφωνα με τους οποίους βαθμολογούνται οι διακομιστές. Ο διαχειριστής εκχωρεί ένα βάρος σε κάθε διακομιστή εφαρμογών με βάση τη σχετική ισχύ επεξεργασίας και τους διαθέσιμους πόρους κάθε διακομιστή. Το LoadMaster<sup>7</sup> λαμβάνει αποφάσεις ΕΦ με βάση τις ενεργές συνδέσεις και τα εκχωρημένα βάρη διακομιστή (π.χ., εάν υπάρχουν δύο διακομιστές με τον μικρότερο αριθμό συνδέσεων, επιλέγεται ο διακομιστής με το μεγαλύτερο βάρος).

---

<sup>7</sup> Το LoadMaster υποστηρίζει ένα πλούσιο σύνολο τεχνικών που κυμαίνονται από την απλή εξισορρόπηση φορτίου κυκλικής ροής έως την προσαρμοστική εξισορρόπηση φορτίου που ανταποκρίνεται στις πληροφορίες κατάστασης που ανακτώνται από τη φάρμα διακομιστών.

**ESCE [2]:** Ο Equally Spread Current Execution (ESCE) είναι ένας δυναμικός αλγόριθμος. Δίνει προτεραιότητα στο μέγεθος της εργασίας και τη διανέμει τυχαία σε μια εικονική μηχανή (VM) με μικρότερο φορτίο. Αυτή η τεχνική είναι επίσης γνωστή ως Spread Spectrum, καθώς κατανέμει το φόρτο εργασίας σε διάφορους κόμβους. Ο αλγόριθμος ESCE χρησιμοποιεί μια ουρά για την αποθήκευση αιτημάτων και τη διανομή του φορτίου σε VM που δεν είναι υπερφορτωμένα. Ωστόσο, το ESCE μπορεί να προκαλέσει επιβάρυνση κατά την ενημέρωση του πίνακα ευρετηρίων λόγω επικοινωνίας μεταξύ του ελεγκτή του Κέντρου δεδομένων και του εξισορροπητή φορτίου.



Εικόνα 3.8 Διάγραμμα Ροής ESCE (πηγή: [2])

### 3.4.3 Άλλοι Υφιστάμενοι Αλγόριθμοι

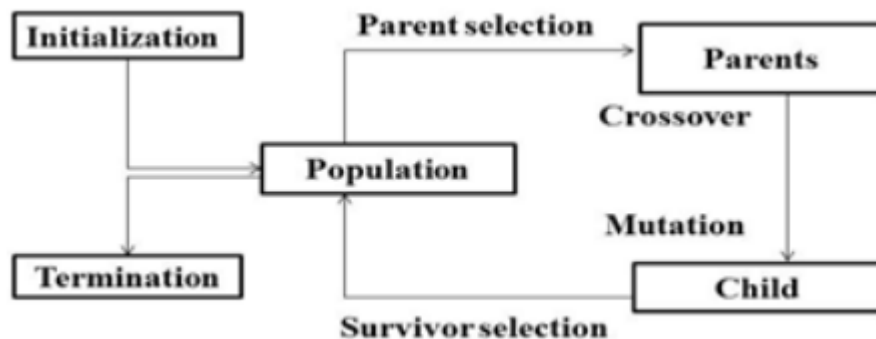
**Particle Swarm Optimization (PSO) [27]:** Οι κοινωνιοβιολόγοι Kennedy και Eberhart εισήγαγαν τον αλγόριθμο PSO το 1995, πρόκειται για μια τεχνική βελτιστοποίησης βασισμένη στο Swarm Intelligence που προσομοιώνει τη συμπεριφορά της εκτροφής ψαριών και των πτηνών. Όπως αναφέρθηκε στην αρχική εργασία, οι κοινωνιοβιολόγοι πιστεύουν ότι ένα κοπάδι ψαριών ή ένα κοπάδι πουλιών που κινείται σε μια ομάδα «μπορεί να επωφεληθεί από την εμπειρία όλων των άλλων μελών». Με άλλα λόγια, ενώ ένα πουλί πετά και ψάχνει τυχαία για τροφή, για παράδειγμα, όλα τα πουλιά στο κοπάδι μπορούν να μοιραστούν την ανακάλυψή τους και να βοηθήσουν ολόκληρο το κοπάδι να πάρει το καλύτερο κυνήγι. Ενώ μπορούμε να προσομοιώσουμε την κίνηση ενός κοπαδιού πουλιών, μπορούμε επίσης να φανταστούμε ότι κάθε πουλί θα μας

βοηθά να βρούμε τη βέλτιστη λύση σε έναν χώρο λύσης υψηλών διαστάσεων και η καλύτερη λύση που βρέθηκε από το κοπάδι είναι η καλύτερη λύση στο χώρο. Αυτή είναι μια ευρετική λύση γιατί δεν μπορούμε ποτέ να αποδείξουμε ότι μπορεί να βρεθεί η πραγματική παγκόσμια βέλτιστη λύση και συνήθως δεν είναι. Ωστόσο, συχνά διαπιστώνουμε ότι η λύση που βρέθηκε από τον PSO είναι αρκετά κοντά στο ολικό βέλτιστο. Ο βασικός αλγόριθμος PSO αποτελείται από τη δημιουργία ταχυτήτων και θέσεων για κάθε σωματίδιο, τον υπολογισμό των τιμών καταλληλότητας κάθε σωματιδίου και την ενημέρωση της θέσης και της ταχύτητας σε κάθε επανάληψη. Ένα σωματίδιο εδώ υποδηλώνει ένα σημείο στο χώρο αναζήτησης. Ανάλογα με τις ενημερώσεις ταχύτητας, το σωματίδιο αλλάζει τη θέση του από τη μια κίνηση στην άλλη. Σε αυτόν τον αλγόριθμο ο νέος χώρος αναζήτησης επηρεάζεται από τις παραμέτρους ελέγχου και συγκεκριμένα, το βάρος αδράνειας ( $w$ ), τον αριθμός πληθυσμού, τον αριθμό επαναλήψεων, τον παράγοντα αυτοπεποίθησης  $c1$  και τον συντελεστή εμπιστοσύνης σμήνος  $c2$ .

**Genetic Algorithm [26]:** Στη διαδικασία των γενετικών αλγορίθμων, ο προγραμματισμός πραγματοποιείται με βάση τη βιολογική έννοια της παραγωγής πληθυσμού. Ο σκοπός της χρήσης αυτού του αλγορίθμου ήταν να βελτιωθεί η κατανομή του φορτίου εντός του Cloud, όπου ο αλγόριθμος αρχίζει να λειτουργεί χρησιμοποιώντας την προκαταρκτική διαδικασία πληθυσμού. Ο αρχικός πληθυσμός αποτελείται από το σύνολο όλων των ατόμων που εμπλέκονται στον προσδιορισμό της βέλτιστης λύσης. Κάθε διάλυμα στον πληθυσμό αναφέρεται ως άτομο και κάθε άτομο περιγράφεται ως χρωμόσωμα για να γίνει κατάλληλο για τη διεξαγωγή γενετικών επεμβάσεων. Οι συγγραφείς του άρθρου υποθέτουν ότι θα πραγματοποιηθεί σύγκριση του πρόσφατα αναπτυγμένου πληθυσμού με τον προηγούμενο. Στη συνέχεια επιλέγονται λύσεις για να ληφθούν αποφάσεις (απόγονοι) με βάση τη συνάρτηση καταλληλότητας. Η συνάρτηση καταλληλότητας χρησιμοποιείται για τον προσδιορισμό της ποιότητας των ατόμων στον πληθυσμό σε σχέση με τον καθορισμένο στόχο βελτιστοποίησης. Για κάθε χρωμόσωμα, υπολογίζεται η συνάρτηση καταλληλότητας, με βάση την οποία επιλέγονται τα καταλληλότερα αποτελέσματα που θα χρησιμοποιηθούν ως γονείς. Στη συνέχεια, η λειτουργία ξεκινά με μια διαδικασία διασταύρωσης. Ένα μέρος κάθε γονέα



χρησιμοποιείται για τη δημιουργία ενός νέου παιδιού. Αυτό το παιδί στη συνέχεια βελτιώνεται χρησιμοποιώντας μια διαδικασία μετάλλαξης. Αυτή η διαδικασία συνεχίζεται μέχρι να επιτευχθούν τα καλύτερα αποτελέσματα. Η βάση των γενετικών αλγορίθμων είναι η τυχειότητα. Ωστόσο, δεν είναι το ίδιο με την τυχαία αναζήτηση στο ότι εγκρίνει τα καταλληλότερα άτομα σε έναν πληθυσμό. Ο ρυθμός διασταύρωσης και οι τιμές πιθανότητας μετάλλαξης έχουν σημαντικό αντίκτυπο στην απόδοση του γενετικού αλγορίθμου. Το γενικό σχήμα του γενετικού αλγορίθμου φαίνεται στο επόμενο σχήμα.



Εικόνα 3.9 Γενικό σχήμα Γενετικού Αλγορίθμου (πηγή: [26])

**Resource Based (Adaptive) load balancing method [24]:** Η εξισορρόπηση φορτίου βάσει πόρων (ή προσαρμοστικής) λαμβάνει αποφάσεις με βάση δείκτες κατάστασης που ανακτά το LoadMaster από τους διακομιστές υποστήριξης. Η ένδειξη κατάστασης καθορίζεται από ένα προσαρμοσμένο πρόγραμμα (ένας «πράκτορας») που εκτελείται σε κάθε διακομιστή. Το LoadMaster υποβάλλει τακτικά ερωτήματα σε κάθε διακομιστή για αυτές τις πληροφορίες κατάστασης και στη συνέχεια ορίζει κατάλληλα το δυναμικό βάρος του πραγματικού διακομιστή.

Με αυτόν τον τρόπο, η μέθοδος εξισορρόπησης φορτίου εκτελεί ουσιαστικά έναν λεπτομερή «έλεγχο υγείας» στον πραγματικό διακομιστή. Η συγκεκριμένη μέθοδος είναι κατάλληλη σε κάθε περίπτωση όπου απαιτούνται λεπτομερείς πληροφορίες ελέγχου υγείας από κάθε διακομιστή για τη λήψη αποφάσεων εξισορρόπησης φορτίου. Για παράδειγμα, αυτή η μέθοδος θα ήταν χρήσιμη για

οποιαδήποτε εφαρμογή όπου ο φόρτος εργασίας ποικίλλει και απαιτείται λεπτομερής απόδοση και κατάσταση της εφαρμογής για την αξιολόγηση της υγείας του διακομιστή. Αυτή η μέθοδος μπορεί επίσης να χρησιμοποιηθεί για την παροχή υγειονομικού ελέγχου με γνώση της εφαρμογής για υπηρεσίες επιπέδου 4 (UDP) μέσω της μεθόδου εξισορρόπησης φορτίου.

**Resource Based (SDN Adaptive) load balancing method [24]:** Το SDN adaptive είναι ένας αλγόριθμος εξισορρόπησης φορτίου που συνδυάζει γνώσεις από τα επίπεδα 2, 3, 4 και 7 και είσοδο από έναν ελεγκτή SDN για τη λήψη πιο βελτιστοποιημένων αποφάσεων διανομής κυκλοφορίας. Αυτό επιτρέπει την παροχή πληροφοριών σχετικά με την κατάσταση των διακομιστών, την κατάσταση των εφαρμογών που εκτελούνται σε αυτούς, την υγεία της υποδομής δικτύου και το επίπεδο συμφόρησης στο δίκτυο να παίζουν ρόλο στη λήψη αποφάσεων εξισορρόπησης φορτίου. Αυτή η μέθοδος είναι κατάλληλη για αναπτύξεις που περιλαμβάνουν ελεγκτή SDN.

**Weighted Response Time load balancing method [24]:** Ο σταθμισμένος αλγόριθμος εξισορρόπησης φόρτου χρόνου απόκρισης χρησιμοποιεί το χρόνο απόκρισης του διακομιστή εφαρμογών για τον υπολογισμό του βάρους του διακομιστή. Ο διακομιστής εφαρμογών που ανταποκρίνεται ταχύτερα λαμβάνει το επόμενο αίτημα. Αυτός ο αλγόριθμος είναι κατάλληλος για σενάρια όπου ο χρόνος απόκρισης της εφαρμογής είναι το πρωταρχικό μέλημα.

**Κατακερματισμός IP προέλευσης [24]:** Σε αυτόν τον αλγόριθμο, η διεύθυνση IP προέλευσης και προορισμού συνδυάζονται σε ένα αίτημα για τη δημιουργία ενός κλειδιού κατακερματισμού, το οποίο στη συνέχεια ορίζεται σε έναν συγκεκριμένο διακομιστή. Αυτό επιτρέπει μια διακοπείσα σύνδεση να επιστραφεί στον ίδιο διακομιστή που την χειριζόταν αρχικά.

**URL Hash load balancing method [24]:** Ο αλγόριθμος ΕΦ κατακερματισμού URL είναι παρόμοιος με τον κατακερματισμό IP πηγής, με τη διαφορά ότι ο κατακερματισμός που δημιουργήθηκε βασίζεται στη διεύθυνση URL στο αίτημα πελάτη. Αυτό διασφαλίζει ότι τα αιτήματα πελατών σε μια συγκεκριμένη διεύθυνση URL αποστέλλονται πάντα στον ίδιο διακομιστή υποστήριξης.

**Generalized Priority Algorithm [26]:** Αυτός ο αλγόριθμος λειτουργεί με ιεράρχηση εργασιών με βάση το μέγεθός τους, ενώ σε μεγαλύτερες εργασίες δίνεται μεγαλύτερη προτεραιότητα. Επιπλέον, τα προεπιλεγμένα μηχανήματα έχουν προτεραιότητα με βάση την ισχύ του επεξεργαστή τους. Στη συνέχεια, ο αλγόριθμος επιλέγει την κατάλληλη εικονική μηχανή με βάση την προτεραιότητα της εργασίας. Για παράδειγμα, εάν εξετάσουμε το παράδειγμα έξι εικονικών μηχανών (VM) που αντιπροσωπεύονται από τα αναγνωριστικά και τις ταχύτητες επεξεργαστή τους, VMs = 0, 850, 1, 1000, 5, 3, 150, 4, 250, 5, 750. Παρατηρούμε ότι, η μεγαλύτερη προτεραιότητα δίνεται στο VM 4 γιατί έχει τον ταχύτερο επεξεργαστή. Μετά το VM 4, προτεραιότητα δίνεται στο VM 2, ακολουθούμενο από το VM 6 και ούτω καθεξής.

### 3.5 Πολιτικές Δυναμικού Αλγόριθμου Εξισορρόπησης Φορτίου

Οι αλγόριθμοι εξισορρόπησης δυναμικού φορτίου αποτελούνται από 6 κύριες πολιτικές [13], [22] :

- **Πολιτική μεταφοράς**

Απαιτείται όταν οποιαδήποτε εργασία πρέπει να μεταφερθεί από έναν κόμβο σε άλλο. Πιο συγκεκριμένα, αυτή η πολιτική ανακαλύπτει τις συνθήκες υπό τις οποίες απαιτείται η μεταφορά εργασιών από έναν τοπικό κόμβο σε έναν άλλο τοπικό ή απομακρυσμένο κόμβο. Αποτελείται από δύο προσεγγίσεις. Όλες τις τρέχουσες εργασίες και την τελευταία ληφθείσα εργασία για τον προσδιορισμό των εργασιών που πρόκειται να μεταφερθούν. Στην τελευταία ληφθείσα προσέγγιση, όλες οι εισερχόμενες εργασίες εισέρχονται στην πολιτική μεταφοράς και η εργασία που φτάνει τελευταία θα μεταφερθεί. Σύμφωνα με όλες τις τρέχουσες προσεγγίσεις εργασιών, η πολιτική μεταφοράς που βασίζεται σε έναν κανόνα αποφασίζει ότι μια εργασία πρέπει να μεταφερθεί (μεταφορά εργασιών) ή την επεξεργάζεται τοπικά (αναπρογραμματισμός εργασιών) και εξαρτάται από τον φόρτο εργασίας κάθε κόμβου.

- **Πολιτική επιλογής**

Αυτή η πολιτική προσδιορίζει τις εργασίες που πρέπει να μεταφερθούν από έναν κόμβο στον άλλο. Επιλέγει τις εργασίες με βάση το ποσό των γενικών εξόδων που

απαιτούνται για τη μετεγκατάσταση, τον αριθμό των μη τοπικών κλήσεων συστήματος και τον χρόνο που απαιτείται για την εκτέλεση της εργασίας. Εδώ δίνεται ιδιαίτερη σημασία στην επιλογή του σωστού επεξεργαστή για μετατόπιση φορτίου, έτσι ώστε να αυξηθεί η απόδοση και ο χρόνος απόκρισης.

- **Πολιτική τοποθεσίας**

Είναι ο εντοπισμός κρίσιμων πόρων για την προσφορά υπηρεσιών και την επιλογή θέσεων πόρων. Πιο αναλυτικά αυτή η πολιτική καθορίζει τους υπολογιστικούς κόμβους που είναι υποφορτωμένοι ή ελεύθεροι και μεταφέρει εργασίες σε αυτούς για επεξεργασία. Επιλέγει τον κόμβο προορισμού αφού προσδιορίσει τη διαθεσιμότητα των απαιτούμενων υπηρεσιών για τη μετεγκατάσταση εργασιών με βάση τις διαθέσιμες προσεγγίσεις: ανίχνευση, διαπραγμάτευση και τύχη. Στην τυχαία προσέγγιση, η πολιτική τοποθεσίας επιλέγει τυχαία τον προορισμό και μεταφέρει τις εργασίες. Στην προσέγγιση ανίχνευσης, ένας κόμβος διερευνά άλλους κόμβους του συστήματος για να επιλέξει τον προορισμό. Τέλος, στην προσέγγιση της διαπραγμάτευσης οι κόμβοι διαπραγματεύονται μεταξύ τους για εξισορρόπηση φορτίου.

