



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Προηγμένη Εφαρμογή Εκχώρησης Πόρων σε Υπολογιστικά Νέφη Μικρής Κλίμακας με τη Χρήση Κινητών Συσκευών Integrated Application for Resource Allocation in a Mobile Cloudlet Environment
Όνοματεπώνυμο Φοιτητή	Δημήτριος Γκότσης
Πατρώνυμο	Ιωάννης
Αριθμός Μητρώου	ΜΠΣΠ 12013
Επιβλέπων	Χρήστος Δουληγέρης, Καθηγητής

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης
Καθηγητής

Δ. Βέργαδος
Επικ. Καθηγητής

Παν. Κοτζανικολάου
Επικ. Καθηγητής

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά τον Υποψήφιο Διδάκτορα κ. Δημήτρη Καλλέργη για την καθοδήγηση και την αμέριστη βοήθειά του κατά την διάρκεια εκπόνηση της μεταπτυχιακής διατριβής μου. Ακόμη, θα ήθελα να ευχαριστήσω ιδιαίτερα όλους τους καθηγητές του Μεταπτυχιακού Προγράμματος Σπουδών «Προηγμένα Συστήματα Πληροφορικής», για τις πολύτιμες γνώσεις που μου προσέφεραν καθ' όλη την διάρκεια των σπουδών.

Περίληψη

Στις μέρες μας αποτελεί κοινή πρακτική η ανάθεση εκτέλεσης απαιτητικών εφαρμογών από τις κινητές συσκευές σε υποδομή υπολογιστικών νεφών. Σκοπός είναι η αποδοτικότερη και ταχύτερη εκτέλεση τους σε διακομιστές υψηλής υπολογιστικής ισχύος. Σε αυτή την περίπτωση κάνουμε λόγο για μια αρχιτεκτονική, η οποία ονομάζεται Mobile Cloud Computing (MCC). Σε αυτή την υποδομή, η κινητή συσκευή επικοινωνεί μέσω 3G/LTE συνδέσεων με ένα απομακρυσμένο Cloud. Μια διαφορετική προσέγγιση προέκυψε αργότερα, ώστε οι κινητές συσκευές να μην επικοινωνούν απευθείας με το Cloud. Αυτή η προσέγγιση οδήγησε στη δημιουργία μιας Cloudlet-Based αρχιτεκτονικής υπολογιστικού νέφους. Σε μια τέτοια υποδομή οι κινητές συσκευές επικοινωνούν μέσω Wi-Fi LAN με το κοντινότερο Cloudlet στο οποίο και στέλνουν tasks για εκτέλεση. Το Cloudlet διαθέτει ισχυρότερους επεξεργαστές σε σύγκριση με ένα smartphone ή ένα tablet, όμως, ανάλογα και με το φόρτο εργασίας οι πόροι επεξεργαστικής ισχύος του μπορούν να εξαντληθούν. Σε αυτή την περίπτωση το Cloudlet αναζητά πόρους από το Cloud με το οποίο είναι συνδεδεμένο. Η παρούσα μεταπτυχιακή διατριβή εστιάζει σε μια εναλλακτική προσέγγιση της Cloudlet-Based υποδομής υπολογιστικού νέφους, στην οποία το Cloudlet να αναζητά επιπλέον υπολογιστικούς πόρους από κινητές συσκευές, οι οποίες είναι συνδεδεμένες στο ίδιο τοπικό ασύρματο δίκτυο με αυτό. Στο πλαίσιο εκπόνησης της εργασίας παρουσιάζεται μία μεθοδολογία για τη δέσμευση και διαχείριση μέρους της CPU μίας κινητής συσκευής μέσω εφαρμογής, προσομοιώνεται μια Cloudlet-Based υποδομή υπολογιστικού νέφους και εκτελούνται πειραματικά σενάρια ανάθεσης εργασιών από το Cloudlet σε έξυπνες κινητές συσκευές.

Λέξεις Κλειδιά: Mobile Cloud Computing, Cloudlet, CloudSim, Resource Allocation, Mobile Devices, Android, Location Based Services.

Abstract

Nowadays, it is a common practice for mobile devices to offload computationally heavy tasks off to a Cloud. In this way, the processing of intensive mobile device jobs takes place in cloud systems, which have rich computational resources. We call such an environment Mobile Cloud Computing (MCC) architecture. In MCC, mobile devices communicate with remote Clouds using 3G/LTE connections. An alternative MCC infrastructure was revealed later in which mobile devices do not communicate directly with Cloud. It had as a result the implementation of a new MCC model, which is based on Cloudlets. In Cloudlet-Based MCC system mobile devices communicate with nearby Cloudlets using Wi-Fi connections and send to them tasks in order to do the required processing and return the final results back. Cloudlets have greater computational resources than smartphones and tablets, but depending on the workload its resources can be exhausted. In this case, Cloudlet asks for extra resources from Cloud. All Cloudlets are connected to a remote Cloud. This dissertation is focused on an alternative approach of the MCC Cloudlet-Based model, where Cloudlet asks for extra computational resources from mobile devices that are connected to the same Wi-Fi LAN. It presents an experimental method, where smart mobile devices offer computational resources to Cloudlets. In the producing context of the dissertation in question, a methodology for resource allocation in android systems is analyzed. Furthermore, a Cloudlet-Based Cloud infrastructure is simulated and experimental scenarios for delegating tasks to mobile devices are executed.

Key Words: Mobile Cloud Computing, Cloudlet, CloudSim, Resource Allocation, Mobile Devices, Android, Location-Based Services.

Πίνακας Περιεχομένων

1. Εισαγωγή	13
1.1 Πρόλογος	13
1.2 Σκοπός και Αντικείμενο Μελέτης.....	15
1.3 Δομή της εργασίας	16
2. Θεωρητική Προσέγγιση	18
2.1 Cloud Computing	18
2.2 Mobile Cloud Computing	20
2.3 Cloudlet-Based MCC Αρχιτεκτονική.....	21
2.4 Mobile Device Clouds	23
2.5 Resource Allocation	24
2.6 Location-Based Services	25
2.7 Η Φύση του Προβλήματος	25
3. Προσέγγιση Λύσης	28
3.1 Μέθοδος.....	28
3.2 Μεθοδολογία.....	29
3.3 Περιγραφή των εμπλεκόμενων ρόλων	32
3.4 Περιγραφή Περιπτώσεων Χρήσης.....	32
3.4.1 Διαγράμματα Περιπτώσεων Χρήσης.....	32
3.4.2 Διαγράμματα Δραστηριοτήτων	34
3.4.3 Περίπτωση χρήσης: CPU Usage	36
3.4.4 Περίπτωση χρήσης: Παραχώρηση πόρων στο Cloudlet.....	36
3.4.5 Περίπτωση χρήσης: Ανάθεση εργασιών από το Cloudlet	36
3.4.6 Περίπτωση χρήσης: Αναζήτηση Μαθημάτων	37
3.4.7 Περίπτωση χρήσης: Αναζήτηση Αίθουσας Διδασκαλίας.....	37
3.4.8 Περίπτωση χρήσης: Αναζήτηση Υλικού Μαθήματος.....	38
3.4.9 Περίπτωση χρήσης: Βοήθεια	38
3.5 Λειτουργικές απαιτήσεις	39
3.5.1 CPU Usage.....	39
3.5.2 Παραχώρηση πόρων στο Cloudlet	39
3.5.3 Ανάθεση task από το Cloudlet	40
3.5.4 Βοήθεια.....	41
3.5.5 Αναζήτηση Μαθημάτων	41

