



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΔΙΑΧΕΙΡΙΣΗ ΠΟΡΩΝ ΚΑΙ ΕΡΓΑΣΙΩΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ
ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΟΜΙΧΛΗΣ**

ΓΕΩΡΓΙΟΣ ΜΠΙΛΙΑΛΗΣ

Επιβλέπουσα : Σπυριδούλα Μαργαρίτη

ΕΔΙΠ, Α

Άρτα, Ιούλιος, 2025

Resource and task management in Fog Computing

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, 07/07/2025

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπουσα καθηγήτρια

Σπυριδούλα Μαργαρίτη

2. Μέλος επιτροπής

Δημήτρης Λιαροκάπης,

3. Μέλος επιτροπής

Ελευθέριος Στεργίου

© Μπιλιάλης, Γεώργιος, 2025.

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.

Μπιλιάλης Γεώργιος

A handwritten signature in black ink, appearing to be 'Miliailis', written over a horizontal line.

Υπογραφή

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει το πεδίο της υπολογιστικής ομίχλης (fog computing), δίνοντας έμφαση στη διαχείριση πόρων και στην κατανομή εργασιών σε κατανεμημένα υπολογιστικά περιβάλλοντα. Αρχικά, παρουσιάζονται οι βασικές αρχές, τα πλεονεκτήματα, οι προκλήσεις και οι εφαρμογές της υπολογιστικής ομίχλης σε σύγκριση με το cloud και το edge computing. Στη συνέχεια, αναλύονται οι κύριοι

αλγόριθμοι κατανομής εργασιών, τα σχετικά κριτήρια αξιολόγησης και τα σημαντικότερα εργαλεία προσομοίωσης, με ιδιαίτερη αναφορά στο FogWorkflowSim. Το πειραματικό μέρος περιλαμβάνει προσομοιώσεις με στόχο τη σύγκριση αλγορίθμων προγραμματισμού εργασιών ως προς τον χρόνο εκτέλεσης, την κατανάλωση ενέργειας και το κόστος. Τα αποτελέσματα δείχνουν ότι οι εξελιγμένοι αλγόριθμοι προσφέρουν βελτιωμένη απόδοση σε δυναμικά περιβάλλοντα, ενώ η υπολογιστική ομίχλη αναδεικνύεται ως κρίσιμη τεχνολογία για σύγχρονες, απαιτητικές εφαρμογές.

Abstract

This thesis examines the field of fog computing, focusing on resource management and task scheduling in distributed computing environments. Initially, the fundamental principles, advantages, challenges, and applications of fog computing are presented, with comparisons to cloud and edge computing. The study analyzes the main task scheduling algorithms, relevant evaluation criteria, and the most important simulation tools, with a particular emphasis on FogWorkflowSim. The experimental part includes simulations aimed at comparing scheduling algorithms in terms of execution time, energy consumption, and cost. The results indicate that advanced algorithms offer improved performance in dynamic environments, while fog computing emerges as a key technology for modern, demanding applications. Overall, this research highlights the benefits and challenges of fog computing, providing insights and useful conclusions for its optimal use in next-generation distributed systems.

Πίνακας περιεχομένων

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	11
ΚΕΦΑΛΑΙΟ 2: ΥΠΟΛΟΓΙΣΤΙΚΗ ΟΜΙΧΛΗ	13
2.1 Ορισμός και Βασικές Έννοιες.....	13
2.2 Σύγκριση της Υπολογιστικής Ομίχλης με το Cloud και το Edge Computing	14
2.3 Αρχιτεκτονική και Τοπολογία Υπολογιστικής Ομίχλης	15
2.4 Πλεονεκτήματα και Περιορισμοί της Υπολογιστικής Ομίχλης.....	17
2.5 Εφαρμογές Υπολογιστικής Ομίχλης (Fog Computing)	18
2.6 Διαχείριση Πόρων σε Περιβάλλοντα Υπολογιστικής Ομίχλης	20
2.7 Ασφάλεια και Προστασία Δεδομένων	21
2.8 Εργαλεία Προσομοίωσης στην Υπολογιστική Ομίχλη.....	23
ΚΕΦΑΛΑΙΟ 3 : ΜΗΧΑΝΙΣΜΟΙ ΚΑΙ ΣΤΡΑΤΗΓΙΚΕΣ ΚΑΤΑΝΟΜΗΣ ΠΟΡΩΝ ΚΑΙ ΕΡΓΑΣΙΩΝ.....	26
3.1 Προκλήσεις Διαχείρισης Πόρων σε Περιβάλλοντα Υπολογιστικής Ομίχλης	26
3.2 Μηχανισμοί Κατανομής Πόρων	27
3.3 Κατανομή Εργασιών (Task Scheduling).....	29
3.4 Στρατηγικές Κατανομής Εργασιών.....	30
3.5 Αλγόριθμοι Κατανομής Εργασιών στην Υπολογιστική Ομίχλη.....	32
3.6 Κριτήρια Αξιολόγησης Κατανομής Εργασιών	34
ΚΕΦΑΛΑΙΟ 4 : FOGWORKFLOWSIM.....	36
4.1 Γενική Περιγραφή και Λειτουργία.....	36
4.2 Κύρια χαρακτηριστικά	36
4.3 Υποστηριζόμενα σενάρια προσομοίωσης.....	38
4.4 Τεχνική Αρχιτεκτονική.....	38
4.5 Βασικές Κλάσεις και Μονάδες του Συστήματος.....	40
4.6 Ροή Δεδομένων και Λειτουργία Προσομοίωσης	43
ΚΕΦΑΛΑΙΟ 5: ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ	47
5.1 Ρύθμιση Περιβάλλοντος & Προετοιμασία	47
5.2 Σενάρια Προσομοίωσης.....	48
5.3 Εκτέλεση & Παρατήρηση.....	50
5.4 Ανάλυση Αποτελεσμάτων	52
5.4.1 Χρόνος Εκτέλεσης.....	53

5.4.2 Κατανάλωση Ενέργειας	55
5.4.3 Κόστος Εκτέλεσης	57
5.4.4 Σύγκριση GA vs PSO και παραδοσιακών αλγορίθμων	58
5.5 Συμπεράσματα Αποδοτικότητας	59
ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ	61
ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ.....	64

Κατάλογος Εικόνων

<u>Εικόνα 1: Αρχιτεκτονική FogWorkflowSim</u>	42
<u>Εικόνα 2: Σύγκριση χρόνων εκτέλεσης</u>	61
<u>Εικόνα 3: Σύγκριση ως προς τη κατανάλωση ενέργειας μεταξύ των PSO και GA</u>	63
<u>Εικόνα 4: Σύγκριση ως προς το κόστος εκτέλεσης μεταξύ των PSO και GA</u>	65

Κατάλογος Πινάκων

<u>Πίνακας 1: Συνοπτικά συγκριτικά αποτελέσματα</u>	60
---	----

1 ΚΕΦΑΛΑΙΟ: ΕΙΣΑΓΩΓΗ

Η αλματώδης πρόοδος των Τεχνολογιών Πληροφορικής και Επικοινωνιών (ΤΠΕ), σε συνδυασμό με την εκρηκτική εξάπλωση του Διαδικτύου των Πραγμάτων (Internet of Things - IoT), έχει οδηγήσει σε μια νέα εποχή υπολογιστικής επεξεργασίας δεδομένων, όπου η ταχύτητα απόκρισης, η τοπική ανάλυση και η αποκεντρωμένη λήψη αποφάσεων αποτελούν θεμελιώδεις απαιτήσεις για τη λειτουργία σύγχρονων εφαρμογών. Σε αυτό το περιβάλλον, η παραδοσιακή προσέγγιση του υπολογιστικού νέφους (cloud computing) εμφανίζει περιορισμούς, ιδιαίτερα σε σενάρια όπου η καθυστέρηση μετάδοσης και η ανάγκη για άμεση επεξεργασία δεδομένων γίνονται κρίσιμες.

Η υπολογιστική ομίχλη (Fog Computing) έρχεται να καλύψει το κενό αυτό, προσφέροντας ένα ενδιάμεσο, κατανεμημένο επίπεδο επεξεργασίας μεταξύ του κέντρου δεδομένων (cloud) και των άκρων του δικτύου (edge). Μέσα από την προσέγγιση αυτή, οι υπολογιστικοί πόροι μεταφέρονται πιο κοντά στις πηγές δεδομένων, επιτρέποντας τη γρήγορη, τοπική ανάλυση και την ελαχιστοποίηση της καθυστέρησης. Η τεχνολογία αυτή έχει βρει εφαρμογή σε ποικίλους τομείς, όπως οι έξυπνες πόλεις, η βιομηχανική αυτοματοποίηση, τα συστήματα υγείας, τα αυτόνομα οχήματα και τα σύγχρονα δίκτυα 5G, όπου η ανάγκη για ταχύτατη και ασφαλή επεξεργασία πληροφοριών είναι πιο επιτακτική από ποτέ.

Ωστόσο, η υιοθέτηση της υπολογιστικής ομίχλης συνοδεύεται και από σημαντικές προκλήσεις. Η ετερογένεια των συσκευών και των δικτυακών υποδομών, η δυναμική κατανομή των πόρων, η διασφάλιση της ποιότητας υπηρεσίας (QoS), η ενεργειακή αποδοτικότητα και η προστασία της ιδιωτικότητας αποτελούν βασικά πεδία έρευνας. Ιδιαίτερο ενδιαφέρον παρουσιάζει η ανάπτυξη αλγορίθμων και στρατηγικών για την αποτελεσματική κατανομή εργασιών (task scheduling) σε περιβάλλοντα fog, λαμβάνοντας υπόψη την ποικιλομορφία των εφαρμογών, τις πραγματικές απαιτήσεις χρόνου απόκρισης και τους περιορισμούς των διαθέσιμων πόρων.

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, επιχειρείται μία εις βάθος μελέτη των θεωρητικών και πρακτικών πτυχών της υπολογιστικής ομίχλης, με ιδιαίτερη έμφαση στους μηχανισμούς διαχείρισης και κατανομής εργασιών. Εξετάζονται οι αρχιτεκτονικές και τα μοντέλα λειτουργίας της ομίχλης, οι βασικοί αλγόριθμοι

κατανομής και τα σχετικά κριτήρια αξιολόγησης, ενώ αναλύονται και τα σημαντικότερα εργαλεία προσομοίωσης που χρησιμοποιούνται στη διεθνή ερευνητική κοινότητα.

Επιπλέον, στο πειραματικό μέρος της εργασίας, αξιοποιείται το εργαλείο FogWorkflowSim για την προσομοίωση και συγκριτική ανάλυση διαφορετικών αλγορίθμων κατανομής εργασιών σε περιβάλλοντα υπολογιστικής ομίχλης. Μέσα από τη δημιουργία και αξιολόγηση ρεαλιστικών σεναρίων, αναδεικνύονται τα πλεονεκτήματα, οι περιορισμοί και οι προοπτικές κάθε προσέγγισης, με στόχο την εξαγωγή χρήσιμων συμπερασμάτων για τη βέλτιστη αξιοποίηση της τεχνολογίας fog computing σε σύγχρονες εφαρμογές.

Τέλος, η εργασία αυτή φιλοδοξεί να συμβάλει τόσο στην καλύτερη κατανόηση των βασικών αρχών και των τεχνικών προκλήσεων της υπολογιστικής ομίχλης, όσο και στην παροχή επιστημονικών εργαλείων και αποτελεσμάτων που μπορούν να αξιοποιηθούν σε μελλοντικές ερευνητικές ή βιομηχανικές εφαρμογές.

2 ΚΕΦΑΛΑΙΟ: ΥΠΟΛΟΓΙΣΤΙΚΗ ΟΜΙΧΛΗ

2.1 Ορισμός και Βασικές Έννοιες

Η υπολογιστική ομίχλη (Fog Computing) αποτελεί ένα αρχιτεκτονικό παράδειγμα που επεκτείνει τις δυνατότητες του υπολογιστικού νέφους (cloud computing) προς την περιφέρεια του δικτύου, φέρνοντας την επεξεργασία δεδομένων πιο κοντά στις πηγές δημιουργίας τους, όπως αισθητήρες, συσκευές IoT και κινητά συστήματα. Πρόκειται για ένα ενδιάμεσο επίπεδο μεταξύ των άκρων του δικτύου (edge) και του κέντρου δεδομένων, το οποίο εξυπηρετεί τη χαμηλή καθυστέρηση, την άμεση απόκριση και τη μείωση του φόρτου στο κεντρικό νέφος. Η έννοια της "ομίχλης" υποδηλώνει αυτήν ακριβώς τη διάχυση των υπολογιστικών πόρων πλησιέστερα στον τελικό χρήστη ή στις συσκευές πεδίου.

Μια από τις βασικές αρχές της υπολογιστικής ομίχλης είναι η αποκεντρωμένη επεξεργασία. Αντί τα δεδομένα να διαβιβάζονται απευθείας σε απομακρυσμένα κέντρα δεδομένων για ανάλυση και λήψη αποφάσεων, η υπολογιστική ομίχλη επιτρέπει την τοπική επεξεργασία μέσω ενδιάμεσων κόμβων, όπως routers, gateways ή μικρά data centers. Αυτοί οι κόμβοι διαθέτουν υπολογιστική ισχύ, αποθηκευτικό χώρο και δυνατότητες δικτύωσης, που τους καθιστούν ικανούς να ανταποκρίνονται σε απαιτήσεις πραγματικού χρόνου.

Η αρχιτεκτονική της ομίχλης χαρακτηρίζεται από τη δυνατότητα κατανομής υπηρεσιών και εφαρμογών σε πολλαπλά επίπεδα του δικτύου. Η προσέγγιση αυτή υποστηρίζει την κλιμακωτή κατανομή των λειτουργιών, επιτρέποντας την ευέλικτη διαχείριση των διαθέσιμων πόρων και την υιοθέτηση διαφορετικών επιπέδων επεξεργασίας ανάλογα με τις ανάγκες της εφαρμογής. Έτσι, ενισχύεται η δυνατότητα υποστήριξης κρίσιμων εφαρμογών, όπως στην υγειονομική περίθαλψη, την αυτόνομη οδήγηση και τις βιομηχανικές αυτοματοποιήσεις, όπου η καθυστέρηση και η αξιοπιστία είναι καίριες.

Επιπλέον, η υπολογιστική ομίχλη ευνοεί την ενισχυμένη ασφάλεια και προστασία της ιδιωτικότητας, καθώς μειώνει την ανάγκη μετάδοσης ευαίσθητων δεδομένων σε απομακρυσμένα κέντρα. Μέσω της τοπικής επεξεργασίας, είναι δυνατή η εφαρμογή

πολιτικών ασφαλείας πλησιέστερα στην πηγή των δεδομένων, ενισχύοντας την εποπτεία και τον έλεγχο από την πλευρά του χρήστη ή του διαχειριστή της συσκευής. Παράλληλα, η δυνατότητα τοπικής αποθήκευσης συμβάλλει στην ανθεκτικότητα του συστήματος, σε περιπτώσεις απώλειας σύνδεσης με το κεντρικό νέφος.

Τέλος, η υπολογιστική ομίχλη συμβάλλει στη μείωση της κατανάλωσης εύρους ζώνης και του ενεργειακού κόστους, καθώς οι τοπικές επεξεργασίες περιορίζουν την ανάγκη συνεχούς αποστολής τεράστιων ποσοτήτων δεδομένων στο cloud. Η αποδοτικότερη χρήση των πόρων και η ελαχιστοποίηση των καθυστερήσεων την καθιστούν ιδανική για περιβάλλοντα με υψηλή κινητικότητα και δυναμικές απαιτήσεις, όπου το κλασικό cloud δεν μπορεί να ανταποκριθεί με την ίδια ταχύτητα και προσαρμοστικότητα.

2.2 Σύγκριση της Υπολογιστικής Ομίχλης με το Cloud και το Edge Computing

Η υπολογιστική ομίχλη (fog computing), αν και συνδέεται στενά τόσο με το cloud computing όσο και με το edge computing, διαμορφώνει μια διακριτή υπολογιστική προσέγγιση που επιχειρεί να γεφυρώσει το χάσμα μεταξύ αυτών των δύο τεχνολογιών. Η διαφοροποίηση βασίζεται κυρίως στην τοποθεσία της επεξεργασίας δεδομένων, στην καθυστέρηση απόκρισης και στη διαχείριση των πόρων και της ασφάλειας.

Σε αντίθεση με την υπολογιστική νέφος (cloud computing), η οποία βασίζεται σε μεγάλα, κεντροποιημένα κέντρα δεδομένων συνήθως απομακρυσμένα από τον τελικό χρήστη, η υπολογιστική ομίχλη προσφέρει ένα κατακευματισμένο μοντέλο επεξεργασίας πιο κοντά στην πηγή των δεδομένων. Το cloud παρέχει σχεδόν απεριόριστους υπολογιστικούς και αποθηκευτικούς πόρους, αλλά υστερεί σε περιπτώσεις που απαιτείται ελάχιστη καθυστέρηση (latency-sensitive applications), όπως στα αυτόνομα οχήματα ή στις βιομηχανικές εφαρμογές. Η ομίχλη έρχεται να καλύψει αυτό το κενό, προσφέροντας ενδιάμεσο επίπεδο επεξεργασίας που συνδυάζει την ευελιξία του cloud με την εγγύτητα του edge.

Σε σύγκριση με το edge computing, η υπολογιστική ομίχλη διαφοροποιείται ως προς το εύρος και τη λειτουργικότητα των ενδιάμεσων κόμβων. Το edge επικεντρώνεται στην απευθείας επεξεργασία δεδομένων στο ίδιο το άκρο του δικτύου – δηλαδή σε συσκευές όπως αισθητήρες ή embedded συστήματα – χωρίς απαραίτητα τη συμμετοχή πιο εξελιγμένων ενδιάμεσων υποδομών. Αντιθέτως, η ομίχλη υποθέτει την ύπαρξη πολλών ενδιάμεσων επιπέδων μεταξύ των edge συσκευών και του cloud, τα οποία έχουν τη δυνατότητα πιο σύνθετης ανάλυσης, αποθήκευσης και επικοινωνίας, διαμορφώνοντας ένα πιο πολυστρωματικό και δυναμικό υπολογιστικό περιβάλλον. Ένα άλλο κρίσιμο στοιχείο σύγκρισης αφορά την κατανομή της νοημοσύνης και των υπηρεσιών. Το cloud τείνει να συγκεντρώνει την "ευφυΐα" στο κέντρο, όπου και πραγματοποιείται η κύρια ανάλυση και λήψη αποφάσεων. Στο edge, η νοημοσύνη περιορίζεται στο επίπεδο της συσκευής ή της εφαρμογής. Η υπολογιστική ομίχλη, ωστόσο, διαχέει τη νοημοσύνη σε διάφορα σημεία του δικτύου, επιτρέποντας ευέλικτες αρχιτεκτονικές με κατανεμημένες αποφάσεις, προσαρμοσμένες στις απαιτήσεις κάθε επιπέδου.

Τέλος, σε θέματα διαχείρισης και ασφάλειας, η υπολογιστική ομίχλη προσφέρει καλύτερη ισορροπία από το edge και περισσότερη ευελιξία από το cloud. Οι fog κόμβοι μπορούν να λειτουργούν ως φίλτρα ή ενδιάμεσοι ελεγκτές, εφαρμόζοντας πολιτικές ασφάλειας, ελέγχου πρόσβασης και προστασίας προσωπικών δεδομένων πιο κοντά στη συσκευή, μειώνοντας παράλληλα τον κίνδυνο έκθεσης που συνδέεται με την αποστολή δεδομένων σε απομακρυσμένα κέντρα.

2.3 Αρχιτεκτονική και Τοπολογία Υπολογιστικής Ομίχλης

Η αρχιτεκτονική της υπολογιστικής ομίχλης είναι πολυεπίπεδη και ιεραρχική, σχεδιασμένη ώστε να υποστηρίξει την αποκεντρωμένη επεξεργασία δεδομένων και την ευέλικτη κατανομή υπηρεσιών μεταξύ των άκρων του δικτύου και των κεντρικών υπολογιστικών υποδομών. Ο κύριος στόχος αυτής της αρχιτεκτονικής είναι να μεταφέρει την υπολογιστική ισχύ όσο το δυνατόν πλησιέστερα στον τελικό χρήστη ή

στις συσκευές που παράγουν δεδομένα, χωρίς να εγκαταλείπει τη δυνατότητα ενοποίησης με το cloud.

Στο βασικό της σχήμα, η αρχιτεκτονική της ομίχλης περιλαμβάνει τρία κύρια επίπεδα: το επίπεδο των edge συσκευών, το ενδιάμεσο επίπεδο των fog κόμβων και το επίπεδο του cloud. Οι edge συσκευές – όπως αισθητήρες, κάμερες, smartphones ή βιομηχανικοί ελεγκτές – παράγουν τεράστιες ποσότητες δεδομένων. Αντί να τα αποστέλλουν απευθείας στο cloud, τα δεδομένα διοχετεύονται πρώτα σε fog κόμβους, που μπορεί να είναι routers, switches, μικροί servers ή ακόμα και τοπικά data centers με υπολογιστική και αποθηκευτική ικανότητα.