- **Πολιτική πληροφοριών**

Αυτή είναι μια άλλη πολιτική δυναμικής εξισορρόπησης φορτίου που διατηρεί όλες τις πληροφορίες πόρων στο σύστημα, οι οποίες χρησιμοποιούνται περαιτέρω από άλλες πολιτικές για τη λήψη των αποφάσεών τους. Αποφασίζει τον χρόνο συλλογής πληροφοριών. Διάφορες μέθοδοι συλλογής πληροφοριών από τους κόμβους είναι η Agent, η Broadcasting και η Centralized polling. Στη μέθοδο μετάδοσης, όλοι οι κόμβοι εκπέμπουν τις πληροφορίες τους, οι οποίες είναι προσβάσιμες από άλλους κόμβους. Η μέθοδος Agent χρησιμοποιείται αυτήν τη στιγμή από κόμβους για τη συλλογή πληροφοριών. Διάφορες πολιτικές πληροφόρησης είναι η πολιτική βάσει ζήτησης, η περιοδική πολιτική και η πολιτική που βασίζεται στην αλλαγή κατάστασης.

- **Πολιτική εκτίμησης φορτίου**

Εδώ υπολογίζεται το συνολικό φορτίο ενός κόμβου μέσα σε ένα σύστημα.

- **Πολιτική μεταφοράς διεργασιών**

Σύμφωνα με αυτήν την πολιτική, η εκτέλεση διεργασίας καθορίζεται είτε είναι τοπικά είτε απομακρυσμένα.

Πίνακας 3.1. Σύνοψη των πολιτικών εξισορρόπησης φορτίου (πηγή:[21])

Policy	Transfer Policy	Selection Policy	Location Policy	Information Policy
<b>Περιγραφή</b>	Περιλαμβάνει: <ul style="list-style-type: none"> <li>✓ Επαναπρογραμματισμός εργασιών</li> <li>✓ Μετανάστευση εργασιών</li> <li>✓ Βασισμένο στα κατώφλια ως προς τις μονάδες φορτίου</li> </ul>	Παράγοντες για την επιλογή μιας εργασίας προς μεταφορά: <ul style="list-style-type: none"> <li>• Γενικά έξοδα μετανάστευσης</li> <li>• Αριθμός απομακρυσμένων κλήσεων συστήματος</li> <li>• Χρόνος εκτέλεσης της εργασίας</li> </ul>	Εύρεση κατάλληλου συνεργάτη για εργασία μεταφοράς  Έλεγχος της διαθεσιμότητας των απαραίτητων υπηρεσιών για μετανάστευση εντός του εταίρου	Προσδιορισμός του χρόνου όταν οι πληροφορίες σχετικά με τους κόμβους πρέπει να συγκεντρωθεί  Τρεις τύποι πολιτικής πληροφορίας: <ol style="list-style-type: none"> <li>1. Demand-driven policy</li> <li>2. Periodic policies</li> <li>3. State-change driven policy</li> </ol>

Επειδή δεν απαιτείται ξεχωριστός κόμβος αφιερωμένος στην κατανομή εργασιών, ο δυναμικός σχεδιασμός εξισορρόπησης φορτίου μπορεί να είναι πιο ευέλικτος. Είναι μια μοναδική ανάθεση όταν οι εργασίες εκχωρούνται σε έναν επεξεργαστή με βάση την κατάστασή του σε μια δεδομένη στιγμή. Η δυναμική ανάθεση, από την άλλη πλευρά, αναφέρεται στη δυνατότητα ανακατανομής καθηκόντων σε συνεχή βάση με βάση την κατάσταση του συστήματος και την πρόοδό του. Προφανώς, ένας αλγόριθμος εξισορρόπησης φορτίου που απαιτεί υπερβολική επικοινωνία για τη λήψη των συμπερασμάτων του διατρέχει τον κίνδυνο να καθυστερήσει την επίλυση του γενικού προβλήματος .

### 3.6 Σύγκριση Αλγορίθμων

Σε αυτή την ενότητα, παρουσιάζεται μια σύγκριση αλγορίθμων εξισορρόπησης φορτίου. Κάθε αλγόριθμος προσδιορίζεται με συγκεκριμένες ιδιότητες και περιορισμούς. Ο πρώτος αλγόριθμος που μελετήθηκε είναι ο Round Robin, ο οποίος

είναι ένας στατικός αλγόριθμος χωρίς επιβάρυνση και απλή πολυπλοκότητα. Τα πλεονεκτήματά του περιλαμβάνουν την απλότητα και την ευκολία εγκατάστασης, αλλά έχει επίσης περιορισμούς, όπως η αδυναμία περαιτέρω βελτίωσης, η απουσία δυνατοτήτων πολλαπλών εργασιών, οι πιθανές επιβαρύνσεις και η αποτυχία να ληφθεί υπόψη η σημασία της χωρητικότητας και η διάρκεια της εργασίας. Το Round Robin έχει χαμηλή απόδοση και δεν προσφέρει ανοχή σε σφάλματα.

Ο Throttled είναι ένας στατικός αλγόριθμος χωρίς επιβάρυνση και μέτριο βαθμό πολυπλοκότητας. Η δύναμή του έγκειται στην ικανότητά του να επιτρέπει στο κέντρο δεδομένων να αναζητά την καταλληλότερη εικονική μηχανή που ταιριάζει με την απαιτούμενη εργασία, ενώ βελτιστοποιεί τη χρήση των πόρων και αυξάνει τον μέσο χρόνο εκτέλεσης. Ωστόσο, ο Throttled έχει περιορισμούς, καθώς δεν θέτει χρονικά όρια και μπορεί να οδηγήσει σε αχρησιμοποίητα VM. Η απόδοσή του είναι μέτρια, αλλά έχει υψηλή απόδοση και ανοχή σφαλμάτων. Συνολικά, ο Throttled είναι ένας χρήσιμος αλγόριθμος που μπορεί να βοηθήσει στη βελτιστοποίηση της χρήσης πόρων στα κέντρα δεδομένων.

Ο Min-Min είναι επίσης ένας στατικός αλγόριθμος με απλό βαθμό πολυπλοκότητας, αλλά έχει μεσαία επιβάρυνση. Η δύναμή του βασίζεται κυρίως στην ταχύτητα και την ευκολία εγκατάστασης. Παρ' όλα αυτά, ο Min-Min έχει περιορισμούς, καθώς μπορεί να οδηγήσει σε «λιμοκτονία». Αυτός ο αλγόριθμος μπορεί να βελτιώσει την απόδοση για μικρότερες εργασίες και να αυξήσει την απόδοση. Δυστυχώς, ο Min-Min δεν έχει ανοχή σφαλμάτων. Συνολικά, ο Min-Min μπορεί να είναι χρήσιμος σε ορισμένα σενάρια, αλλά είναι σημαντική η γνώση των περιορισμών του προκειμένου να διασφαλιστεί η επιτυχή εφαρμογή του.

Ο αλγόριθμος Max-Min είναι στατικός αλγόριθμος με απλό βαθμό πολυπλοκότητας και μέτρια επιβάρυνση. Ένα από τα δυνατά του σημεία είναι η ικανότητά του να εκτελεί μεγάλες εργασίες παράλληλα με σύντομες εργασίες, κάτι που μπορεί να οδηγήσει σε βελτιωμένη απόδοση σε σύγκριση με τον αλγόριθμο Min-Min. Ο συγκεκριμένος αλγόριθμος έχει περιορισμούς καθώς μπορεί να οδηγήσει σε «λιμοκτονία». Ο αλγόριθμος Max-Min δεν παρέχει ανοχή σφαλμάτων, πράγμα που σημαίνει ότι τυχόν σφάλματα ή αστοχίες που προκύψουν δεν θα αντιμετωπιστούν από τον αλγόριθμο. Συνολικά, ο αλγόριθμος Max-Min είναι ένα χρήσιμο εργαλείο για τον προγραμματισμό εργασιών και μπορεί να προσφέρει καλύτερη απόδοση από τον

αλγόριθμο Min-Min σε ορισμένες περιπτώσεις, αλλά θα πρέπει επίσης να ληφθούν υπόψη οι περιορισμοί του.

Ο ομορτυνιστικός αλγόριθμος είναι εξίσου ένας στατικός αλγόριθμος με απλό βαθμό πολυπλοκότητας και χαμηλό κόστος. Το πλεονέκτημά του έγκειται στην απλότητα και στην ικανότητά του να επιτυγχάνει ισορροπία φορτίου. Όμως, ο αλγόριθμος έχει αρκετούς περιορισμούς οι οποίοι θα πρέπει να ληφθούν υπόψη. Ένας από αυτούς είναι ότι δεν υπάρχει δικαιοσύνη στην ανάθεση εργασιών, κάτι που μπορεί να οδηγήσει σε ορισμένες εργασίες να ανατίθενται στον ίδιο κόμβο επανειλημμένα ενώ άλλες να μην ανατίθενται καθόλου. Επιπλέον, ο αλγόριθμος σχεδιάζει εργασίες αργά, επειδή δεν λαμβάνει υπόψη τον τρέχοντα χρόνο εκτέλεσης των κόμβων, γεγονός που μπορεί να οδηγήσει σε κακή απόδοση. Ολόκληρος ο χρόνος ολοκλήρωσης, γνωστός και ως χρόνος κατασκευής, είναι επίσης πολύ φτωχός. Ως αποτέλεσμα, ο αλγόριθμος έχει περιορισμένη απόδοση και κακή απόδοση. Τέλος, δεν υπάρχει ανοχή σφαλμάτων, που σημαίνει ότι τυχόν σφάλματα ή αστοχίες που προκύψουν δεν θα αντιμετωπιστούν από τον αλγόριθμο.

Ο αλγόριθμος AMLB (Active Monitoring Load Balancing) είναι ένας τύπος στατικού αλγορίθμου που αναγνωρίζεται για την υψηλή απόδοση και απόδοση, παρά το γεγονός ότι έχει υψηλό κόστος. Το επίπεδο πολυπλοκότητάς του είναι μέτριο και είναι ιδιαίτερα χρήσιμο για τον εντοπισμό της λιγότερο φορτωμένης εικονικής μηχανής (VM) και τη βελτιστοποίηση της απόδοσης με τη συχνή ενημέρωση του πίνακα ευρετηρίων. Ωστόσο, ο υπολογισμός του πίνακα ευρετηρίων μπορεί να είναι χρονοβόρος και είναι ο καταλληλότερος για σενάρια όπου ο φόρτος εργασίας έχει μικρή διακύμανση. Ένα αξιοσημείωτο πλεονέκτημα του αλγορίθμου AMLB είναι ότι είναι ανεκτικός σε σφάλματα, όπως υποδεικνύεται από την κατάσταση του FT. Παρά τους περιορισμούς του, εξακολουθεί να είναι μια αξιόπιστη επιλογή για καταστάσεις όπου η απόδοση και η απόδοση είναι πρωταρχικής σημασίας.

Ο αλγόριθμος Ant Colony Optimization είναι ένας δυναμικός αλγόριθμος με υψηλό βαθμό πολυπλοκότητας και υψηλό κόστος. Ένα από τα δυνατά του σημεία όπως είδαμε παραπάνω, είναι ότι μιμείται το στυλ των ζωντανών μυρμηγκιών. Αυτό βοηθά στη μετάβαση από μια εικονική μηχανή (VM) σε μια άλλη VM, έχοντας ως αποτέλεσμα καλύτερη απόδοση και μείωση του χρόνου λειτουργίας. Επιπλέον, ο αλγόριθμος Ant Colony Optimization μπορεί να συνδυαστεί με ευρετικούς

αλγόριθμους αναζήτησης για περαιτέρω βελτίωση της απόδοσης. Παρόλα αυτά, υπάρχουν ορισμένοι περιορισμοί. Ένας από αυτούς είναι ότι είναι πολύπλοκος και μπορεί να επιφέρει μεγάλο φορτίο στη μονάδα συστήματος. Επιπλέον, μπορεί να μην είναι απολύτως αξιόπιστος στην πραγματικότητα. Ωστόσο, ο αλγόριθμος Ant Colony Optimization έχει υψηλή απόδοση και υψηλή ανοχή σφαλμάτων, που σημαίνει ότι μπορεί να χειριστεί τυχόν σφάλματα ή αστοχίες που μπορεί να προκύψουν. Συμπερασματικά, ενώ ο αλγόριθμος Ant Colony Optimization έχει ορισμένους περιορισμούς, τα δυνατά του σημεία τον καθιστούν ένα ισχυρό εργαλείο για τον προγραμματισμό εργασιών και τη βελτιστοποίηση σε πολλές περιπτώσεις.

Ο αλγόριθμος Honey Bee είναι ένας σύνθετος αλγόριθμος που έχει σχεδιαστεί για να επιλέγει δυναμικά την πιο οικονομική εικονική μηχανή (VM) για μια δεδομένη εργασία. Αυτός ο αλγόριθμος είναι ιδιαίτερα χρήσιμος επειδή λαμβάνει υπόψη τόσο τους υπερχρησιμοποιημένους όσο και τους υποχρησιμοποιημένους πόρους κατά την επιλογή ενός και αναθέτει λιγότερες εργασίες σε έναν υπολογιστή που μπορεί να τις χειριστεί γρήγορα, μειώνοντας έτσι το χρόνο εκτέλεσης. Παρά τα οφέλη του, ο αλγόριθμος μελισσών περιορίζεται από την πολυπλοκότητά του και η αύξηση του μεγέθους της μηχανής δεν βελτιώνει την απόδοσή του. Ωστόσο, η δύναμη του αλγόριθμου έγκειται στην υψηλή απόδοση και ανοχή ελαττωμάτων, καθιστώντας τον ελκυστική επιλογή για συγκεκριμένες περιπτώσεις χρήσης, παρά τη χαμηλή απόδοσή του.

Ο γενετικός αλγόριθμος είναι ένας δυναμικός αλγόριθμος με υψηλή πολυπλοκότητα και υψηλό γενικό κόστος. Τα βασικά πλεονεκτήματα του βρίσκονται στην ικανότητά του να εκτελεί εργασίες (cloudlets<sup>8</sup>) σε λιγότερο χρόνο, να βελτιώνει την κατανομή φορτίου στο Cloud και να βελτιστοποιεί τη χρήση πόρων. Ακόμη, είναι αποτελεσματικός στην επιλογή της καλύτερης ευρετηρίασης πληθυσμού σε εικονικούς υπολογιστές. Ωστόσο, ο αλγόριθμος περιορίζεται από την πολυπλοκότητά του και τις υψηλές απαιτήσεις πόρων για τη δημιουργία και τον υπολογισμό του πληθυσμού. Επιπρόσθετα, η αποτελεσματικότητα της μετάλλαξης διασταύρωσης εξαρτάται σε μεγάλο βαθμό από τις παραμέτρους πιθανότητας. Παρά τους περιορισμούς αυτούς, ο

---

<sup>8</sup> Το cloudlet είναι ένα κέντρο δεδομένων μικρής κλίμακας ή ένα σύμπλεγμα υπολογιστών που έχει σχεδιαστεί για να παρέχει γρήγορα υπηρεσίες υπολογιστικού νέφους σε κινητές συσκευές, όπως smartphone, tablet και φορητές συσκευές, σε κοντινή γεωγραφική εγγύτητα.



γενετικός αλγόριθμος παρέχει υψηλή απόδοση, καθιστώντας τον αρκετά καλή επιλογή για ορισμένες εφαρμογές. Επιπλέον, διαθέτει δυνατότητες ανοχής σφαλμάτων, εξασφαλίζοντας ότι ο αλγόριθμος μπορεί να αντέξει σφάλματα ή αποτυχίες.

Ο αλγόριθμος FCFS (First Come First Served) είναι ένας τύπος δυναμικού αλγορίθμου που είναι σχετικά απλός ως προς την πολυπλοκότητά του. Η κύρια λειτουργία του είναι να διανέμει εργασίες σε όλα τα διαθέσιμα VM με τη σειρά με την οποία λαμβάνονται. Όμως, υπάρχουν ορισμένοι περιορισμοί στον αλγόριθμο FCFS. Για παράδειγμα, δεν είναι προληπτικός αλγόριθμος, που σημαίνει ότι δεν μπορεί να προβλέψει μελλοντικά σενάρια. Επιπλέον, αναθέτει εργασίες σε όλες τις εικονικές μηχανές ανεξάρτητα από τις προδιαγραφές τους και δεν λαμβάνει υπόψη τον χρόνο που θα χρειαστεί για να ολοκληρωθούν οι εργασίες που βρίσκονται ήδη στην ουρά. Παρά αυτούς τους περιορισμούς, ο αλγόριθμος μπορεί να αποδώσει αρκετά καλά, με μεσαία απόδοση και απόδοση. Ωστόσο, δεν έχει ανοχή σφαλμάτων. Συνολικά, ενώ ο αλγόριθμος έχει κάποια επιβάρυνση, η απλότητά του τον καθιστά ελκυστική επιλογή για απλούστερα σενάρια προγραμματισμού εργασιών.