3.5.6 Αναζήτηση Αίθουσας Διδασκαλίας	42
3.5.7 Αναζήτηση Υλικού Μαθημάτων.....	43
3.6 Μη Λειτουργικές απαιτήσεις	43
3.6.1 Περιβάλλον λειτουργίας	43
3.6.2 Περιορισμοί στη σχεδίαση και την υλοποίηση	44
3.6.3 Προϋποθέσεις - Απαιτήσεις	45
3.7 Απαιτήσεις για τις εξωτερικές διεπαφές.....	45
3.7.1 Διεπαφή χρήστη	45
3.7.2 Διεπαφές υλικού	47
3.7.3 Διεπαφές Λογισμικού	47
3.7.4 Διεπαφές επικοινωνιών.....	47
3.8 Τεχνολογίες που Χρησιμοποιήθηκαν	47
3.8.1 Android OS	47
3.8.2 Java	48
3.8.3 CloudSim	48
3.8.4 Eclipse IDE	49
3.8.5 Linpack Benchmark.....	49
3.9 Ανάλυση Κώδικα	50
4. Αποτελέσματα	59
4.1 Παρουσίαση Εφαρμογής	59
4.2 Αξιολόγηση Συστήματος.....	71
4.3 Περιορισμοί Συστήματος.....	72
5. Επίλογος	74
5.1 Σύνοψη	74
5.2 Συμπεράσματα	75
5.3 Μελλοντική Εργασία.....	75
Βιβλιογραφία	78

Λίστα Εικόνων

Εικόνα 2.1: Mobile Cloud Computing Αρχιτεκτονική [31].....	21
Εικόνα 2.2: Mobile Cloud Computing & Mobile Cloudlet Cloud Computing αρχιτεκτονικές [3].....	22
Εικόνα 2.3: Mobile Device Clouds [13]	24
Εικόνα 3.1: MCC Cloudlet-Based Μοντέλο [11].....	31
Εικόνα 4.1: Εμβόλιμη Οθόνη.....	59
Εικόνα 4.2: Κύρια Οθόνη.....	60
Εικόνα 4.3: Οθόνη Βοήθειας.....	60
Εικόνα 4.4: Χρήση της CPU	61
Εικόνα 4.5: Alert Λάθος Επιλογή.....	61
Εικόνα 4.6: Δέσμευση Πόρων Πείραμα 1.....	62
Εικόνα 4.7: Ειδοποίηση Δέσμευσης Πόρων	62
Εικόνα 4.8: Αποτυχία Εκτέλεσης task.....	63
Εικόνα 4.9: Cloudlets Αποτελέσματα Εκτέλεσης Πειράματος 1	64
Εικόνα 4.10: Δέσμευση Πόρων Πείραμα 2.....	64
Εικόνα 4.11: Επιτυχής Εκτέλεση task	65
Εικόνα 4.12: Cloudlets Αποτελέσματα Εκτέλεσης Πειράματος 2	65
Εικόνα 4.13: Δέσμευση Πόρων Πείραμα 3.....	66
Εικόνα 4.14: Δεν Ανατέθηκε task.....	66
Εικόνα 4.15: Cloudlets Αποτελέσματα Εκτέλεσης Πειράματος 3	67
Εικόνα 4.16: Δεν Υπάρχει Σύνδεση	67
Εικόνα 4.17: Δεν βρέθηκαν Μαθήματα.....	68
Εικόνα 4.18: Λίστα Μαθημάτων.....	68
Εικόνα 4.19: Επιλογές Μαθήματος	69
Εικόνα 4.20: Προβολή Αίθουσας	69
Εικόνα 4.21: Εισαγωγή Κωδικού	70
Εικόνα 4.22: Λάθος Κωδικός.....	70
Εικόνα 4.23: Υλικό Μαθήματος.....	71

Λίστα Διαγραμμάτων

Διάγραμμα 3.1: Κύριο μενού επιλογών	33
Διάγραμμα 3.2: Αναζήτηση πληροφοριών για τα μαθήματα.....	33
Διάγραμμα 3.3: Δέσμευση υπολογιστικών πόρων	34
Διάγραμμα 3.4: Ανάθεση task από το Cloudlet.....	34
Διάγραμμα 3.5: Αναζήτηση Βοήθειας.....	34
Διάγραμμα 3.6: Δέσμευση πόρων συσκευής και ενημέρωση του Cloudlet	35
Διάγραμμα 3.7: Αναζήτηση Πόρων σε συσκευές από το Cloudlet και ανάθεση tasks	35

Λίστα Πινάκων

Κώδικας 3.1: Διαχείριση αδειών εφαρμογή για πόρους δικτύου	50
Κώδικας 3.2: Έλεγχος σύνδεσης στο Wi-Fi δίκτυο	51
Κώδικας 3.3: Εκτέλεση της εντολής Top του Linux	53
Κώδικας 3.4: Υπολογισμός χρήσης της CPU	54
Κώδικας 3.5: Δέσμευση υπολογιστικών πόρων	55
Κώδικας 3.6: Κώδικας από την πλευρά του server	56
Κώδικας 3.7: Κώδικας από την πλευρά του client	57
Κώδικας 3.8: Δημιουργία cloudlet στο CloudSim	58
Κώδικας 3.9: Δημιουργία Virtual Machine στο CloudSim	58

Συνομογραφίες

ADT: Android Development Tools

AOSP: Android Open Source Project

AP: Access Point

API: Application Programming Interface

BSSID: Basic Service Set Identification

FLOPS: Floating Operations per Second

GB: Giga Byte

GNU: General Public License

CPU: Central Processing Unit

GPS: Global Positioning System

GUI: Graphical User Interface

HTML: Hyper Text Markup Language

IaaS: Infrastructure as a Service

IDE: Integrated Development Environment

IP: Internet Protocol

JVM: Java Virtual Machine

JSE: Java Standard Edition

LAN: Local Area Network

LBS: Location Based Services

LTE: Long Term Evolution

MAC: Media Access Control

MCC: Mobile Cloud Computing

MDC: Mobile Device Cloud

MFLOPS: Millions Floating Operations per Second

NDK: Native Development Kit

OS: Operating System

PaaS: Platform as a Service

RAM: Random Access Memory

SaaS: Software as a Service

SDK: Software Development Kit

TCP: Transmission Control Protocol

UI: User Interface

UML: Unified Modeling Language

VM: Virtual Machine

Wi-Fi: Wireless Fidelity

WLAN: Wireless Local Area Network

1. Εισαγωγή

1.1 Πρόλογος

Στις μέρες μας οι έξυπνες κινητές συσκευές, όπως τα smartphones και τα tablets, αποτελούν τις πλέον περιζήτητες και ευρέως χρησιμοποιούμενες φορητές ηλεκτρονικές συσκευές. Το υλικό και το λογισμικό τους διαρκώς εξελίσσεται και οι βιομηχανίες της πληροφορικής και της κατασκευής ηλεκτρονικών μικροσυσκευών έχουν δώσει σε αυτές ιδιαίτερη βαρύτητα. Πλέον, το κινητό τηλέφωνο δεν είναι σχεδιασμένο μόνο για να κάνει κλήσεις, ή να λαμβάνει και να στέλνει μηνύματα. Μια τέτοια συσκευή αποτελεί πραγματικά έναν εξελιγμένο υπολογιστή χειρός με λειτουργικό σύστημα και τη δυνατότητα να εκτελεί περίπλοκες και απαιτητικές εφαρμογές. Οι έξυπνες φορητές συσκευές διατηρούν τα τελευταία χρόνια έναν κυρίαρχο ρόλο στην καθημερινότητα του σύγχρονου ανθρώπου. Οποιοσδήποτε πλέον, με τη χρήση της κινητής του συσκευής μπορεί, να συνδεθεί στο Διαδίκτυο (Internet) και να χρησιμοποιήσει τις υπηρεσίες του, να δημιουργήσει και να επεξεργαστεί έγγραφα, να πραγματοποιήσει βίντεο κλήσεις, να χρησιμοποιήσει υπηρεσίες παροχής πληροφοριών ανάλογα με τη θέση του (Location-Based Services – LBS), να αποθηκεύσει στην συσκευή του μεγάλο όγκο δεδομένων, όπως φωτογραφίες, βίντεο, τραγούδια κ.ά. Τα παραπάνω είναι ενδεικτικά παραδείγματα της χρήσης και των δυνατοτήτων των σημερινών smartphones.