Οι fog κόμβοι λειτουργούν ως ενδιάμεσοι φορείς επεξεργασίας, προσφέροντας υπηρεσίες όπως προεπεξεργασία, ανάλυση σε πραγματικό χρόνο, προσωρινή αποθήκευση και δρομολόγηση δεδομένων. Ανάλογα με τη φύση της εφαρμογής, οι κόμβοι αυτοί μπορούν είτε να διαχειριστούν πλήρως τα δεδομένα επί τόπου είτε να τα προωθήσουν προς τα ανώτερα επίπεδα για περαιτέρω επεξεργασία. Αυτή η ικανότητα δυναμικής ανάθεσης καθιστά το σύστημα πιο ευπροσάρμοστο και αποδοτικό. Η τοπολογία των fog υποδομών ποικίλλει ανάλογα με την περίπτωση χρήσης, αλλά συχνά ακολουθεί κατανομημένα ή υβριδικά πρότυπα. Σε μία ιεραρχική τοπολογία, οι fog κόμβοι οργανώνονται σε επίπεδα – τοπικοί, περιφερειακοί και υπερ-τοπικοί – που επιτρέπουν κλιμακωτή μεταφορά και επεξεργασία δεδομένων. Σε ορισμένες περιπτώσεις, εφαρμόζεται οριζόντια επικοινωνία μεταξύ κόμβων (peer-to-peer), επιτρέποντας την άμεση ανταλλαγή δεδομένων χωρίς την ανάγκη επιστροφής στο cloud ή σε ανώτερο κόμβο. Επιπρόσθετα, η αρχιτεκτονική fog computing χαρακτηρίζεται από την υποστήριξη virtualization και containerization τεχνολογιών, που επιτρέπουν την απομόνωση, διαχείριση και ταχεία ανάπτυξη εφαρμογών και υπηρεσιών σε διαφορετικούς κόμβους. Η δυνατότητα παροχής "fog-as-a-service" μοντέλων αναδεικνύει τον ρόλο των fog υποδομών ως ευέλικτες, δυναμικές πλατφόρμες που υποστηρίζουν πλήθος εφαρμογών, από απλή τηλεμετρία έως πολύπλοκες αναλύσεις μηχανικής μάθησης.

2.4 Πλεονεκτήματα και Περιορισμοί της Υπολογιστικής Ομίχλης

Η υπολογιστική ομίχλη έχει αναδειχθεί ως κρίσιμη τεχνολογία για την υποστήριξη σύγχρονων εφαρμογών που απαιτούν επεξεργασία σε πραγματικό χρόνο, αποκεντρωμένη λήψη αποφάσεων και ελαχιστοποίηση της καθυστέρησης. Ωστόσο, παρά τα σημαντικά πλεονεκτήματά της, η προσέγγιση αυτή συνοδεύεται και από τεχνικές, λειτουργικές και οργανωτικές προκλήσεις.

Ένα από τα κύρια πλεονεκτήματα της υπολογιστικής ομίχλης είναι η μείωση της καθυστέρησης. Φέρνοντας την επεξεργασία πιο κοντά στην πηγή παραγωγής δεδομένων, μειώνεται σημαντικά ο χρόνος απόκρισης, καθιστώντας την κατάλληλη για εφαρμογές όπως αυτόνομα οχήματα, ρομποτική, βιομηχανικά δίκτυα και ιατρική παρακολούθηση. Η δυνατότητα λήψης αποφάσεων σε ελάχιστο χρόνο αποτελεί καθοριστικό παράγοντα για περιβάλλοντα με υψηλές απαιτήσεις ακρίβειας και ασφάλειας. Επιπλέον, η αποσυμφόρηση του δικτύου και η μείωση της εξάρτησης από το cloud αποτελούν σημαντικά οφέλη. Με την τοπική προεπεξεργασία και φιλτράρισμα των δεδομένων, περιορίζεται η ανάγκη συνεχούς αποστολής μεγάλων όγκων πληροφορίας στα κεντρικά data centers. Αυτό οδηγεί σε οικονομία εύρους ζώνης, εξοικονόμηση ενέργειας και μεγαλύτερη αποδοτικότητα του συνολικού συστήματος.

Άλλο σημαντικό πλεονέκτημα είναι η βελτιωμένη ασφάλεια και προστασία ιδιωτικότητας. Οι fog κόμβοι επιτρέπουν την εφαρμογή πολιτικών ασφαλείας πιο κοντά στις συσκευές, μειώνοντας τον κίνδυνο υποκλοπής ή απώλειας δεδομένων κατά τη μεταφορά. Παράλληλα, διευκολύνεται η συμμόρφωση με κανονιστικά πλαίσια που απαιτούν τοπική αποθήκευση ή περιορισμό διακίνησης ευαίσθητων πληροφοριών. Η ευελιξία και επεκτασιμότητα της ομίχλης επιτρέπουν την ανάπτυξη κατανεμημένων εφαρμογών με δυνατότητα δυναμικής διαχείρισης πόρων ανάλογα με τις ανάγκες. Οι fog υποδομές μπορούν να προσαρμόζονται σε μεταβαλλόμενα φορτία, υποστηρίζοντας την ταχεία ανάπτυξη νέων υπηρεσιών και την ενοποίηση με υφιστάμενες cloud και edge λύσεις.

Ωστόσο, παρά τα παραπάνω, η υπολογιστική ομίχλη αντιμετωπίζει και σημαντικούς περιορισμούς. Πρώτον, η πολυπλοκότητα της διαχείρισης αυξάνεται λόγω της ετερογένειας των συσκευών, της δυναμικής τοπολογίας και της ανάγκης για

ενορχήστρωση υπηρεσιών σε πολλαπλά επίπεδα. Η ενοποίηση fog κόμβων με τα υπόλοιπα υπολογιστικά στρώματα απαιτεί σύνθετους μηχανισμούς συντονισμού και αυτοματισμού. Ένα ακόμη πρόβλημα είναι η έλλειψη τυποποίησης. Η απουσία καθολικά αποδεκτών προτύπων για την αρχιτεκτονική, την ασφάλεια και τη διαλειτουργικότητα δυσχεραίνει την ευρεία υιοθέτηση της τεχνολογίας και περιορίζει τη συμβατότητα μεταξύ διαφορετικών κατασκευαστών και παρόχων. Η ασφάλεια, αν και πλεονέκτημα σε τοπικό επίπεδο, αποτελεί επίσης πρόκληση, καθώς η αποκέντρωση αυξάνει την επιφάνεια επίθεσης. Οι fog κόμβοι, ιδίως όταν είναι εγκατεστημένοι σε μη προστατευμένα περιβάλλοντα, είναι ευάλωτοι σε φυσικές επιθέσεις, παραβιάσεις ή μη εξουσιοδοτημένη πρόσβαση. Τέλος, η ανάγκη για υπολογιστικούς πόρους σε κάθε σημείο του δικτύου δημιουργεί κόστη σε εξοπλισμό, συντήρηση και ενεργειακή κατανάλωση, ιδίως όταν συγκρίνεται με την πλήρως κεντροποιημένη προσέγγιση του cloud. Το οικονομικό μοντέλο της ομίχλης παραμένει ακόμη υπό διαμόρφωση, ιδίως για μικρούς παρόχους ή αναπτυσσόμενες αγορές.

2.5 Εφαρμογές Υπολογιστικής Ομίχλης (Fog Computing)

Η υπολογιστική ομίχλη έχει αναδειχθεί σε κρίσιμο τεχνολογικό πυλώνα για την υποστήριξη εφαρμογών που απαιτούν υψηλή ταχύτητα απόκρισης, διαχείριση μεγάλου όγκου δεδομένων σε πραγματικό χρόνο και λειτουργία κοντά στις πηγές πληροφορίας. Η αξιοποίησή της επεκτείνεται σε πολλαπλούς τομείς, με χαρακτηριστικά παραδείγματα τις έξυπνες πόλεις, τη Βιομηχανία 4.0, τον χώρο της υγείας, τα δίκτυα IoT και τα αυτόνομα οχήματα, καθώς και τις εφαρμογές που βασίζονται στο δίκτυο 5G.

Στο πλαίσιο των έξυπνων πόλεων, η υπολογιστική ομίχλη παρέχει την απαραίτητη υποδομή για την επεξεργασία δεδομένων από αισθητήρες, κάμερες παρακολούθησης, σταθμούς μέτρησης ρύπανσης και έξυπνες μεταφορές, προσφέροντας άμεση ανάλυση και λήψη αποφάσεων σε τοπικό επίπεδο. Για παράδειγμα, σε ένα σύστημα διαχείρισης κυκλοφορίας, οι fog κόμβοι μπορούν να επεξεργάζονται εικόνες και σήματα από φανάρια και οχήματα σε πραγματικό χρόνο, βελτιστοποιώντας τη ροή της κυκλοφορίας ή αποτρέποντας ατυχήματα χωρίς την ανάγκη αποστολής δεδομένων στο cloud. Παράλληλα, υποστηρίζεται η ευφυής διαχείριση ενέργειας, φωτισμού και απορριμμάτων με μεγαλύτερη αποδοτικότητα και χαμηλότερο κόστος.

Η Βιομηχανία 4.0, που χαρακτηρίζεται από την υιοθέτηση τεχνολογιών όπως το Industrial Internet of Things (IIoT), η ρομποτική και η ανάλυση μεγάλων δεδομένων, βασίζεται έντονα στην υπολογιστική ομίχλη για την υποστήριξη κρίσιμων διεργασιών παραγωγής. Οι fog υποδομές επιτρέπουν την αποκεντρωμένη παρακολούθηση και αυτοματοποίηση γραμμών παραγωγής, επιθεώρηση σε πραγματικό χρόνο, καθώς και την ανίχνευση και διόρθωση σφαλμάτων σε πρώιμο στάδιο. Μέσω της τοπικής επεξεργασίας δεδομένων από αισθητήρες και μηχανές, εξασφαλίζεται η συνεχής λειτουργία ακόμα και σε περιπτώσεις προσωρινής απώλειας συνδεσιμότητας με το cloud, γεγονός που ενισχύει τη σταθερότητα και την αξιοπιστία των βιομηχανικών διαδικασιών.

Στον τομέα της υγείας, η ανάγκη για ταχύτατη επεξεργασία ευαίσθητων δεδομένων καθιστά την υπολογιστική ομίχλη ιδανική. Οι fog κόμβοι μπορούν να συλλέγουν και να αναλύουν βιομετρικά σήματα από φορητές συσκευές ή ιατρικά όργανα, παρέχοντας έγκαιρες ειδοποιήσεις για καταστάσεις έκτακτης ανάγκης, όπως καρδιακές αρρυθμίες ή κρίσεις σε ασθενείς με χρόνιες παθήσεις. Επιπλέον, υποστηρίζεται η συμμόρφωση με κανονιστικά πλαίσια προστασίας προσωπικών δεδομένων, καθώς τα ευαίσθητα δεδομένα δεν χρειάζεται να εγκαταλείπουν τον χώρο του ασθενή ή του νοσοκομείου. Παρόμοιες αρχές ισχύουν και στα συστήματα IoT, όπου η διαχείριση δεδομένων από εκατομμύρια συσκευές απαιτεί κατανεμημένη και αποδοτική υποδομή. Οι fog κόμβοι λειτουργούν ως ενδιάμεσοι "κόμβοι ευφυΐας", προσφέροντας επεξεργασία, φιλτράρισμα και αποθήκευση δεδομένων από διάσπαρτες συσκευές, προτού αυτά φτάσουν στο cloud για μακροπρόθεσμη ανάλυση ή αποθήκευση.

Ένα από τα πιο εντυπωσιακά και απαιτητικά παραδείγματα εφαρμογής της υπολογιστικής ομίχλης είναι τα αυτόνομα οχήματα. Τα συστήματα αυτά βασίζονται σε τεράστιες ποσότητες δεδομένων που προέρχονται από ραντάρ, κάμερες, LIDAR και άλλους αισθητήρες, τα οποία πρέπει να αναλυθούν σε κλάσματα δευτερολέπτου για την ασφαλή πλοήγηση του οχήματος. Η ύπαρξη fog κόμβων εντός ή πλησίον του οχήματος επιτρέπει την τοπική λήψη αποφάσεων για την αποφυγή εμποδίων ή αλλαγή πορείας, χωρίς την καθυστέρηση που συνεπάγεται η αποστολή των δεδομένων στο cloud. Επιπλέον, επιτρέπεται η συνεργασία μεταξύ πολλών οχημάτων ή μεταξύ οχημάτων και υποδομής (V2X), με βάση την τοπική ανταλλαγή και επεξεργασία πληροφορίας.

Τέλος, η υπολογιστική ομίχλη συνδυάζεται αρμονικά με τα δίκτυα 5G, προσφέροντας ένα υπερσύγχρονο περιβάλλον για εφαρμογές που απαιτούν εξαιρετικά χαμηλή καθυστέρηση και υψηλό ρυθμό μετάδοσης δεδομένων. Εφαρμογές όπως επαυξημένη και εικονική πραγματικότητα (AR/VR), απομακρυσμένες ιατρικές επεμβάσεις, ή βιομηχανικά συστήματα ακριβείας μπορούν να αξιοποιούν fog κόμβους τοποθετημένους κοντά στους σταθμούς βάσης 5G για να επιτυγχάνουν σχεδόν μηδενική καθυστέρηση. Η σύζευξη fog computing και 5G δημιουργεί ένα ισχυρό τεχνολογικό υπόβαθρο για το "υπολογιστικό συνεχές", όπου οι εφαρμογές μπορούν να μετακινούνται ευέλικτα από το edge προς το cloud, ανάλογα με τις ανάγκες και τους περιορισμούς κάθε περίπτωσης.

2.6 Διαχείριση Πόρων σε Περιβάλλοντα Υπολογιστικής Ομίχλης

Η αποτελεσματική διαχείριση πόρων αποτελεί κεντρικό άξονα στην επιτυχή λειτουργία ενός περιβάλλοντος υπολογιστικής ομίχλης. Σε αντίθεση με το παραδοσιακό cloud computing, όπου η εστίαση είναι στην κεντροποιημένη επεξεργασία με υψηλή διαθεσιμότητα υπολογιστικών πόρων, η υπολογιστική ομίχλη απαιτεί ευφυή και προσαρμοστική κατανομή πόρων που είναι περιορισμένοι, ετερογενείς και διασκορπισμένοι σε γεωγραφικά διαφορετικά σημεία. Οι βασικοί τύποι πόρων που πρέπει να διαχειρίζονται είναι η επεξεργαστική ισχύς (CPU), η κύρια μνήμη (RAM), το δικτυακό εύρος ζώνης (Bandwidth) και η ενεργειακή κατανάλωση (Power).

Η επεξεργαστική ισχύς (CPU) αποτελεί κρίσιμο πόρο για την εκτέλεση εφαρμογών σε fog κόμβους, καθώς καθορίζει τη δυνατότητα επεξεργασίας των εισερχόμενων δεδομένων σε πραγματικό χρόνο. Η κατανομή της CPU απαιτεί δυναμικούς αλγορίθμους προγραμματισμού και προτεραιοποίησης, ώστε να εξυπηρετούνται τόσο κρίσιμες όσο και λιγότερο επείγουσες εργασίες. Δεδομένου ότι οι fog κόμβοι έχουν περιορισμένη υπολογιστική ισχύ σε σχέση με τα cloud data centers, η απόδοση εξαρτάται από την ορθολογική εκμετάλλευση των διαθέσιμων πυρήνων και την αποφυγή υπερφόρτωσης.

Η μνήμη RAM είναι εξίσου σημαντική, ιδιαίτερα σε εφαρμογές που απαιτούν προσωρινή αποθήκευση και ταχεία προσπέλαση δεδομένων. Οι fog εφαρμογές συχνά επεξεργάζονται ροές δεδομένων αντί για στατικά αρχεία, κάτι που επιβάλλει αποτελεσματική διαχείριση του διαθέσιμου χώρου μνήμης, ώστε να διατηρούνται μόνο τα απαραίτητα δεδομένα και να αποφεύγεται η υπερχειλίση. Τεχνικές όπως η έξυπνη εκκαθάριση cache και η εικονική μνήμη (virtual memory management) αξιοποιούνται για την ενίσχυση της απόδοσης σε περιορισμένα περιβάλλοντα.

Το δικτυακό εύρος ζώνης (Bandwidth) συνιστά έναν καθοριστικό πόρο σε fog περιβάλλοντα, λόγω της συνεχούς μεταφοράς δεδομένων μεταξύ edge συσκευών, fog κόμβων και, ενδεχομένως, του cloud. Η διαχείριση του bandwidth είναι απαραίτητη για την αποφυγή συμφόρησης και την εγγύηση ποιότητας υπηρεσίας, ειδικά σε εφαρμογές με υψηλές απαιτήσεις μετάδοσης, όπως video streaming, τηλεχειρισμός ρομποτικών συστημάτων ή συστήματα παρακολούθησης. Η κατανομή εύρους ζώνης πρέπει να είναι δυναμική και προσαρμοζόμενη, με χρήση τεχνικών δρομολόγησης και προτεραιοποίησης πακέτων βάσει της σημασίας και του επιπέδου καθυστέρησης των δεδομένων.

Τέλος, η ενεργειακή κατανάλωση (Power) παίζει ιδιαίτερα κρίσιμο ρόλο, καθώς πολλοί fog κόμβοι βρίσκονται σε περιβάλλοντα χωρίς σταθερή παροχή ρεύματος ή εξαρτώνται από μπαταρίες (όπως σε κινητές υποδομές ή απομακρυσμένους αισθητήρες). Η εξοικονόμηση ενέργειας επιτυγχάνεται μέσω ενεργειακά αποδοτικών αλγορίθμων, adaptive λειτουργιών λειτουργικού συστήματος (όπως dynamic voltage and frequency scaling - DVFS), και στρατηγικών offloading, όπου ορισμένες εργασίες μεταφέρονται στο cloud όταν η κατανάλωση ενέργειας είναι απαγορευτική. Η επίτευξη ενεργειακής ισορροπίας αποτελεί βασική πρόκληση για τη μακροχρόνια λειτουργία των fog υποδομών.

2.7 Ασφάλεια και Προστασία Δεδομένων

Η υπολογιστική ομίχλη, λόγω της αποκεντρωμένης και πολυεπίπεδης φύσης της, εισάγει νέες και πολύπλοκες προκλήσεις στον τομέα της ασφάλειας και της προστασίας

προσωπικών δεδομένων. Σε αντίθεση με τα κεντροποιημένα συστήματα του cloud, όπου η ασφάλεια μπορεί να ελέγχεται κεντρικά και με σταθερές υποδομές, το fog περιβάλλον περιλαμβάνει μεγάλο αριθμό ετερογενών κόμβων, πολλές φορές εγκατεστημένων σε μη ασφαλείς ή δημόσιες τοποθεσίες, γεγονός που αυξάνει την επιφάνεια επίθεσης και τη δυσκολία εποπτείας.

Οι απειλές ασφάλειας στο fog computing καλύπτουν ένα ευρύ φάσμα επιπέδων – από το φυσικό έως το λογισμικό και το δίκτυο. Φυσικές επιθέσεις περιλαμβάνουν την παραβίαση ή καταστροφή fog κόμβων, ειδικά όταν αυτοί βρίσκονται σε μη ελεγχόμενους χώρους. Σε λογισμικό επίπεδο, κακόβουλοι παράγοντες μπορούν να επιτεθούν μέσω malware, rootkits ή trojans, στοχεύοντας στον έλεγχο των συσκευών ή την υποκλοπή δεδομένων. Δικτυακές επιθέσεις, όπως Man-in-the-Middle (MitM), spoofing και denial-of-service (DoS), είναι ιδιαίτερα επικίνδυνες, καθώς επηρεάζουν άμεσα τη διαθεσιμότητα, την εμπιστευτικότητα και την ακεραιότητα των δεδομένων που μεταφέρονται ή υποβάλλονται σε επεξεργασία στο fog περιβάλλον. Επιπλέον, λόγω της φύσης του fog, μπορεί να προκύψουν και εσωτερικές απειλές από μη αξιόπιστους κόμβους ή από κακόβουλους χρήστες εντός του συστήματος.

Για την ενίσχυση της ασφάλειας, είναι απαραίτητη η υιοθέτηση ισχυρών μηχανισμών αυθεντικοποίησης (authentication), ώστε να διασφαλίζεται ότι κάθε χρήστης, συσκευή ή υπηρεσία που επιχειρεί να εισέλθει στο δίκτυο είναι πιστοποιημένος και αξιόπιστος. Οι παραδοσιακές μέθοδοι (π.χ. κωδικοί πρόσβασης) δεν επαρκούν στο πολυσχιδές περιβάλλον του fog, γι' αυτό προτιμώνται ισχυρότερα σχήματα όπως η πολυπαραγοντική ταυτοποίηση (MFA), η πιστοποίηση βάσει PKI (Public Key Infrastructure), ή ακόμα και μέθοδοι βασισμένες σε blockchain για αποκεντρωμένη επιβεβαίωση ταυτότητας. Επιπρόσθετα, είναι κρίσιμο να εφαρμόζονται πολιτικές αυστηρού ελέγχου πρόσβασης (access control) ώστε κάθε κόμβος ή χρήστης να διαθέτει μόνο τα δικαιώματα που αντιστοιχούν στον ρόλο του.