Ο αλγόριθμος γενικευμένης προτεραιότητας (Generalized Priority Algorithm) είναι ένας τύπος δυναμικού αλγορίθμου ο οποίος εκχωρεί προτεραιότητα σε εργασίες ανάλογα με το μέγεθός τους και την επεξεργαστική ισχύ του μηχανήματος. Αυτός ο αλγόριθμος είναι μέτριας πολυπλοκότητας και δεν προσθέτει επιπλέον έξοδα. Το κύριο πλεονέκτημα αυτού του αλγορίθμου είναι η ικανότητά του να βελτιώνει την απόδοση και να επιτυγχάνει υψηλή απόδοση. Είναι σημαντικό να σημειωθεί ότι υπάρχουν μειονεκτήματα σε αυτόν τον αλγόριθμο, όπως ο αυξημένος χρόνος εκτέλεσης και η πιθανότητα πρόκλησης αιτίας. Παρά τους περιορισμούς αυτούς, ο Αλγόριθμος Γενικευμένης Προτεραιότητας θεωρείται ανεκτικός σε σφάλματα, γεγονός που τον καθιστά αξιόπιστη επιλογή για συγκεκριμένες περιπτώσεις χρήσης. Συνολικά, αυτός ο αλγόριθμος μπορεί να είναι ένα ισχυρό εργαλείο για τη βελτιστοποίηση της ιεράρχησης εργασιών και τη βελτίωση της αποτελεσματικότητας.

Ο δυναμικός αλγόριθμος Biased Random Sampling αποδίδει καλύτερα με μεγάλο και παρόμοιο πληθυσμό πόρων. Αλλά υποβαθμίζεται καθώς αυξάνεται η ποικιλία του πληθυσμού.

Ο αλγόριθμος PSO έχει πολλά πλεονεκτήματα. Έχει λίγες παραμέτρους που πρέπει να ρυθμιστούν, επομένως είναι απλό στην εφαρμογή. Επίσης, μπορεί να χρησιμοποιηθεί για ταυτόχρονη επεξεργασία. Είναι σημαντικός στην παγκόσμια αναζήτηση, καθώς δεν επηρεάζεται από την κλιμάκωση των μεταβλητών σχεδιασμού. Ωστόσο, ο PSO τείνει να οδηγεί σε γρήγορη και πρόωρη σύγκλιση. Τέλος έχει αρκετές εφαρμογές στον τομέα της επεξεργασίας εικόνας, στα συστήματα ελέγχου, στη δικτύωση, στη μηχανική μάθηση, στα συστήματα ισχύος και σε πολλά άλλα.

Πίνακας 3.2 Συγκριτικός πίνακας αλγορίθμων ΕΦ

Algorithm Name	Type	Overhead	Complexity	Performance	Throughput	Fault Tolerance
<b>Round Robin [23]</b>	Static	No	Simple	Low	Low	No
<b>Throttled [26]</b>	Static	No	Medium	Medium	High	Yes
<b>Min-Min [22]</b>	Static	Medium	Simple	Increased performance for smaller tasks	Improved	No
<b>Max-Min [22]</b>	Static	Medium	Simple	Better performance than Min-Min	Improved	No
<b>Opportunistic [23], [22]</b>	Static	Low	Simple	Poor	Limited	No
<b>Ant Colony Optimization [25]</b>	Dynamic	High	High	High	High	Yes
<b>Honey Bee [25]</b>	Dynamic	High	High	High	Low	Yes
<b>AMLB [26]</b>	Static	High	Medium	High	High	Yes
<b>Genetic Algorithm [26]</b>	Dynamic	High	High	High	High	Yes
<b>First come first serve [26]</b>	Dynamic	Medium	Simple	Medium	Medium	No
<b>Generalized Priority Algorithm [26]</b>	Dynamic	No	Medium	Increased	High	Yes
<b>ESCE [13]</b>	Dynamic	No	High	Medium	Low	No
<b>PSO [22],[27]</b>	Dynamic	No	High	Low	Medium	No

## Κεφάλαιο 4°

### 4 Αξιολόγηση Απόδοσης

Για την αξιολόγηση της απόδοσης των αλγορίθμων εξισορρόπησης σε περιβάλλον Cloud, οι ερευνητές προτείνουν είτε τη χρήση ρεαλιστικών πλατφορμών είτε τη χρήση εργαλείων προσομοίωσης.

Υπάρχουν διάφορες πλατφόρμες ειδικά σχεδιασμένες για δοκιμές απόδοσης. Οι πιο γνωστές είναι το OpenNebula, η ElasticHosts και το EC2 (Elastic Compute Cloud). Το OpenNebula είναι μια ανοιχτή πλατφόρμα που μπορεί να χειριστεί αποτελεσματικά περίπλοκες, μεγάλης κλίμακας κατανεμημένες υποδομές με διαφορετικά χαρακτηριστικά. Επιπλέον, παρέχει ένα ευρύ φάσμα προηγμένων χαρακτηριστικών, ευέλικτων επιλογών και βελτιωμένης διαλειτουργικότητας για τη διευκόλυνση της κατασκευής ισχυρών και προσαρμόσιμων περιβαλλόντων Cloud. Από την άλλη πλευρά η ElasticHosts παρέχει μια φιλική προς το χρήστη πλατφόρμα που προσφέρει διακομιστές Cloud με άμεσες και προσαρμόσιμες υπολογιστικές δυνατότητες. Τέλος, το EC2 προσφέρει στους πελάτες τη δυνατότητα να μισθώνουν υπολογιστικούς πόρους από την υποδομή Cloud. Αυτοί οι πόροι περιλαμβάνουν την αποθήκευση, την επεξεργαστική ισχύ και μια σειρά από υπηρεσίες Ιστού [28].

Στα εργαλεία προσομοίωσης μεταξύ άλλων περιλαμβάνονται το CloudSim, το CloudSched, το FlexCloud, το Cloud Analyst [28], [29]. Το CloudSim είναι ένας προσομοιωτής που βασίζεται σε Java. Ο αντικειμενοστραφής προγραμματισμός επιτρέπει στους ερευνητές να επεκτείνουν και να ορίζουν πολιτικές σε όλα τα στοιχεία λογισμικού. Αυτό καθιστά το CloudSim ιδανικό εργαλείο για τη μελέτη και την εξομοίωση σε διάφορα περιβάλλοντα. Όμως έχουν δημιουργηθεί κι άλλα εργαλεία προσομοίωσης όπως το CloudSched, το οποίο δίνει τη δυνατότητα στους χρήστες να συγκρίνουν διαφορετικούς αλγόριθμους προγραμματισμού πόρων σε περιβάλλον ανάπτυξης IaaS. Διευκολύνει την ανάλυση τόσο των κεντρικών υπολογιστών όσο και του φόρτου εργασίας, βοηθώντας τους προγραμματιστές να εντοπίσουν και να εξερευνήσουν κατάλληλες λύσεις για διαφορετικούς αλγόριθμους προγραμματισμού πόρων. Για όσους αναζητούν μια ευέλικτη και επεκτάσιμη πλατφόρμα προσομοίωσης, το FlexCloud αποτελεί μια καλή επιλογή. Επιτρέπει στους χρήστες να προσομοιώνουν ολόκληρη τη διαδικασία προετοιμασίας κέντρων δεδομένων Cloud, χειρισμού

αιτημάτων εικονικής μηχανής και αξιολόγησης της απόδοσης για διαφορετικούς αλγόριθμους προγραμματισμού.

#### 4.1 Το Λογισμικό Προσομοίωσης Cloud Analyst

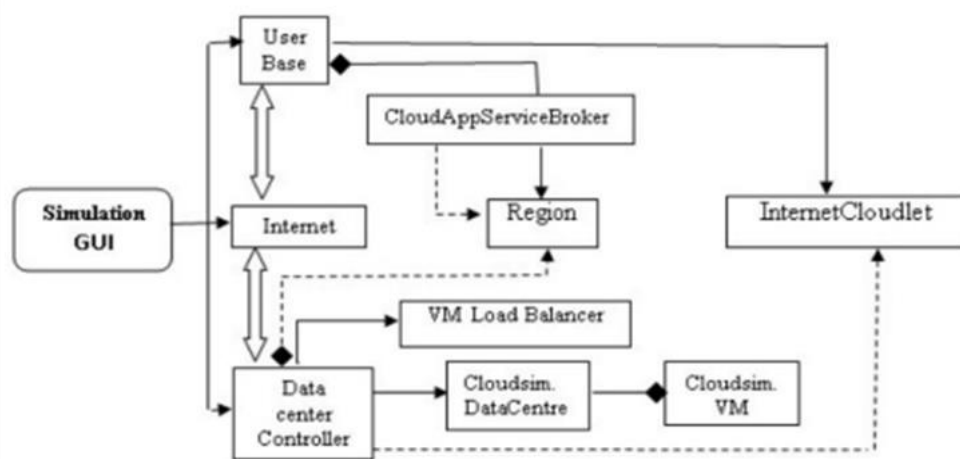
Υπάρχουν αρκετά εξαιρετικά καλά εργαλεία που μπορούν να χρησιμοποιηθούν για να μοντελοποιήσουν ένα προσομοιωμένο περιβάλλον για τη μελέτη της συμπεριφοράς μιας μεγάλης κλίμακας εφαρμογής στο διαδίκτυο. Αλλά η ύπαρξη ενός εύχρηστου εργαλείου με ένα επίπεδο ικανότητας απεικόνισης είναι ακόμα καλύτερο από μια εργαλειοθήκη. Ένα τέτοιο εργαλείο διαχωρίζει την άσκηση του πειράματος προσομοίωσης από μια άσκηση προγραμματισμού και επιτρέπει σε έναν μοντελοποιητή να επικεντρωθεί στην προσομοίωση και της παραμέτρους και όχι της τεχνικές λεπτομέρειες του προγραμματισμού. Η γραφική παραγωγή των αποτελεσμάτων προσομοίωσης επιτρέπει την ευκολότερη ανάλυση των αποτελεσμάτων και μπορεί επίσης να βοηθήσει στην γρήγορη επισήμανση τυχόν προβλημάτων με την απόδοση και ακρίβεια της λογικής προσομοίωσης [29].

Το εργαλείο Cloud Analyst χρησιμοποιείται για τον εντοπισμό του πιο αποδοτικού αλγόριθμου εξισορρόπησης φορτίου μεταξύ διαφορετικών τεχνικών προγραμματισμού και εξισορρόπησης φορτίου. Αυτή η γνώση παρέχει πολύτιμες πληροφορίες για το σχεδιασμό υπηρεσιών υποδομής Cloud. Λαμβάνονται υπόψη διάφορες πτυχές, συμπεριλαμβανομένου του συντονισμού μεταξύ διαφορετικών κέντρων δεδομένων, αλγορίθμων ΕΦ και άλλων υπηρεσιών προστιθέμενης αξίας. Αυτά τα μέτρα στοχεύουν στον αποτελεσματικό συγχρονισμό των λειτουργιών μεταξύ των κέντρων δεδομένων, βελτιώνοντας τελικά το κόστος και την απόδοση για τους ιδιοκτήτες. Επιπρόσθετα, το CloudSim είναι ένα αποτελεσματικό εργαλείο που μπορεί να χρησιμοποιηθεί και για τη δημιουργία μιας καλής δομής του Cloud. Όλοι οι προτεινόμενοι αλγόριθμοι μπορούν να εκτελεστούν με αυτό το εργαλείο. Χρησιμοποιείται επίσης για την εκχώρηση κάθε αιτήματος πελάτη σε διαθέσιμα VM. Αυτά τα VM υπάρχουν στον κεντρικό υπολογιστή και διαχειρίζονται από κέντρα δεδομένων [30].

##### 4.1.1 Cloud Analyst Simulator

Ο προσομοιωτής Cloud Analyst είναι μια βελτιωμένη έκδοση του εργαλείου CloudSim, η οποία ενσωματώνει μια γραφική διεπαφή χρήστη (GUI). Περιλαμβάνει

τρία κύρια στοιχεία: το κέντρο δεδομένων (DC), τη βάση χρηστών (UB) και το διαδίκτυο (IR). Η Εικόνα 4.1 παρέχει μια οπτική αναπαράσταση των βασικών συστατικών του Cloud Analyst. Είναι ένα εργαλείο λογισμικού που διευκολύνει την προσομοίωση της ανάπτυξης DC, τη διαθεσιμότητα της βάσης χρηστών και την κατανομή εργασιών σε συγκεκριμένα κέντρα δεδομένων, λαμβάνοντας υπόψη το διαφορετικό κόστος. Τέλος, βοηθά σημαντικά στην αποτελεσματική εξισορρόπηση του φόρτου εργασίας σε βάσεις χρηστών, μαζί με τα σχετικά κέντρα δεδομένων και τη συνδεσιμότητα στο διαδίκτυο [31].



Εικόνα 4.1 Κύρια στοιχεία του Cloud Analyst Simulator (πηγή: [31])

#### 4.1.2 Χαρακτηριστικά του Cloud Analyst

Τα βασικά χαρακτηριστικά του Cloud Analyst είναι [32]:

1. Ευκολία στη χρήση: η προσομοίωση δεν απαιτεί περισσότερο από ένα πάτημα πλήκτρων.
2. Έξοδος που βασίζεται σε GUI: Όπου η έξοδος είναι με τη μορφή πινάκων και διαγραμμάτων που βοηθούν στην κατανόηση και σύγκριση των αποτελεσμάτων.
3. Δυνατότητα επανάληψης: Η δυνατότητα επανάληψης της προσομοίωσης με της της παραμέτρους για να δίνονται τα ίδια αποτελέσματα, ώστε τα αποτελέσματα της προσομοίωσης να μην είναι τυχαία.

4. Δυνατότητα αποθήκευσης των αποτελεσμάτων: Η δυνατότητα αποθήκευσης πειραμάτων με της εισόδους της και τα αποτελέσματα σε αρχεία που μπορούν να μεταφερθούν σε της τοποθεσίες υπολογιστών.

Υπάρχουν αρκετά στοιχεία στο Cloud Analyst που κάνουν τη χρήση του εύκολη και ευέλικτη, τα οποία αναλύονται παρακάτω [26].

**Εργαλειοθήκη Cloud Analyst:** Η εργαλειοθήκη Cloud Analyst χωρίζει τον κόσμο σε έξι περιοχές, που αναπαρίστανται με τους αριθμούς από 0 έως 5, οι οποίες αντιστοιχούν σε έξι ηπείρους. Το 0 αντιπροσωπεύει τη Βόρεια Αμερική, το 1 τη Νότια Αμερική, το 2 την Ευρώπη, το 3 την Ασία, το 4 την Αφρική και το 5 την Αυστραλία. Μέσα σε αυτές της περιοχές, βρίσκονται τα στοιχεία UserBase και κέντρου δεδομένων του προσομοιωτή Cloud Analyst. Αυτό σημαίνει ότι οι χρήστες και τα κέντρα δεδομένων που συμμετέχουν σε προσομοιώσεις χρησιμοποιώντας το Cloud Analyst βρίσκονται σε αυτές της έξι περιοχές.

**Βάση χρήστη:** Ο Αναλυτής Cloud θεωρεί μια ομάδα χρηστών ως μια ενιαία οντότητα που ονομάζεται UserBase ή αλλιώς βάση χρήστη, η οποία αποτελεί βασικό στοιχείο της προσομοίωσης. Ο κύριος ρόλος της βάσης χρήστη είναι να δημιουργεί κίνηση, πράγμα που σημαίνει ότι προσομοιώνει της ενέργειες πραγματικών χρηστών που θα έχουν πρόσβαση και θα χρησιμοποιούν της υπηρεσίες που παρέχονται από την υποδομή Cloud. Στην ουσία, το UserBase αντιπροσωπεύει τη συλλογική συμπεριφορά των χρηστών σε ένα προσομοιωμένο περιβάλλον.

**Ελεγκτής Κέντρου Δεδομένων:** Ο Ελεγκτής Κέντρου Δεδομένων (DCC) είναι ένα κρίσιμο στοιχείο του Προσομοιωτή Cloud Analyst που αντιστοιχίζεται σε μία μόνο παρουσία της προσομοίωσης Cloud. Το DCC είναι υπεύθυνο για τη διαχείριση όλων των δραστηριοτήτων του κέντρου δεδομένων, συμπεριλαμβανομένης της δρομολόγησης αιτημάτων cloudlet και της δημιουργίας και διαγραφής εικονικών μηχανών (VM). Ουσιαστικά, το DCC ενεργεί ως κεντρικός συντονιστής των διαφόρων πόρων και διαδικασιών εντός του κέντρου δεδομένων, διασφαλίζοντας ότι συνεργάζονται αποτελεσματικά για την παροχή των απαιτούμενων υπηρεσιών.

**VM Load Balancer:** Ο VM Load Balancer (VMLB) είναι ένα άλλο βασικό στοιχείο του προσομοιωτή Cloud Analyst με τον οποίο επικοινωνεί το DCC

σχετικά με την κατανομή των πόρων. Το VMLB είναι υπεύθυνο για να αποφασίσει ποιες εικονικές μηχανές (VM) θα χειριστούν ποια cloudlets, διασφαλίζοντας ότι το φορτίο εξισορροπείται ομοιόμορφα σε όλους της διαθέσιμους πόρους. Το Cloud Analyst έχει επί του παρόντος τρεις διαφορετικές επιλογές VMLB. Round Robin, Throttled και Active Monitoring Load Balancer. Αυτές οι επιλογές VMLB παρέχουν διαφορετικούς τρόπους κατανομής του φόρτου εργασίας και μπορούν να επιλεγούν με βάση της συγκεκριμένες απαιτήσεις της προσομοίωσης που εκτελείται.