Σε όλα τα πλεονεκτήματα των έξυπνων φορητών συσκευών, που περιγράφηκαν στην προηγούμενη παράγραφο, έρχεται να προστεθεί και η δυνατότητα τους να συνδέονται και να χρησιμοποιούν πόρους από υπολογιστικά νέφη (Clouds). Μια κινητή συσκευή μπορεί πολύ εύκολα να χρησιμοποιήσει μια εφαρμογή ή υπηρεσία, η οποία παρέχεται από μια υποδομή υπολογιστικού νέφους σε επίπεδο Software as a Service (SaaS). Σε αυτή την περίπτωση, η client εφαρμογή, που τρέχει στην κινητή συσκευή, αποτελεί τη διεπαφή (interface) για την επικοινωνία της συσκευής με το Σύννεφο και τη χρήση της υπηρεσίας από τον ενδιαφερόμενο. Επίσης, οι κινητές συσκευές έχουν τη δυνατότητα να αναθέτουν την εκτέλεση των ιδιαίτερα απαιτητικών – εξειδικευμένων (native) εφαρμογών τους στο Σύννεφο. Με αυτόν τον τρόπο η εκτέλεση των εφαρμογών είναι ταχύτερη και οι φορητές συσκευές εξοικονομούν ενέργεια και πόρους υπολογιστικής ισχύος.

Όπως γίνεται αντιληπτό από τα παραπάνω, η έννοια του Σύννεφου σχετίζεται άμεσα με τη διαχείριση και τον διαμοιρασμό πόρων μέσω διαδικτύου. Η επικοινωνία των mobile συσκευών και η αναζήτηση πόρων σε αυτό οδήγησε σε αρχιτεκτονικές κινητών υπολογιστικών νεφών (Mobile Cloud Computing – MCC). Για την έννοια του MCC έχουν πραγματοποιηθεί διάφορες μελέτες και προσεγγίσεις, μια εκ των οποίων είναι η Cloudlet-Based αρχιτεκτονική υπολογιστικού νέφους. Η Cloudlet-Based αρχιτεκτονική υπολογιστικού νέφους είναι μια υποδομή MCC, της οποίας η λειτουργία στηρίζεται στην χρήση των Cloudlets. Τα Cloudlets είναι μικρά υπολογιστικά συστήματα (Data Centers), τα οποία αποτελούν ένα ενδιάμεσο επίπεδο στην επικοινωνία των φορητών συσκευών με το Σύννεφο. Οι φορητές συσκευές σε ένα Cloudlet-Based μοντέλο MCC επικοινωνούν σε πρώτο επίπεδο με ένα Cloudlet, το οποίο διαχειρίζεται και εξυπηρετεί τα αιτήματά τους (requests). Σε δεύτερο επίπεδο το Cloudlet επικοινωνεί με το Σύννεφο, με σκοπό την προώθηση των requests των κινητών συσκευών, ή την αναζήτηση υπολογιστικών πόρων ή ακόμη και την πρόσβαση σε βάσεις δεδομένων ευρείας κλίμακας. Στην Cloudlet-Based αρχιτεκτονική MCC κυρίαρχο ρόλο κατέχουν αυτά τα μικρά Data Centers (Cloudlets), τα οποία μιμούνται την βασική υποδομή του Σύννεφου. Οι φορητές συσκευές συνδέονται μέσω wi-fi ή Bluetooth με ένα Cloudlet.

Στη Cloudlet-Based αρχιτεκτονική υπολογιστικού νέφους το Cloudlet δεσμεύει, διαχειρίζεται και μοιράζει τους πόρους του σε κινητές συσκευές. Σε περίπτωση όπου οι πόροι του εξαντληθούν, αναζητά με την σειρά του επιπλέον πόρους από το Σύννεφο με το οποίο επικοινωνεί.

Όμως, όπως αναφέρθηκε στην αρχή αυτής της ενότητας οι έξυπνες κινητές συσκευές διαρκώς εξελίσσονται και σήμερα διαθέτουν ισχυρούς πολυπύρηνους επεξεργαστές, μεγαλύτερες μνήμες και χώρους αποθήκευσης δεδομένων, που φτάνουν ακόμη και τα 128 GB. Έτσι, βάσει των παραπάνω δυνατοτήτων των σύγχρονων κινητών συσκευών είναι εφικτή σχεδίαση και υλοποίηση μιας εναλλακτικής Cloudlet-Based υποδομής υπολογιστικού νέφους. Σε αυτήν την υποδομή οι κινητές συσκευές, που είναι συνδεδεμένες μαζί της, θα μπορούν, όχι μόνο να αναζητούν, αλλά παράλληλα και να προσφέρουν πόρους στην υποδομή. Συνεπώς, το Cloudlet σε μια τέτοια εκδοχή θα έχει την ευελιξία, να αναζητά επιπλέον πόρους πρώτα από τις κινητές συσκευές, οι οποίες είναι συνδεδεμένες μαζί του και στην συνέχεια εάν είναι απαραίτητο από το Σύννεφο. Σε μια

τέτοια υποδομή, οι κινητές συσκευές συμμετέχουν ενεργά στην δημιουργία ενός Mobile Cloud Computing (MCC) περιβάλλοντος. Κάθε συσκευή μπορεί να προσφέρει υπολογιστικούς πόρους ή/και πόρους για την αποθήκευση δεδομένων.

Η εκπόνηση της παρούσας μεταπτυχιακής διατριβής βασίζεται στην προσέγγιση της Cloudlet-Based υποδομής υπολογιστικού νέφους. Ως αντικείμενο έρευνας έχει τη δέσμευση και διαχείριση υπολογιστικών πόρων από κινητές συσκευές (smartphones) και τη διάθεση τους σε Cloudlets. Επιπλέον, με χρήση πληροφοριών από το Wi-Fi δίκτυο, στο οποίο είναι συνδεδεμένη η φορητή συσκευή και το Cloudlet, υλοποιείται LBS υπηρεσία, η οποία τελικά αποτελεί μέρος της mobile εφαρμογής.

1.2 Σκοπός και Αντικείμενο Μελέτης

Η εκπόνηση της παρούσας μεταπτυχιακής διατριβής εστιάζει στη μελέτη και ανάπτυξη εφαρμογής δέσμευσης πόρων σε υπολογιστικά νέφη μικρής κλίμακας με τη χρήση κινητών συσκευών. Στόχος της είναι η υλοποίηση και παρουσίαση μέθοδο δυναμικής δέσμευσης υπολογιστικών πόρων από κινητές συσκευές μέσω μιας απλής εφαρμογής και η διάθεση τους στο Cloudlet με το οποίο επικοινωνούν. Οι κινητές συσκευές με αυτόν τον τρόπο θα παραχωρούν ένα τμήμα από την κεντρική μονάδα επεξεργασίας τους (CPU) στη Cloudlet-Based υποδομή υπολογιστικού νέφους, στην οποία συμμετέχουν, για την εκτέλεση εργασιών αυτής.

Επίσης, στόχο της εργασίας αποτελεί η ανάλυση και παρουσίαση μεθόδου εντοπισμού της θέσης του χρήστη μιας κινητής συσκευής εντός κλειστού χώρου χωρίς προφανώς τη χρήση GPS. Το A-GPS, με το οποίο είναι εξοπλισμένα όλα τα καινούργια smartphones και tablets, δεν είναι αποτελεσματικό μέσα σε κτίρια και γενικότερα κλειστούς χώρους. Στις επόμενες δυο ενότητες παρουσιάζονται αναλυτικά οι στόχοι και τα αντικείμενα μελέτης της μεταπτυχιακής διατριβής.