Η κρυπτογράφηση (encryption) αποτελεί ακρογωνιαίο λίθο για την προστασία των δεδομένων τόσο κατά τη μετάδοση όσο και κατά την αποθήκευση. Η χρήση κρυπτογράφησης end-to-end εξασφαλίζει ότι τα δεδομένα δεν μπορούν να αναγνωσθούν ή να παραβιαστούν κατά τη διαδρομή τους στο δίκτυο, ακόμη και αν η σύνδεση παρεμβληθεί από τρίτους. Σε τοπικό επίπεδο, οι fog κόμβοι πρέπει να αποθηκεύουν δεδομένα με ισχυρή κρυπτογράφηση και να εφαρμόζουν μηχανισμούς

ασφαλούς διαχείρισης κλειδιών. Για επιπλέον προστασία, μπορούν να ενσωματωθούν τεχνικές όπως η ομομορφική κρυπτογράφηση (homomorphic encryption), η οποία επιτρέπει την επεξεργασία δεδομένων χωρίς να απαιτείται η αποκρυπτογράφησή τους, μειώνοντας έτσι τον κίνδυνο διαρροής.

Ιδιαίτερα σημαντικό στο πλαίσιο της προστασίας προσωπικών δεδομένων είναι η συμμόρφωση με κανονιστικά πλαίσια όπως ο Γενικός Κανονισμός για την Προστασία Δεδομένων (GDPR). Η γεωγραφική διασπορά των fog κόμβων δημιουργεί δυσκολίες στον έλεγχο του πού ακριβώς αποθηκεύονται ή επεξεργάζονται τα δεδομένα, κάτι που μπορεί να έρχεται σε σύγκρουση με εθνικούς ή διεθνείς κανόνες. Ο GDPR απαιτεί, μεταξύ άλλων, τη ρητή συγκατάθεση των υποκειμένων των δεδομένων, την ελαχιστοποίηση των συλλεγόμενων πληροφοριών, τη δυνατότητα διαγραφής ("δικαίωμα στη λήθη") και τη διαφάνεια στον τρόπο επεξεργασίας. Το fog computing οφείλει να ενσωματώνει από τον σχεδιασμό ("privacy by design") μηχανισμούς που εξασφαλίζουν τη συμμόρφωση με αυτούς τους κανόνες, όπως τοπική επεξεργασία, περιορισμένη αποθήκευση ή γεωεντοπισμός δεδομένων.

2.8 Εργαλεία Προσομοίωσης στην Υπολογιστική Ομίχλη

Η συνεχώς αυξανόμενη σημασία της υπολογιστικής ομίχλης ως υποδομή επεξεργασίας δεδομένων κοντά στο σημείο παραγωγής τους, δημιούργησε την ανάγκη για εξειδικευμένα εργαλεία προσομοίωσης που να μπορούν να μοντελοποιήσουν και να αξιολογήσουν με ακρίβεια τα τεχνικά και λειτουργικά χαρακτηριστικά αυτής της τεχνολογίας. Λόγω της φύσης του fog computing – με την έντονη ετερογένεια, την αποκέντρωση, την κατανομή πόρων και τη δυναμική των εφαρμογών – η ανάπτυξη, υλοποίηση και πειραματική αξιολόγηση σε πραγματικές υποδομές είναι συχνά χρονοβόρα, δαπανηρή και τεχνικά απαιτητική. Για τον λόγο αυτό, η χρήση προσομοιωτών αποτελεί κρίσιμο εργαλείο στην ερευνητική διαδικασία, καθώς επιτρέπει την αναπαραγωγή ρεαλιστικών σεναρίων, τη δοκιμή στρατηγικών κατανομής εργασιών, την παρακολούθηση μετρικών απόδοσης και την πειραματική αξιολόγηση αλγορίθμων σε ελεγχόμενο περιβάλλον.

Το FogSim υπήρξε ένα από τα πρώτα εργαλεία που αναπτύχθηκαν αποκλειστικά για την προσομοίωση fog περιβαλλόντων. Βασισμένο στο CloudSim, το FogSim

προσαρμόστηκε ώστε να συμπεριλάβει fog κόμβους, τοπικές επεξεργασίες, χαμηλές καθυστερήσεις και στοιχειώδη διαχείριση ενέργειας. Επιτρέπει την απεικόνιση απλών τοπολογιών υπολογιστικής ομίχλης, με δυνατότητα μοντελοποίησης κόμβων διαφορετικών χαρακτηριστικών, χρηστών και εφαρμογών με προκαθορισμένες ρυθμίσεις. Αν και είναι ιδανικό για αρχική κατανόηση της λειτουργίας fog υποδομών, παρουσιάζει περιορισμούς στην υποστήριξη πολύπλοκων εφαρμογών, ιδιαίτερα εκείνων που βασίζονται σε αλληλεξαρτώμενες διεργασίες (όπως DAGs ή workflows), καθώς και σε σενάρια με δυναμική κατανομή και προσαρμοστικότητα.

Αντιθέτως, το iFogSim αποτελεί πιο ολοκληρωμένο και διαδεδομένο εργαλείο, επίσης βασισμένο στο CloudSim, αλλά με επεκτάσεις που επιτρέπουν την πλήρη μοντελοποίηση fog και edge υποδομών. Υποστηρίζει την αναπαράσταση εφαρμογών ως κατευθυνόμενων άκυκλων γραφών (DAGs), όπου κάθε κόμβος του γραφήματος αντιπροσωπεύει μια επιμέρους εργασία και κάθε ακμή εκφράζει μια εξάρτηση. Το iFogSim επιτρέπει την παραμετροποίηση κόμβων σε διαφορετικά επίπεδα (edge, fog, cloud), την απεικόνιση καθυστερήσεων μετάδοσης, την παρακολούθηση της ενεργειακής κατανάλωσης και την αξιολόγηση πολιτικών κατανομής και offloading. Χάρη στη modular αρχιτεκτονική του, είναι εύκολα επεκτάσιμο για τις ανάγκες κάθε ερευνητικού σεναρίου και έχει χρησιμοποιηθεί ευρέως σε μελέτες που σχετίζονται με IoT εφαρμογές, έξυπνες πόλεις, συστήματα υγείας και βιομηχανικές πλατφόρμες. Ωστόσο, δεν παρέχει εκ φύσεως υποστήριξη για πλήρεις workflows με ροές ελέγχου, δυναμική αναδρομή ή προσαρμοστική λογική, κάτι που περιορίζει τη χρήση του σε σύνθετες εφαρμογές επιχειρησιακού τύπου.

Για την κάλυψη αυτού του κενού, αναπτύχθηκε το FogWorkflowSim, ένα εργαλείο που επεκτείνει τις δυνατότητες του iFogSim, ενσωματώνοντας πλήρως τη μοντελοποίηση και προσομοίωση ροών εργασιών (workflows) σε fog περιβάλλοντα. Το FogWorkflowSim υποστηρίζει τον καθορισμό πολύπλοκων εφαρμογών με εξαρτώμενα βήματα, δυνατότητες χρονοδρομολόγησης, και δυναμική προσαρμογή ανάλογα με τις συνθήκες εκτέλεσης. Ενσωματώνει πολλαπλούς αλγορίθμους κατανομής, όπως HEFT και GA-based scheduling, και παρακολουθεί με ακρίβεια βασικές μετρικές, όπως makespan, κόστος, καθυστέρηση, κατανομή φόρτου και αξιοποίηση πόρων. Αποτελεί εξαιρετικό εργαλείο για την ανάλυση εφαρμογών που βασίζονται σε πολυσταδιακές διαδικασίες, όπως η βιοϊατρική επεξεργασία, η επιστημονική υπολογιστική, ή η βιομηχανική παρακολούθηση. Παρ' όλα αυτά, απαιτεί μεγαλύτερη εμπειρία στη

ρύθμιση και τη χρήση του, καθώς προϋποθέτει βαθύτερη κατανόηση της αρχιτεκτονικής του iFogSim και πιο σύνθετο σχεδιασμό σεναρίων.

Η συγκριτική αξιολόγηση των τριών εργαλείων αποκαλύπτει ότι το FogSim είναι κατάλληλο για απλές, αρχικές προσεγγίσεις, προσφέροντας ευκολία χρήσης και βασική λειτουργικότητα. Το iFogSim είναι περισσότερο ισορροπημένο, συνδυάζοντας ευελιξία και δυνατότητα επέκτασης, και αποτελεί το πιο διαδεδομένο εργαλείο στην ερευνητική κοινότητα. Το FogWorkflowSim απευθύνεται κυρίως σε προχωρημένες ερευνητικές εφαρμογές, με έμφαση στη μοντελοποίηση πολύπλοκων εξαρτήσεων και προγραμματιστικών απαιτήσεων. Η επιλογή του κατάλληλου εργαλείου εξαρτάται σε μεγάλο βαθμό από τη φύση του πειραματικού σεναρίου, τη δομή των εργασιών και τους στόχους της μελέτης.

Ο ρόλος των εργαλείων αυτών στην ερευνητική διαδικασία είναι καθοριστικός, καθώς επιτρέπουν την εκτέλεση πειραμάτων μεγάλης κλίμακας χωρίς την ανάγκη φυσικής υποδομής. Μέσω αυτών, ερευνητές μπορούν να δοκιμάσουν διαφορετικές στρατηγικές κατανομής εργασιών, να μελετήσουν την επίδραση παραμέτρων όπως η καθυστέρηση, η κατανάλωση ενέργειας ή η κινητικότητα, και να εντοπίσουν σημεία βελτίωσης ή προτάσεις καινοτομίας. Επιπλέον, η δυνατότητα αναπαραγωγής των πειραμάτων και η ανοικτή φύση των εργαλείων διευκολύνει την επιστημονική αξιοπιστία, τη δημοσίευση αποτελεσμάτων και τη σύγκριση διαφορετικών προσεγγίσεων.

3 ΚΕΦΑΛΑΙΟ: ΜΗΧΑΝΙΣΜΟΙ ΚΑΙ ΣΤΡΑΤΗΓΙΚΕΣ ΚΑΤΑΝΟΜΗΣ ΠΟΡΩΝ ΚΑΙ ΕΡΓΑΣΙΩΝ

3.1 Προκλήσεις Διαχείρισης Πόρων σε Περιβάλλοντα Υπολογιστικής Ομίχλης

Η διαχείριση πόρων σε συστήματα υπολογιστικής ομίχλης (fog computing) παρουσιάζει σημαντικές προκλήσεις, οι οποίες απορρέουν από τη φύση των κατανεμημένων, ετερογενών και δυναμικά μεταβαλλόμενων υποδομών που τη χαρακτηρίζουν. Σε αντίθεση με τα ομοιογενή, ελεγχόμενα και στατικά περιβάλλοντα του cloud, το fog περιβάλλον απαιτεί ευφυείς, εύελικτους και προσαρμόσιμους μηχανισμούς διαχείρισης πόρων, ικανούς να λειτουργούν σε συνθήκες απροσδιοριστίας και με έντονους περιορισμούς.

Μία από τις θεμελιώδεις προκλήσεις είναι η ετερογένεια των συσκευών, των πλατφορμών και των τεχνολογιών που συμμετέχουν σε ένα fog οικοσύστημα. Οι fog κόμβοι μπορεί να είναι απλοί δρομολογητές, πύλες πρόσβασης, micro-servers ή ακόμα και ενσωματωμένα συστήματα με άνισες δυνατότητες σε CPU, RAM, αποθήκευση και ενέργεια. Παράλληλα, κάθε κόμβος μπορεί να χρησιμοποιεί διαφορετικό λειτουργικό σύστημα, πρωτόκολλο επικοινωνίας ή αρχιτεκτονική, καθιστώντας τη διαλειτουργικότητα και την ενοποίηση πόρων μια σύνθετη διαδικασία. Η ανάπτυξη ομοιογενών μηχανισμών διαχείρισης που μπορούν να εφαρμοστούν σε τόσο ποικιλόμορφα περιβάλλοντα παραμένει ένα από τα πιο σύνθετα προβλήματα του πεδίου.

Η δυναμικότητα του fog περιβάλλοντος αποτελεί επίσης σημαντικό εμπόδιο για τη σταθερή και αποδοτική διαχείριση πόρων. Σε ένα δυναμικό δίκτυο, οι συνθήκες αλλάζουν συνεχώς: οι κόμβοι μπορούν να εισέρχονται ή να εξέρχονται από το δίκτυο, το φορτίο μεταβάλλεται ακανόνιστα, οι απαιτήσεις των εφαρμογών αυξομειώνονται, ενώ η ποιότητα της σύνδεσης και η διαθεσιμότητα πόρων μπορεί να επηρεάζονται από εξωτερικούς παράγοντες. Αυτή η εγγενής αστάθεια απαιτεί την υιοθέτηση προγνωστικών ή αυτοπροσαρμοζόμενων μεθόδων διαχείρισης, που να μπορούν να ανταποκρίνονται σε πραγματικό χρόνο χωρίς να διακόπτουν την παροχή υπηρεσιών.

Επιπλέον, τα fog περιβάλλοντα αντιμετωπίζουν λειτουργικούς και υλικούς περιορισμούς, οι οποίοι περιορίζουν σημαντικά το περιθώριο δράσης των μηχανισμών

διαχείρισης. Οι πόροι που διαθέτουν οι fog κόμβοι είναι περιορισμένοι συγκριτικά με τα cloud data centers. Η χαμηλότερη υπολογιστική ισχύς, ο περιορισμένος χώρος αποθήκευσης και η μειωμένη μνήμη περιορίζουν τις δυνατότητες επεξεργασίας εντατικών εφαρμογών, ιδίως όταν πολλαπλές εφαρμογές συνυπάρχουν σε έναν κόμβο. Η ανάγκη για χαμηλή κατανάλωση ενέργειας καθιστά αναγκαία την ανάπτυξη ενεργειακά αποδοτικών στρατηγικών, που ενίοτε έρχονται σε σύγκρουση με στόχους υψηλής απόδοσης ή χαμηλής καθυστέρησης.

Η διαχείριση των πολιτικών ποιότητας υπηρεσίας (QoS) αποτελεί επιπλέον πρόκληση, καθώς οι fog εφαρμογές διαφέρουν δραματικά ως προς τις απαιτήσεις τους: άλλες απαιτούν άμεση απόκριση και χαμηλή καθυστέρηση (π.χ. αυτόνομα οχήματα), ενώ άλλες δίνουν έμφαση στη διατήρηση ενεργειακής αποδοτικότητας ή στην ασφάλεια δεδομένων. Η εξισορρόπηση αυτών των στόχων, ιδιαίτερα όταν οι πόροι είναι περιορισμένοι, προϋποθέτει σύνθετους αλγορίθμους προτεραιοποίησης και βελτιστοποίησης.

Τέλος, η έλλειψη τυποποίησης και ενοποιημένων πρωτοκόλλων καθιστά δυσχερή την εφαρμογή καθολικών μηχανισμών διαχείρισης. Η ετερογένεια σε επίπεδο κατασκευαστών, λειτουργικών συστημάτων, εφαρμογών και πρωτοκόλλων επικοινωνίας δημιουργεί ένα κατακερματισμένο οικοσύστημα όπου η ολιστική διαχείριση πόρων είναι δύσκολα υλοποιήσιμη. Οι ερευνητικές προσπάθειες εστιάζουν πλέον στην ανάπτυξη ανοικτών προτύπων, πλαισίων ορθής πρακτικής και ευέλικτων συστημάτων ορχήστρωσης που να μπορούν να λειτουργούν ανεξαρτήτως υποδομής και παρόχου.

3.2 Μηχανισμοί Κατανομής Πόρων

Η αποτελεσματική κατανομή πόρων αποτελεί κρίσιμο παράγοντα για την απόδοση και αξιοπιστία των υπολογιστικών περιβαλλόντων ομίχλης. Σε αντίθεση με τα κεντροποιημένα μοντέλα του cloud, όπου η κατανομή γίνεται με βάση προβλέψιμους και στατικούς πόρους, στο fog computing η διαδικασία πρέπει να είναι δυναμική, αποκεντρωμένη και ευέλικτη, προσαρμοσμένη στις τοπικές ανάγκες και στις ιδιαιτερότητες των ετερογενών κόμβων. Οι μηχανισμοί κατανομής πόρων αποσκοπούν

στο να εξασφαλίσουν ισορροπία μεταξύ των απαιτήσεων των εφαρμογών (latency, QoS, ενέργεια) και της διαθεσιμότητας πόρων (CPU, RAM, bandwidth, ισχύς).

Μία βασική κατηγορία μηχανισμών κατανομής είναι οι αλγόριθμοι προγραμματισμού εργασιών (task scheduling). Αυτοί οι αλγόριθμοι αξιολογούν τις απαιτήσεις των εισερχόμενων εργασιών και τις αντιστοιχούν σε κατάλληλους fog κόμβους βάσει διαθέσιμων πόρων, προτεραιότητας, τοποθεσίας ή ενεργειακής κατανάλωσης. Προσεγγίσεις όπως το *First-Come-First-Serve (FCFS)*, *Min-Min*, *Max-Min* ή πιο εξελιγμένες μέθοδοι, όπως *genetic algorithms* και *fuzzy logic*, αξιοποιούνται για την επίτευξη βέλτιστης απόδοσης σε σύνθετα περιβάλλοντα με περιορισμένους πόρους.

Ένας άλλος σημαντικός μηχανισμός είναι η δυναμική κατανομή φορτίου (load balancing), η οποία επιδιώκει τη διανομή της επεξεργασίας μεταξύ πολλαπλών κόμβων, αποτρέποντας τη συμφόρηση ή την υπερφόρτωση ορισμένων κόμβων ενώ άλλοι παραμένουν ανενεργοί. Οι τεχνικές αυτές βασίζονται είτε σε στατικά κριτήρια είτε σε δυναμικά μοντέλα που παρακολουθούν την κατάσταση του συστήματος σε πραγματικό χρόνο και ανακατανέμουν εργασίες όταν ανιχνεύονται ανισορροπίες. Η κατανομή πόρων βάσει ποιότητας υπηρεσίας (QoS-aware allocation) εισάγει πρόσθετα κριτήρια στη διαδικασία απόφασης, όπως η καθυστέρηση, η αξιοπιστία, η ασφάλεια και η ενεργειακή απόδοση. Ο μηχανισμός αυτός είναι ιδιαίτερα σημαντικός σε εφαρμογές κρίσιμης σημασίας – π.χ. αυτόνομα οχήματα ή ιατρικές υπηρεσίες – όπου δεν αρκεί μόνο η κατανομή με βάση τη διαθεσιμότητα, αλλά απαιτείται συμμόρφωση με αυστηρές προδιαγραφές απόδοσης.

Ιδιαίτερα χρήσιμος είναι και ο μηχανισμός offloading, όπου εργασίες μεταφέρονται (ολικά ή μερικά) από τοπικές edge συσκευές σε fog κόμβους ή από fog κόμβους στο cloud, ανάλογα με την τρέχουσα κατάσταση του συστήματος και τη φύση της εργασίας. Η απόφαση για offloading στηρίζεται σε παραμέτρους όπως η διαθέσιμη CPU, η ένταση δικτύου, η κατανάλωση ενέργειας και ο απαιτούμενος χρόνος απόκρισης. Η υιοθέτηση υβριδικών στρατηγικών επιτρέπει μεγαλύτερη ευελιξία στην εκτέλεση εφαρμογών σε δυναμικά περιβάλλοντα. Τέλος, αναδύονται συστήματα ορχήστρωσης πόρων (resource orchestration frameworks), τα οποία αυτοματοποιούν τη διαδικασία κατανομής, παρακολούθησης και αναδιάταξης πόρων σε ευρεία κλίμακα. Τέτοια συστήματα συνδυάζουν τεχνολογίες containerization (π.χ. Docker, Kubernetes με fog

επεκτάσεις), monitoring εργαλείων και έξυπνων αλγορίθμων πρόβλεψης για την αποτελεσματική διαχείριση πόρων σε πραγματικό χρόνο.

3.3 Κατανομή Εργασιών (Task Scheduling)

Η κατανομή εργασιών (task scheduling) αποτελεί μία από τις πιο κρίσιμες διεργασίες στα υπολογιστικά περιβάλλοντα ομίχλης, καθώς καθορίζει πώς και πού θα εκτελεστούν οι εισερχόμενες διεργασίες, με στόχο τη βέλτιστη χρήση των πόρων, τη μείωση της καθυστέρησης και την επίτευξη των απαιτήσεων ποιότητας υπηρεσίας (QoS). Η πολυπλοκότητα αυτής της διαδικασίας εξαρτάται άμεσα από τη φύση και τη δομή των εργασιών, οι οποίες ταξινομούνται σε τρία βασικά μοντέλα: απλές (independent tasks), κατευθυνόμενοι άκυκλοι γράφοι (DAGs) και ροές εργασιών (workflows).