**Cloud Application Service Broker:** Το Cloud Application Service Broker είναι υπεύθυνο για τη διαχείριση της ροής κίνησης μεταξύ των βάσεων χρηστών και των κέντρων δεδομένων. Λειτουργεί ως μεσολαβητής μεταξύ αυτών των δύο στοιχείων και διασφαλίζει ότι παρέχονται οι κατάλληλες υπηρεσίες της χρήστες. Ο προσομοιωτής Cloud Analyst έχει τρεις διαφορετικές πολιτικές μεσολάβησης υπηρεσιών που μπορούν να χρησιμοποιηθούν για τη διαχείριση αυτής της κυκλοφορίας. Βελτιστοποίηση του χρόνου απόκρισης, πλησιέστερο κέντρο δεδομένων και δυναμική αναδιαμόρφωση. Κάθε μία από αυτές της πολιτικές παρέχει διαφορετική προσέγγιση για τη διαχείριση της ροής κυκλοφορίας, ανάλογα με της συγκεκριμένες ανάγκες της προσομοίωσης.

Οι πολιτικές μεσίτη (broker policies) που είναι διαθέσιμες στον προσομοιωτή Cloud Analyst περιλαμβάνουν τρεις διαφορετικές επιλογές:

- Βελτιστοποίηση χρόνου απόκρισης (Optimize Response Time):
- Πλησιέστερο κέντρο δεδομένων (Closest Data Center)
- Δυναμική αναδιαμόρφωση (Reconfigure Dynamically)

Η πολιτική βελτιστοποίησης χρόνου απόκρισης δίνει προτεραιότητα στην ελαχιστοποίηση του χρόνου που απαιτείται για να λάβουν απάντηση οι χρήστες από της υπηρεσίες Cloud στις οποίες έχουν πρόσβαση. Η πολιτική του πλησιέστερου κέντρου δεδομένων δίνει προτεραιότητα στην καθοδήγηση των χρηστών στο κέντρο δεδομένων που είναι γεωγραφικά πλησιέστερα σε αυτούς, προκειμένου να μειωθεί η καθυστέρηση και να βελτιωθεί η συνολική εμπειρία χρήστη. Τέλος, η δυναμικά αναδιαμορφωμένη πολιτική έχει σχεδιαστεί για να



προσαρμόζει και να βελτιστοποιεί τη ροή της κυκλοφορίας σε πραγματικό χρόνο, με βάση τις μεταβαλλόμενες συνθήκες εντός της προσομοίωσης.

#### 4.2 Το περιβάλλον προσομοίωσης

Για την προσομοίωση των αλγορίθμων εξισορρόπησης φορτίου σε περιβάλλον Cloud, χρησιμοποιήθηκε επιτραπέζιος υπολογιστής με τα ακόλουθα χαρακτηριστικά:

##### **Χαρακτηριστικά Μηχανής εκτέλεσης προσομοιώσεων:**

- Λειτουργικό Σύστημα : Windows 10 Pro
- Επεξεργαστής: Intel Core i3 9100F
- Μνήμη: 16 GB DDR4
- Σκληρός δίσκος : SSD 500 GB

##### **Έκδοση εργαλείου Cloudsim:**

- CloudSim 3.0.3

Χρησιμοποιώντας το εργαλείο Cloud Analyst είναι δυνατή τόσο η εναλλαγή των αλγορίθμων εξισορρόπησης όσο και των εφαρμοζόμενων πολιτικών μεσίτη. Πιο συγκεκριμένα, στη μελέτη αυτή εφαρμόστηκαν οι παρακάτω αλγόριθμοι και πολιτικές:

##### **Αλγόριθμοι:**

1. Round Ronin
2. ESCE
3. Throttled
4. Honey Bee
5. PSO

##### **Πολιτικές μεσίτη υπηρεσιών:**

- a. Βελτιστοποίηση χρόνου απόκρισης (Optimize Response Time)
- b. Πλησιέστερο κέντρο δεδομένων ( Closest Data Center)
- c. Δυναμική αναδιαμόρφωση (Reconfigure Dynamically)

Εκτελέστηκαν διάφορα σενάρια με κέντρα δεδομένων, από 1 έως 6, να τοποθετούνται σε διάφορες περιοχές. Η προσομοίωση εκτελέστηκε για χρονικό 60 λεπτών. Το πλήθος το χρηστών διαφοροποιείται ανάλογα με το σενάριο. Οι υπόλοιπες παράμετροι που χρησιμοποιήθηκαν στην προσομοίωση παρουσιάζονται στον Πίνακα 4.1

Πίνακας 4.1 Ρυθμίσεις παραμέτρων

<b>Παράμετροι</b>	<b>Εύρος τιμών/Προδιαγραφές</b>
VM-image Size	10000
VM-memory	512 MB
VM-Bandwidth	1000
Data center architecture	X86
Data center - OS	Linux
Data center No of VMs	5
Virtual Machine Manager (VMM)	Xen
Number of region use	6
Request per user per Hour	60

## Κεφάλαιο 5°

### 5 Σενάρια προσομοίωσης

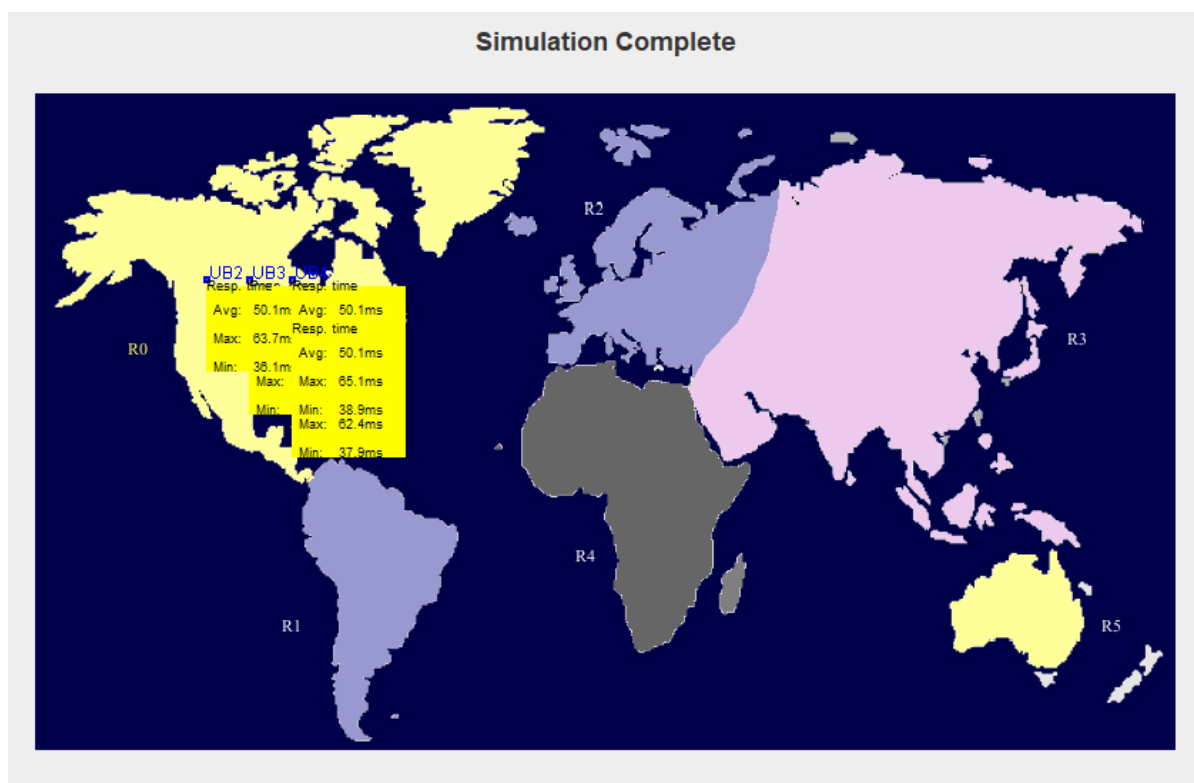
#### 5.1 Σενάριο 1- Προσομοίωση με 1 DC & 6 UB στην ίδια περιοχή

##### 5.1.1 Διαμόρφωση σεναρίου

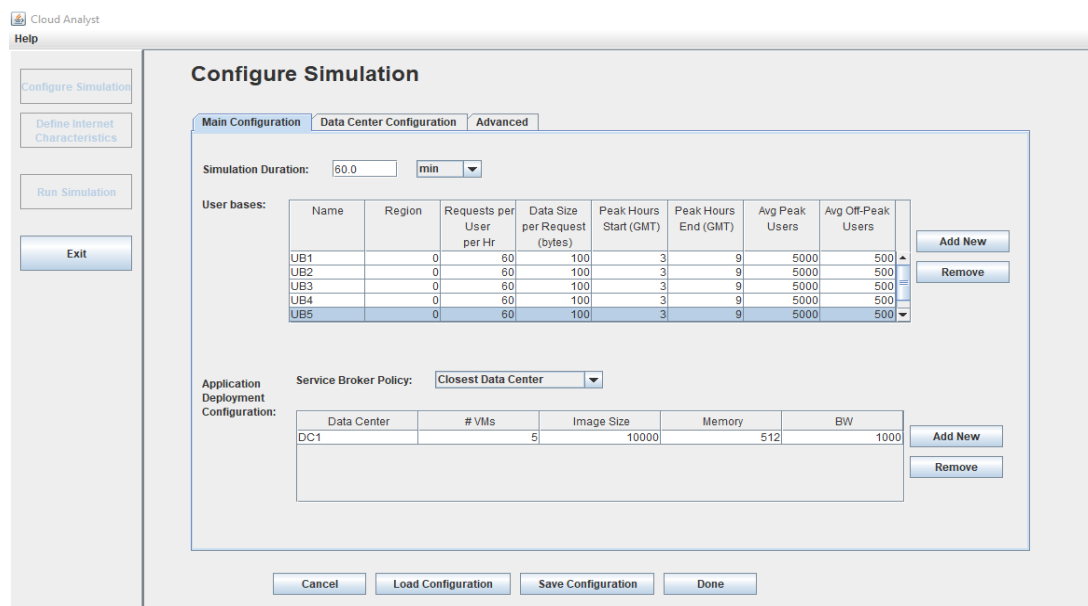
**Σκοπός:** Σκοπός αυτού του σεναρίου είναι η αναπαραγωγή των αποτελεσμάτων από της εργασίας των Ahmad και Hammo [32], χρησιμοποιώντας τον προσομοιωτή Cloud Analyst για τη σύγκριση των αλγορίθμων Round Robin, ESCE και Throttled. Ο κύριος στόχος είναι η σε βάθος κατανόηση της λειτουργίας και της συμπεριφοράς των αλγορίθμων μέσω της αναδημιουργίας του σεναρίου, της επανάληψης των πειραμάτων και τη συλλογή αποτελεσμάτων για καθορισμένες μετρικές. Επιπλέον, συμπεριλαμβάνονται στην αξιολόγηση οι αλγόριθμοι Honey Bee και PSO.

**Διαμόρφωση:** Οι βάσεις χρηστών και τα κέντρα δεδομένων τοποθετούνται και τα δύο στην ίδια περιοχή προκειμένου να αγνοηθεί η επίδραση της γεωγραφικής απόστασης στα αποτελέσματα. Επιλέγεται ένα κέντρο δεδομένων και πέντε βάσεις χρηστών χρησιμοποιούνται που τοποθετούνται όλα στην γεωγραφικά στην περιοχή της Βόρειας Αμερικής (περιοχή (0)), όπως παρατηρούμε στην Εικόνα 5.1. Για τα χαρακτηριστικά του κέντρου δεδομένων όπως καθυστέρηση μετάδοσης, διαθέσιμο εύρος ζώνης μεταξύ των περιοχών και κόστος, διατηρήθηκαν οι προεπιλεγμένες ρυθμίσεις του εργαλείου. Το πείραμα επαναλαμβάνεται αρκετές φορές μεταβάλλοντας:

- a) τον αριθμό των χρηστών από 5000 χρήστες έως 125000 χρήστες ανά βάση χρηστών,
- b) την πολιτική μεσίτη



Εικόνα 5.1: Γραφική οθόνη εξόδου



Εικόνα 5.2 Γραφική διεπαφή για τον καθορισμό των βάσεων χρηστών

**Μετρικές:** Για το σενάριο αυτό συλλέχθηκαν αποτελέσματα για το μέσο χρόνο απόκρισης, το μέσο χρόνος επεξεργασίας του κέντρου δεδομένων, το κόστος και την υπεραπόδοση.

Ως υπεραπόδοση ορίζεται ως

$$PO = \frac{(MB - MA)}{MA} * 100$$

όπου  $M_A$  η υπολογιζόμενη μετρική ενός αλγορίθμου A και  $M_B$  η υπολογιζόμενη μετρική ενός αλγορίθμου B.

### 5.1.2 Αποτελέσματα

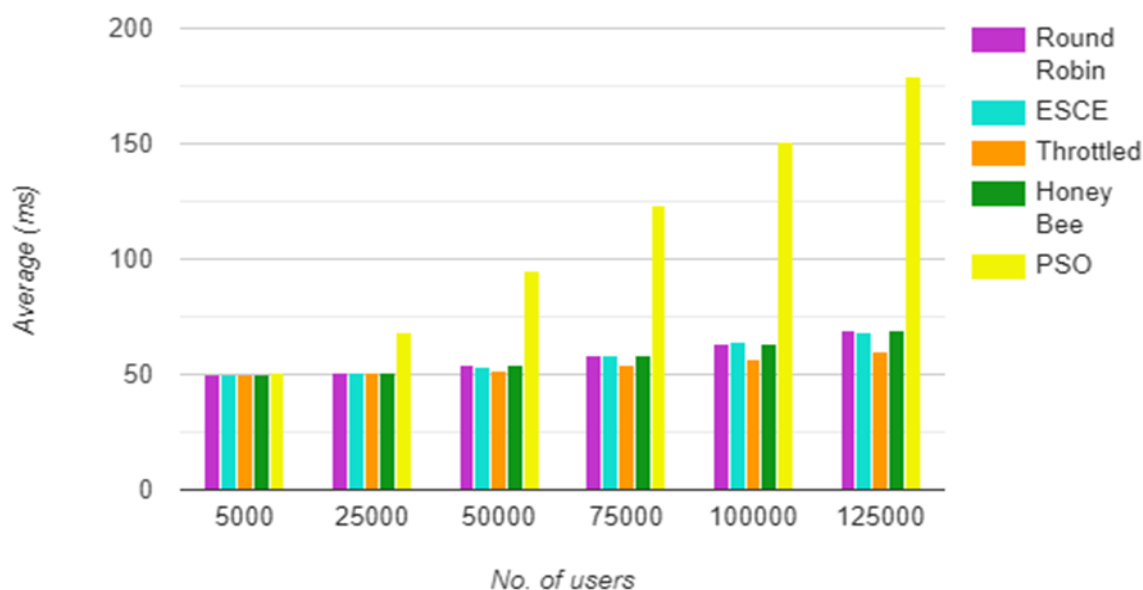
#### A) Χρόνος απόκρισης

Τα αποτελέσματα για τον μέσο συνολικό χρόνο απόκρισης κάθε πολιτικής απεικονίζονται στα Γραφήματα 4-1, 4-2, 4-3.