Σκοπός

- Μελέτη και ανάπτυξη εφαρμογής για την πλατφόρμα του λειτουργικού συστήματος Android, η οποία να έχει την δυνατότητα σύνδεσης με Cloudlet-Based υποδομή υπολογιστικού νέφους.

- Υλοποίηση μιας Cloudlet υποδομής με χρήση εργαλείου προσομοίωσης.
- Δυναμική δέσμευση μέρους της CPU μιας κινητής συσκευής, προκειμένου αυτή να προσφέρει υπολογιστικούς πόρους σε ένα Cloudlet.
- Αποτελεσματική διαχείριση και διαμοιρασμός των υπολογιστικών πόρων της κινητής συσκευής.
- Επικοινωνία της mobile εφαρμογής με την Cloudlet-based υποδομή υπολογιστικού νέφους και λήψη απόφασης για την ανάθεση εργασίας (task) στη κινητή συσκευή που συνδράμει.
- Εντοπισμός της τρέχουσας θέσης του χρήστη ανάλογα με το Access Point με το οποίο είναι συνδεδεμένη η φορητή του συσκευή.
- Υλοποίηση LBS υπηρεσίας.

Αντικείμενα Μελέτης

- ❖ Η μελέτη της Mobile – Cloudlet – Cloud Computing αρχιτεκτονικής.
- ❖ Η εύρεση μεθόδου για την δέσμευση και διαχείριση μέρους της CPU μιας Android κινητής συσκευής.
- ❖ Η επικοινωνία της mobile εφαρμογής με το Cloudlet με στόχο την ανταλλαγή δεδομένων.
- ❖ Λήψη απόφασης από το Cloudlet για το εάν και πότε ένα task θα ανατεθεί για εκτέλεση σε μια συσκευή, η οποία προσφέρει υπολογιστικούς πόρους σε αυτό.
- ❖ Η εκτέλεση ή απόρριψη ενός task από μια κινητή συσκευή, ανάλογα με τον εάν μπορεί να το διαχειριστεί.
- ❖ Η μέθοδος εντοπισμού της θέσης μιας φορητής συσκευής εντός κλειστού χώρου και η παροχή πληροφοριών βάσει αυτής.

1.3 Δομή της εργασίας

Το κεφάλαιο 1 περιέχει την εισαγωγή στην παρούσα μεταπτυχιακή διατριβή, καθώς και τους στόχους και τα αντικείμενα μελέτης.

Στο κεφάλαιο 2 περιγράφεται σε θεωρητικό επίπεδο η φύση του προβλήματος, το οποίο η διπλωματική διατριβή καλείται να επιλύσει και επιπλέον αναφέρονται τα μέρη από τα οποία αποτελείται. Επίσης, παρουσιάζονται κι άλλες προσεγγίσεις του θέματος, οι οποίες εντοπίζονται στην σχετική βιβλιογραφία. Ακόμη, γίνεται μια πρώτη εισαγωγή, ως προς τη μέθοδο που θα ακολουθηθεί, προκειμένου να προσεγγίσουμε το θέμα από την δική μας πλευρά.

Στο κεφάλαιο 3 γίνεται μια γενική περιγραφή της δομής του συστήματος και αναλύεται λεπτομερώς, η μέθοδος που ακολουθήθηκε για την επίλυση του προβλήματος. Επιπλέον, παρουσιάζονται οι εμπλεκόμενοι ρόλοι του συστήματος, περιγράφονται τα σενάρια χρήσης της εφαρμογής και αναλύονται λεπτομερώς οι λειτουργικές και μη λειτουργικές απαιτήσεις, οι οποίες περιγράφουν επακριβώς τις ιδιότητές της. Επιπλέον, παρουσιάζονται τα εργαλεία και οι τεχνολογίες, που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής και αναλύονται κρίσιμα τμήματα κώδικα.

Στο κεφάλαιο 4 παρουσιάζεται η εφαρμογή, που αναπτύχθηκε καθώς και τα αποτελέσματα του πειραματικού μέρους της εργασίας. Έπειτα, αξιολογούνται τα αποτελέσματα και γίνεται μια αποτίμηση της συνολικής προσέγγισης στο θέμα.

Στο κεφάλαιο 5 γίνεται μια σύνοψη της όλης προσέγγισης, διατυπώνονται τα συμπεράσματα στα οποία καταλήξαμε και παρατίθενται προτάσεις για μελλοντικές επεκτάσεις – βελτιώσεις της παρούσας εργασίας.

2. Θεωρητική Προσέγγιση

2.1 Cloud Computing

Το Cloud είναι ένα παράλληλο και κατανεμημένο σύνολο υπολογιστικών συστημάτων υψηλής υπολογιστικής ισχύος το οποίο διαθέτει τεράστιους αποθηκευτικούς χώρους. Τα συστήματα αυτά συνδέονται μεταξύ τους και είναι προσβάσιμα μέσω του Διαδικτύου από υπολογιστές και έξυπνες κινητές συσκευές. Η έννοια του Cloud βασίζεται στην διαχείριση και στον διαμοιρασμό πόρων με χρήση μιας Internet – based υποδομής, με στόχο την αύξηση της διαθεσιμότητας τους σε τρίτους χρήστες. [15] Οι πόροι του Cloud προσφέρονται στους τελικούς χρήστες με την ευελιξία να πληρώνουν μόνο για ότι χρησιμοποιούν. Το υπολογιστικό νέφος λόγω της εικονικοποίησης (virtualization) δίνει την ψευδαίσθηση στους χρήστες ότι οι πόροι του είναι ανεξάντλητοι. Το Cloud Computing αποτελεί επανάσταση στην Επιστήμη της Πληροφορικής και εξελίσσεται διαρκώς. [2][6]

Τα βασικότερα πλεονεκτήματα [3] του Cloud Computing για τους καταναλωτές συνοψίζονται παρακάτω:

- ✓ **Ανεξαρτησία συσκευής.** Οι χρήστες μπορούν να έχουν πρόσβαση στις υπηρεσίες και τις εφαρμογές που παρέχονται μέσω Cloud υποδομών, χρησιμοποιώντας οποιαδήποτε ηλεκτρονική συσκευή με πρόσβαση στο Διαδίκτυο, όπως για παράδειγμα ένα smartphone ή ένα tablet.
- ✓ **Ανεξαρτησία θέσης.** Το μόνο που απαιτείται για χρήση των Cloud υπηρεσιών είναι η δυνατότητα πρόσβασης στο Internet.
- ✓ **Διαμοιρασμός πόρων.** Οι χρήστες έχουν τη δυνατότητα να χρησιμοποιούν υπολογιστικού και όχι μόνο πόρους από τους υπέρ-υπολογιστές και τα μεγάλα κέντρα δεδομένων που φιλοξενούνται στο Cloud.
- ✓ **Οικονομικά οφέλη.** Οι επιχειρήσεις πλέον δεν είναι υποχρεωμένες να δαπανούν τεράστια ποσά για την αγορά υπολογιστικών συστημάτων και λογισμικού, αλλά μπορούν να «νοικιάζουν» ότι χρειάζονται από μια Cloud υποδομή.
- ✓ **Ελεκτασιμότητα.** Οι καταναλωτές μπορούν πολύ εύκολα να αποκτήσουν, επιπλέον πόρους σε περίπτωση όπου οι ανάγκες τους το απαιτούν.

- ✓ **Αδιάλειπτη διαθεσιμότητα αποθηκευτικού χώρου.** Το Cloud είναι επίσης υπεύθυνο για τη συνεχή διάθεση πόρων αποθήκευσης, οποιουδήποτε είδους δεδομένων και διατήρηση αρχείων.