Οι απλές εργασίες (independent tasks) είναι αυτόνομα υπολογιστικά φορτία που δεν εξαρτώνται μεταξύ τους και μπορούν να εκτελεστούν ανεξάρτητα σε οποιονδήποτε κατάλληλο fog κόμβο. Η απουσία εξαρτήσεων επιτρέπει την ταχεία και παράλληλη εκτέλεση, γεγονός που διευκολύνει την εφαρμογή κλασικών αλγορίθμων κατανομής, όπως FCFS (First-Come-First-Served), Round-Robin, ή Greedy-based approaches. Το scheduling σε αυτό το μοντέλο είναι λιγότερο πολύπλοκο, αλλά η αποδοτικότητα του εξαρτάται από τον σωστό συνδυασμό φόρτου και διαθέσιμων πόρων, καθώς και από το latency μεταξύ των edge συσκευών και των fog κόμβων.

Αντιθέτως, οι κατευθυνόμενοι άκυκλοι γράφοι (Directed Acyclic Graphs – DAGs) μοντελοποιούν σύνθετες εργασίες που αποτελούνται από πολλαπλά βήματα με εξαρτήσεις μεταξύ τους. Σε ένα DAG, οι κόμβοι αναπαριστούν υπο-εργασίες και οι ακμές εκφράζουν εξαρτήσεις εκτέλεσης – δηλαδή, μια εργασία δεν μπορεί να ξεκινήσει πριν ολοκληρωθεί η προηγούμενη από την οποία εξαρτάται. Το scheduling DAGs σε fog περιβάλλοντα είναι πιο απαιτητικό, καθώς πρέπει να ληφθεί υπόψη όχι μόνο η διαθεσιμότητα πόρων αλλά και η τοπολογική σειρά εκτέλεσης, το κόστος επικοινωνίας μεταξύ κόμβων και οι χρονικοί περιορισμοί. Τεχνικές όπως *list scheduling*, *critical path methods* και *HEFT (Heterogeneous Earliest Finish Time)* χρησιμοποιούνται ευρέως για τη βελτιστοποίηση αυτών των περιπτώσεων.

Το τρίτο και πιο σύνθετο μοντέλο αφορά τις ροές εργασιών (workflows), οι οποίες συχνά περιγράφονται ως πολυεπίπεδες ή πολυσταδιακές διαδικασίες που εμπλέκουν πολλαπλές DAGs, χρονικούς περιορισμούς, επαναληπτικά μοτίβα, και ενίοτε δυναμικές αλληλεξαρτήσεις μεταξύ των εργασιών. Τα workflows χρησιμοποιούνται συχνά σε εφαρμογές όπως η ανάλυση βιοϊατρικών δεδομένων, τα έξυπνα εργοστάσια και η περιβαλλοντική παρακολούθηση. Η εκτέλεση workflows απαιτεί εξελιγμένα συστήματα οργάνωσης και αλγορίθμους που υποστηρίζουν τόσο την προκαθορισμένη λογική της ροής όσο και την ευελιξία για επαναπρογραμματισμό όταν αλλάζουν οι συνθήκες (π.χ., αποτυχία ενός κόμβου ή συμφόρηση).

Σε fog περιβάλλοντα, η πολυπλοκότητα του scheduling αυξάνεται εξαιτίας των δυναμικών συνθηκών (μεταβαλλόμενη διαθεσιμότητα κόμβων, περιορισμένοι πόροι, ετερογένεια), αλλά και λόγω της ανάγκης για κατανεμημένες αποφάσεις. Συχνά εφαρμόζονται κατανεμημένοι ή ημι-κατανεμημένοι μηχανισμοί κατανομής, στους οποίους οι fog κόμβοι λαμβάνουν τοπικές αποφάσεις βασισμένοι σε περιορισμένες πληροφορίες, ή συνεργάζονται με έναν κεντρικό ελεγκτή για συντονισμό. Επιπλέον, η ενσωμάτωση τεχνικών από τη μηχανική μάθηση, όπως reinforcement learning ή metaheuristic approaches (π.χ. genetic algorithms, ant colony optimization), έχει προταθεί για την αντιμετώπιση των δυσκολιών που παρουσιάζουν τα δυναμικά και μεγάλης κλίμακας workflows.

3.4 Στρατηγικές Κατανομής Εργασιών

Η επιλογή της κατάλληλης στρατηγικής κατανομής εργασιών (scheduling) αποτελεί κρίσιμο βήμα για την αποδοτική λειτουργία ενός fog περιβάλλοντος. Λόγω της ετερογένειας, της γεωγραφικής διασποράς των κόμβων και της δυναμικής φύσης των εφαρμογών, απαιτούνται εξελιγμένες προσεγγίσεις που να εξισορροπούν την απόδοση, την καθυστέρηση και την κατανάλωση πόρων. Οι στρατηγικές scheduling μπορούν να ταξινομηθούν σε δύο βασικές διαστάσεις: στατικές έναντι δυναμικών, και κεντρικές έναντι κατανεμημένων.

Οι στατικές στρατηγικές (static scheduling) βασίζονται στην υπόθεση ότι οι πληροφορίες σχετικά με τις εργασίες και τους διαθέσιμους πόρους είναι γνωστές εκ των προτέρων και παραμένουν σταθερές κατά την εκτέλεση. Οι αποφάσεις λήψης κατανομής γίνονται πριν την έναρξη των εργασιών, με βάση μοντέλα πρόβλεψης, ιστορικά δεδομένα ή αναλυτικά σενάρια. Τέτοιες στρατηγικές είναι κατάλληλες για περιβάλλοντα με προβλέψιμα φορτία ή για εφαρμογές με αυστηρές χρονικές απαιτήσεις, καθώς προσφέρουν προγραμματισμό υψηλής ακρίβειας. Ωστόσο, μειονεκτούν σε περιβάλλοντα με έντονη μεταβλητότητα, αφού δεν μπορούν να ανταποκριθούν σε απρόβλεπτες αλλαγές, όπως υπερφόρτωση κόμβων ή αποτυχίες.

Αντίθετα, οι δυναμικές στρατηγικές (dynamic scheduling) λαμβάνουν αποφάσεις σε πραγματικό χρόνο, καθώς οι εργασίες φτάνουν στο σύστημα και οι συνθήκες μεταβάλλονται. Η δυναμική κατανομή βασίζεται σε συνεχή παρακολούθηση της κατάστασης των κόμβων (π.χ. CPU, μνήμη, latency, φόρτος δικτύου), και προσαρμόζει τις αναθέσεις με στόχο την ισορροπία και τη βελτιστοποίηση της απόδοσης. Αυτές οι στρατηγικές είναι ιδιαίτερα κατάλληλες για fog περιβάλλοντα, όπου η ρευστότητα των συνθηκών αποτελεί κανόνα. Το μειονέκτημά τους είναι η ανάγκη για αυξημένο υπολογιστικό κόστος και πιο σύνθετους μηχανισμούς λήψης αποφάσεων.

Αναφορικά με την αρχιτεκτονική της λήψης αποφάσεων, οι στρατηγικές διακρίνονται σε κεντρικές (centralized) και κατανεμημένες (distributed). Στις κεντρικές στρατηγικές, η ευθύνη για τη λήψη αποφάσεων ανήκει σε έναν κεντρικό ελεγκτή ή orchestrator, ο οποίος έχει συνολική εικόνα του συστήματος και μπορεί να εφαρμόσει παγκόσμια βέλτιστες πολιτικές κατανομής. Αν και προσφέρουν έλεγχο και συνέπεια, οι κεντρικές προσεγγίσεις υφίστανται καθυστερήσεις, αποτελούν σημείο συμφόρησης (bottleneck) και είναι λιγότερο ανθεκτικές σε αποτυχίες – γεγονός που έρχεται σε αντίθεση με την αποκεντρωμένη φύση του fog computing.

Οι κατανεμημένες στρατηγικές, από την άλλη, μεταθέτουν τη λήψη αποφάσεων στους ίδιους τους fog κόμβους ή σε τοπικές συστάδες κόμβων (clusters). Κάθε κόμβος έχει μερική γνώση της κατάστασης και λαμβάνει αποφάσεις τοπικά, με στόχο την αυτονομία, την κλιμακωσιμότητα και την ανθεκτικότητα του συστήματος. Η χρήση peer-to-peer επικοινωνίας, μηχανισμών συναίνεσης ή αποκεντρωμένων πρωτοκόλλων επιτρέπει την αυτο-οργάνωση των κόμβων. Παρότι μειώνεται ο έλεγχος και η δυνατότητα παγκόσμιας βελτιστοποίησης, οι κατανεμημένες προσεγγίσεις είναι

καλύτερα προσαρμοσμένες στις ιδιαιτερότητες του fog περιβάλλοντος και πιο ευέλικτες σε σενάρια αποτυχίας.

Σε πολλές περιπτώσεις υιοθετούνται υβριδικά μοντέλα, τα οποία συνδυάζουν δυναμικές με στατικές και κεντρικές με κατανεμημένες στρατηγικές. Για παράδειγμα, ένας κεντρικός orchestrator μπορεί να καταναίμει αρχικά τους πόρους (στατικά), ενώ επιτρέπει στους fog κόμβους να αναπροσαρμόζουν δυναμικά την εκτέλεση βάσει τοπικών συνθηκών. Παράλληλα, η χρήση τεχνητής νοημοσύνης και ενισχυτικής μάθησης ενισχύει την ικανότητα πρόβλεψης και προσαρμογής των μηχανισμών κατανομής.

3.5 Αλγόριθμοι Κατανομής Εργασιών στην Υπολογιστική Ομίχλη

Ο αλγόριθμος Min-Min επιλέγει πρώτα τις εργασίες με τον μικρότερο εκτιμώμενο χρόνο εκτέλεσης (ETC) και τις αναθέτει στους κόμβους που μπορούν να τις ολοκληρώσουν πιο γρήγορα. Είναι αποτελεσματικός σε περιβάλλοντα με μικρές εργασίες και επιδιώκει τη γρήγορη εξυπηρέτηση όσο το δυνατόν περισσότερων εργασιών σε σύντομο χρονικό διάστημα. Ωστόσο, δεν αποδίδει καλά όταν υπάρχουν εργασίες με μεγάλη διάρκεια, καθώς αυτές τείνουν να καθυστερούν. Επιπλέον, ο αλγόριθμος Max-Min αντιστρέφει τη λογική του Min-Min: επιλέγει την εργασία με τον μέγιστο χρόνο εκτέλεσης και τη διανέμει στον κόμβο που μπορεί να την επεξεργαστεί γρηγορότερα. Στόχος του είναι να μειώσει τη συνολική διάρκεια (makespan) εκτέλεσης και να αποφευχθεί η υπερφόρτωση μεμονωμένων κόμβων. Ενδείκνυται για καταστάσεις όπου οι εργασίες είναι άνισα κατανεμημένες ή υπάρχουν πολλές "βαριές" διεργασίες.

Οι γενετικοί αλγόριθμοι βασίζονται στις αρχές της φυσικής επιλογής και της εξέλιξης. Δημιουργούν πληθυσμούς λύσεων (χρωμοσώματα), που εξελίσσονται με τελεστές όπως διασταύρωση (crossover) και μετάλλαξη (mutation). Κάθε λύση αξιολογείται με μια συνάρτηση καταλληλότητας (fitness function), π.χ. ελαχιστοποίηση καθυστέρησης ή κατανάλωσης ενέργειας. Οι GA είναι εξαιρετικά αποτελεσματικοί σε πολύπλοκα, δυναμικά περιβάλλοντα, αν και απαιτούν περισσότερο υπολογιστικό χρόνο.

Ο αλγόριθμος PSO εμπνέεται από τη συλλογική συμπεριφορά σμηνών (π.χ. πουλιά ή ψάρια). Κάθε "σωματίδιο" αντιστοιχεί σε μια πιθανή λύση και κινείται στον χώρο αναζήτησης με βάση τη δική του εμπειρία και την εμπειρία των άλλων σωματιδίων. Στο scheduling, χρησιμοποιείται για την εύρεση της βέλτιστης αντιστοίχισης εργασιών-κόμβων. Έχει γρήγορη σύγκλιση και χαμηλότερο υπολογιστικό κόστος από τους GA, αλλά μπορεί να παγιδευτεί σε τοπικά άκρα.

Επίσης, ο HEFT είναι ένας ευρετικός αλγόριθμος σχεδιασμένος για DAGs σε ετερογενή περιβάλλοντα. Υπολογίζει το κρίσιμο μονοπάτι της εργασίας και προσπαθεί να ελαχιστοποιήσει τον χρόνο ολοκλήρωσης κάθε υπο-εργασίας (subtask). Είναι ευρέως αποδεκτός σε fog συστήματα λόγω της ισορροπίας που προσφέρει ανάμεσα σε ταχύτητα και ποιότητα λύσης. Τέλος, ο αλγόριθμος ACO μιμείται τη συμπεριφορά των μυρμηγκιών στην εύρεση μονοπατιών. Κάθε "μυρμήγκι" είναι ένας πράκτορας που εξερευνά πιθανές λύσεις (διαδρομές κατανομής εργασιών) και αφήνει "φερομόνες" στις βέλτιστες επιλογές, επηρεάζοντας μελλοντικές αποφάσεις. Το ACO είναι κατάλληλο για μεγάλα, δυναμικά συστήματα, αλλά παρουσιάζει αυξημένο υπολογιστικό κόστος.

Αλγόριθμος	Πλεονεκτήματα	Μειονεκτήματα	Ιδανικές Περιπτώσεις
Min-Min	Γρήγορη εξυπηρέτηση μικρών εργασιών	Καθυστερεί μεγάλες εργασίες	Ομοιόμορφα φορτία
Max-Min	Ισορροπεί μεγάλες και μικρές εργασίες	Μπορεί να αυξήσει καθυστέρηση σε μικρές εργασίες	Ανισοβαρή workloads
GA	Παγκόσμια βελτιστοποίηση, ευελιξία	Υψηλή πολυπλοκότητα	Μεγάλα, δυναμικά fog συστήματα
PSO	Γρήγορη σύγκλιση, αποδοτικότητα	Κίνδυνος τοπικού βέλτιστου	Scheduling με πολλούς περιορισμούς
HEFT	Ιδανικό για DAGs, καλή απόδοση	Όχι κατάλληλο για ανεξάρτητες εργασίες	Ετερογενή συστήματα με εξαρτήσεις
ACO	Προσαρμοστικότητα, ανακάλυψη νέων λύσεων	Αργή σύγκλιση, υψηλό κόστος	Δυναμικά, μεταβαλλόμενα περιβάλλοντα
Greedy/RR	Ταχύτητα, ευκολία υλοποίησης	Χαμηλή ποιότητα λύσης	Μικρής κλίμακας ή απλές εφαρμογές

3.6 Κριτήρια Αξιολόγησης Κατανομής Εργασιών

Η αξιολόγηση της αποτελεσματικότητας των μηχανισμών κατανομής εργασιών σε περιβάλλοντα υπολογιστικής ομίχλης απαιτεί τη χρήση συγκεκριμένων μετρικών απόδοσης που αντανakλούν τόσο την ποιότητα εξυπηρέτησης των εφαρμογών όσο και τη συνολική απόδοση του συστήματος. Σε αντίθεση με το cloud computing, όπου η έμφαση δίνεται κυρίως στη διαθεσιμότητα και την επεκτασιμότητα, η υπολογιστική ομίχλη δίνει προτεραιότητα στην ταχύτητα απόκρισης, την τοπική επεξεργασία και την αποδοτική χρήση περιορισμένων πόρων. Τα βασικότερα και συχνότερα χρησιμοποιούμενα κριτήρια αξιολόγησης είναι η καθυστέρηση (latency), ο ρυθμός επεξεργασίας (throughput) και η αξιοποίηση πόρων (utilization).

Η καθυστέρηση (latency) αποτελεί ίσως το πιο σημαντικό κριτήριο για την αξιολόγηση των αλγορίθμων κατανομής σε fog περιβάλλοντα. Αναφέρεται στον συνολικό χρόνο που μεσολαβεί από τη στιγμή που μια εργασία δημιουργείται ή αιτείται (από συσκευή ή χρήστη) έως τη στιγμή που αυτή ολοκληρώνεται. Περιλαμβάνει τόσο τον χρόνο μεταφοράς των δεδομένων προς τον fog κόμβο όσο και τον χρόνο εκτέλεσης της ίδιας της εργασίας. Σε εφαρμογές όπως αυτόνομα οχήματα, βιομηχανικός αυτοματισμός, ή τηλεϊατρική, η καθυστέρηση πρέπει να είναι εξαιρετικά μικρή (χαμηλού επιπέδου millisecond), γεγονός που καθιστά το latency βασική προτεραιότητα. Οι αλγόριθμοι κατανομής καλούνται να μειώνουν δραστικά τη χρονική υστέρηση με έξυπνη επιλογή κόμβων, πρόβλεψη συμφόρησης και αποφυγή επεξεργασίας σε απομακρυσμένα σημεία του δικτύου.

Ο ρυθμός επεξεργασίας (throughput) είναι το δεύτερο βασικό κριτήριο, το οποίο μετρά τον συνολικό αριθμό εργασιών που ολοκληρώνονται σε μια δεδομένη χρονική περίοδο. Αντιπροσωπεύει την ικανότητα του συστήματος να εξυπηρετεί υψηλό φόρτο εργασίας, ιδίως σε σενάρια όπου πολλές αιτήσεις υποβάλλονται ταυτόχρονα. Ένας αλγόριθμος κατανομής θεωρείται αποτελεσματικός όταν διατηρεί υψηλό throughput ακόμα και υπό συνθήκες συμφόρησης ή μεταβαλλόμενων πόρων. Στα περιβάλλοντα ομίχλης, αυτό το κριτήριο αξιολογεί την ευστάθεια και την επεκτασιμότητα του συστήματος, καθώς και την ικανότητά του να ανταποκρίνεται σε δυναμικά φορτία χωρίς να καταρρέει η απόδοση. Η επίτευξη υψηλού throughput προϋποθέτει καλή εξισορρόπηση φορτίου,

αποφυγή συμφόρησης και αποδοτικό προγραμματισμό εργασιών με βάση προτεραιότητες ή προθεσμίες.

Η αξιοποίηση πόρων (resource utilization) αποτελεί εξίσου σημαντικό κριτήριο, καθώς αντανακλά την αποδοτικότητα με την οποία χρησιμοποιούνται οι διαθέσιμοι υπολογιστικοί, αποθηκευτικοί και δικτυακοί πόροι. Σε ένα fog περιβάλλον, όπου οι πόροι είναι συχνά περιορισμένοι και ανομοιογενείς, η μέγιστη αξιοποίησή τους χωρίς υπερφόρτωση είναι απαραίτητη για τη διατήρηση βιώσιμης λειτουργίας. Το συγκεκριμένο κριτήριο μετρά πόσο πλήρως χρησιμοποιούνται πόροι όπως η CPU, η μνήμη, το bandwidth και η ενέργεια, σε σύγκριση με τη συνολική τους διαθεσιμότητα. Η χαμηλή αξιοποίηση συνεπάγεται ανεκμετάλλευτες δυνατότητες, ενώ η υπερβολικά υψηλή οδηγεί σε επιδείνωση της καθυστέρησης, αυξημένη πιθανότητα σφαλμάτων ή αποτυχίας κόμβων. Οι πιο εξελιγμένοι αλγόριθμοι κατανομής επιδιώκουν ισορροπία μεταξύ αξιοποίησης και σταθερότητας, διατηρώντας τους κόμβους ενεργούς σε επίπεδα που προσφέρουν απόδοση χωρίς να ξεπερνούν κρίσιμα όρια.

4 ΚΕΦΑΛΑΙΟ: FOGWORKFLOWSIM

4.1 Γενική Περιγραφή και Λειτουργία

Το FogWorkflowSim είναι ένα σύγχρονο εργαλείο προσομοίωσης που στοχεύει στην αξιολόγηση της εκτέλεσης ροών εργασιών (workflows) σε περιβάλλοντα υπολογιστικής ομίχλης (fog computing). Αναπτύχθηκε για να γεφυρώσει το κενό μεταξύ υπαρχόντων εργαλείων προσομοίωσης νέφους και ομίχλης, επιτρέποντας την μελέτη σεναρίων όπου εργασίες από συσκευές IoT κατανέμονται σε πόρους κοντά στο άκρο του δικτύου (edge/fog nodes) καθώς και σε απομακρυσμένους διακομιστές νέφους. Η ανάγκη για τέτοια εργαλεία προέκυψε επειδή τα παραδοσιακά συστήματα workflow που βασίζονται αποκλειστικά στο cloud δεν μπορούν να υποστηρίξουν εφαρμογές πραγματικού χρόνου σε IoT, λόγω των καθυστερήσεων δικτύου. Η αρχιτεκτονική fog computing προσθέτει ένα ενδιάμεσο επίπεδο υπολογιστικών πόρων κοντά στις συσκευές (κόμβοι ομίχλης) ώστε να μειώσει την καθυστέρηση απόκρισης. Χαρακτηριστικά παραδείγματα σεναρίων είναι οι έξυπνες εφαρμογές IoT όπως η απομακρυσμένη παρακολούθηση υγείας ή η διαχείριση κυκλοφορίας, όπου απαιτείται ταχεία τοπική επεξεργασία δεδομένων (π.χ. φιλτράρισμα σημάτων από αισθητήρες σε κόμβους ομίχλης) πριν τα δεδομένα αποσταλούν στο νέφος για αποθήκευση ή περαιτέρω ανάλυση. Το FogWorkflowSim επιτρέπει την προσομοίωση τέτοιων καταστάσεων, αξιολογώντας πώς η κατανομή των εργασιών μεταξύ συσκευών άκρου, ενδιάμεσων κόμβων ομίχλης και κέντρων νέφους επηρεάζει την απόδοση του συστήματος.