#### A1) Χρόνος απόκρισης για την πολιτική *Closest Data Center*

Πίνακας 5.1 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users in each UB	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	50.11	50.11	50.11	50.31	50.73
25000	50.73	50.72	50.50	51.11	68.19
50000	53.90	53.47	51.98	53.98	95.40
75000	58.45	58.17	54.40	58.61	123.09
100000	63.36	63.92	57.02	63.72	151.13
125000	68.95	68.56	59.73	69.01	179.28

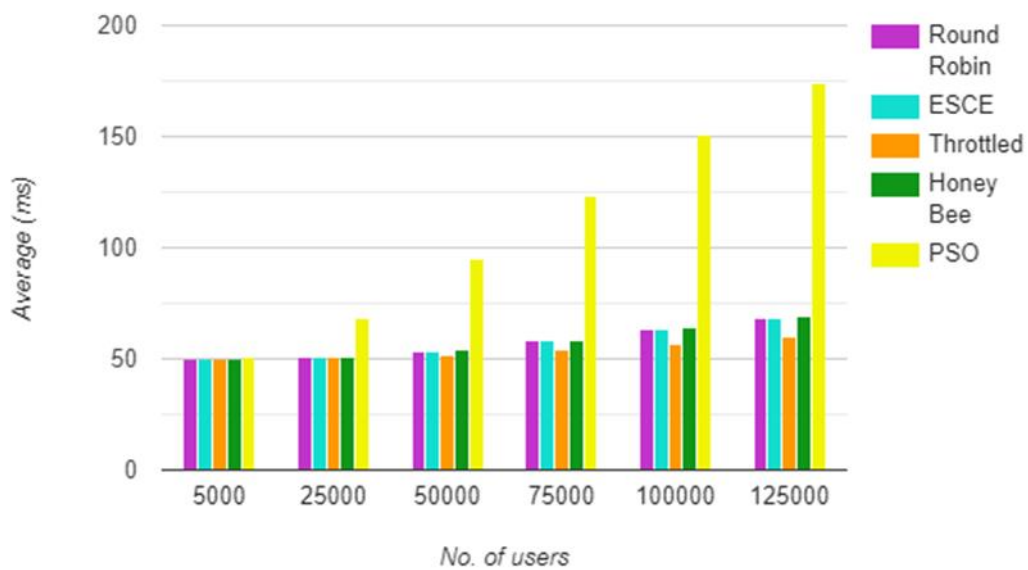


Γράφημα 5-1 : Μέσος όρος του χρόνου απόκρισης ORT (ms) συγκριτικά με τον αριθμό των χρηστών

## A2) Χρόνος απόκρισης για την πολιτική *Optimise Response Time*

Πίνακας 5.2 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users in each UB	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	50.14	50.08	50.08	50.29	50.75
25000	50.73	50.70	50.49	51.11	68.25
50000	53.70	53.47	51.97	53.91	95.30
75000	58.25	58.14	54.38	58.58	123.09
100000	63.37	63.28	57.02	64.38	151.15
125000	68.63	68.54	59.71	68.99	174.25

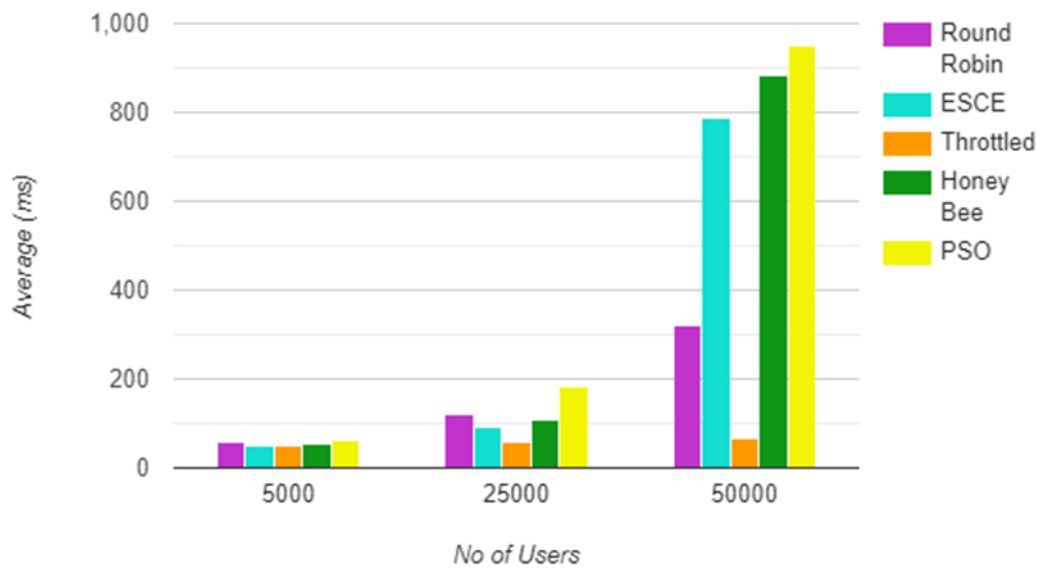


Γράφημα 5-2 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### A3) Χρόνος απόκρισης για την πολιτική *Reconfigure Dynamically with L*

Πίνακας 5.3 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users in each UB	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	59.27	51.08	51.08	52.22	61.23
25000	121.72	89.62	58.08	109.82	184.12
50000	321.62	787.46	68.06	882.98	949.62



Γράφημα 5-3 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

**B) Χρόνος επεξεργασίας κέντρου δεδομένων**

**B1) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Closest Data Center***

Πίνακας 5.4 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Average of Data Center processing time (ms)</b>					
<b>No. of users in each UB</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>5000</b>	0.49	0.49	0.49	0.69	1.11
<b>25000</b>	0.98	0.97	0.75	1.36	18.81
<b>50000</b>	4.03	3.61	2.10	4.13	45.63
<b>75000</b>	8.45	8.18	4.38	8.63	73.18
<b>100000</b>	13.26	13.80	6.87	13.63	101.10
<b>125000</b>	18.72	18.34	9.45	18.79	129.13

**B2) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Optimise Response Time***

Πίνακας 5.5 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Average of Data Center processing time (ms)</b>					
<b>No. of users in each UB</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>5000</b>	0.49	0.49	0.49	0.70	1.12
<b>25000</b>	0.99	0.96	0.74	1.37	18.56
<b>50000</b>	3.84	3.62	2.10	4.06	45.54
<b>75000</b>	8.26	8.15	4.36	8.60	73.19
<b>100000</b>	13.26	13.17	6.87	14.27	101.12
<b>125000</b>	18.41	18.33	9.45	18.78	124.11



**B3) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Reconfigure Dynamically with L***

Πίνακας 5.6 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Average of Data Center processing time (ms)</b>					
<b>No. of users in each UB</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>5000</b>	9.64	1.46	1.46	2.60	11.60
<b>25000</b>	72.01	39.91	8.36	60.11	134.40
<b>50000</b>	271.80	737.55	18.27	833.06	899.69

**Γ) Κόστος**

**Γ1)Κόστος για την πολιτική *Closest Data Center***

Πίνακας 5.7 Κόστος κάθε αλγορίθμου για 50000 χρήστες σε κάθε User Base

<b>Name of Algorithm</b>	<b>Data Center</b>	<b>VM Cost \$</b>	<b>Data Transfer Cost \$</b>	<b>Total \$</b>
<b>Round Robin</b>	DC1	0.50	19.15	19.65
<b>ESCE</b>	DC1	0.50	16.03	16.53
<b>Throttled</b>	DC1	0.50	16.03	16.53
<b>Honeybee</b>	DC1	0.50	16.03	16.53
<b>PSO</b>	DC1	0.50	16.03	16.53

**Γ2)Κόστος για την πολιτική *Optimise Response Time***

Πίνακας 5.8 Κόστος κάθε αλγορίθμου για 50000 χρήστες σε κάθε User Base

<b>Name of Algorithm</b>	<b>Data Center</b>	<b>VM Cost \$</b>	<b>Data Transfer Cost \$</b>	<b>Total \$</b>
<b>Round Robin</b>	DC1	0.50	16.04	16.54
<b>ESCE</b>	DC1	0.50	16.03	16.53
<b>Throttled</b>	DC1	0.50	16.03	16.53
<b>Honeybee</b>	DC1	0.50	16.03	16.53
<b>PSO</b>	DC1	0.50	16.03	16.53

### Γ3)Κόστος για την πολιτική *Reconfigure Dynamically with L*

Πίνακας 5.9 Κόστος κάθε αλγορίθμου για 50000 χρήστες σε κάθε User Base

Name of Algorithm	Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
Round Robin	DC1	3.25	16.03	19.28
ESCE	DC1	3.16	16.03	19.19
Throttled	DC1	3.16	16.03	19.19
Honeybee	DC1	3.16	16.03	19.19
PSO	DC1	3.083	16.029	19.113

#### Δ) Υπεραπόδοση

Σύμφωνα με τα παραπάνω αποτελέσματα ο αλγόριθμος Throttled έχει την καλύτερη απόδοση σε όλες τις κατηγορίες μετρήσεων. Για το λόγο αυτό, εδώ εξετάζουμε την υπεραπόδοση του σε σχέση με τους υπόλοιπους αλγορίθμους. Η υπεραπόδοση υπολογίζεται σύμφωνα με τον τύπο (1). Τα αποτελέσματα, δηλαδή το ποσοστό βελτίωσης του αλγορίθμου Throttled σε σχέση με τους υπόλοιπους, παρουσιάζονται στον Πίνακα 4.2 για 5000, 75000 και 125000 χρήστες αντίστοιχα.

Πίνακας 5.10 Η υπεραπόδοση (Outperformance) των αλγορίθμων σε ms

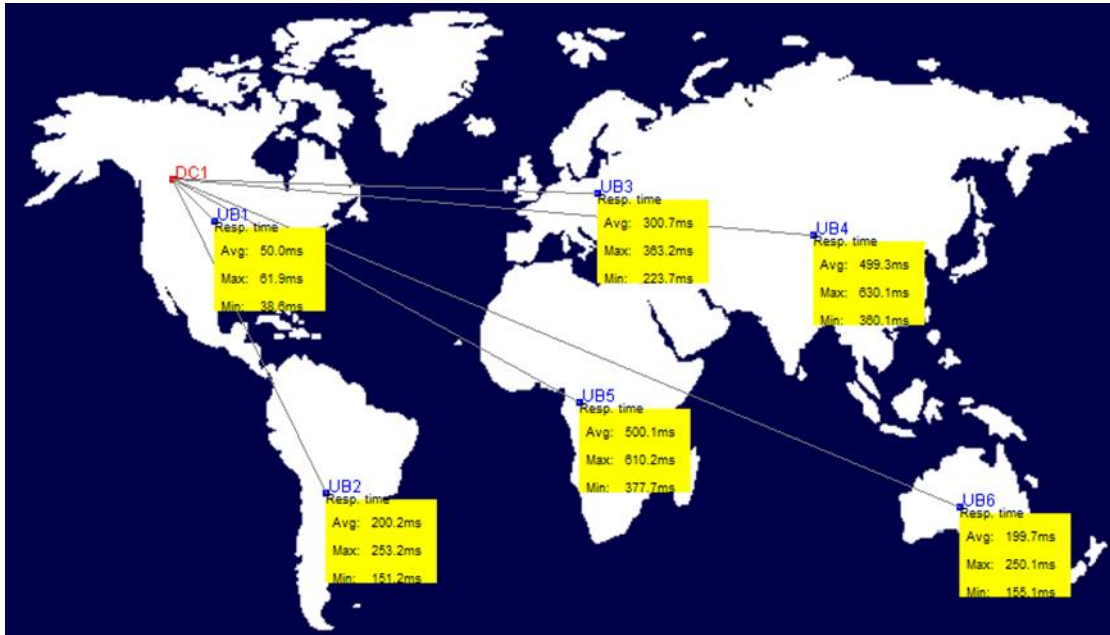
No. of users in each UB	Throttled to RR	Throttled to ESCE	Throttled to Honeybee	Throttled to PSO
5000	0%	0%	0.39%	1.23%
75000	7.27%	6.75%	7.73%	126.26%
125000	15.43%	14.78%	15.53%	200.15%

## 5.2 Σενάριο 2- Προσομοίωση με 1 DC & 5 UB σε διαφορετικές περιοχές

### 5.2.1 Διαμόρφωση σεναρίου

**Σκοπός:** Σκοπός του σεναρίου είναι η κατανόηση της λειτουργίας των αλγορίθμων με τη χρήση ενός κέντρου δεδομένων και έξι βάσεις χρηστών, οι οποίες είναι ομοιόμορφα κατανομημένες σε κάθε ήπειρο όπως φαίνεται και στην Εικόνα 5.3.

**Διαμόρφωση:** Το κέντρο δεδομένων βρίσκεται τοποθετημένο στη βόρεια Αμερική (region 0). Σε κάθε ήπειρο έχει τοποθετηθεί μια βάση χρηστών. Επιπλέον έγινε επανάληψη του πειράματος για κάθε πολιτική μεσίτη και για τους πέντε αλγορίθμους για 5000 χρήστες.



Εικόνα 5.3 Γραφική οθόνη εξόδου

### Configure Simulation

Main Configuration | Data Center Configuration | Advanced

Simulation Duration: 60.0 min

User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	0	60	100	3	9	5000	500
UB2	1	60	100	3	9	5000	500
UB3	2	60	100	3	9	5000	500
UB4	3	60	100	3	9	5000	500
UB5	4	60	100	3	9	5000	500

Service Broker Policy: Closest Data Center

Application Deployment Configuration:

Data Center	# VMs	Image Size	Memory	BW
DC1	5	10000	512	1000

Buttons: Cancel, Load Configuration, Save Configuration, Done

Εικόνα 5.4 Κόρια διαμόρφωση

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC1		x86	Linux	Xen	0.1	0.05	0.1	0.1	2

Εικόνα 5.5 Διαμόρφωση του κέντρου δεδομένων

**Μετρικές:** Για το σενάριο αυτό συλλέχθηκαν αποτελέσματα για το μέσο χρόνο απόκρισης, το μέσο χρόνος επεξεργασίας του κέντρου δεδομένων και το κόστος .

Η Εικόνα 5.4 αντιπροσωπεύει τις ρυθμίσεις της καρτέλας της κύριας διαμόρφωσης. Σε αυτό το σημείο ο χρήστης έχει τη δυνατότητα να επιλέξει και να εισάγει τα δεδομένα που αντιπροσωπεύουν τους χρήστες καθώς και τα αιτήματα αυτών Αντίστοιχα η Εικόνα 5.5 αντιπροσωπεύει την καρτέλα για τις ρυθμίσεις του κέντρου δεδομένων.

## 5.2.2 Αποτελέσματα

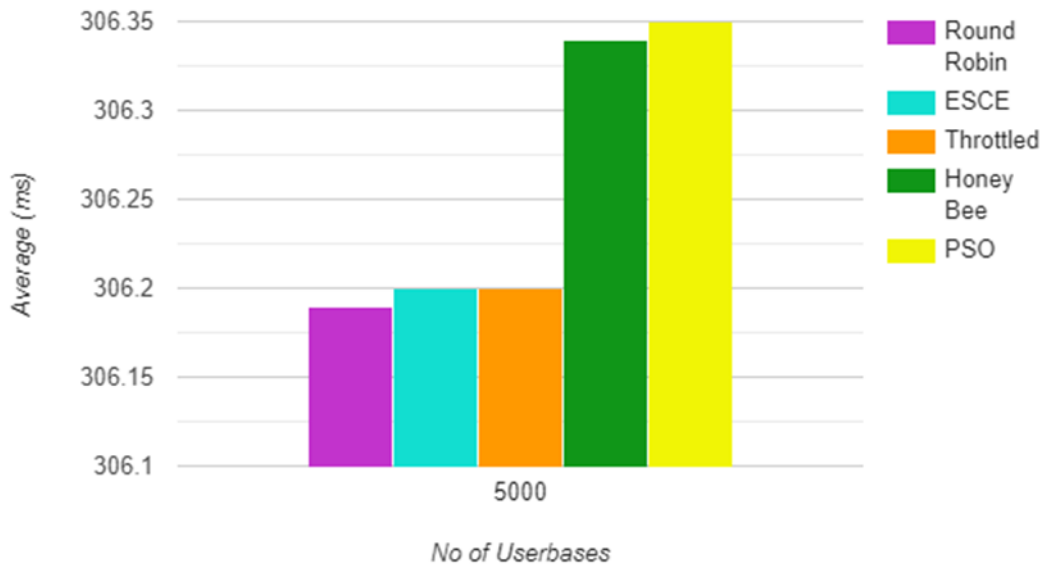
### A) Χρόνος απόκρισης

Τα αποτελέσματα για τον μέσο συνολικό χρόνο απόκρισης κάθε πολιτικής απεικονίζονται στα Γραφήματά 4-4, 4-5, 4-6.

### A1) Χρόνος απόκρισης για την πολιτική *Closest Data Center*

Πίνακας 5.11 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	306.19	306.20	306.20	306.34	306.35

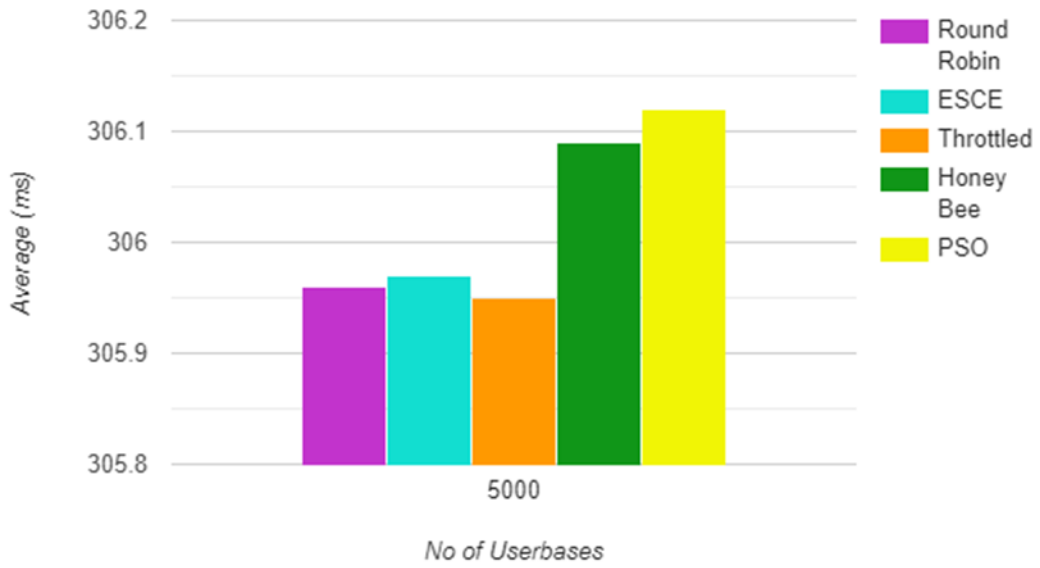


Γράφημα 5-4 Σύγκριση του μέσου όρου χρόνου απόκρισης *ORT* (ms)

## A2) Χρόνος απόκρισης για την πολιτική *Optimise Response Time*

Πίνακας 5.12 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (*ORT*) σε ms

Average of <i>ORT</i> (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
<b>5000</b>	305.96	305.97	305.95	306.09	306.12