Τα παραπάνω πλεονεκτήματα της υπολογιστικής νέφους συντέλεσαν στην ανάπτυξη πλήθους εφαρμογών, κυρίως εμπορικών καθώς και εφαρμογών κοινωνικής δικτύωσης. Οι χρήστες απλά χρησιμοποιούν τις προσφερόμενες υπηρεσίες, αποθηκεύουν τα δεδομένα τους στις υποδομές του Cloud, χωρίς να τους απασχολούν θέματα, τα οποία αφορούν την εγκατάσταση λογισμικού, την αναβάθμιση του, ή τη δημιουργία αρχείων backup κλπ.

Τα μοντέλα υπηρεσιών του Cloud Computing για τους καταναλωτές – χρήστες, είναι τρία και αναλύονται παρακάτω:

- **Software as a Service (SaaS).** Σε αυτό το μοντέλο ο καταναλωτής χρησιμοποιεί τις προσφερόμενες από το Cloud υπηρεσίες ή/και εφαρμογές.
- **Platform as a Service (PaaS).** Καταναλωτής σε αυτό το μοντέλο αναπτύσσει, υλοποιεί και παράλληλα ελέγχει και διαχειρίζεται τις εφαρμογές, οι οποίες φιλοξενούνται στο υπολογιστικό νέφος.
- **Infrastructure as a Service (IaaS).** Στο μοντέλο υπηρεσιών υποδομή ένας Cloud περιβάλλοντος, ο χρήστης εγκαθιστά, παρακολουθεί και διαχειρίζεται τις υποδομές των πληροφοριακών του συστημάτων.

Επίσης, στο Cloud Computing διακρίνονται τέσσερα διαφορετικά μοντέλα λειτουργίας [4], τα οποία τα εξής:

- **Ιδιωτικό.** Η υποδομή του υπολογιστικού νέφους λειτουργεί αποκλειστικά και μόνο για έναν οργανισμό ή εταιρεία. Η πρόσβαση στις υπηρεσίες του από τρίτους δεν είναι εφικτή. Η διαχείριση του στις περισσότερες περιπτώσεις γίνεται και οι υποδομές είναι εγκατεστημένες εντός του χώρου του Οργανισμού ή τις επιχείρησης.
- **Δημόσιο.** Σε αντίθεση με το Ιδιωτικό Cloud η υποδομή και οι υπηρεσίες του Δημοσίου είναι διαθέσιμες στο ευρύτερο κοινό ή σε ένα πολύ μεγάλο σύνολο εταιρειών.

- **Κοινοτικό.** Η υποδομή αυτού του μοντέλου λειτουργίας του υπολογιστικού νέφους είναι μοιρασμένη σε συγκεκριμένους οργανισμούς και υποστηρίζει μια κοινότητα χρηστών, οι οποίοι έχουν κάποιους κοινούς σκοπούς.
- **Υβριδικό.** Αυτή η υποδομή υπολογιστικού νέφους προκύπτει από τη σύνθεση δύο ή και περισσότερων υποδομών από τα μοντέλα που περιγράφηκαν παραπάνω. Με αυτόν τον τρόπο προκύπτει μια Cloud αρχιτεκτονική με συνδυασμένα χαρακτηριστικά.

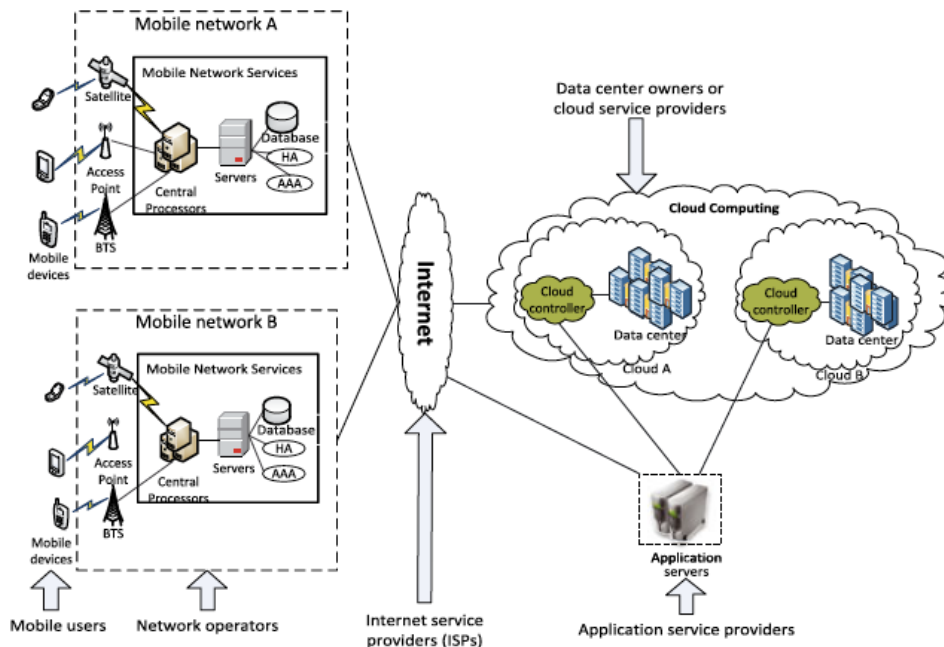
Η υπολογιστική νέφους μπορεί να χρησιμοποιηθεί όχι μόνο από υπολογιστές και εξυπηρετητές, αλλά και από έξυπνες κινητές συσκευές.

2.2 Mobile Cloud Computing

Ο συνδυασμός της υπολογιστικής νέφους με το Mobile Computing οδήγησε στην έννοια του Mobile Cloud Computing (MCC). Η υποδομή του Mobile Cloud Computing ουσιαστικά επεκτείνει τις επεξεργαστικές δυνατότητες των φορητών συσκευών και επιπλέον προσφέρει σε αυτές χώρους για την αποθήκευση δεδομένων. Με αυτόν τον τρόπο μπορούν να ξεπεραστούν κάποιες από τις αδυναμίες των κινητών συσκευών, όπως είναι εξάντληση της μπαταρίας, οι χαμηλών δυνατοτήτων επεξεργαστές σε σύγκριση με αυτούς των υπολογιστών, η περιορισμένη διάθεση πόρων για μόνιμη αποθήκευση δεδομένων, όπως βίντεο, φωτογραφιών κ.ά. [15].

Πλέον, οποιαδήποτε φορητή συσκευή μπορεί να συνδεθεί με το Cloud με χρήση μιας thin client εφαρμογής και να χρησιμοποιήσει υπηρεσίες – εφαρμογές, οι οποίες προσφέρονται στους τελικούς καταναλωτές σε επίπεδο SaaS. Επίσης, οι κινητές συσκευές εκτός από τη χρήση υπηρεσιών, έχουν τη δυνατότητα μεταφοράς και αποθήκευσης μεγάλου όγκου δεδομένων στο Cloud. Τα smartphones και τα tablets καλούνται πλέον, να εκτελέσουν πολύπλοκες εφαρμογές πραγματικού χρόνου και υψηλής ανάλυσης γραφικών. Οι εφαρμογές αυτές έχουν υψηλές απαιτήσεις σε υπολογιστικούς πόρους, τους οποίους σε πολλές περιπτώσεις οι φορητές συσκευές δε διαθέτουν. Σε αυτήν την περίπτωση τη λύση έρχεται να δώσει το MCC. Μια κινητή συσκευή έχει τη δυνατότητα να χρησιμοποιήσει υπολογιστικούς πόρους από το Σύννεφο, με στόχο την αποδοτικότερη εκτέλεση των απαιτητικών εφαρμογών. Με τη χρήση του MCC πολύπλοκες και πολυμεσικές εφαρμογές εφαρμογές, μπορούν να μεταφέρονται και να εκτελούνται στους διακομιστές (servers) του

Νέφους, αντί να εκτελούνται τοπικά στην συσκευή. Το παραπάνω έχει ως αποτέλεσμα την ταχύτερη εκτέλεση των εφαρμογών, την μικρότερη δέσμευση υπολογιστικών πόρων της συσκευής, καθώς και τη χαμηλότερη κατανάλωση μπαταρίας, αφού η κεντρική μονάδα επεξεργασίας (CPU) χρησιμοποιείται σε μικρότερο βαθμό [4] [5].