4.2 Κύρια χαρακτηριστικά

Το FogWorkflowSim έχει σχεδιαστεί ως ένα αυτοματοποιημένο, επεκτάσιμο εργαλείο που διευκολύνει τους ερευνητές να πειραματιστούν με διάφορες πολιτικές διαχείρισης πόρων και εργασιών σε περιβάλλον ομίχλης. Συγκεκριμένα υποστηρίζει: 1) αυτόματη δημιουργία μιας προσομοιωμένης υποδομής Fog (διαμόρφωση πολυεπίπεδου περιβάλλοντος με συσκευές IoT, κόμβους ομίχλης και διακομιστές cloud), 2) αυτόματη

εκτέλεση των εργασιών μιας εισαγόμενης ροής εργασιών πάνω σε αυτή την υποδομή, και 3) αυτόματη αξιολόγηση και σύγκριση της απόδοσης διαφορετικών στρατηγικών κατανομής εργασιών (offloading) και αλγορίθμων χρονοπρογραμματισμού με βάση βασικές μετρικές όπως ο χρόνος ολοκλήρωσης, η κατανάλωση ενέργειας και το κόστος. Το εργαλείο διαθέτει γραφικό περιβάλλον χρήστη (GUI) που καθοδηγεί τον χρήστη βήμα-βήμα στη διαδικασία πειραματισμού. Πιο συγκεκριμένα, ο χρήστης αρχικά διαμορφώνει το περιβάλλον ομίχλης, ορίζοντας τον αριθμό και τα χαρακτηριστικά των συσκευών σε κάθε επίπεδο (τερματικές συσκευές, κόμβοι ομίχλης, διακομιστές νέφους). Στη συνέχεια φορτώνει ή ορίζει μια workflow εφαρμογή (σενάριο εργασιών με τις μεταξύ τους εξαρτήσεις) – το FogWorkflowSim υποστηρίζει εισαγωγή workflows σε μορφή XML, π.χ. παραγόμενα από το εργαλείο WorkflowGenerator. Ακολουθώντας, μέσω του GUI, επιλέγονται οι στρατηγικές καταμερισμού φορτίου (computation offloading) και οι αλγόριθμοι χρονοπρογραμματισμού που θα χρησιμοποιηθούν, καθώς και οι στόχοι βελτιστοποίησης (χρόνος, ενεργειακή κατανάλωση, κόστος) που ενδιαφέρουν.

Το FogWorkflowSim παρέχει βιβλιοθήκη έτοιμων επιλογών: ενδεικτικά, περιλαμβάνονται απλοί αλγόριθμοι όπως *Min-Min*, *Max-Min*, *First-Come First-Served (FCFS)* και *Round-Robin* (για βελτιστοποίηση χρόνου), καθώς και πιο σύνθετες μεταερευνητικές μέθοδοι όπως *Particle Swarm Optimization (PSO)* και *Genetic Algorithm (GA)* που μπορούν να χρησιμοποιηθούν για πολυκριτηριακή βελτιστοποίηση (χρόνος-ενέργεια-κόστος). Επιπλέον, το σύστημα επιτρέπει την προσθήκη δικών του αλγορίθμων από τον χρήστη για χρονοπρογραμματισμό ή offloading, επεκτείνοντας τις βιβλιοθήκες του.

Τέλος, ο χρήστης εκκινεί την προσομοίωση και συλλέγει αποτελέσματα, με το εργαλείο να παράγει αναφορές για τις επιλεγμένες μετρικές απόδοσης. Τα αποτελέσματα μπορούν να απεικονιστούν μέσω του GUI (π.χ. σε γραφήματα) και να εξαχθούν για περαιτέρω ανάλυση. Χάρη σε αυτές τις αυτοματοποιημένες λειτουργίες, το FogWorkflowSim λειτουργεί ως πειραματική πλατφόρμα για την έρευνα πάνω σε συστήματα fog-workflows. Είναι γραμμένο σε γλώσσα Java (συμβατό με JDK 1.8) και διατίθεται τόσο ως εφαρμογή desktop (Java project) όσο και μέσω web διεπαφής (Web UI) για online εκτέλεση προσομοιώσεων.

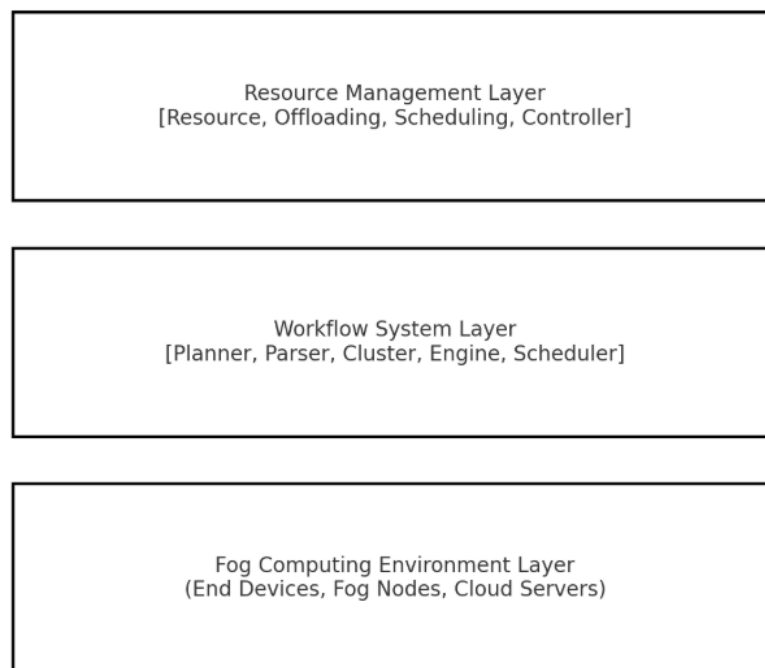
4.3 Υποστηριζόμενα σενάρια προσομοίωσης

Το FogWorkflowSim έχει τη δυνατότητα προσομοίωσης ποικίλων σεναρίων που εμπίπτουν στα πεδία του Internet of Things, του edge computing και των υβριδικών συστημάτων cloud-fog. Βασικό σενάριο χρήσης είναι αυτό όπου πλήθος αισθητήρων/συσκευών IoT παράγουν εργασίες (tasks) που πρέπει να επεξεργαστούν είτε τοπικά είτε σε ανώτερα επίπεδα υπολογισμού. Το εργαλείο μοντελοποιεί ρητά την ιεραρχία συσκευές τέλους – κόμβοι ομίχλης – νέφος, επιτρέποντας την μελέτη διαφόρων τοπολογιών: από μια καθαρά τοπική επεξεργασία (All-in-Fog) όπου όλες οι εργασίες εκτελούνται πλησίον των δεδομένων, μέχρι μια παραδοσιακή cloud υλοποίηση (All-in-Cloud) όπου το σύνολο των εργασιών μεταφορτώνεται σε απομακρυσμένους servers. Φυσικά υποστηρίζονται και ενδιάμεσες πολιτικές, όπου αποφασίζεται δυναμικά ποια εργασία θα τρέξει σε ποιο επίπεδο ανάλογα με κριτήρια. Για παράδειγμα, στο σενάριο της έξυπνης υγείας (Smart Healthcare) που αναφέρθηκε, το FogWorkflowSim μπορεί να προσομοιώσει πολλούς κόμβους ομίχλης (π.χ. τοπικούς servers σε κλινικές) που επεξεργάζονται παράλληλα δεδομένα ασθενών από φορητούς αισθητήρες, μειώνοντας την καθυστέρηση και τον όγκο δεδομένων προς το cloud. Έτσι, δίνεται η δυνατότητα αξιολόγησης του πόσο αποφορτίζεται το δίκτυο και οι cloud πόροι, και πόσο βελτιώνεται ο χρόνος απόκρισης, υπό διάφορες παραμετροποιήσεις (αριθμός συσκευών, ισχύς επεξεργαστών, διαθέσιμο bandwidth, κ.ά.). Συνοψίζοντας, το FogWorkflowSim είναι κατάλληλο για την προσομοίωση σύνθετων workflow εφαρμογών σε κατανομημένα περιβάλλοντα ομίχλης, όπου εμπλέκονται ετερογενείς υπολογιστικοί πόροι κοντά στις πηγές δεδομένων, προσφέροντας ένα ευέλικτο εργαστήριο δοκιμών για ερευνητικά σενάρια στο IoT και το edge computing.

4.4 Τεχνική Αρχιτεκτονική

Στο Επίπεδο Υπολογιστικού Περιβάλλοντος Ομίχλης (Fog Computing Environment layer) μοντελοποιείται η φυσική υποδομή: τερματικές συσκευές (End Devices), ενδιάμεσοι κόμβοι ομίχλης (Fog Nodes) και διακομιστές νέφους (Cloud Servers). Πάνω από αυτό, το Επίπεδο Συστήματος Ροής Εργασιών (Workflow System layer)

περιλαμβάνει τις βασικές μονάδες λογισμικού για τον προγραμματισμό και την εκτέλεση των workflows: Planner, Parser, Cluster, Engine και Scheduler. Στην κορυφή, το Επίπεδο Διαχείρισης Πόρων (Resource Management layer) παρέχει επιπλέον λειτουργικότητα μέσω των ενοτήτων Resource, Offloading, Scheduling και Controller, που αντιστοιχούν στη διαχείριση των πόρων/υποδομών, στις στρατηγικές offloading, στους αλγορίθμους χρονοπρογραμματισμού και στον έλεγχο/μέτρηση της απόδοσης αντίστοιχα.



Εικόνα 1: Αρχιτεκτονική FogWorkflowSim

Η σχεδίαση του FogWorkflowSim βασίστηκε στην ολοκλήρωση δύο υπαρχόντων πλαισίων προσομοίωσης, του iFogSim (για προσομοίωση κατανεμημένων περιβαλλόντων ομίχλης/IoT) και του WorkflowSim (για προσομοίωση workflows πάνω από το CloudSim). Με άλλα λόγια, οι λειτουργίες μοντελοποίησης φυσικών πόρων από το iFogSim συνδυάστηκαν με τις λειτουργίες διαχείρισης ροών εργασιών από το WorkflowSim, με τις απαραίτητες τροποποιήσεις ώστε τα δύο συστήματα να συνεργαστούν ομαλά. Το αποτέλεσμα είναι μια πολυεπίπεδη αρχιτεκτονική, όπου κάθε επίπεδο προσφέρει συγκεκριμένες υπηρεσίες προς τα ανώτερα επίπεδα. Στη συνέχεια περιγράφονται οι κύριες κλάσεις και μονάδες του FogWorkflowSim ανά επίπεδο, καθώς και η ροή δεδομένων κατά την προσομοίωση.

4.5 Βασικές Κλάσεις και Μονάδες του Συστήματος

Επίπεδο Υπολογιστικού Περιβάλλοντος (Fog Computing Environment): Το κατώτερο επίπεδο αφορά την προσομοίωση της υλικής υποδομής σε ένα σενάριο ομίχλης. Το FogWorkflowSim ορίζει μια γενική κλάση FogDevice η οποία χρησιμοποιείται για να εκπροσωπήσει κάθε τύπο υπολογιστικού πόρου στο περιβάλλον ομίχλης. Ένα αντικείμενο FogDevice μπορεί, μέσω παραμετροποίησης, να προσομοιώσει είτε μια απλή συσκευή IoT/αισθητήρα, είτε έναν ισχυρό edge server, είτε έναν απομακρυσμένο server στο cloud. Η ευελιξία αυτή επιτυγχάνεται με τον ορισμό διαφόρων χαρακτηριστικών υλικού (όπως η επεξεργαστική ισχύς σε MIPS, η διαθέσιμη μνήμη και αποθήκευση, το εύρος ζώνης uplink/downlink, η κατανάλωση ενέργειας ανά κατάσταση λειτουργίας κ.ά.), τα οποία μπορεί να ρυθμίσει ο χρήστης για κάθε FogDevice ώστε να αντικατοπτρίζει το αντίστοιχο πραγματικό στοιχείο. Επιπλέον, η κλάση αυτή (κληρονομεί συμπεριφορά από αντίστοιχες κλάσεις του iFogSim/CloudSim) παρέχει τις μεθόδους για διαχείριση και κατανομή πόρων (π.χ. κατανομή υπολογιστικών δυνατοτήτων σε διεργασίες) καθώς και για την προσομοίωση της εκτέλεσης εργασιών πάνω στο συγκεκριμένο device. Στο μοντέλο του FogWorkflowSim, όλοι οι φυσικοί κόμβοι (αισθητήρες, συσκευές χρηστών, κόμβοι ομίχλης, data centers νέφους) αναπαρίστανται από στιγμιότυπα της FogDevice, οργανωμένα ιεραρχικά σε τρία υπο-επίπεδα: στρώμα συσκευών τέλους, στρώμα κόμβων ομίχλης και στρώμα διακομιστών cloud. Ο χρήστης μπορεί να καθορίσει τον αριθμό των κόμβων σε κάθε στρώμα (π.χ. πλήθος αισθητήρων, αριθμό fog nodes), καθώς και τις μεταξύ τους συνδέσεις δικτύου (στην τρέχουσα έκδοση υποστηρίζεται στατικό δίκτυο με προκαθορισμένο bandwidth/latency ανά σύνδεση όπως στο iFogSim).

Επίπεδο Συστήματος Ροής Εργασιών (Workflow System): Το μεσαίο επίπεδο αρχιτεκτονικής αφορά το λογισμικό που διαχειρίζεται την φόρτωση, προγραμματισμό και εκτέλεση των workflow tasks στο προσομοιωμένο περιβάλλον. Αποτελείται από πέντε κύριες μονάδες (modules) που αλληλεπιδρούν μεταξύ τους σε μια σωληνωτή ροή (pipeline) από την είσοδο μιας ροής εργασιών μέχρι την ολοκλήρωση όλων των εργασιών της:

- *Planner*: Αναλαμβάνει την έναρξη της προσομοίωσης και τον γενικό συντονισμό. Μπορεί να θεωρηθεί ως ο ελεγκτής που ενεργοποιεί τα υπόλοιπα βήματα μόλις το περιβάλλον και οι παράμετροι έχουν οριστεί.
- *Parser*: Είναι υπεύθυνος για την ανάγνωση/παράλυση του αρχείου workflow (XML) που δίνει ο χρήστης. Διαβάζει τον ορισμό των tasks, των δεδομένων εισόδου/εξόδου τους και των μεταξύ τους εξαρτήσεων, και δημιουργεί αντίστοιχα αντικείμενα Task μέσα στο σύστημα. Κάθε Task αντιπροσωπεύει μια αυτοτελή εργασία της ροής (π.χ. ένα βήμα υπολογισμού σε μια εφαρμογή επιστημονικού workflow).
- *Cluster*: Η μονάδα αυτή (προαιρετικά) μπορεί να ομαδοποιήσει πολλαπλές tasks σε ενιαίες δουλειές (jobs) προτού προγραμματιστούν, ανάλογα με συγκεκριμένο αλγόριθμο συσσωμάτωσης. Για παράδειγμα, σε workflows με πολύ μικρές εργασίες, ομαδοποιώντας τις σε μεγαλύτερες δουλειές μειώνεται το scheduling overhead. Ο χρήστης μπορεί να επιλέξει ή να εφαρμόσει clustering αλγόριθμο, διαφορετικά η κάθε task μπορεί να αντιμετωπιστεί ξεχωριστά.
- *Engine*: Αποτελεί τη μηχανή εκτέλεσης του workflow. Λαμβάνει τις έτοιμες προς εκτέλεση εργασίες/δουλειές (από τον Parser ή/και το Cluster) και τις υποβάλλει στον Scheduler για εκχώρηση σε πόρους, ακολουθώντας τις εξαρτήσεις (δηλ. μόνο όταν όλες οι προαπαιτούμενες tasks μιας task έχουν ολοκληρωθεί, τότε την προωθεί για εκτέλεση). Επίσης, παρακολουθεί την πορεία εκτέλεσης και μπορεί να επαναπρογραμματίσει ή να επανυποβάλει μια task σε περίπτωση αποτυχίας (failure) ή καθυστέρησης. Τέλος, το Engine σηματοδοτεί την ολοκλήρωση της προσομοίωσης όταν όλες οι εργασίες του workflow έχουν τερματιστεί.
- *Scheduler*: Είναι ίσως η σημαντικότερη μονάδα, καθώς υλοποιεί τον αλγόριθμο χρονικού προγραμματισμού των εργασιών πάνω στους διαθέσιμους πόρους. Ο Scheduler αποτελεί το σημείο εισαγωγής στο σύστημα κάθε στρατηγικής scheduling που θέλουμε να δοκιμάσουμε – το FogWorkflowSim διαθέτει πολλούς έτοιμους schedulers (FCFS, Round Robin, Min-Min, κτλ όπως προαναφέρθηκε, υλοποιημένους ως κλάσεις) και δίνει τη δυνατότητα εύκολης ενσωμάτωσης νέων. Η βασική λειτουργία του Scheduler είναι να

αποφασίζει ποια εργασία θα εκτελεστεί σε ποιον Fog device και πότε, με βάση τον επιλεγμένο αλγόριθμο. Για τον σκοπό αυτό, ο Scheduler αλληλεπιδρά με το υποκείμενο επίπεδο πόρων: δημιουργεί εάν χρειάζεται νέες Εικονικές Μηχανές (VMs) στους διαθέσιμους κόμβους (FogDevices) και αναθέτει τις εργασίες σε συγκεκριμένες VMs για εκτέλεση. Κατά την ανάθεση, ενημερώνει την κατάσταση των πόρων (π.χ. φόρτο CPU κάθε Fog node) και όταν ολοκληρωθεί η εκτέλεση μιας task σε έναν κόμβο, λαμβάνει τα αποτελέσματα/ενημερώσεις (callbacks) και τα περνάει πίσω στο Engine. Αξίζει να σημειωθεί ότι στον Scheduler μπορεί να ενσωματώνεται και η στρατηγική offloading, δηλ. οι κανόνες που καθορίζουν σε ποια στιβάδα (edge ή cloud) θα σταλεί κάθε εργασία. Στο FogWorkflowSim, το κομμάτι του offloading μπορεί να υλοποιηθεί είτε ως μέρος του scheduler (π.χ. ένας scheduler που πρώτα αποφασίζει fog/cloud και μετά διαλέγει συγκεκριμένο κόμβο), είτε μέσω του ξεχωριστού *Offloading Module* (περιγράφεται πιο κάτω) που συνεργάζεται με έναν απλούστερο scheduler.

Επίπεδο Διαχείρισης Πόρων (Resource Management): Το ανώτερο επίπεδο συγκεντρώνει διάφορες βοηθητικές λειτουργίες που επιτρέπουν την ευέλικτη πειραματική αξιολόγηση του συστήματος. Χωρίζεται σε τέσσερις κύριες ενότητες (βιβλιοθήκες κλάσεων):

- *Resource*: Αντιπροσωπεύει τη δεξαμενή των υπολογιστικών πόρων που είναι διαθέσιμοι στο σύστημα. Σε επίπεδο υλοποίησης, πρόκειται για μια εικονική στρώση πάνω από τα επιμέρους FogDevices, η οποία παρέχει ενιαία διαχείριση των κατανεμημένων πόρων (CPU, μνήμη, αποθήκευση) σαν να επρόκειτο για ένα κοινόχρηστο σύνολο. Ουσιαστικά, εδώ υλοποιούνται οι μηχανισμοί κατανομής/δέσμευσης πόρων για τις νέες διεργασίες (π.χ. δημιουργία VM σε συγκεκριμένο host, δέσμευση πυρήνων CPU για μια task κλπ).
- *Offloading*: Πρόκειται για μια βιβλιοθήκη στρατηγικών μεταφόρτωσης υπολογισμού. Παρέχει έτοιμες υλοποιήσεις πολιτικών που καθορίζουν αν μια εργασία θα εκτελεστεί τοπικά (στο άκρο) ή θα αποσταλεί σε ανώτερο επίπεδο (π.χ. cloud). Όπως έχει ήδη αναφερθεί, το FogWorkflowSim περιλαμβάνει μερικές βασικές στρατηγικές (All-in-Fog, All-in-Cloud, καθώς και πιο έξυπνες στρατηγικές που λαμβάνουν υπόψη τους περιορισμούς χρόνου απόκρισης ή

ενεργειακής κατανάλωσης). Ο χρήστης μπορεί να επεκτείνει τη βιβλιοθήκη προσθέτοντας νέες στρατηγικές offloading, ώστε να δοκιμάσει π.χ. κάποιο βέλτιστο αλγόριθμο κατανομής φορτίου που ανέπτυξε.