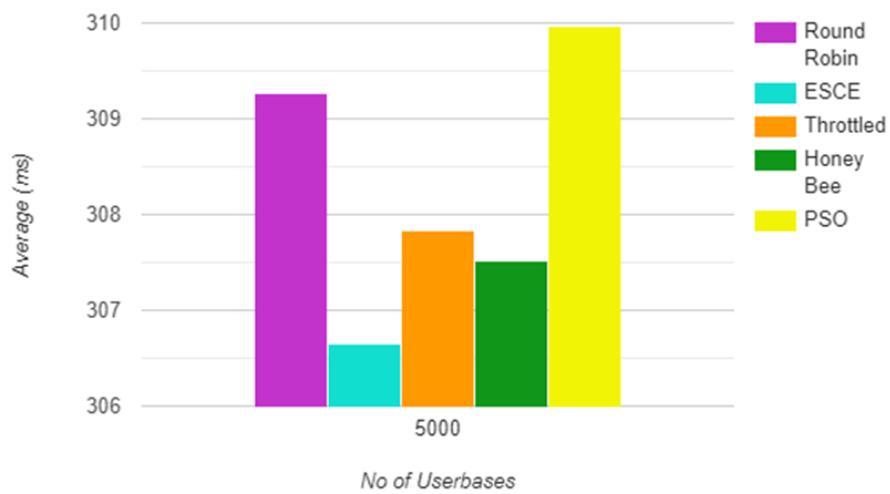


Γράφημα 5-5 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### A3) Χρόνος απόκρισης για την πολιτική Reconfigure Dynamically with L

Πίνακας 5.13 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	309.27	306.65	307.84	307.51	309.96



Γράφημα 5-6 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

**B) Χρόνος επεξεργασίας κέντρου δεδομένων**

**B1) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Closest Data Center***

Πίνακας 5.14 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Avg of Data Center processing time (ms)</b>					
<b>Name of Algorithm</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>Data Center Processing Time</b>	0.34	0.34	0.27	0.48	0.49

**B2) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Optimise Response Time***

Πίνακας 5.15 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Avg of Data Center processing time (ms)</b>					
<b>Name of Algorithm</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>Data Center Processing Time</b>	0.34	0.34	0.34	0.49	0.50

**B3) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Reconfigure Dynamically with L***

Πίνακας 5.16 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Avg of Data Center processing time (ms)</b>					
<b>Name of Algorithm</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>Data Center Processing Time</b>	3.40	0.78	1.96	1.63	4.11

### *Γ) Κόστος*

#### *Γ1)Κόστος για την πολιτική Closest Data Center*

*Πίνακας 5.17 Κόστος κάθε αλγορίθμου*

<b>Name of Algorithm</b>	<b>Data Center</b>	<b>VM Cost \$</b>	<b>Data Transfer Cost \$</b>	<b>Total \$</b>
<b>Round Robin</b>	DC1	0.50	1.66	2.16
<b>ESCE</b>	DC1	0.50	1.66	2.16
<b>Throttled</b>	DC1	0.50	1.66	2.16
<b>Honeybee</b>	DC1	0.50	1.66	2.16
<b>PSO</b>	DC1	0.50	1.66	2.16

#### *Γ2)Κόστος για την πολιτική Optimise Response Time*

*Πίνακας 5.18 Κόστος κάθε αλγορίθμου*

<b>Name of Algorithm</b>	<b>Data Center</b>	<b>VM Cost \$</b>	<b>Data Transfer Cost \$</b>	<b>Total \$</b>
<b>Round Robin</b>	DC1	0.50	1.66	2.16
<b>ESCE</b>	DC1	0.50	1.66	2.16
<b>Throttled</b>	DC1	0.50	1.66	2.16
<b>Honeybee</b>	DC1	0.50	1.66	2.16
<b>PSO</b>	DC1	0.50	1.66	2.16

#### *Γ3)Κόστος για την πολιτική Reconfigure Dynamically with L*

*Πίνακας 5.19 Κόστος κάθε αλγορίθμου*

<b>Name of Algorithm</b>	<b>Data Center</b>	<b>VM Cost \$</b>	<b>Data Transfer Cost \$</b>	<b>Total \$</b>
<b>Round Robin</b>	DC1	3.35	1.66	5.01
<b>ESCE</b>	DC1	3.35	1.66	5.01
<b>Throttled</b>	DC1	3.35	1.66	5.01
<b>Honeybee</b>	DC1	3.35	1.66	5.01
<b>PSO</b>	DC1	3.35	1.66	5.01

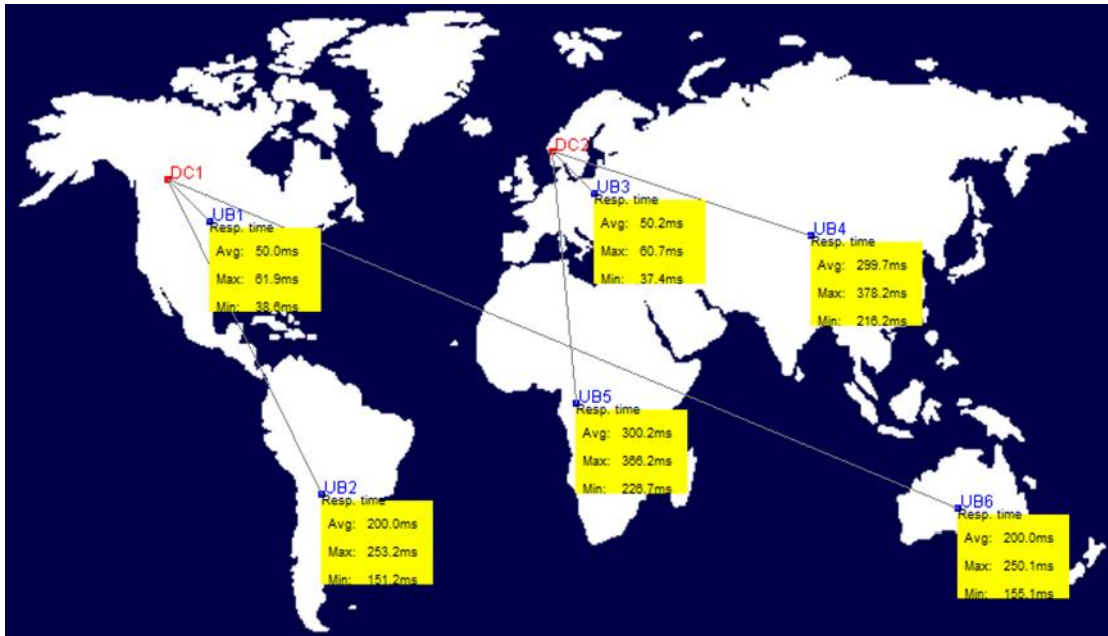


### 5.3 Σενάριο 3- Προσομοίωση με 2 DC & 5 UB σε διαφορετικές περιοχές

#### 5.3.1 Διαμόρφωση σεναρίου

**Σκοπός:** Σκοπός του σεναρίου είναι η κατανόηση της λειτουργίας των αλγορίθμων με τη χρήση δυο κέντρων δεδομένων και έξι βάσεων χρηστών.

**Διαμόρφωση:** Στο τρίτο σενάριο πραγματοποιήσαμε την προσομοίωση, τοποθετώντας 2 κέντρα δεδομένων και 6 βάσεις χρηστών με τις ίδιες τιμές που θέσαμε στο προηγούμενο σενάριο, καθώς και με τον ίδιο αριθμό χρηστών. Το DC1 βρίσκεται στη βόρεια Αμερική (region 0), και το DC2 στην Ευρώπη (region2). Τέλος, έγινε επανάληψη του πειράματος για κάθε πολιτική μεσίτη.



Εικόνα 5.6 Γραφική οθόνη εξόδου

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC1	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC2	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Εικόνα 5.7 Διαμόρφωση του κέντρου δεδομένων

### 5.3.2 Αποτελέσματα

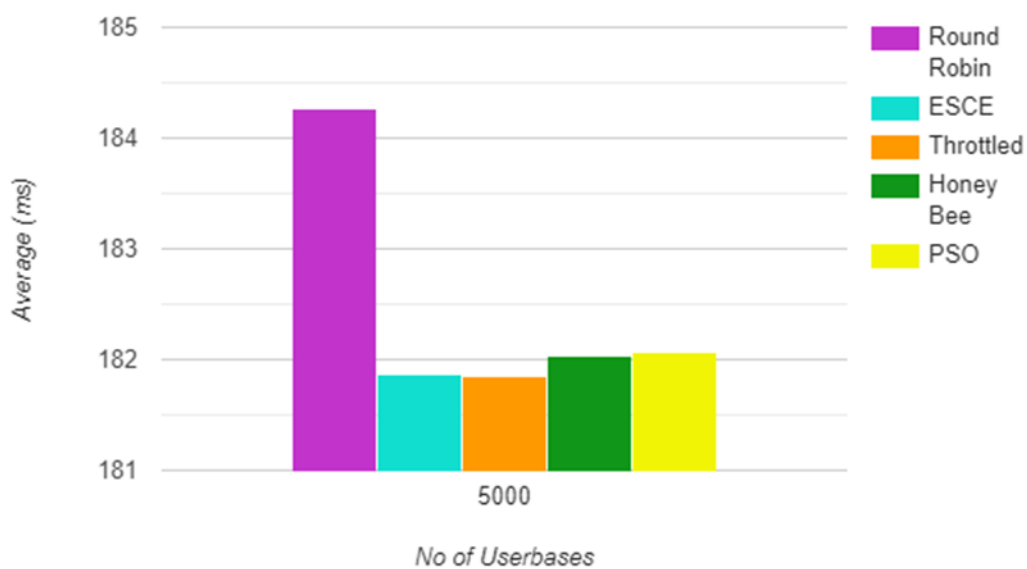
#### A) Χρόνος απόκρισης

Τα αποτελέσματα για τον μέσο συνολικό χρόνο απόκρισης κάθε πολιτικής απεικονίζονται στα Γραφήματα 4-7, 4-8, 4-9.

#### A1) Χρόνος απόκρισης για την πολιτική *Closest Data Center*

Πίνακας 5.20 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	184.27	181.86	181.85	182.04	182.07

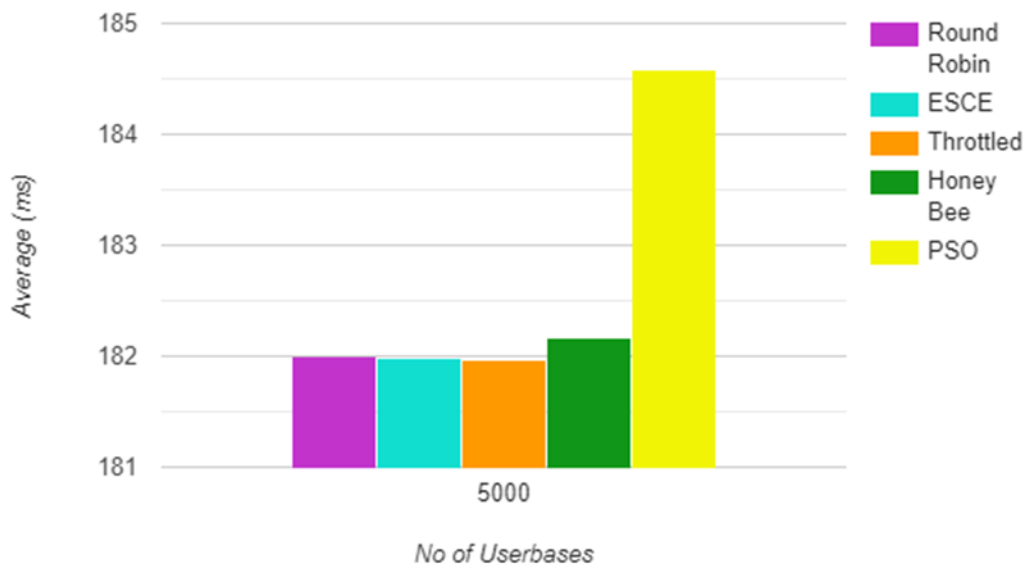


Γράφημα 5-7 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### A2) Χρόνος απόκρισης για την πολιτική *Optimise Response Time*

Πίνακας 5.21 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	182.00	181.98	181.97	182.16	184.58

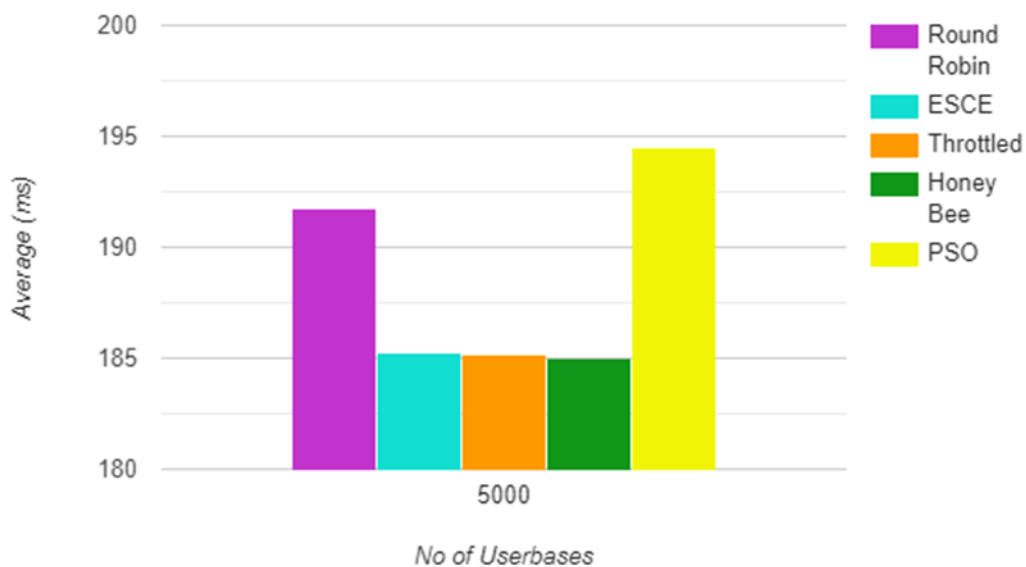


Γράφημα 5-8 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### A3) Χρόνος απόκρισης για την πολιτική *Reconfigure Dynamically with L*

Πίνακας 5.22 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
5000	191.74	185.23	185.19	185.03	194.52



Γράφημα 5-9 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### B) Χρόνος επεξεργασίας κέντρου δεδομένων

#### B1) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική Closest Data Center

Πίνακας 5.23 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

Avg of Data Center processing time (ms)					
Name of Algorithm	Round Robin	ESCE	Throttled	Honeybee	PSO
Data Center Processing Time	0.34	0.37	0.37	0.56	0.60

#### B2) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική Optimise Response Time

Πίνακας 5.24 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

Avg of Data Center processing time (ms)					
Name of Algorithm	Round Robin	ESCE	Throttled	Honeybee	PSO
Data Center Processing Time	0.37	0.37	0.37	0.56	3.35

**B3) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική Reconfigure Dynamically with L**

Πίνακας 5.25 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Avg of Data Center processing time (ms)</b>					
<b>Name of Algorithm</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>Data Center Processing Time</b>	0.37	0.37	0.37	0.56	3.35

**Γ) Κόστος**

**Γ1) Κόστος για την πολιτική Closest Data Center**

Πίνακας 5.26 Κόστος κάθε αλγορίθμου

<b>Name of Algorithm</b>	<b>Data Center</b>	<b>VM Cost \$</b>	<b>Data Transfer Cost \$</b>	<b>Total \$</b>
<b>Round Robin</b>	DC1	0.50	0.96	1.46
	DC2	0.50	0.95	1.46
<b>ESCE</b>	DC1	0.50	0.71	1.21
	DC2	0.50	0.95	1.45
<b>Throttled</b>	DC1	0.50	0.71	1.21
	DC2	0.50	0.95	1.45
<b>Honeybee</b>	DC1	0.50	0.71	1.21
	DC2	0.50	0.95	1.45
<b>PSO</b>	DC1	0.50	0.71	1.21

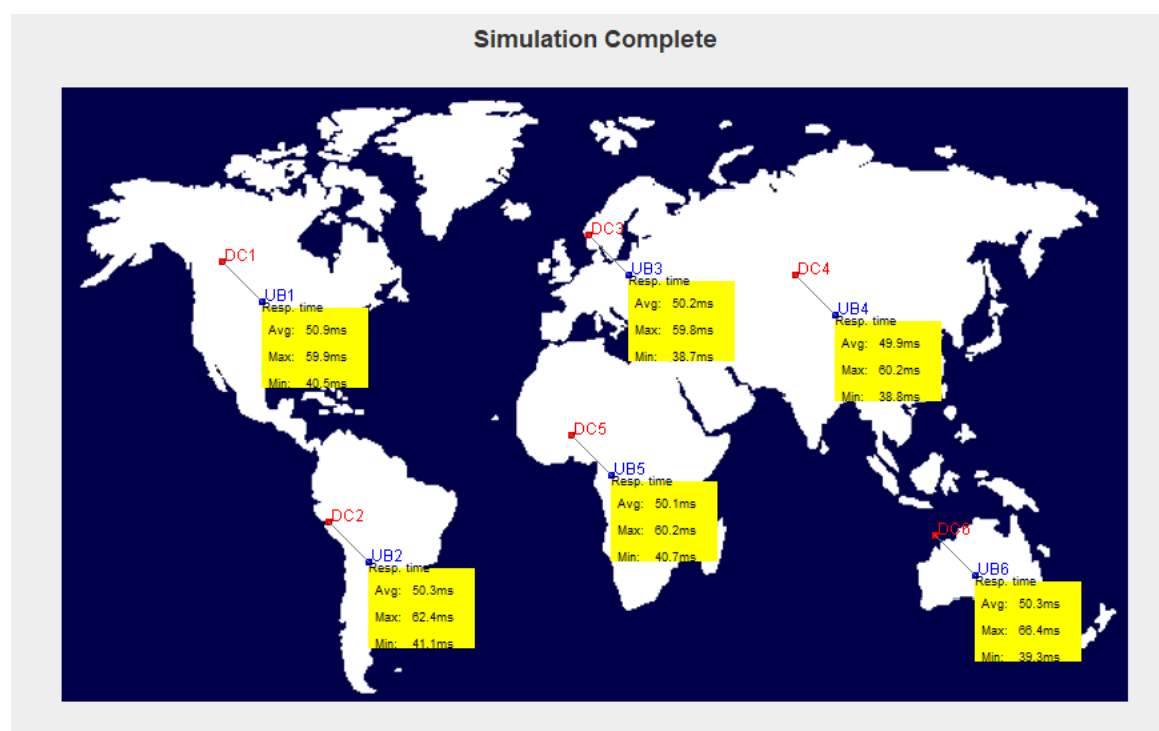
Το κόστος για τις πολιτικές Optimise Response Time και Reconfigure Dynamically with L παραμένει το ίδιο.