Εικόνα 2.1: Mobile Cloud Computing Αρχιτεκτονική [31]

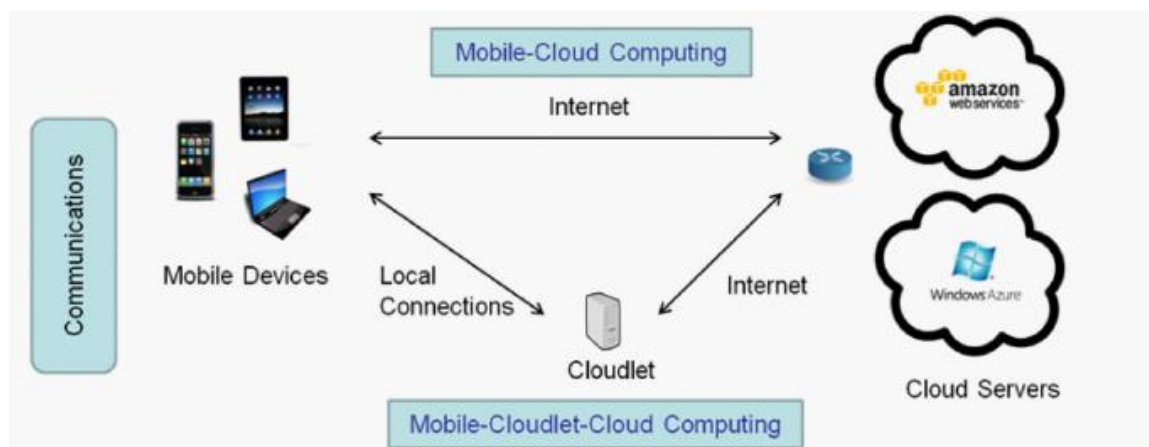
Οι συσκευές μπορούν να συνδεθούν με μια Cloud υποδομή, προκειμένου να μεταφέρουν εργασίες και δεδομένα μέσω του Διαδικτύου, με χρήση δικτύων κινητής τηλεφωνίας (3G/4G συνδέσεις) ή μέσω ενός Wi-Fi hotspot. Σε αυτό το σημείο δημιουργούνται κάποια ουσιαστικά προβλήματα για την αρχιτεκτονική του Mobile Cloud Computing, τα οποία σχετίζονται κυρίως με το μικρό εύρος ζώνης, τις καθυστερήσεις, καθώς και την υψηλή κατανάλωση ενέργειας των 3G/4G συνδέσεων. Επομένως, σε περιπτώσεις όπου πρέπει να μεταφερθεί μεγάλος όγκος δεδομένων μέσω Διαδικτύου, όπως μια εικόνα για επεξεργασία, η απόδοση αυτής της υποδομής δεν είναι η αναμενόμενη. [8]

2.3 Cloudlet-Based MCC Αρχιτεκτονική

Μια νέα αρχιτεκτονική Mobile Cloud Computing βασισμένη στη χρήση των Cloudlets έρχεται να δώσει τη λύση στο πρόβλημα, που περιγράφηκε στην προηγούμενη ενότητα. Μια Cloudlet-Based υποδομή υπολογιστικού νέφους, αποτελείται από τη σύνθεση τριών συστημάτων του Mobile Computing, των Cloudlets και του υπολογιστικού νέφους. [9]

Το Cloudlet είναι ένα υπολογιστικό σύστημα (datacenter) μικρής κλίμακας, το οποίο αποτελείται από έναν τουλάχιστον υπολογιστή με τη δική του ανεξάρτητη CPU, GPU και μνήμη. Το Cloudlet μπορεί να είναι και μια συστάδα (cluster) υπολογιστών. Τα Cloudlets αποτελούν ένα ενδιάμεσο επίπεδο στην υποδομή του MCC, τα οποία παρεμβάλλονται ανάμεσα στην επικοινωνία των κινητών συσκευών με το Cloud. Οι κινητές συσκευές συνδέονται με τα Cloudlets μέσω ενός Wi-Fi τοπικού δικτύου ή/και σπανιότερα με τη χρήση Bluetooth. Σε μια τέτοια Cloud υποδομή όπως γίνεται αντιληπτό, προϋπόθεση για τις κινητές συσκευές είναι να βρίσκονται σε μικρή απόσταση από το Cloudlet [10][12]. Όσες συσκευές συνδέονται με ένα Cloudlet, μπορούν να του αναθέτουν εργασίες για εκτέλεση και δεδομένα εκμεταλλευόμενες τα πλεονεκτήματα μιας one-hop Wi-Fi σύνδεσης, δηλαδή, το μεγάλο εύρος ζώνης, τις μικρότερες καθυστερήσεις του δικτύου και τη χαμηλότερη κατανάλωση ενέργειας. Η χρήση των Cloudlets στην αρχιτεκτονική του MCC συμβάλει στη μείωση του χρόνου απόκρισης των εφαρμογών και στη γενικότερη αύξηση της απόδοσης του συστήματος, προσφέροντας έτσι τη βέλτιστη εμπειρία χρήσης στους καταναλωτές υπηρεσιών [11]. Η χρήση του Cloudlet στοχεύει [7][8]:

- ✓ Στην επεξεργασία δεδομένων και στην εκτέλεση εργασιών, οι οποίες στέλνονται από τις κινητές συσκευές.
- ✓ Στην προσωρινή αποθήκευση δεδομένων είτε από τις συσκευές, είτε από το Cloud (π.χ. φόρτωση τμήματος μιας μεγάλης βάσης δεδομένων).
- ✓ Στην προώθηση εργασιών στο Cloud, στην περίπτωση όπου αυτό δε διαθέτει τους απαιτούμενους πόρους για τη διαχείριση όλων των tasks, που του έχουν ανατεθεί.



Εικόνα 2.2: Mobile Cloud Computing & Mobile Cloudlet Cloud Computing αρχιτεκτονικές [3]

Τα Cloudlets μιμούνται κατά κάποιο τρόπο το μοντέλο λειτουργίας του υπολογιστικού νέφους, αλλά αποτελούν διαφορετικά συστήματα, με άλλους σκοπούς το καθένα και για αυτό το λόγο δεν πρέπει να συγχέονται. Στην εικόνα 2.2 απεικονίζεται η αρχιτεκτονική του Mobile Cloud Computing, καθώς και του Cloudlet-Based μοντέλου υπολογιστικού νέφους.

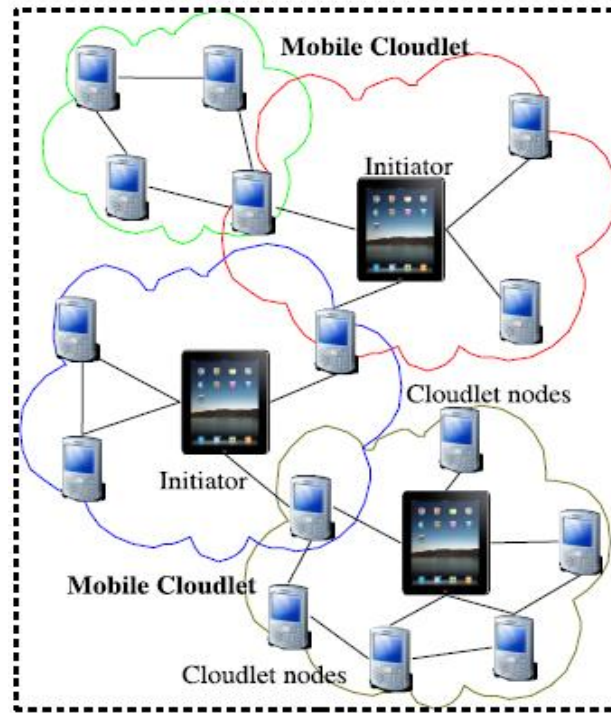
2.4 Mobile Device Clouds

Μια Cloudlet-based υποδομή Cloud στοχεύει στη διάθεση πόρων σε κινητές συσκευές ακολουθώντας έναν αποδοτικότερο τρόπο. *Τι γίνεται όμως στη περίπτωση, όπου μια κινητή συσκευή χρειάζεται επιπλέον υπολογιστικούς πόρους και δεν έχει την δυνατότητα, χρήσης ενός Cloudlet ;* Σε αυτό το ερώτημα έρχεται να δώσει απάντηση η ανάπτυξη των Mobile Device Clouds.