- *Scheduling*: Αντίστοιχα, αυτή η ενότητα αποτελεί μια βιβλιοθήκη εναλλακτικών αλγορίθμων χρονοπρογραμματισμού (scheduling algorithms) για το Workflow. Περιλαμβάνει τους κλασικούς αλγορίθμους (FCFS, RoundRobin, MinMin, κ.α.) καθώς και τους πιο εξελιγμένους (GA, PSO) που προσφέρονται από το εργαλείο. Η ενσωμάτωση νέων αλγορίθμων γίνεται εύκολα, καθιστώντας το FogWorkflowSim επεκτάσιμο και προσαρμόσιμο στις ανάγκες του εκάστοτε πειράματος.
- *Controller*: Η ενότητα αυτή αντιστοιχεί στον μηχανισμό συλλογής μετρικών απόδοσης και γενικότερα παρακολούθησης της προσομοίωσης. Ο Controller υπολογίζει τις χρονικές και ενεργειακές μετρήσεις κατά την εκτέλεση κάθε εργασίας και συνολικά του workflow, καθώς και τυχόν κόστος (αν ορίζονται μοντέλα κόστους χρήσης πόρων). Στο FogWorkflowSim έχουν υλοποιηθεί τρεις βασικές κατηγορίες μετρικών (χρόνος, ενέργεια, οικονομικό κόστος), αλλά το σύστημα υποστηρίζει επεκτασιμότητα επιτρέποντας στον ερευνητή να εισαγάγει και επιπλέον δείκτες απόδοσης προς μέτρηση, ανάλογα με τις απαιτήσεις της μελέτης του.

4.6 Ροή Δεδομένων και Λειτουργία Προσομοίωσης

Η εσωτερική λειτουργία του FogWorkflowSim ακολουθεί τα βήματα που περιγράφηκαν στην ενότητα της γενικής χρήσης, με την κάθε μονάδα να παίζει τον ρόλο της στην ροή δεδομένων από την αρχή μέχρι το τέλος της προσομοίωσης. Συνοπτικά, η αλληλουχία έχει ως εξής:

1. Εκκίνηση και Προετοιμασία: Ο Planner ενεργοποιείται μόλις ο χρήστης καθορίσει το σενάριο και τις παραμέτρους. Το σύστημα δημιουργεί τα

αντικείμενα FogDevice που αντιστοιχούν στους διαθέσιμους κόμβους (με βάση τις ρυθμίσεις του χρήστη για το πόσοι κόμβοι υπάρχουν σε κάθε επίπεδο και τι δυνατότητες έχουν). Σε αυτό το στάδιο διαμορφώνεται και το εικονικό δίκτυο επικοινωνίας μεταξύ των κόμβων (π.χ. ιεραρχική δενδροειδής τοπολογία με καθορισμένο bandwidth ανά σύνδεση).

2. Φόρτωση Workflow: Ο Parser διαβάζει το αρχείο XML του workflow και δημιουργεί τις task. Για κάθε task καταγράφονται ιδιότητες όπως ο απαιτούμενος χρόνος CPU (σε MI – Million Instructions), τα δεδομένα που παράγει ή καταναλώνει (σε MB, σχετικό με latency δικτύου) και η λίστα των προαπαιτούμενων tasks (dependencies). Όλες αυτές οι πληροφορίες αποθηκεύονται στα αντίστοιχα αντικείμενα Task που εισάγονται στο σύστημα. Αν έχει καθοριστεί διαδικασία clustering, τότε οι task ομαδοποιούνται σε jobs σύμφωνα με τον επιλεγμένο αλγόριθμο από το module Cluster.
3. Προγραμματισμός Εργασιών: Το Engine λαμβάνει τις έτοιμες προς εκτέλεση εργασίες (δηλ. όσες δεν έχουν ανεκπλήρωτες εξαρτήσεις) και τις περνά στον Scheduler για να προγραμματιστούν. Ο Scheduler τότε αποφασίζει την ανάθεση κάθε εργασίας σε συγκεκριμένο FogDevice. Εδώ λαμβάνονται υπόψη οι πολιτικές offloading (αν μια εργασία επιτρέπεται ή ενδείκνυται να τρέξει στο cloud ή πρέπει να μείνει στο fog) καθώς και το ποιός ακριβώς κόμβος έχει τους επαρκείς πόρους τη δεδομένη στιγμή. Ο Scheduler μπορεί να δημιουργήσει νέες VM σε έναν κόμβο-στόχο και αναθέτει την εργασία σε μια VM (συσχετίζοντας έτσι την task με έναν πυρήνα CPU, συγκεκριμένη μνήμη κ.λπ. στον κόμβο αυτό). Η ανάθεση αυτή αποτελεί γεγονός στο επίπεδο προσομοίωσης (CloudSim event) που ξεκινά την εκτέλεση της task.
4. Εκτέλεση και Offloading: Η κάθε εργασία εκτελείται πλέον στον προσομοιωμένο πόρο που της ανατέθηκε. Το FogWorkflowSim, χρησιμοποιώντας τις κλάσεις του CloudSim/iFogSim, προσομοιώνει την κατανάλωση CPU και χρόνου: μειώνει τον υπολειπόμενο όγκο υπολογισμού (MI) της task όσο περνά ο προσομοιωμένος χρόνος και οι κύκλοι CPU του FogDevice διατίθενται σε αυτήν. Αν κατά την εκτέλεση η task χρειάζεται να μεταφέρει δεδομένα (π.χ. να στείλει ενδιάμεσα αποτελέσματα σε επόμενη task), το σύστημα προσομοιώνει και την καθυστέρηση δικτύου ανάλογα με τα

χαρακτηριστικά σύνδεσης μεταξύ των εμπλεκόμενων FogDevices. Η στρατηγική offloading υπεισέρχεται όταν μια task *θα μπορούσε* να εκτελεστεί σε διαφορετικά επίπεδα: π.χ. σε ένα υβριδικό σενάριο, ο Scheduler/Offloading module μπορεί να αποφασίσει να στείλει μια βαριά εργασία στο cloud για να μην επιβαρυνθεί ο τοπικός κόμβος – σε αυτή την περίπτωση, δημιουργείται ένα event μεταφοράς της task προς έναν FogDevice του επιπέδου cloud και η εκτέλεση συνεχίζεται εκεί. Όλα αυτά συμβαίνουν διαφανώς κατά την προσομοίωση, βάσει του σεναρίου που έχει στηθεί.

5. Συλλογή Αποτελεσμάτων: Μόλις μια task ολοκληρωθεί, ο αντίστοιχος FogDevice ενημερώνει τον Scheduler/Engine ότι τελείωσε (πέρασ event). Το Engine τότε ελέγχει αν απελευθερώνονται άλλες task που εξαρτώνταν από αυτήν (οπότε τώρα όλες οι προαπαιτούμενες έχουν ολοκληρωθεί) και αν ναι, τις προωθεί και αυτές για εκτέλεση (επιστροφή στο βήμα 3). Το σύστημα συνεχίζει τον κύκλο προγραμματισμού–εκτέλεσης μέχρι να τερματίσουν όλες οι εργασίες. Παράλληλα, το module Controller καταγράφει τις μετρικές απόδοσης: για κάθε task υπολογίζεται ο συνολικός χρόνος που μεσολάβησε από την υποβολή μέχρι την ολοκλήρωση (Makespan), η συνολική ενέργεια που κατανάλωσαν οι συσκευές όπου έτρεξε (π.χ. άθροισμα ενέργειας CPU, μνήμης και μεταδόσεων δικτύου στους εμπλεκόμενους FogDevices) και ενδεχομένως το κόστος εκτέλεσης (αν έχει οριστεί μοντέλο κόστους ανά χρήση CPU ή μεταφορά δεδομένων στο cloud). Μόλις το Engine διαπιστώσει ότι δεν απομένουν άλλες εργασίες, ολοκληρώνει την προσομοίωση και ο Planner/Controller παράγει τα τελικά συγκεντρωτικά αποτελέσματα (συνολικός χρόνος ολοκλήρωσης όλου του workflow, μέση καθυστέρηση, συνολική ενέργεια που δαπανήθηκε, κόστος, κ.ά.).

Τα αποτελέσματα αυτά εμφανίζονται στον χρήστη μέσω του γραφικού περιβάλλοντος, όπου μπορεί να συγκρίνει διαφορετικές στρατηγικές πειραματισμού δίπλα-δίπλα. Για παράδειγμα, μια διπλωματική εργασία *θα μπορούσε* να χρησιμοποιήσει το FogWorkflowSim για να συγκρίνει τον χρόνο ολοκλήρωσης και την ενεργειακή κατανάλωση ενός δεδομένου workflow υπό διαφορετικές πολιτικές scheduling (π.χ. Round-Robin έναντι GA) και διαφορετικά επίπεδα offloading (π.χ. όλα τα tasks στο fog έναντι ορισμένων tasks στο cloud). Το εργαλείο αυτοματοποιεί όλη τη διαδικασία, επιτρέποντας στον ερευνητή να εστιάσει στην ανάλυση των αποτελεσμάτων και στην

εξαγωγή συμπερασμάτων σχετικά με τη βέλτιστη διαχείριση ροών εργασιών σε περιβάλλοντα ομίχλης.

5 ΚΕΦΑΛΑΙΟ: ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

5.1 Ρύθμιση Περιβάλλοντος & Προετοιμασία

Στο παρόν κεφάλαιο περιγράφεται το πειραματικό μέρος της εργασίας, όπως εκτελέστηκε με χρήση του FogWorkflowSim, ενός εργαλείου προσομοίωσης για αξιολόγηση απόδοσης workflows σε περιβάλλον Fog Computing (παρουσιάστηκε λεπτομερώς σε προηγούμενο κεφάλαιο). Η πειραματική διάταξη βασίστηκε σε προσωπικό υπολογιστή με επεξεργαστή Intel Core i5 (3.2GHz), μνήμη RAM 16GB και λειτουργικό σύστημα Windows 10. Όλες οι εφαρμογές υλοποιήθηκαν σε γλώσσα Java (JDK 1.8) και η εκτέλεση των προσομοιώσεων πραγματοποιήθηκε μέσω του περιβάλλοντος Eclipse IDE. Το FogWorkflowSim εγκαταστάθηκε ως Java project στο Eclipse, με εισαγωγή του πηγαιού κώδικα από το αποθετήριο GitHub των δημιουργών. Η εγκατάσταση απαιτήσε τη ρύθμιση των απαραίτητων βιβλιοθηκών (όπως των CloudSim και WorkflowSim, τις οποίες επεκτείνει το FogWorkflowSim) και την εξασφάλιση ότι χρησιμοποιείται η έκδοση Java 1.8 για συμβατότητα.

Για την προετοιμασία των εισόδων της προσομοίωσης χρησιμοποιήθηκαν αρχεία περιγραφής workflow σε μορφή XML (DAX). Το FogWorkflowSim παρέχει ενσωματωμένη υποστήριξη για workflows όπως το Montage και επιτρέπει την εισαγωγή αρχείων ορισμού workflow από τον χρήστη. Τα αρχεία XML του Montage περιγράφουν λεπτομερώς τα tasks (εργασίες) του workflow, τα δεδομένα εισόδου/εξόδου τους και τις μεταξύ τους εξαρτήσεις. Στα πλαίσια του πειράματος, προετοιμάστηκαν πολλαπλές εκδόσεις του Montage workflow με διαφορετικό πλήθος tasks (20, 40, 60, 80 και 100 tasks), ώστε να εξεταστεί η επίδραση της κλίμακας του workflow στην απόδοση των αλγορίθμων χρονοπρογραμματισμού που εξετάστηκαν. Κάθε αρχείο εισόδου αντιστοιχούσε σε μια παραλλαγή του Montage με τον συγκεκριμένο αριθμό εργασιών, διατηρώντας παρόμοια δομή αλλά διαφορετική κλίμακα. Πριν την εκτέλεση, επαληθεύτηκε ότι το FogWorkflowSim μπορούσε να φορτώσει σωστά τα εν λόγω αρχεία και να δημιουργήσει το αντίστοιχο γράφημα workflow στο περιβάλλον προσομοίωσης.

5.2 Σενάρια Προσομοίωσης

Για την αξιολόγηση χρησιμοποιήθηκαν πέντε διαφορετικά σενάρια προσομοίωσης, ένα για κάθε πλήθος εργασιών (tasks) που αναφέρθηκε ανωτέρω: 20, 40, 60, 80 και 100 tasks. Σε όλα τα σενάρια, το περιβάλλον Fog Computing παρέμεινε σταθερό και συμβατό με εκείνο που περιγράφεται στο FogWorkflowSim. Ειδικότερα, υιοθετήθηκε μια τοπολογία με 1 τερματική συσκευή (End Device), 2 κόμβους ομίχλης (Fog Nodes) και 3 διακομιστές νέφους (Cloud Servers). Οι υπολογιστικοί πόροι κάθε επιπέδου ρυθμίστηκαν σύμφωνα με τις προδιαγραφές του εργαλείου: η τερματική συσκευή διαθέτει υπολογιστική ισχύ 1000 MIPS, κάθε κόμβος ομίχλης 1300 MIPS και κάθε διακομιστής του νέφους 1600 MIPS. Επιπλέον, ορίστηκαν παράμετροι κατανάλωσης ενέργειας για την τερματική συσκευή (π.χ. 700 mW κατανάλωση σε ενεργό κατάσταση και 30 mW σε αδράνεια), ενώ για λόγους απλότητας αγνοήθηκε η κατανάλωση ενέργειας στους κόμβους ομίχλης και στο νέφος. Αυτό ευθυγραμμίζεται με την υπόθεση ότι η κύρια πηγή ενεργειακής επιβάρυνσης είναι η συσκευή τελικού χρήστη, την οποία επιδιώκουμε να προστατεύσουμε ενεργειακά μέσω offloading. Οι ταχύτητες δικτύου ρυθμίστηκαν επίσης βάσει της βιβλιογραφίας: για τη σύνδεση της συσκευής με το επίπεδο ομίχλης/νέφους χρησιμοποιήθηκε uplink bandwidth 20 Mbps και download bandwidth 40 Mbps, ώστε να προσομοιωθεί ρεαλιστικά η μετάδοση δεδομένων από την συσκευή προς το Fog και το Cloud.

Σε κάθε σενάριο εκτελέστηκαν και αξιολογήθηκαν έξι διαφορετικοί αλγόριθμοι προγραμματισμού εργασιών (scheduling), οι οποίοι υποστηρίζονται εγγενώς από το FogWorkflowSim. Οι αλγόριθμοι αυτοί αντιπροσωπεύουν τόσο απλές ουρές προτεραιότητας όσο και πιο εξελιγμένες μεθόδους βελτιστοποίησης:

- FCFS (First-Come, First-Served) – Πρώτη εισαγωγή, πρώτη εκτέλεση: οι εργασίες εκτελούνται με τη σειρά που καθίστανται διαθέσιμες, χωρίς περαιτέρω βελτιστοποίηση.
- Min-Min – Ελαχιστοποίηση του ελάχιστου χρόνου: σε κάθε βήμα προγραμματίζεται πρώτα η διαθέσιμη εργασία που μπορεί να ολοκληρωθεί συντομότερα (λαμβάνοντας υπόψη τους διαθέσιμους πόρους), με στόχο τη γρήγορη ολοκλήρωση όσο το δυνατόν περισσότερων εργασιών.

- Max-Min – Ελαχιστοποίηση του μέγιστου χρόνου: προτεραιοποιεί τις μεγαλύτερες σε διάρκεια εργασίες πρώτα, ώστε να αρχίσουν νωρίς και να μη μείνουν όλες προς το τέλος, επιδιώκοντας τη μείωση του χρόνου ολοκλήρωσης του πιο αργού task.
- Round Robin – Κυκλική κατανομή: οι εργασίες ανατίθενται κυκλικά στους διαθέσιμους πόρους χωρίς να λαμβάνεται υπόψη η διάρκειά τους, διασφαλίζοντας μια μορφή ισοκατανομής φορτίου αλλά χωρίς βέλτιστη αξιοποίηση των πόρων.
- PSO (Particle Swarm Optimization) – Αλγόριθμος σμήνους σωματιδίων: μια μετα-ευριστική μέθοδος που μοντελοποιεί πιθανές λύσεις ως «σωματίδια» και επαναληπτικά βελτιώνει τον προγραμματισμό των εργασιών με βάση την συλλογική ευφυΐα σμήνους. Στόχος του PSO είναι η προσέγγιση της βέλτιστης εκχώρησης εργασιών σε πόρους, ελαχιστοποιώντας το χρόνο ολοκλήρωσης (ή άλλους στόχους απόδοσης).
- GA (Genetic Algorithm) – Γενετικός αλγόριθμος: μετα-ευριστική εμπνευσμένη από τη φυσική εξέλιξη, που διατηρεί έναν πληθυσμό πιθανών λύσεων (χρονοπρογραμματισμών) και τις βελτιώνει μέσω γενετικών τελεστών (διασταύρωση, μετάλλαξη) προς ένα βέλτιστο ή εγγύς βέλτιστο προγραμματισμό των εργασιών.

Οι παράμετροι των μετα-ευριστικών αλγορίθμων ρυθμίστηκαν σύμφωνα με τις προτεινόμενες τιμές των δημιουργών του FogWorkflowSim. Συγκεκριμένα, για τον αλγόριθμο PSO χρησιμοποιήθηκε πληθυσμός 30 σωματιδίων, συντελεστές μάθησης $C1 = C2 = 2$, αδράνεια (inertia weight) ίση με 1, και εκτελέστηκαν 100 επαναλήψεις βελτιστοποίησης σε κάθε προσομοίωση. Αντίστοιχα, ο GA εκτελέστηκε με πληθυσμό 50 ατόμων, λόγο διασταύρωσης 0.8, πιθανότητα μετάλλαξης 0.1 και 100 γενεές εξέλιξης. Αυτές οι ρυθμίσεις διασφαλίζουν ότι οι δύο μετα-ευριστικοί αλγόριθμοι έχουν επαρκή χρόνο για να βελτιστοποιήσουν τον προγραμματισμό, ώστε τα αποτελέσματά τους να είναι συγκρίσιμα και (σχεδόν) συγκλίνοντα.

Ανεξάρτητα από τον αλγόριθμο χρονοπρογραμματισμού, σε όλα τα σενάρια εφαρμόστηκε η ίδια στρατηγική offloading (μεταφοράς εργασιών) για την κατανομή των εργασιών στα διαθέσιμα επίπεδα (συσκευή, ομίχλη, νέφος). Επιλέχθηκε η προκαθορισμένη στρατηγική «Simple» που παρέχει το FogWorkflowSim. Η

στρατηγική Simple στοχεύει στη μείωση της ενεργειακής κατανάλωσης της τερματικής συσκευής υπό περιορισμούς χρόνου απόκρισης – ουσιαστικά, αποφασίζει αν μια εργασία θα εκτελεστεί τοπικά στη συσκευή, σε κοντινό κόμβο ομίχλης ή θα αποφορτιστεί στο νέφος, με κριτήριο την τήρηση μιας προδιαγεγραμμένης προθεσμίας (deadline) του workflow και την εξοικονόμηση ενέργειας στη συσκευή. Πρακτικά, η Simple offloading υλοποιεί μια μίξη των δύο άκρων: All-in-Fog (όλες οι εργασίες στην ομίχλη, χωρίς χρήση νέφους) και All-in-Cloud (όλες οι εργασίες στο νέφος, χωρίς χρήση ενδιάμεσων κόμβων ομίχλης). Αντίθετα με αυτές τις ακραίες περιπτώσεις, η Simple επιτρέπει σε κάποιες εργασίες να εκτελεστούν στην τερματική συσκευή αν είναι ελαφριές, ενώ μεταφέρει βαρύτερες ή χρονικά κρίσιμες εργασίες στους πόρους του fog ή του cloud ώστε να μην παραβιαστεί το deadline.

Με τον τρόπο αυτό, εξισορροπείται το φορτίο: η συσκευή επωμίζεται όσο το δυνατόν λιγότερη επεξεργασία (γλιτώνοντας ενέργεια), χωρίς όμως το workflow να καθυστερήσει υπερβολικά. Σημειώνεται ότι η ίδια στρατηγική offloading διατηρήθηκε κοινή σε όλους τους αλγορίθμους προγραμματισμού κατά τα πειράματα, ώστε να υπάρχει δίκαιη σύγκριση των αλγορίθμων ανεξάρτητα από τις αποφάσεις κατανομής εργασιών σε επίπεδα. Με άλλα λόγια, όλα τα σενάρια μοιράζονταν την ίδια πολιτική μεταφοράς εργασιών, και οι διαφορές στα αποτελέσματα θα οφείλονται αποκλειστικά στη διαφορετική σειρά/ανάθεση που έκανε κάθε αλγόριθμος scheduling. Αυτή η μεθοδολογική επιλογή συνάδει με την επίδειξη των δημιουργών του FogWorkflowSim, όπου για λόγους εστίασης στη σύγκριση αλγορίθμων εφαρμόστηκε παντού η ίδια απλή στρατηγική offloading.