## 5.4 Σενάριο 4 - Περίπτωση Μελέτης με τη χρήση έξι Data Center

### 5.4.1 Διαμόρφωση σεναρίου

**Σκοπός:** Σύμφωνα με την μελέτη των Singh, κ.ά. [33], οι οποίοι για την υλοποίηση των διαφόρων αλγορίθμων, προσομοίωσαν το περιβάλλον λαμβάνοντας 6 βάσεις χρηστών και 6 κέντρα δεδομένων, με 6 VM σε κάθε κέντρο δεδομένων, δοκιμάζουμε να εκτελέσουμε με τις ίδιες παραμέτρους την συγκεκριμένη προσομοίωση.

**Διαμόρφωση:** Κάθε προσομοίωση εκτελείται για 60 ώρες με αριθμό χρηστών ίσο με 10000 και τους μέσους χρήστες εκτός αιχμής ως 100 σε κάθε βάση χρηστών. Η πολιτική μεσίτη υπηρεσιών που χρησιμοποιείται για την πρώτη προσομοίωση είναι η Closest Data Center και έπειτα ακολουθούν και οι επόμενες δυο.



Εικόνα 5.8 Γραφική οθόνη εξόδου με 6 διαφορετικά Data Centers και 6 User Bases αντίστοιχα

### 5.4.2 Αποτελέσματα

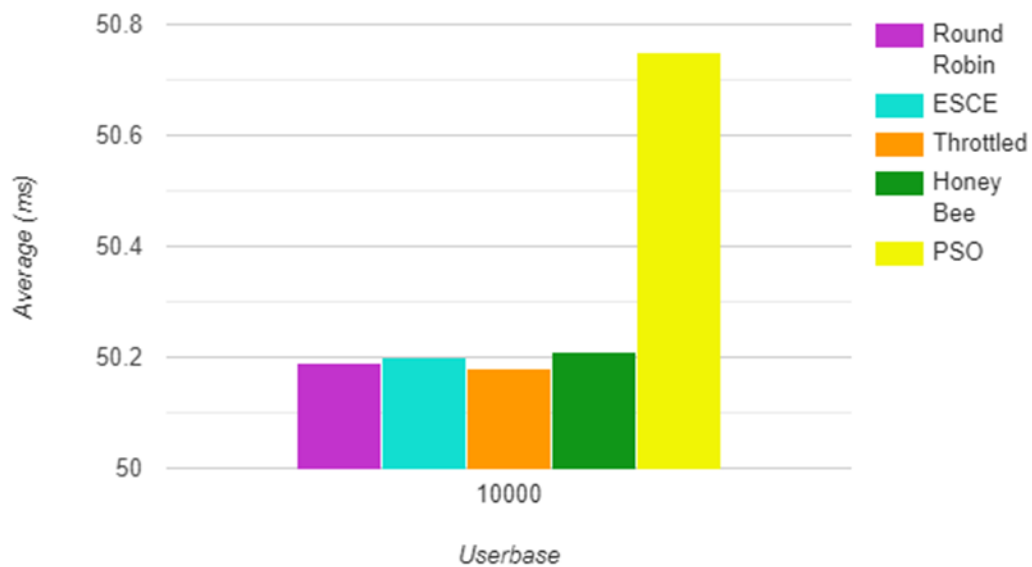
#### A) Χρόνος απόκρισης

Τα αποτελέσματα για τον μέσο συνολικό χρόνο απόκρισης κάθε πολιτικής απεικονίζονται στα Γραφήματα 4-10, 4-11, 4-12.

### A1) Χρόνος απόκρισης για την πολιτική *Closest Data Center*

Πίνακας 5.27 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
10000	50.19	50.20	50.18	50.21	50.75

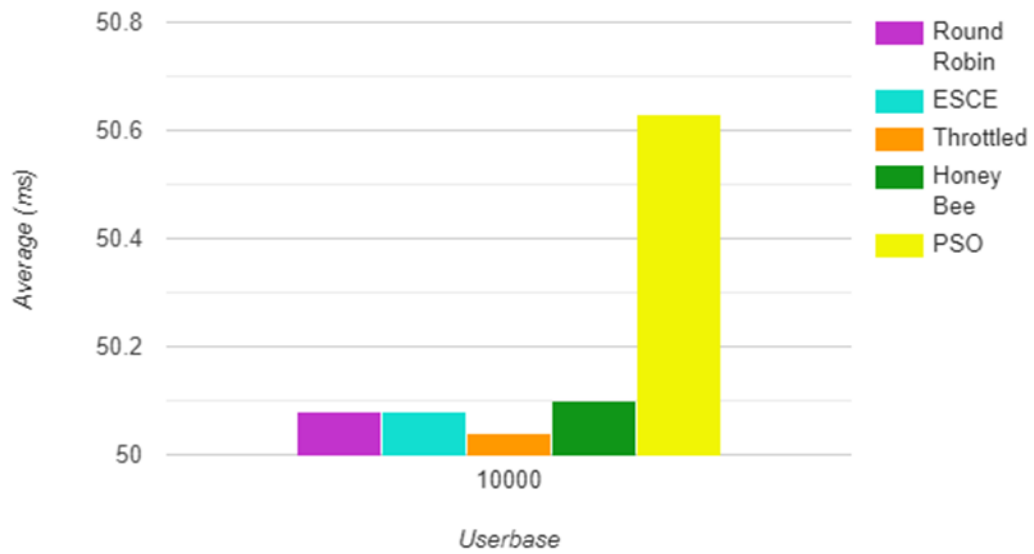


Γράφημα 5-10 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### A2) Χρόνος απόκρισης για την πολιτική *Optimise Response Time*

Πίνακας 5.28 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
10000	50.08	50.08	50.04	50.10	50.63



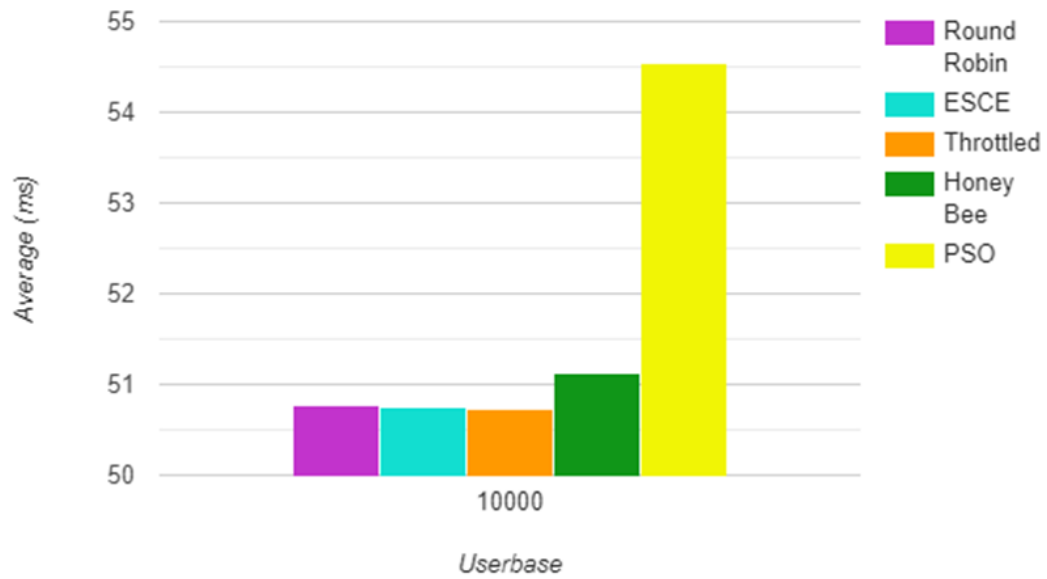
Γράφημα 5-11 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

### A3) Χρόνος απόκρισης για την πολιτική *Reconfigure Dynamically with L*

Πίνακας 5.29 Σύνοψη του μέσου όρου του συνολικού χρόνου απόκρισης (ORT) σε ms

Average of ORT (ms)					
No. of users	Round Robin	ESCE	Throttled	Honeybee	PSO
<b>10000</b>	50.77	50.76	50.72	51.13	54.55





Γράφημα 5-12 Σύγκριση του μέσου όρου χρόνου απόκρισης ORT (ms)

**B) Χρόνος επεξεργασίας κέντρον δεδομένων**

**B1) Χρόνος επεξεργασίας κέντρον δεδομένων για την πολιτική Closest Data Center**

Πίνακας 5.30 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρον δεδομένων

Avg of Data Center processing time (ms)					
Name of Algorithm	Round Robin	ESCE	Throttled	Honeybee	PSO
Data Center Processing Time	0.50	0.53	0.50	0.54	1.08

**B2) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Optimise Response Time***

Πίνακας 5.31 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Avg of Data Center processing time (ms)</b>					
<b>Name of Algorithm</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>Data Center Processing Time</b>	0.49	0.52	0.48	0.54	1.07

**B3) Χρόνος επεξεργασίας κέντρου δεδομένων για την πολιτική *Reconfigure Dynamically with L***

Πίνακας 5.32 Μέσος όρος του χρόνου επεξεργασίας(ms) κέντρου δεδομένων

<b>Avg of Data Center processing time (ms)</b>					
<b>Name of Algorithm</b>	<b>Round Robin</b>	<b>ESCE</b>	<b>Throttled</b>	<b>Honeybee</b>	<b>PSO</b>
<b>Data Center Processing Time</b>	1.09	1.09	1.05	1.46	4.88

**Γ) Κόστος**

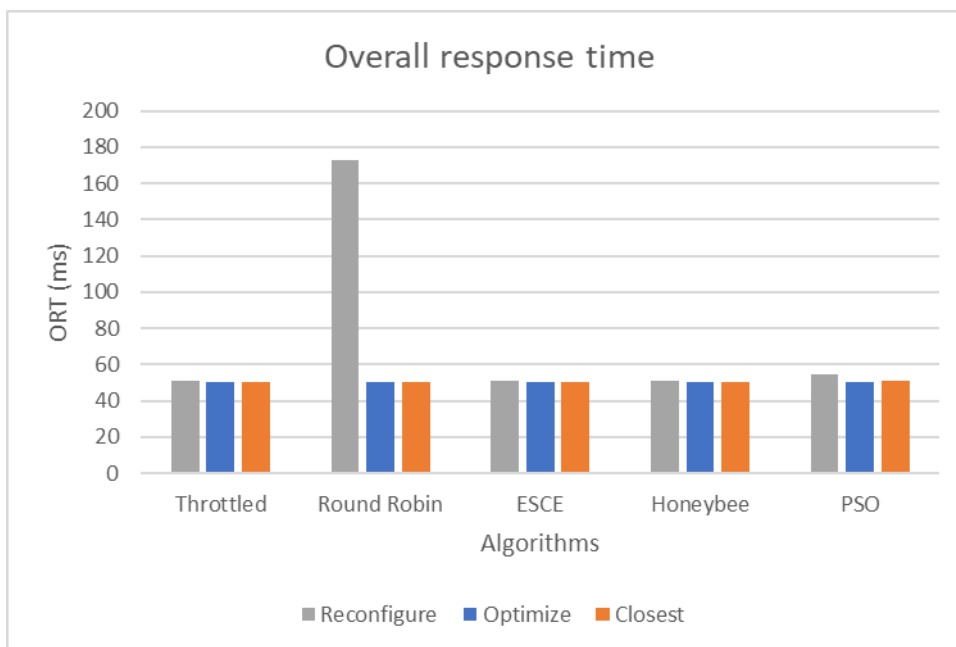
**Γ1) Κόστος και για τις τρεις πολιτικές**

Πίνακας 5.33 Κόστος κάθε αλγορίθμου

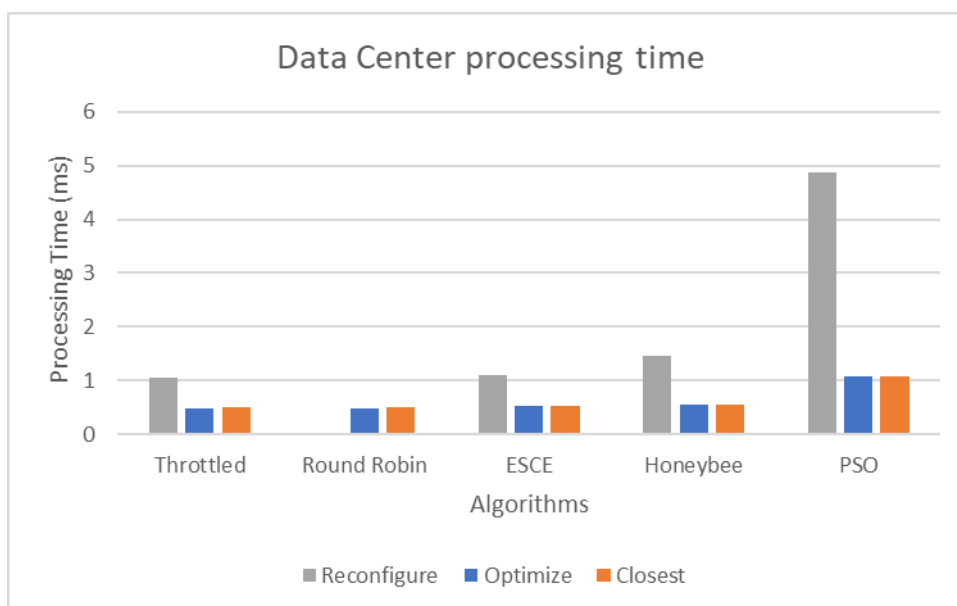
<b>Name</b>	<b>CDC</b>	<b>ORT</b>	<b>RD</b>
<b>Round Robin</b>	5.25\$	3.40\$	18.77\$
<b>ESCE</b>	4.65\$	3.36\$	19.10\$
<b>Throttled</b>	5.15\$	3.36\$	19.47\$
<b>Honeybee</b>	3.36\$	3.36\$	19.16\$
<b>PSO</b>	3.36\$	3.36\$	18.60\$

#### Δ) Σύγκριση πολιτικών

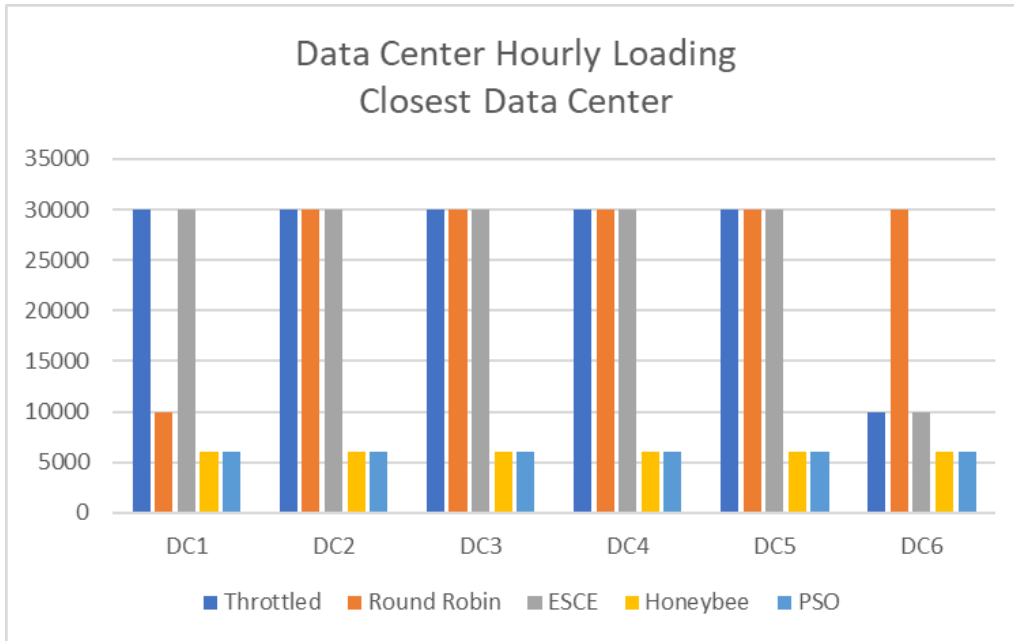
Στα επόμενα γραφήματα ακολουθεί η σύγκριση των τριών πολιτικών για το συνολικό χρόνο απόκρισης, το χρόνο επεξεργασίας του κέντρου δεδομένων, τον ωριαίο φόρτο του κέντρου δεδομένων και το κόστος.