Σε αυτή την προσέγγιση, οι κινητές συσκευές, οι οποίες βρίσκονται στον ίδιο χώρο και μπορούν να επικοινωνήσουν μέσω Wi-Fi ή Bluetooth, δημιουργούν μια εικονική (virtual) υποδομή υπολογιστικού νέφους. Αυτή η εικονική υποδομή του υπολογιστικού νέφους είναι γνωστή ως Mobile Device Cloud – (MDC). Σε αυτό το σύστημα μια συσκευή αναζητά υπολογιστικούς πόρους σε άλλες κινητές συσκευές προκειμένου, να εκτελέσει ταχύτερα και με τη λιγότερο δυνατή κατανάλωση ενέργειας μια απαιτητική σε πόρους και χρόνο εφαρμογή. Οι υπόλοιπες συσκευές, που συμμετέχουν στο MDC, αναμένουν την ανάθεση ενός task. Αυτές οι συσκευές προσφέρουν μέρος από τη CPU τους προκειμένου να εκτελέσουν παράλληλα διαφορετικά tasks της ίδιας εφαρμογής. Το διαμοιρασμό των εργασιών αναλαμβάνει η ίδια η συσκευή, η οποία πρώτη ζήτησε πόρους από τις υπόλοιπες. Επίσης, αυτή η συσκευή είναι υπεύθυνη για τη συλλογή και αποθήκευση των αποτελεσμάτων. Τα αποτελέσματα της εκτέλεσης κατά αυτόν τον τρόπο θα είναι διαθέσιμα και για τις υπόλοιπες συσκευές, σε περίπτωση που τα χρειαστούν. Επομένως, οι υπόλοιπες συσκευές δε θα χρειαστεί, να εκτελέσουν ξανά την ίδια εφαρμογή. Το παραπάνω έχει ως αποτέλεσμα, όλες οι συσκευές να εξοικονομούν ενέργεια και να εκτελούν ταχύτερα εφαρμογές, τα αποτελέσματα των οποίων αποτελούν κοινό στόχο των χρηστών τους. [1]

Ένα παράδειγμα αυτής της προσέγγισης του MCC είναι η μετάφραση ενός περιγραφικού κειμένου, που έχει τραβηχτεί σε φωτογραφία σε μια άλλη γλώσσα. Σε αυτή την περίπτωση, η εκτέλεση μιας τέτοιου είδους εφαρμογής είναι ιδιαίτερα απαιτητική για μια

και μόνο συσκευή. Αντιθέτως, ο διαμοιρασμός και η παράλληλη εκτέλεση της σε συσκευές απόμων της ίδια ομάδας, αποτελεί ευκολότερη υπόθεση. Βασική προϋπόθεση είναι η συμμετοχή όσο το δυνατόν περισσότερων συσκευών.[23]



Εικόνα 2.3: Mobile Device Clouds [13]

2.5 Resource Allocation

Με τον όρο resource allocation εννοούμε τη δέσμευση και το διαμοιρασμό των πόρων ενός συστήματος σε άλλα συστήματα ή εφαρμογές. Οι πόροι ενός συστήματος, οι οποίοι μπορούν να δεσμευτούν και να κατανεμηθούν κατάλληλα, μπορεί να υπολογιστικοί, πόροι αποθήκευσης δεδομένων, πόροι κύριας μνήμης κ.ά. Προϋπόθεση για την αποτελεσματική δέσμευση και κατανομή πόρων, είναι το σύστημα να γνωρίζει κάθε στιγμή την διαθεσιμότητα τους. [19]

Το υπολογιστικό νέφος αποτελεί λαμπρό παράδειγμα όσων περιγράφηκαν στην προηγούμενη παράγραφο. Οι Cloud υποδομές σχεδιάστηκαν με στόχο τη δέσμευση και τη διάθεση πόρων σε άλλα συστήματα, όπως για παράδειγμα σε υπολογιστές, διακομιστές, καθώς και σε έξυπνες κινητές συσκευές. Το Σύννεφο δέχεται και διαχειρίζεται τα αιτήματα (requests) από άλλα συστήματα και τα κατανέμει κατάλληλα διαθέσιμους πόρους.

Η πολιτική δέσμευσης πόρων διαφέρει ανάλογα με τις απαιτήσεις του συστήματος και τις ανάγκες των καταναλωτών. Μέρος των πόρων ενός συστήματος μπορεί, να δεσμευτεί για την εκτέλεση μιας ειδικής κατηγορίας εφαρμογών ή για την εξυπηρέτηση μιας συγκεκριμένης ομάδας χρηστών. [18]

Για την αποτελεσματική δέσμευση και διαμοιρασμό των πόρων απαιτείται ο σχεδιασμός και η υλοποίηση κατάλληλου αλγορίθμου. Ο αλγόριθμος θα πρέπει να ενημερώνει συνεχώς το σύστημα για τη διαθεσιμότητα των πόρων του. Έτσι, το σύστημα με την σειρά θα είναι σε θέση οποιαδήποτε στιγμή να εξυπηρετεί ή να απορρίπτει τα αιτήματα.

2.6 Location-Based Services

Οι εφαρμογές παροχής υπηρεσιών θέσης, ως στόχο έχουν την παροχή πληροφοριών ανάλογα με το χώρο, στον οποίο κινείται ο χρήστης της συσκευής. Πλέον, όλες οι καινούργιες κινητές συσκευές διαθέτουν δέκτη GPS, κάτι το οποίο διευκολύνει σημαντικά την ανάπτυξη LBS εφαρμογών, στην περίπτωση όπου ο χρήστης βρίσκεται σε ανοικτό χώρο π.χ. στο δρόμο. Σε περίπτωση όμως, όπου ο χρήστης βρίσκεται σε κλειστό χώρο π.χ. σε ένα μεγάλο εμπορικό κέντρο, η μέθοδος εντοπισμού της θέσης του βασίζεται στη χρήση και επεξεργασία πληροφοριών από το δίκτυο. Τέτοιου είδους πληροφορίες είναι οι IP και MAC διευθύνσεις των δικτυακών συσκευών. [16][17]

2.7 Η Φύση του Προβλήματος

Το πρόβλημα το οποίο μελετάται στην παρούσα διπλωματική διατριβή αφορά τους τομείς του Resource Allocation και της Cloudlet-Based αρχιτεκτονικής του MCC. Όπως αναλύθηκε στις παραπάνω ενότητες, όταν οι υπολογιστικοί πόροι του Cloudlet εξαντληθούν, λόγω του υψηλού φόρτου εργασιών, υπάρχουν δυο επιλογές:

1. Το Cloudlet, θα πρέπει να αναζητήσει επιπλέον υπολογιστικούς πόρους από το Cloud με το οποίο επικοινωνεί, προωθώντας σε αυτό τα tasks των κινητών συσκευών.
2. Το Cloudlet ανάλογα με την πολιτική προτεραιότητας εκτέλεσης των εργασιών, θα πρέπει να προχωρήσει στην απόρριψη κάποιων εξ' αυτών.