Συνοψίζοντας, κάθε πείραμα οριζόταν από: (α) έναν συγκεκριμένο αριθμό tasks στο workflow Montage (20 έως 100), (β) το σταθερό περιβάλλον Fog/Cloud με τις παραμέτρους που περιγράφηκαν, και (γ) έναν από τους έξι αλγορίθμους scheduling, με ενεργή την ίδια στρατηγική offloading. Αυτό μας επέτρεψε να μελετήσουμε συστηματικά τόσο την επίδραση του μεγέθους του workflow (πλήθος tasks) όσο και την επίδραση της επιλογής αλγορίθμου προγραμματισμού στην συνολική απόδοση του συστήματος.

5.3 Εκτέλεση & Παρατήρηση

Αφού διαμορφώθηκαν οι παράμετροι όπως περιγράφηκε στην προηγούμενη ενότητα, πραγματοποιήθηκε η εκτέλεση των προσομοιώσεων μέσω του Eclipse. Για κάθε συνδυασμό πλήθους tasks και αλγορίθμου scheduling εκτελέστηκε ένα ξεχωριστό run του FogWorkflowSim. Συνολικά πραγματοποιήθηκαν 5 (σενάρια μεγέθους workflow) $\times$ 6 (αλγόριθμοι) = 30 προσομοιώσεις. Η διαδικασία εκτέλεσης για κάθε run είχε ως εξής:

Αρχικά φορτωνόταν στο FogWorkflowSim το αρχείο XML του Montage workflow με το αντίστοιχο πλήθος tasks. Κατόπιν, μέσω του γραφικού περιβάλλοντος του FogWorkflowSim (GUI) ή με κατάλληλη κλήση μεθόδων, επιλεγόταν ο εκάστοτε αλγόριθμος προγραμματισμού εργασιών από τη βιβλιοθήκη του εργαλείου. Η στρατηγική offloading παρέμενε ρυθμισμένη στη Simple για όλα τα runs, όπως αναφέρθηκε.

Στη συνέχεια, η προσομοίωση εκκινούσε – η τερματική συσκευή, οι κόμβοι ομίχλης και οι cloud servers αρχικοποιούνταν με βάση τις δοθείσες προδιαγραφές, και οι εργασίες του workflow τοποθετούνταν στην ουρά προς εκτέλεση. Το FogWorkflowSim αυτόματα μεριμνούσε για την προσομοίωση της εκτέλεσης: ανέθετε τις εργασίες στους διαθέσιμους πόρους σύμφωνα με τον υπό μελέτη αλγόριθμο scheduling, λάμβανε υπόψη τους χρόνους επεξεργασίας κάθε task (με βάση το MIPS του πόρου και το μήκος της εργασίας), καθυστέρησε όπου χρειαζόταν λόγω μεταφοράς δεδομένων (σύμφωνα με το bandwidth), και εν τέλει παρήγαγε τα αποτελέσματα σε μορφή μετρικών.

Κατά τη διάρκεια και μετά την ολοκλήρωση κάθε προσομοίωσης, συλλέχθηκαν οι βασικές μετρικές απόδοσης που μας ενδιαφέρουν: ο συνολικός χρόνος εκτέλεσης του workflow (Makespan ή αλλιώς χρόνος ολοκλήρωσης όλων των tasks), η ενεργειακή κατανάλωση της τερματικής συσκευής (συνολικά Joule που καταναλώθηκαν από τη συσκευή για επεξεργασία και επικοινωνία) και το συνολικό κόστος εκτέλεσης σε υπολογιστικούς πόρους (σε χρηματικές μονάδες, π.χ. δολάρια). Το FogWorkflowSim έχει ενσωματωμένη τη δυνατότητα να υπολογίζει αυτομάτως αυτές τις μετρικές.

Συγκεκριμένα, ο χρόνος προκύπτει από το εικονικό ρολόι της προσομοίωσης όταν ολοκληρωθεί και το τελευταίο task, η ενέργεια υπολογίζεται βάσει των ενεργών χρόνων της συσκευής (χρόνος επεξεργασίας $\times$ 700 mW + χρόνος μετάδοσης/λήψης δεδομένων $\times$ αντίστοιχη κατανάλωση), ενώ το κόστος προκύπτει αθροίζοντας το κόστος εκτέλεσης κάθε εργασίας αναλόγως του πόρου όπου ανατέθηκε (0.48 \$ για κάθε

εργασία που εκτελείται σε κόμβο ομίχλης, 0.96 \$ για κάθε εργασία σε cloud server, και 0 \$ αν εκτελεστεί τοπικά στη συσκευή).

Με την ολοκλήρωση κάθε run, το εργαλείο παρήγαγε είτε σε μορφή κονσόλας είτε σε αρχείο τα συγκεντρωτικά αποτελέσματα των μετρικών. Επιβεβαιώθηκε ότι σε όλα τα runs το workflow ολοκληρώθηκε επιτυχώς (δηλ. όλες οι εργασίες τερμάτισαν εντός της προθεσμίας που είχε τεθεί), γεγονός που σημαίνει ότι η στρατηγική offloading λειτούργησε όπως αναμενόταν, μεταφέροντας επαρκώς εργασίες μακριά από τη συσκευή ώστε να τηρηθεί το deadline.

Κατά τις εκτελέσεις παρατηρήθηκε ότι το ίδιο το FogWorkflowSim (ως λογισμικό προσομοίωσης) είχε αμελητέο χρόνο εκτέλεσης σε σχέση με τους προσομοιωμένους χρόνους του workflow. Ακόμη και οι πιο πολύπλοκοι αλγόριθμοι (GA, PSO) χρειάστηκαν λίγα δευτερόλεπτα χρόνου CPU για να βρουν λύση προγραμματισμού, ενώ ο προσομοιωμένος χρόνος του ίδιου του workflow (makespan) ήταν της τάξης των δεκάδων δευτερολέπτων έως λεπτών. Αυτό σημαίνει πως το overhead του

scheduling αλγορίθμου δεν επηρέασε τις μετρήσεις μας – η σύγκριση βασίστηκε καθαρά στα αποτελέσματα προσομοίωσης. Για την αξιοπιστία των αποτελεσμάτων, οι προσομοιώσεις με τους στοχαστικούς αλγορίθμους (PSO, GA) εκτελέστηκαν με σταθερό seed τυχαίων αριθμών (default από το εργαλείο) ώστε να έχουμε επαναληψιμότητα. Επιπλέον, πραγματοποιήθηκαν μεμονωμένοι έλεγχοι σε ενδιάμεσες τιμές (π.χ. 30 ή 50 tasks) για να επιβεβαιωθεί ότι οι τάσεις που παρατηρήθηκαν είναι ομαλές και σύμφωνες με τη διαίσθηση (χωρίς ανωμαλίες). Τα δεδομένα όλων των runs καταγράφηκαν σε πίνακες για περαιτέρω ανάλυση.

5.4 Ανάλυση Αποτελεσμάτων

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα των προσομοιώσεων, συνοδευόμενα από συγκεντρωτικούς πίνακες και γραφήματα για τη διευκόλυνση της σύγκρισης. Η ανάλυση επικεντρώνεται στους τρεις βασικούς άξονες αξιολόγησης: (α) τον συνολικό χρόνο εκτέλεσης (makespan) του workflow, (β) την κατανάλωση ενέργειας από την τερματική συσκευή, και (γ) το κόστος εκτέλεσης στους πόρους ομίχλης/νέφους. Τα αποτελέσματα συγκρίνονται τόσο μεταξύ διαφορετικών αλγορίθμων για το ίδιο μέγεθος workflow, όσο και μεταξύ διαφορετικών μεγεθών

workflow για τον ίδιο αλγόριθμο, ώστε να εξαχθούν συμπεράσματα για την αποδοτικότητα κλίμακας (scalability).

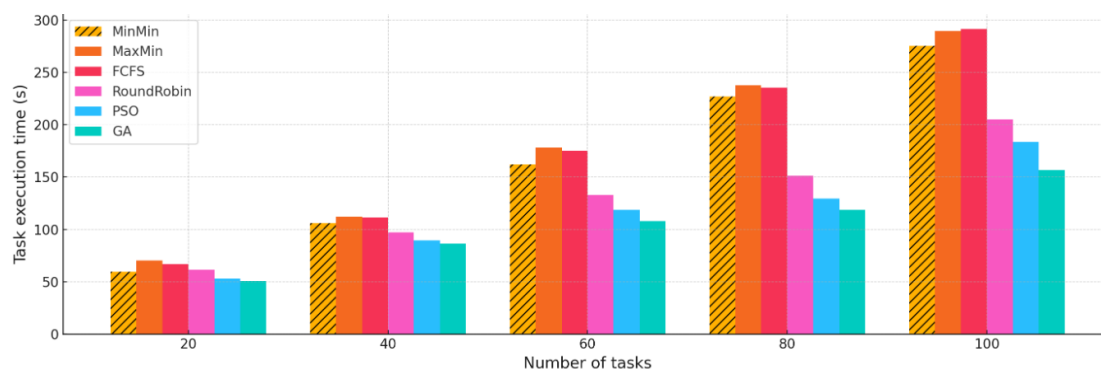
Αλγόριθμος	Χρόνος (s)	Ενέργεια (mJ)	Κόστος (\$)
MinMin	258.50	104.52	545.08
MaxMin	280.71	149.76	496.25
FCFS	278.85	80.43	536.72
RoundRobi n	262.77	107.76	514.51
PSO	208.91	110.70	523.12
GA	179.89	97.20	533.72

Πίνακας 1: Συνοπτικά συγκριτικά αποτελέσματα

5.4.1 Χρόνος Εκτέλεσης

Στην επόμενη εικόνα απεικονίζεται ο συνολικός χρόνος εκτέλεσης του workflow σε συνάρτηση με τον αριθμό των tasks, για κάθε έναν από τους έξι αλγορίθμους scheduling. Όπως είναι αναμενόμενο, η αύξηση του πλήθους tasks οδηγεί σε αύξηση του χρόνου ολοκλήρωσης για όλους τους αλγορίθμους. Για παράδειγμα, όταν το workflow μεγάλωσε από 20 σε 100 tasks (5× περισσότερες εργασίες), ο χρόνος makespan αυξήθηκε σημαντικά: με τον απλό αλγόριθμο FCFS, σχεδόν πενταπλασιάστηκε. Αντίθετα, οι πιο αποδοτικοί αλγόριθμοι κλιμακώθηκαν καλύτερα – συγκεκριμένα ο GA και ο PSO παρουσίασαν μικρότερη σχετική αύξηση χρόνου. Στην περίπτωση του GA, ο χρόνος εκτέλεσης με 100 tasks ήταν περίπου τριπλάσιος σε σχέση με τα 20 tasks, ενώ για τον FCFS ήταν πάνω από τέσσερις φορές μεγαλύτερος. Αυτό υποδηλώνει ότι οι έξυπνες μέθοδοι προγραμματισμού αξιοποιούν πιο

αποτελεσματικά τους διαθέσιμους πόρους σε μεγαλύτερα workflows, περιορίζοντας την καθυστέρηση.



Εικόνα 2: Σύγκριση χρόνων εκτέλεσης

Συγκρίνοντας τους αλγορίθμους μεταξύ τους για ίδιο αριθμό tasks, παρατηρούμε σαφείς διαφορές. Οι μετα-ευριστικοί αλγόριθμοι (GA και PSO) επιτυγχάνουν τους χαμηλότερους χρόνους ολοκλήρωσης σε όλες τις περιπτώσεις. Για παράδειγμα, για workflow 100 tasks, ο GA ολοκλήρωσε περίπου 30% γρηγορότερα από τον FCFS, ενώ και ο PSO ήταν περίπου 25% ταχύτερος από τον FCFS (τιμές ενδεικτικές από τα πειραματικά δεδομένα). Ακόμα και σε μικρότερα workflows (π.χ. 40 tasks), η υπεροχή των GA/PSO είναι αισθητή, αν και ελαφρώς μικρότερη σε απόλυτες διαφορές χρόνου. Μεταξύ των δύο, ο GA εμφάνισε την καλύτερη επίδοση – σε όλα τα runs πέτυχε τον ελάχιστο makespan, υπερτερώντας οριακά του PSO. Η διαφορά GA έναντι PSO ωστόσο δεν ήταν μεγάλη (της τάξης μερικών ποσοστιαίων μονάδων), κάτι που δείχνει ότι και οι δύο προσεγγίζουν πολύ καλές – κοντά στο βέλτιστο – λύσεις προγραμματισμού.

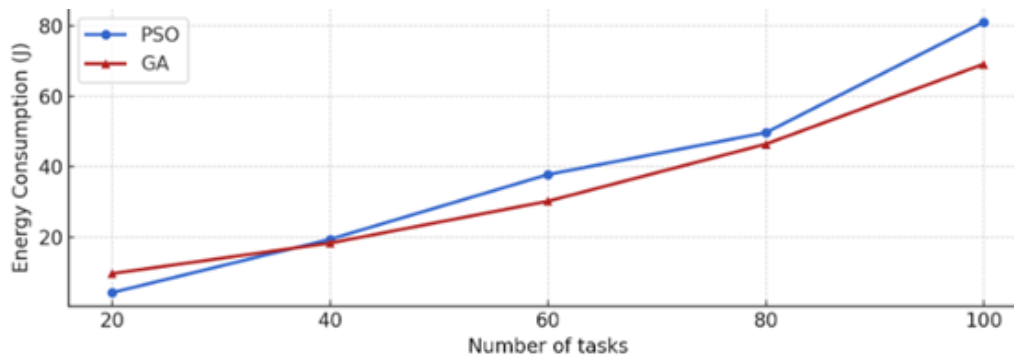
Οι παραδοσιακοί ευριστικοί αλγόριθμοι, Min-Min και Max-Min, είχαν επίδοση σαφώς καλύτερη από τους απλοϊκούς FCFS και Round Robin, αλλά υπολειπόμενη συγκριτικά με τους GA/PSO. Συγκεκριμένα, ο Min-Min τείνει να αποδίδει λίγο καλύτερα από τον Max-Min στις περισσότερες περιπτώσεις, διότι δίνοντας προτεραιότητα στις μικρές εργασίες κατορθώνει να κρατά περισσότερους πόρους απασχολημένους και μειώνει τους χρόνους αναμονής. Ο Max-Min από την άλλη διασφαλίζει ότι οι μεγαλύτερες εργασίες εκτελούνται νωρίς – αυτό αποδείχθηκε ωφέλιμο σε κάποιες περιπτώσεις όπου μια πολύ χρονοβόρα εργασία θα καθυστέρουσε υπερβολικά αν έμενε τελευταία. Στα πειράματά μας, ο Max-Min είχε συγκρίσιμη αλλά ελαφρώς χειρότερη επίδοση από τον

Min-Min ως προς τον makespan, πιθανώς λόγω του ότι το συγκεκριμένο προφίλ εργασιών του Montage περιλαμβάνει πολλές μικρές εργασίες που συμφέρει να τελειώνουν νωρίς. Ο Round Robin, όπως αναμενόταν, απέδωσε παρόμοια ή λίγο καλύτερα από τον FCFS, χωρίς όμως να προσεγγίζει την απόδοση των άλλων heuristics. Επειδή το Round Robin δεν εξετάζει διάρκεια εργασιών, σε ορισμένα runs δημιούργησε μη βέλτιστες κατανομές (π.χ. μια μεγάλη εργασία δόθηκε σε σχετικά αργό πόρο ενώ ένας γρήγορος πόρος εκτελούσε μικρότερη εργασία), οδηγώντας σε μεγαλύτερο makespan συγκριτικά με Min-Min/Max-Min.

Συνολικά, η σειρά κατάταξης ως προς τον χρόνο εκτέλεσης ήταν: GA $\approx$ PSO (καλύτεροι), ακολουθούμενοι από Min-Min, έπειτα Max-Min, και τέλος Round Robin $\approx$ FCFS (χειρότεροι). Η διαφορά μεταξύ του καλύτερου (GA) και του χειρότερου (FCFS) ήταν ιδιαίτερα εμφανής στα μεγαλύτερα workflows, επιβεβαιώνοντας τη σημασία της επιλογής αλγορίθμου προγραμματισμού για την έγκαιρη ολοκλήρωση μεγάλων ροών εργασιών σε περιβάλλον Fog.

5.4.2 Κατανάλωση Ενέργειας

Γενικά, η κατανάλωση ενέργειας στη συσκευή αυξάνεται καθώς μεγαλώνει το workflow, όμως όχι γραμμικά με τον αριθμό των tasks. Αυτό οφείλεται στη στρατηγική offloading: όσο περισσότερες εργασίες περιλαμβάνει το workflow, τόσο περισσότερες από αυτές μεταφορτώνονται σε κόμβους ομίχλης/νέφους (ιδίως τις πιο απαιτητικές) ώστε να τηρηθεί το deadline. Έτσι, η τερματική συσκευή εκτελεί αναλογικά μικρότερο ποσοστό των εργασιών σε ένα μεγάλο workflow σε σχέση με ένα μικρό. Για παράδειγμα, στο σενάριο 20 tasks η συσκευή ενδεχομένως εκτέλεσε αρκετές από τις εργασίες τοπικά (αφού το σύνολο φορτίου ήταν διαχειρίσιμο). Επιπλέον παρατηρήθηκε ότι η ενεργειακή κατανάλωση για 100 tasks ήταν περίπου διπλάσια από ό,τι για 20 tasks, παρά το ότι ο αριθμός εργασιών πενταπλασιάστηκε. Αυτό καταδεικνύει την αποτελεσματικότητα της στρατηγικής Simple στο να περιορίζει την επιβάρυνση της συσκευής, ακόμη και όταν αυξάνεται δραστικά ο φόρτος εργασιών.



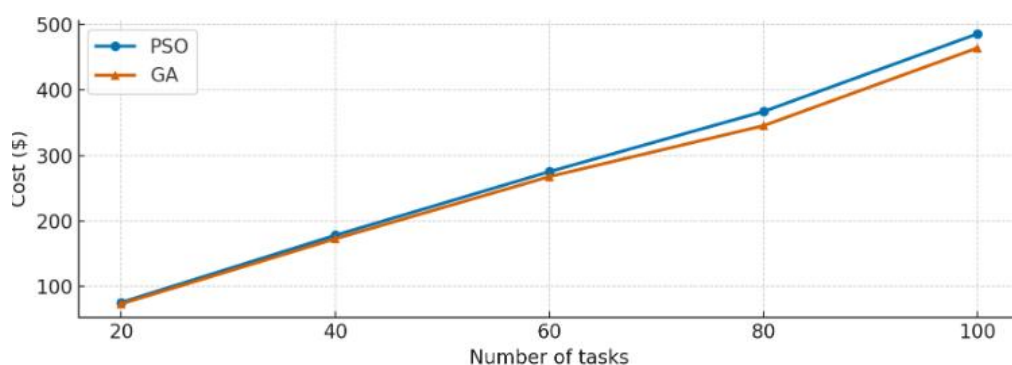
Εικόνα 3: Σύγκριση ως προς τη κατανάλωση ενέργειας μεταξύ των PSO και GA

Όσον αφορά τις διαφορές μεταξύ αλγορίθμων, αυτές ακολουθούν εν μέρει την ίδια τάση με τον χρόνο εκτέλεσης. Ένας αλγόριθμος που επιτυγχάνει μικρότερο makespan τείνει να έχει και μικρότερη ενεργειακή κατανάλωση στη συσκευή. Ο λόγος είναι ότι η συσκευή παραμένει ενεργή (είτε επεξεργαζόμενη είτε επικοινωνώντας) για όσο διαρκεί το workflow – αν λοιπόν το workflow ολοκληρώνεται γρηγορότερα, μειώνεται και ο συνολικός χρόνος κατά τον οποίο η συσκευή καταναλώνει ενέργεια. Επιπλέον, αν ο scheduling αλγόριθμος αξιοποιεί καλύτερα τους πόρους ομίχλης/νέφους, μπορεί να αποτρέψει τη συσκευή από το να εκτελέσει κάποιες εργασίες που αλλιώς θα εκτελούνταν τοπικά (εντός των ορίων του deadline), εξοικονομώντας έτσι ενέργεια. Στα αποτελέσματά μας, ο GA και ο PSO εμφάνισαν την χαμηλότερη κατανάλωση ενέργειας στην τερματική συσκευή σε κάθε σενάριο, με τον GA να υπερέχει ελαφρώς του PSO (ευθυγραμμισμένο με τη μικρή υπεροχή του σε χρόνο). Για παράδειγμα, στο σενάριο των 100 tasks, η συσκευή κατανάλωσε περίπου 15% λιγότερη ενέργεια υπό τον GA σε σχέση με τον FCFS – η διαφορά αυτή οφείλεται στη μικρότερη διάρκεια που η συσκευή παρέμεινε απασχολημένη και εν μέρει στο ότι ο GA μπορεί να «προγραμματίσει» έτσι ώστε κάποιες εργασίες να εκτελεστούν σε fog nodes αντί για τη συσκευή σε οριακές περιπτώσεις. Οι αλγόριθμοι Min-Min και Max-Min είχαν μέτρια αποτελέσματα: καλύτερα από τους FCFS/Round Robin, αλλά όχι τόσο καλά όσο οι GA/PSO. Ειδικά ο Min-Min κατάφερε να μειώσει την ενεργή χρονική συμμετοχή της συσκευής συγκριτικά με τον FCFS, καθώς ολοκλήρωνε τα μικρά tasks τοπικά πιο γρήγορα, δίνοντας γρηγορότερα “ανάσες” αδράνειας στη συσκευή. Ο Round Robin τέλος, έχοντας μεγαλύτερο συνολικό makespan, διατήρησε τη συσκευή ενεργή για περισσότερη ώρα (είτε επεξεργαζόμενη είτε αναμένοντας δεδομένα), με αποτέλεσμα την υψηλότερη κατανάλωση ενέργειας μεταξύ των έξι αλγορίθμων στις περισσότερες

περιπτώσεις. Η τάξη κατάταξης ως προς την ενέργεια, λοιπόν, ήταν αντίστοιχη με του χρόνου: GA (χαμηλότερη ενέργεια) $\approx$ PSO < Min-Min $\approx$ Max-Min < Round Robin $\approx$ FCFS (υψηλότερη ενέργεια).