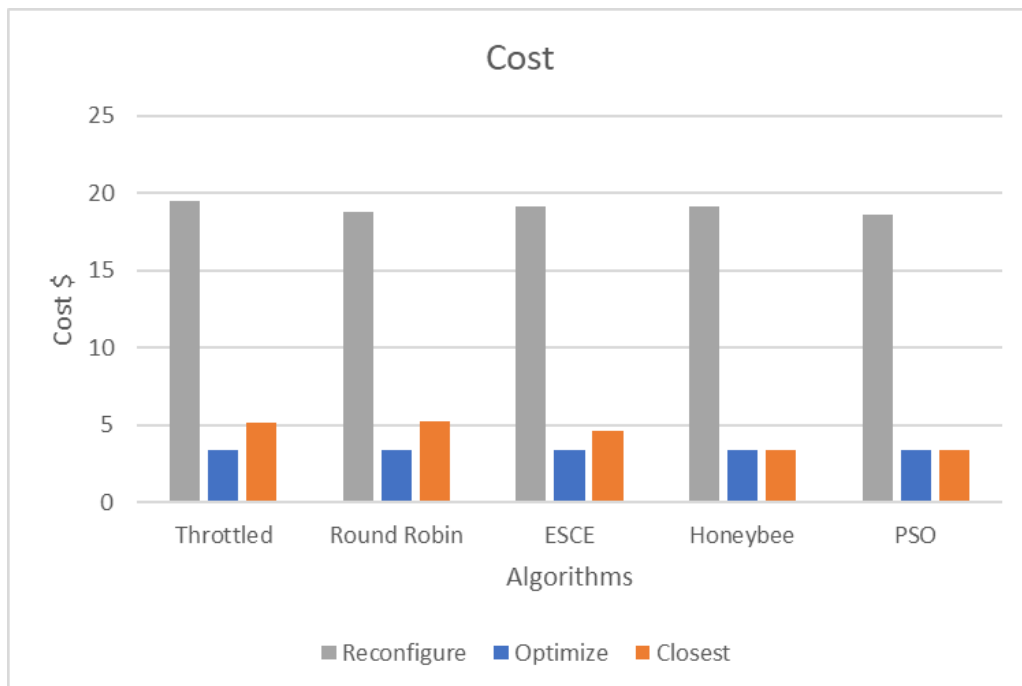
Γράφημα 5-13 Σύγκριση του συνολικού χρόνου απόκρισης και για τις τρεις πολιτικές



Γράφημα 5-14 Σύγκριση του χρόνου επεξεργασία του κέντρου δεδομένων



Γράφημα 5-15 Σύγκριση του ωριαίου φόρτου για την πολιτική Closest Data Center



Γράφημα 5-16 Σύγκριση κόστους για όλες τις πολιτικές

## 5.5 Αποτελέσματα και Συγκριτική Ανάλυση

### **Σενάριο 1:**

Στο πρώτο σενάριο εξετάσαμε τη συμπεριφορά των αλγορίθμων για ένα κέντρο δεδομένων και έξι βάσεις χρηστών, τοποθετώντας τα όλα σε μια ήπειρο, τη βόρεια Αμερική. Με βάση τα αποτελέσματα των γραφημάτων, είμαστε σε θέση να εξάγουμε ορισμένα συμπεράσματα. Για την πολιτική Closest Data Center παρατηρούμε ότι για λιγότερους από 5000 χρήστες δεν υπάρχει σχεδόν καμία διαφορά που μπορεί να αναφερθεί στην απόδοση μεταξύ των αλγορίθμων. Με μια πρώτη ματιά παρατηρούμε ότι ο χρόνος απόκρισης του αλγόριθμου PSO από τους 25000 χρήστες και μετά είναι ο χειρότερος συγκριτικά με της υπόλοιπους τέσσερις. Επιπρόσθετα, όσο αυξάνεται ο αριθμός των χρηστών, ο αλγόριθμος Throttled φαίνεται να είναι αυτός που ανταποκρίνεται καλύτερα από όλους. Ακόμα, ο μέσος όρος του χρόνου επεξεργασίας του κέντρου δεδομένων διαφέρει σε κάθε πολιτική ΕΦ. Επίσης, και με τη χρήση πολιτικής μεσίτη Optimise Response Time καθώς και με την Reconfigure Dynamically with L, ο αλγόριθμος PSO παραμένει ο χειρότερος. Στη τελευταία περίπτωση για την πολιτική Reconfigure Dynamically with L η προσομοίωση πραγματοποιήθηκε έως και για 50000 χρήστες, διότι για μεγαλύτερο αριθμό χρηστών, σχεδόν όλοι αλγόριθμοι εκτός από τον Throttled, καθυστερούσαν υπερβολικά να δώσουν αποτελέσματα.

### **Σενάριο 2:**

Στο δεύτερο σενάριο μελετήσαμε τη συμπεριφορά των αλγορίθμων πάλι για ένα κέντρο δεδομένων και έξι βάσεις χρηστών, όμως αυτή τη φορά οι βάσεις δεν ήταν συγκεντρωμένες όλες σε μια μόνο ήπειρο, αλλά ήταν ομοιόμορφα κατανομημένες και στις έξι διαφορετικές περιοχές ( region0 – region5). Αυτό που παρατηρούμε είναι για την πρώτη πολιτική που εξετάσαμε, δηλαδή την Closest Data Center είναι ότι ο χειρότερος αλγόριθμος, αυτός δηλαδή με τον μεγαλύτερο χρόνο απόκρισης, άρα και τη χειρότερη απόδοση είναι ο PSO. Στη συνέχεια ακολουθεί ο αλγόριθμος Honeybee. Επιπλέον οι αλγόριθμοι Throttled και ESCE δίνουν ίδια αποτελέσματα. Τέλος, ο αλγόριθμος Round Robin φαίνεται να είναι αυτός που αποδίδει καλύτερα συγκριτικά με τους υπόλοιπους. Για την πολιτική Optimise Response Time ο χειρότερος

αλγόριθμος είναι για ακόμη μια φορά ο PSO, και ο καλύτερος ο Throttled. Επιπλέον το κόστος παραμένει το ίδιο με την προηγούμενη πολιτική. Τέλος υπάρχουν μικρές διαφοροποιήσεις στο μέσο όρο του χρόνου επεξεργασίας. Για την Τρίτη και τελευταία πολιτική δηλαδή την Reconfigure Dynamically with L, τα αποτελέσματα είναι αρκετά ενδιαφέροντα καθώς παρατηρούνται ορισμένες διαφοροποιήσεις. Οι PSO & Round Robin είναι οι δυο χειρότεροι αλγόριθμοι όσον αφορά το μέσο όρο του χρόνου απόκρισης. Επίσης στη συγκεκριμένη προσομοίωση ο βέλτιστος αλγόριθμος είναι ο ESCE και όχι ο Throttled όπως αναμέναμε. Τέλος τα κόστη είναι αυξημένα συγκριτικά με τις δυο προηγούμενες περιπτώσεις, το ίδιο ισχύει και το χρόνο επεξεργασίας του κέντρου δεδομένων.

### ***Σενάριο 3:***

Όσον αφορά το τρίτο σενάριο, επιλέξαμε να χρησιμοποιήσουμε δυο κέντρα δεδομένων, όπου το πρώτο βρίσκεται στη βόρεια Αμερική και το δεύτερο στην Ευρώπη, καθώς και έξι βάσεις χρηστών ομοιόμορφα κατανεμημένες. Αυτό που συμπεράναμε συγκριτικά με τα δυο προηγούμενα σενάρια είναι ότι η ύπαρξη δυο κέντρων δεδομένων μειώνει σημαντικά το μέσο όρο του χρόνου απόκρισης, πράγμα το οποίο είναι απόλυτα λογικό, εφόσον ο φόρτος των βάσεων χρηστών διαμοιράζεται. Επίσης, αξίζει να σημειωθεί ότι με τη χρήση της πολιτικής Optimise Response Time, οι βάσεις χρηστών UB4 και UB5 εξυπηρετούνται και από τα δύο κέντρα δεδομένων.

### ***Σενάριο 4:***

Στο τέταρτο και τελευταίο σενάριο, δοκιμάσαμε να εκτελέσουμε το πείραμα τοποθετώντας έξι κέντρα δεδομένων και έξι βάσεις χρηστών. Οι χρόνοι ήταν σαφώς καλύτεροι και για τις τρεις πολιτικές μεσίτη, αφού κάθε βάση χρηστών εξυπηρετείται από το δικό της κέντρο δεδομένων. Για την Optimise Response Time ο χρόνος απόκρισης, ο χρόνος επεξεργασίας και το κόστος είναι καλύτεροι σε σχέση με τις άλλες δυο πολιτικές, εφόσον ο φόρτος εργασίας ισομοιράζεται. Τέλος, όσον αφορά τη σύγκριση πολιτικών αυτό που παρατηρήσαμε είναι ότι για τη πολιτική, Reconfigure Dynamically with L, ο αλγόριθμος RR έχει τον μεγαλύτερο χρόνο απόκρισης. Αντίστοιχα, ο PSO έδωσε τον χειρότερο χρόνο επεξεργασίας κέντρου δεδομένων.

## Κεφάλαιο 6°

### 6 Σύνοψη

Συνοψίζοντας η Εξισορρόπηση Φορτίου είναι μια σημαντική πτυχή στον τομέα του Cloud Computing, για τη βελτίωση της κατανομής του φόρτου εργασίας και την αποτελεσματική χρήση των πόρων, γεγονός που σε αντάλλαγμα μειώνει τον συνολικό χρόνο απόκρισης του συστήματος. Πολλές προσεγγίσεις και αλγόριθμοι έχουν προταθεί για την επίλυση ζητημάτων που σχετίζονται με την ΕΦ όπως για παράδειγμα ο προγραμματισμός εργασιών, η χρήση πόρων κ.λπ. Αυτή η μελέτη παρουσίασε τα προβλήματα που σχετίζονται με την ΕΦ τα οποία συζητήθηκαν μέσω συγκριτικής ανάλυσης των προτεινόμενων αλγορίθμων από ερευνητές τα τελευταία χρόνια. Ωστόσο, εξακολουθούν να υπάρχουν ορισμένα ζητήματα στο περιβάλλον του Cloud, όπως η μετεγκατάσταση των VM, τα ζητήματα ανοχής σφαλμάτων που δεν έχουν ακόμη αντιμετωπιστεί πλήρως.

Όσον αφορά το κομμάτι της προσομοίωσης, συμπεραίνουμε ότι η απόδοση του κάθε αλγορίθμου δεν είναι σταθερή. Εξαρτάται σε πολύ μεγάλο βαθμό από τις τιμές διαφορετικών παραμέτρων που διαμορφώνουν το περιβάλλον του υπολογιστικού νέφους. Εξαρτάται επίσης, αλλά σε μικρότερο βαθμό, από τις υπολογιστικές και επεξεργαστικές δυνατότητες του κάθε υπολογιστή. Προκειμένου να μελετήσει και να αναλύσει πτυχές του Cloud Computing, ο ερευνητής πρέπει να γνωρίζει ότι πολλές διαφορετικές παράμετροι επηρεάζουν την εργασία του. Έτσι, πρέπει να επικεντρωθεί σε όχι περισσότερες από δύο παραμέτρους σε ένα πείραμα. Τέλος, αυτή η μελέτη θα βοηθήσει τους ερευνητές να εντοπίσουν προβλήματα που σχετίζονται με την ΕΦ, ειδικά για να μειωθεί περαιτέρω ο χρόνος απόκρισης και να αποφευχθούν αστοχίες στον διακομιστή.

## Βιβλιογραφία

1. Khiyaita, A., Bakkali, H.E., Zbakh, M., and Kettani, D.E., "Load balancing cloud computing: State of art", 2012 National Days of Network Security and Systems, 2012, σελ. 106-109, doi: 10.1109/JNS2.24923.
2. Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2021). Load balancing techniques in cloud computing environment: A review. *Journal of King Saud University-Computer and Information Sciences*
3. Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: a big picture. *Journal of King Saud University-Computer and Information Sciences*, 32(2), 149-158.
4. Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing—A hierarchical taxonomical classification. *Journal of Cloud Computing*, 8(1), 1-24.
5. Volkova, V. N., Chemenkaya, L. V., Desyatirikova, E. N., Hajali, M., Khodar, A., & Osama, A. (2018, January). Load balancing in cloud computing. In *2018 IEEE conference of russian young researchers in electrical and electronic engineering (EIconRus)* (pp. 387-390). IEEE.
6. Nist: <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>
7. Singh, D., & Shukla, N. K. (2022) Load Balancing in Cloud Computing.
8. Jawhar, M. M., & Osman, H. M. (2022). Quality of Service and Load Balancing in Cloud Computing: A Review. *AL-Rafidain Journal of Computer Sciences and Mathematics*, 16(1), 15-22.
9. Μοντέλα υπηρεσιών Cloud. Διαθέσιμο στο: <https://www-javatpoint-com.translate.goog/cloud-service-models? x tr sl=en& x tr tl=el& x tr hl=el& x tr pto=sc>
10. Μοντέλα ανάπτυξης Cloud. Διαθέσιμο στο: <https://www-guru99-com.translate.goog/cloud-deployment-models.html? x tr sl=en& x tr tl=el& x tr hl=el& x tr pto=sc>
11. Chaczko, Z., Mahadevan, V., Aslanzadeh, S., & Mcdermid, C. (2011, September). Availability and load balancing in cloud computing.



- In *International Conference on Computer and Software Modeling, Singapore* (Vol. 14, pp. 134-140). IACSIT Press.
12. Banisakher, M., Mohammed, D., & Simeon, J. (2018, March). Security analysis of the workload distribution and resource pooling architecture in cloud systems. In *2018 UKSim-AMSS 20th International Conference on Computer Modelling and Simulation (UKSim)* (pp. 114-120). IEEE.
  13. Kumar, P., & Kumar, R. (2019). Issues and challenges of load balancing techniques in cloud computing: A survey. *ACM Computing Surveys (CSUR)*, 51(6), 1-35.
  14. Kumar, A., Kumar, A., Mozammil, S. M., & Vinothini, M. C. (2022) A Literature Survey on Algorithms for the Optimal Load Balancing in Cloud Computing Environments.
  15. Types of Load Balancers. Διαθέσιμο στο:  
<https://www.appviewx.com/education-center/load-balancer-and-types/>
  16. Load balancing in Cloud Computing. Διαθέσιμο στο:  
<https://www.geeksforgeeks.org/load-balancing-in-cloud-computing/>
  17. cloud load balancing. Διαθέσιμο στο:  
<https://www.techtarget.com/searchcloudcomputing/definition/cloud-load-balancing>
  18. Pradhan, A., Bisoy, S. K., & Mallick, P. K. (2020). Load balancing in cloud computing: Survey. In *Innovation in Electrical Power Engineering, Communication, and Computing Technology* (pp. 99-111). Springer, Singapore.
  19. Sriram, G. S. (2022). Challenges of cloud compute load balancing algorithms. *International Research Journal of Modernization in Engineering Technology and Science*, 4(1), 1186-1190.
  20. What is load balancing? Διαθέσιμο στο:  
<https://www.nginx.com/resources/glossary/load-balancing/>
  21. Ghomi, E. J., Rahmani, A. M., & Qader, N. N. (2017). Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, 88, 50-71.
  22. Paul, S. K., & Dhal, S. K. (2022). A Critical analysis on load balancing algorithms in Cloud Environment.

23. Types of LB Algorithms Διαθέσιμο στο:  
[https://www.enjoyalgorithms.com/blog/types-of-load-balancing-algorithms? x\\_tr\\_sl=en& x\\_tr\\_tl=el& x\\_tr\\_hl=el& x\\_tr\\_pto=sc](https://www.enjoyalgorithms.com/blog/types-of-load-balancing-algorithms? x_tr_sl=en& x_tr_tl=el& x_tr_hl=el& x_tr_pto=sc)
24. LB techniques Διαθέσιμο στο:  
[https://kemptechnologies-com.translate.google/load-balancer/load-balancing-algorithms-techniques? x\\_tr\\_sl=en& x\\_tr\\_tl=el& x\\_tr\\_hl=el& x\\_tr\\_pto=sc](https://kemptechnologies-com.translate.google/load-balancer/load-balancing-algorithms-techniques? x_tr_sl=en& x_tr_tl=el& x_tr_hl=el& x_tr_pto=sc)
25. Deepa, T., & Cheelu, D. (2017, August). A comparative study of static and dynamic load balancing algorithms in cloud computing. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 3375-3378). IEEE.
26. Alkhatib, A. A., Alsabbagh, A., Maraqa, R., & Alzubi, S. (2021). Load balancing techniques in cloud computing: Extensive review. *Advances in Science, Technology and Engineering Systems Journal*, 6(2), 860-870.
27. Handur, V. S. (2021). Particle swarm optimization for load balancing in distributed computing systems—a survey. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(1S), 257-265.
28. Xu, M., Tian, W., & Buyya, R. (2017). A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurrency and Computation: Practice and Experience*, 29(12), e4123.
29. Cloud Analyst Διαθέσιμο στο:  
[http://www.cloudbus.org/students/MEDC\\_Project\\_Report\\_Bhathiya\\_318282.pdf](http://www.cloudbus.org/students/MEDC_Project_Report_Bhathiya_318282.pdf)
30. Pradhan, A., Bisoy, S. K., & Mallick, P. K. (2020). Load balancing in cloud computing: Survey. In *Innovation in Electrical Power Engineering, Communication, and Computing Technology* (pp. 99-111). Springer, Singapore.
31. Arulkumar, V., & Bhalaji, N. (2021). Performance analysis of nature inspired load balancing algorithm in cloud environment. *Journal of Ambient Intelligence and Humanized Computing*, 12, 3735-3742.
32. Ahmad, A. Y., & Hammo, A. Y. (2022). A Comparative Study of the Performance of Load Balancing Algorithms Using Cloud Analyst. *Webology*, 19(1), 4898-4911.
33. Singh, S. P., Sharma, A., & Kumar, R. (2016). Analysis of load balancing algorithms using cloud analyst. *International Journal of Grid and Distributed Computing*, 9(9), 11-24.