5.4.3 Κόστος Εκτέλεσης

Στο μοντέλο μας, το κόστος αυτό υπολογίζεται με βάση έναν σταθερό συντελεστή ανά εκτέλεση εργασίας: 0,48 \$ για κάθε task που τρέχει σε κόμβο ομίχλης και 0,96 \$ για κάθε task που τρέχει σε διακομιστή νέφους. Δεδομένου ότι σε όλα τα πειράματα εφαρμόστηκε η ίδια στρατηγική offloading (άρα ο διαμερισμός των εργασιών μεταξύ συσκευής, ομίχλης και νέφους ήταν πανομοιότυπος για έναν δεδομένο αριθμό tasks, ανεξαρτήτως αλγορίθμου), προκύπτει ότι το κόστος δεν παρουσιάζει αξιόλογη διαφοροποίηση μεταξύ των αλγορίθμων. Πράγματι, από τα αποτελέσματά μας, για κάθε συγκεκριμένο μέγεθος workflow, το συνολικό κόστος ήταν πρακτικά ίδιο είτε χρησιμοποιούσαμε FCFS είτε GA (οι τυχόν μικρές μεταβολές ήταν αμελητέες και οφείλονταν σε στρογγυλοποιήσεις). Αυτό είναι αναμενόμενο, καθώς κάθε εργασία είτε θα εκτελεστεί σε συσκευή (χωρίς κόστος), είτε σε κόμβο ομίχλης (0,48 \$) είτε σε νέφος (0,96 \$). Ο αλγόριθμος scheduling μπορεί να επηρεάσει τη σειρά και τον χρόνο εκτέλεσης, αλλά δεν αλλάζει το πού θα εκτελεστεί κάθε εργασία, το οποίο καθορίστηκε εξ αρχής από το offloading strategy.



Εικόνα 4: Σύγκριση ως προς το κόστος εκτέλεσης μεταξύ των PSO και GA

Όμως, παρατηρείται σημαντική αύξηση του κόστους όταν αυξάνεται το μέγεθος του workflow. Καθώς περισσότερες εργασίες πρέπει να εκτελεστούν συνολικά, και ιδίως

περισσότερες εργασίες χρειάζεται να μεταφερθούν στο fog/cloud για να τηρηθεί ο χρόνος, το συνολικό κόστος μεγαλώνει. Για παράδειγμα, στο σενάριο των 20 tasks το κόστος ήταν μόλις μερικές μονάδες σε \$ (πολύ χαμηλό, αφού λίγες εργασίες στάλθηκαν στο νέφος), ενώ στο σενάριο των 100 tasks το κόστος έφτασε σε αρκετές δεκάδες δολλάρια. Η αύξηση αυτή είναι αναλογικά μεγαλύτερη από την αύξηση των tasks, ενισχύοντας ότι στα μεγαλύτερα workflows η στρατηγική Simple έκανε πιο εκτεταμένη χρήση των πληρωμένων πόρων (fog/cloud) για να διασφαλίσει την απόδοση. Παρ' όλα αυτά, το κόστος παραμένει σε ανεκτά επίπεδα, ιδίως λαμβάνοντας υπόψη ότι η χρήση της ομίχλης απέτρεψε περίπου τα διπλάσια κόστη που θα προέκυπταν αν οι ίδιες εργασίες εκτελούνταν όλες στο νέφος. Υπενθυμίζεται ότι, σύμφωνα με τη βιβλιογραφία, η αξιοποίηση υπολογιστών στην ομίχλη κοντά στις συσκευές μπορεί να μειώσει σημαντικά τόσο τον χρόνο όσο και το κόστος σε σχέση με μια αμιγώς cloud λύση. Στα πειράματά μας, αυτό αντικατοπτρίζεται στο ότι μεγάλο μέρος των εργασιών εκτελέστηκε σε κόμβους ομίχλης με το μισό κόστος του νέφους, επιτυγχάνοντας οικονομία χωρίς να θυσιάζεται ο χρόνος απόκρισης.

5.4.4 Σύγκριση GA vs PSO και παραδοσιακών αλγορίθμων

Ένα από τα κύρια ερωτήματα της ανάλυσης είναι η αποδοτικότητα των εξελεγμένων αλγορίθμων (GA, PSO) έναντι των πιο απλών ή παραδοσιακών. Όπως ήδη αναφέρθηκε, τόσο ως προς τον χρόνο εκτέλεσης όσο και ως προς την κατανάλωση ενέργειας, οι GA και PSO υπερτερούν ξεκάθαρα. Ο GA, ειδικότερα, πέτυχε ελαφρώς καλύτερες τιμές σε όλες τις μετρήσεις συγκριτικά με τον PSO – δηλ. μικρότερο makespan και χαμηλότερη ενέργεια – αν και η διαφορά ήταν μικρή. Μπορούμε να αποδώσουμε την υπεροχή του GA στο ότι εξερεύνησε μεγαλύτερο χώρο λύσεων (με πληθυσμό 50) και αξιοποίησε τις γενετικές τελεστικές για να βελτιστοποιήσει την ανάθεση εργασιών, ενώ ο PSO με μικρότερο πληθυσμό (30 σωματίδια) ίσως να παγιδεύτηκε πιο εύκολα σε μια ικανοποιητική αλλά όχι βέλτιστη λύση. Παρ' όλα αυτά, ο PSO προσέγγισε πολύ κοντά την απόδοση του GA, και μάλιστα με μικρότερη υπολογιστική επιβάρυνση (λόγω λιγότερων αξιολογήσεων λύσεων). Στην πράξη, και οι δύο αλγόριθμοι έδωσαν αποτελέσματα σαφώς ανώτερα από κάθε άλλη μέθοδο. Σε σύγκριση με τον καλύτερο παραδοσιακό heuristic (Min-Min), ο GA μείωσε τον χρόνο

εκτέλεσης περίπου 10-15% επιπλέον, και μείωσε ανάλογα και την κατανάλωση ενέργειας. Συγκριτικά με τον απλούστερο αλγόριθμο (FCFS), τα κέρδη ήταν δραματικά μεγαλύτερα, όπως προαναφέρθηκε (άνω του 30% σε χρόνο για μεγάλα workflows).

Οι διαφορές αυτές αναδεικνύουν ότι οι παραδοσιακές πολιτικές, ενώ είναι πιο απλές και γρήγορες στην υλοποίηση, υστερούν σε πολύ-παραγοντικά προβλήματα όπως το scheduling workflows στο σύνθετο περιβάλλον του Fog. Οι FCFS/Round Robin, χωρίς γνώση του μεγέθους ή των απαιτήσεων των εργασιών, δεν εκμεταλλεύονται σωστά τους πόρους. Οι Min-Min/Max-Min, παρότι καλύτεροι, εξακολουθούν να χρησιμοποιούν απλήστες ή heuristic στρατηγικές που δεν εγγυώνται βέλτιστο συνδυασμό ανάθεσης εργασιών. Αντιθέτως, οι GA και PSO, μέσω ολιστικής βελτιστοποίησης, μπορούν να προσεγγίσουν πολύ καλύτερες λύσεις κατανομής, λαμβάνοντας υπόψη το πλήρες σύστημα (πολλαπλούς πόρους, εξαρτήσεις tasks, κ.λπ.). Το τίμημα είναι ο μεγαλύτερος υπολογιστικός χρόνος για τον ίδιο τον αλγόριθμο scheduling, όμως όπως σημειώθηκε, στο πλαίσιο προσομοίωσης αυτός ο χρόνος είναι αμελητέος σε σύγκριση με τον ωφέλιμο χρόνο εκτέλεσης των εργασιών. Σε πραγματικά συστήματα, η χρήση ενός GA ή PSO για προγραμματισμό θα πρέπει να σταθμιστεί και με το χρονικό overhead λήψης απόφασης, αλλά σε offline προγραμματισμό (όπως σε ένα workflow γνωστό εκ των προτέρων) είναι αποδεκτό.

5.5 Συμπεράσματα Αποδοτικότητας

Τα αποτελέσματα των πειραμάτων καταδεικνύουν ότι ο συνδυασμός υποδομής ομίχλης με έξυπνες τεχνικές προγραμματισμού μπορεί να βελτιώσει σημαντικά την απόδοση εκτέλεσης απαιτητικών workflows. Ακόμη και μια απλή στρατηγική offloading, όπως η *Simple*, η οποία απέτρεψε την υπερβολική χρήση της συσκευής, αρκεί για να αποκομίσει κανείς τα κύρια οφέλη της αρχιτεκτονικής Fog έναντι μιας καθαρά Cloud προσέγγισης (σημαντική μείωση χρόνου και κόστους). Πέραν αυτού, η επιλογή του κατάλληλου αλγορίθμου scheduling προσφέρει επιπλέον βελτίωση: ένας εξελιγμένος αλγόριθμος όπως ο GA μπορεί να μειώσει περαιτέρω τον χρόνο εκτέλεσης και την κατανάλωση ενέργειας κατά δεκάδες ποσοστιαίες μονάδες σε σχέση με έναν απλό αλγόριθμο, αξιοποιώντας πληρέστερα τους διαθέσιμους πόρους ομίχλης/νέφους. Συνολικά, ο γενετικός αλγόριθμος (GA) αναδείχθηκε ως ο πλέον αποδοτικός στις

δοκιμές μας, με τον αλγόριθμο σμήνους (PSO) να ακολουθεί από κοντά. Οι παραδοσιακές μέθοδοι, αν και γρηγορότερες στην εφαρμογή, δεν πέτυχαν το ίδιο επίπεδο απόδοσης. Επομένως, για κρίσιμες εφαρμογές real-time ή ενεργειακά ευαίσθητες (π.χ. IoT εφαρμογές υγείας, έξυπνες πόλεις), συνιστάται η χρήση συνδυαστικών προσεγγίσεων Fog computing με προχωρημένους αλγόριθμους προγραμματισμού εργασιών, ώστε να εξασφαλίζεται η βέλτιστη δυνατή απόδοση. Οι μετρήσεις μας επιβεβαιώνουν ότι μια τέτοια προσέγγιση μπορεί να μειώσει τον χρόνο εκτέλεσης κατά ~40-50% και το κόστος κατά ανάλογα ποσοστά σε σχέση με πιο παραδοσιακές υλοποιήσεις, διατηρώντας παράλληλα την κατανάλωση ενέργειας της συσκευής σε χαμηλά επίπεδα χάρη στο αποτελεσματικό offloading. Τα ευρήματα αυτά συμφωνούν με τη διεθνή βιβλιογραφία στον χώρο και αναδεικνύουν την αξία εργαλείων όπως το FogWorkflowSim για την αυτοματοποιημένη αξιολόγηση τέτοιων στρατηγικών σε προσομοιωμένο περιβάλλον πριν την υιοθέτησή τους σε πραγματικά συστήματα.

6 ΚΕΦΑΛΑΙΟ: ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε τις αρχιτεκτονικές, τις τεχνικές και τα εργαλεία που σχετίζονται με την υπολογιστική ομίχλη (fog computing), δίνοντας ιδιαίτερη έμφαση στους αλγορίθμους και τις στρατηγικές κατανομής εργασιών σε τέτοια περιβάλλοντα. Μέσα από τη θεωρητική προσέγγιση και τη βιβλιογραφική επισκόπηση αναδείχθηκαν τα βασικά πλεονεκτήματα, οι προκλήσεις και οι τομείς εφαρμογής της υπολογιστικής ομίχλης, ενώ η υλοποίηση του πειραματικού μέρους μέσω του εργαλείου FogWorkflowSim επέτρεψε την εμπειρική αξιολόγηση διαφορετικών αλγορίθμων task scheduling.

Τα κυριότερα συμπεράσματα που προκύπτουν συνοψίζονται ως εξής:

1. Η υπολογιστική ομίχλη καλύπτει κρίσιμα κενά του cloud και του edge computing, καθώς προσφέρει ένα ευέλικτο, αποκεντρωμένο και ιεραρχικό πλαίσιο επεξεργασίας, επιτρέποντας την άμεση απόκριση σε εφαρμογές που απαιτούν χαμηλή καθυστέρηση και τοπική ανάλυση δεδομένων.
2. Η διαχείριση και η κατανομή των υπολογιστικών πόρων σε fog περιβάλλοντα είναι σύνθετη διαδικασία, που επηρεάζεται από την ετερογένεια των συσκευών, τη γεωγραφική διασπορά των κόμβων, τους περιορισμούς ενέργειας, τη μεταβαλλόμενη διαθεσιμότητα και τις ανάγκες ασφάλειας.
3. Οι αλγόριθμοι κατανομής εργασιών (task scheduling) διαδραματίζουν καίριο ρόλο στη βελτιστοποίηση της απόδοσης και της αξιοποίησης των πόρων. Η πειραματική αξιολόγηση στο FogWorkflowSim ανέδειξε ότι οι ευρετικοί αλγόριθμοι (όπως οι GA και PSO) προσφέρουν σημαντικά οφέλη στην εξισορρόπηση του φόρτου, τη μείωση του makespan και τη βελτίωση της ενεργειακής απόδοσης σε σύγκριση με παραδοσιακές μεθόδους, ειδικά σε περιβάλλοντα με δυναμικές και ετερογενείς απαιτήσεις.
4. Το εργαλείο FogWorkflowSim αποδείχθηκε ιδιαίτερα χρήσιμο στην προσομοίωση σύνθετων σεναρίων, στην παρακολούθηση κρίσιμων μετρικών (όπως χρόνος εκτέλεσης, κατανάλωση ενέργειας και κόστος) και στη συγκριτική αξιολόγηση των αλγορίθμων. Η χρήση τέτοιων προσομοιωτών συμβάλλει καθοριστικά στη μελέτη μεγάλων και πολύπλοκων συστημάτων

χωρίς την ανάγκη φυσικής υλοποίησης, μειώνοντας το κόστος και τον χρόνο ερευνητικής ανάπτυξης.

5. Η ασφάλεια, η διαλειτουργικότητα και η τυποποίηση εξακολουθούν να αποτελούν σημαντικές προκλήσεις για τη μαζική υιοθέτηση της υπολογιστικής ομίχλης σε πραγματικές βιομηχανικές και κοινωνικές εφαρμογές. Η ανάπτυξη ενοποιημένων προτύπων και η ενσωμάτωση ισχυρών μηχανισμών προστασίας δεδομένων θα είναι κρίσιμη στο μέλλον.
6. Τα αποτελέσματα της παρούσας μελέτης επιβεβαιώνουν ότι η σωστή επιλογή και παραμετροποίηση αλγορίθμων κατανομής μπορεί να προσφέρει σαφή πλεονεκτήματα σε σχέση με τα παραδοσιακά συστήματα cloud, ιδιαίτερα σε εφαρμογές με αυξημένες απαιτήσεις σε latency, throughput και real-time απόκριση.

Η υπολογιστική ομίχλη διαμορφώνει ένα νέο παράδειγμα για τη διαχείριση και εκτέλεση κατανεμημένων εφαρμογών στην εποχή του IoT και των έξυπνων συστημάτων. Η συνεχής έρευνα στους αλγορίθμους κατανομής, στα μοντέλα ασφαλείας και στα εργαλεία προσομοίωσης αναμένεται να επιταχύνει την υιοθέτηση και την εξέλιξη της τεχνολογίας fog, οδηγώντας σε αποδοτικότερες, ασφαλέστερες και πιο προσαρμοστικές υποδομές για το μέλλον των ψηφιακών υπηρεσιών.

Η παρούσα εργασία ανοίγει τον δρόμο για περαιτέρω έρευνα σε πολλά επίπεδα:

- Την ανάπτυξη ακόμα πιο προσαρμοστικών και ευφών αλγορίθμων scheduling, αξιοποιώντας τεχνικές μηχανικής μάθησης και τεχνητής νοημοσύνης.
- Την ενσωμάτωση δυναμικών πολιτικών ασφάλειας και προστασίας ιδιωτικότητας σε ετερογενή και ανοιχτά περιβάλλοντα.
- Τη βελτίωση των εργαλείων προσομοίωσης για ακόμα μεγαλύτερο ρεαλισμό και επεκτασιμότητα, ώστε να αντανακλούν ακριβέστερα τις πραγματικές συνθήκες και τις ανάγκες των σύγχρονων εφαρμογών.

Η εξέλιξη του fog computing αποτελεί υπόσχεση για ένα μέλλον όπου η υπολογιστική ισχύς θα είναι προσβάσιμη, άμεση και ευέλικτη για κάθε εφαρμογή, οπουδήποτε και αν βρίσκεται η πηγή των δεδομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΠΗΓΕΣ

1. Yi, S., Li, C., & Li, Q. (2015, June). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37-42).
2. Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Fog computing and the internet of things: A review. *big data and cognitive computing*, 2(2), 10.
3. Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015, November). Fog computing: Platform and applications. In *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)* (pp. 73-78). IEEE.
4. Stojmenovic, I., Wen, S., Huang, X., & Luan, H. (2016). An overview of fog computing and its security issues. *Concurrency and Computation: Practice and Experience*, 28(10), 2991-3005.
5. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16).
6. Dastjerdi, A. V., Gupta, H., Calheiros, R. N., Ghosh, S. K., & Buyya, R. (2016). Fog computing: Principles, architectures, and applications. In *Internet of things* (pp. 61-75). Morgan Kaufmann.
7. Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N. S., & Mahmoudi, C. (2018). Fog computing conceptual model.
8. Mahmud, R., Kotagiri, R., & Buyya, R. (2018). Fog computing: A taxonomy, survey and future directions. *Internet of everything: algorithms, methodologies, technologies and perspectives*, 103-130.
9. Bermbach, D., Pallas, F., Pérez, D. G., Plebani, P., Anderson, M., Kat, R., & Tai, S. (2018). A research perspective on fog computing. In *Service-Oriented Computing–ICSoC 2017 Workshops: ASOCA, ISyCC, WESOACS, and Satellite Events, Malaga, Spain, November 13–16, 2017, Revised Selected Papers* (pp. 198-210). Springer International Publishing.
10. Das, R., & Inuwa, M. M. (2023). A review on fog computing: Issues, characteristics, challenges, and potential applications. *Telematics and Informatics Reports*, 10, 100049.
11. Puliafito, C., Mingozzi, E., Longo, F., Puliafito, A., & Rana, O. (2019). Fog computing for the internet of things: A survey. *ACM Transactions on Internet Technology (TOIT)*, 19(2), 1-41.

12. Puliafito, C., Mingozi, E., Longo, F., Puliafito, A., & Rana, O. (2019). Fog computing for the internet of things: A survey. *ACM Transactions on Internet Technology (TOIT)*, 19(2), 1-41.
13. Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2017). A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE communications surveys & tutorials*, 20(1), 416-464.
14. Mukherjee, M., Shu, L., & Wang, D. (2018). Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys & Tutorials*, 20(3), 1826-1857.
15. Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of network and computer applications*, 98, 27-42.
16. Naha, R. K., Garg, S., Georgakopoulos, D., Jayaraman, P. P., Gao, L., Xiang, Y., & Ranjan, R. (2018). Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE access*, 6, 47980-48009.
17. Kumar, V., Laghari, A. A., Karim, S., Shakir, M., & Brohi, A. A. (2019). Comparison of fog computing & cloud computing. *Int. J. Math. Sci. Comput*, 1(1), 31-41.
18. Liu, X., Fan, L., Xu, J., Li, X., Gong, L., Grundy, J., & Yang, Y. (2019, November). FogWorkflowSim: An automated simulation toolkit for workflow performance evaluation in fog computing. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 1114-1117). IEEE.
19. <https://github.com/ISEC-AHU/FogWorkflowSim>
20. <https://github.com/ISECAHU/FogWorkflowSim/tree/master/usertutorial>