Τα παραπάνω έχουν ως αποτέλεσμα είτε οι χρήστες των κινητών συσκευών να μη μπορούν να εξυπηρετηθούν, είτε ο χρόνος εκτέλεσης των tasks, που ανατίθενται στο Cloudlet να αυξάνεται, λόγω προώθησης τους για εξυπηρέτηση στο Σύννεφο.

Στην ενότητα 2.4 περιγράφηκε η έννοια της δημιουργίας μιας εικονικής υποδομής υπολογιστικού νέφους με χρήση κινητών συσκευών. Σε αυτή την ενότητα αναλύθηκε μια μέθοδος κατά την οποία οι κινητές συσκευές δίνουν υπολογιστικούς πόρους η μια στην άλλη και εκτελούν παράλληλα εργασίες για την ταχύτερη εκτέλεση μιας κοινής εφαρμογής. Σε αυτή την περίπτωση οι κινητές συσκευές επικοινωνούν μεταξύ μέσω ενός ad hoc ασύρματου δικτύου.

Μια ανάλογη λύση μπορεί να υλοποιηθεί και για την περίπτωση των Cloudlets. Σε αυτήν την προσέγγιση οι κινητές συσκευές, οι οποίες επικοινωνούν με ένα Cloudlet να έχουν τη δυνατότητα εκτός από τα ζητάνε πόρους από αυτό, αλλά παράλληλα και να του προσφέρουν. Με αυτόν τον τρόπο το Cloudlet θα μπορεί να αναθέτει tasks σε κινητές, οι οποίες είναι συνδεδεμένες στο ίδιο τοπικό ασύρματο δίκτυο με αυτό. Τα tasks, τα οποία καλείται να εκτελέσει ένα Cloudlet προέρχονται κυρίως μικρές φορητές συσκευές, όπως είναι τα smartphones, επομένως, ένα τέτοιο task μπορεί εύκολα να ανατεθεί σε άλλο smartphone ή να διαμοιραστεί σε περισσότερα.

Στη βιβλιογραφία δεν υπάρχει άλλη αντίστοιχη προσέγγιση, στην οποία mobile συσκευές να προσφέρουν υπολογιστικούς πόρους σε μια Cloudlet-Based αρχιτεκτονική υπολογιστικού νέφους. Η μόνη προσέγγιση για την επίλυση του προβλήματος με τους πόρους των Cloudlets, είναι η εξυπηρέτηση τους από το ίδιο Σύννεφο.

Ένα ακόμη ζήτημα, το οποίο καλείται να επιλύσει η παρούσα μεταπτυχιακή διατριβή, είναι ανεύρεση μεθόδου με την οποία θα δεσμεύονται υπολογιστικοί πόροι από μια κινητή συσκευή. Η κινητή συσκευή θα πρέπει, να δεσμεύει μέρος από τους υπολογιστικούς της πόρους με σκοπό την διάθεση τους για εκτέλεση εργασιών από το Cloudlet. Η δέσμευση των πόρων θα γίνεται δυναμικά κατά την ώρα εκτέλεσης (runtime) της εφαρμογής με επιλογή του ίδιου του χρήστη της συσκευής. Το Cloudlet θα ενημερώνεται απευθείας για τους επιπλέον πόρους που θα έχει στη διάθεση του.

Στη βιβλιογραφία υπάρχουν αρκετές μελέτες και προσεγγίσεις στον τομέα του resource allocation και του resource management σε διάφορα υπολογιστικά συστήματα. Δεν υπάρχει όμως προσέγγιση για δυναμική δέσμευση υπολογιστικών πόρων σε κινητές συσκευές και διάθεσης τους σε υποδομή υπολογιστικού νέφους.

3. Προσέγγιση Λύσης

3.1 Μέθοδος

Στο πλαίσιο υλοποίησης της παρούσας μεταπτυχιακής διατριβής αναπτύχθηκε εφαρμογή για την πλατφόρμα του λειτουργικού συστήματος Android, η οποία ως στόχο έχει την παρουσίαση λύσης για το πρόβλημα της έλλειψης υπολογιστικών πόρων σε μια Cloudlet-Based αρχιτεκτονική υπολογιστικού νέφους. Με τη χρήση αυτής της εφαρμογής ο χρήστης έχει την δυνατότητα της δυναμικής δέσμευσης των υπολογιστικών πόρων της κινητής του συσκευής, προκειμένου αυτή να συνδράμει σε ένα Cloudlet, που αναζητά επιπλέον πόρους για την εκτέλεση των διεργασιών που του έχουν ανατεθεί. Ο χρήστης μέσω της εφαρμογής έχει την ελευθερία, να αποφασίσει το εάν και το πότε η συσκευή του θα συμμετέχει στην Cloudlet υποδομή, καθώς επίσης και το πόσους υπολογιστικούς πόρους θα προσφέρει. Επίσης, η εφαρμογή εντοπίζει αυτόματα την θέση του χρήστη εντός του κλειστού χώρου ενός κτιρίου με χρήση πληροφοριών από Wi-Fi δίκτυο στο οποίο είναι συνδεδεμένη η φορητή συσκευή, όπως είναι η MAC διευθύνσεις των Access Points. Το παραπάνω αποτελεί ουσιαστική λύση για χώρους και περιπτώσεις χρήσης όπου το GPS, δεν μπορεί να χρησιμοποιηθεί αποτελεσματικά. Τα εργαλεία, τα οποία χρησιμοποιήθηκαν για τις ανάγκες της ανάπτυξης καθώς και της διεξαγωγής των πειραμάτων, είναι το Android API¹, το CloudSim API² και η εφαρμογή Linpack for Android³.

Το Android API χρησιμοποιήθηκε για την ανάπτυξη της mobile εφαρμογής, με την οποία ο χρήστης μπορεί να χρησιμοποιεί την LBS υπηρεσία, να δεσμεύει υπολογιστικούς πόρους, τους οποίους στην συνέχεια θα παραχωρεί στο Cloudlet με το οποίο επικοινωνεί η φορητή του συσκευή του. Το CloudSim API είναι Πλαίσιο – (Framework), το οποίο χρησιμοποιείται για την μοντελοποίηση και προσομοίωση Cloud υποδομών σε ένα εικονικό τεστ περιβάλλον. Η χρήση εργαλείων προσομοίωσης αποτελεί μια οικονομική, αξιόπιστη και εξίσου αποτελεσματική λύση για τον έλεγχο εφαρμογών, την αξιολόγηση των αποτελεσμάτων εκτέλεσης, καθώς και την εξαγωγή κατάλληλων συμπερασμάτων. Θα πρέπει σε αυτό το σημείο να διασαφηνιστεί ότι, για τον έλεγχο της εφαρμογής και την

¹ <https://developer.android.com/sdk/index.html>

² <http://www.cloudbus.org/cloudsim/doc/api/>

³ <https://play.google.com/store/apps/details?id=rs.pedjaapps.Linpack>

εκτέλεση του πειραματικού μέρους της μεταπτυχιακής διατριβής χρησιμοποιήθηκε ο Android Emulator, ο οποίος αποτελεί εικονική συσκευή και παρέχεται από το Android API. Τέλος, η εφαρμογή Linpack for Android χρησιμοποιήθηκε για την μέτρηση της υπολογιστικής απόδοσης μιας Android κινητής συσκευής, τα αποτελέσματα της οποίας συνέβαλαν ουσιαστικά στην διαδικασία υπολογισμού των πόρων, που θα προσφέρει μια κινητή συσκευή στο Cloudlet. Στην ενότητα 3.8 περιγράφονται αναλυτικά τα εργαλεία που χρησιμοποιήθηκαν στην παρούσα μεταπτυχιακή διατριβή.

3.2 Μεθοδολογία

Για τις ανάγκες της εργασίας υλοποιήθηκε μια Cloudlet υποδομή με χρήση του εργαλείου προσομοίωσης Cloud υποδομών CloudSim. Αυτή η υποδομή αποτελείται από ένα Broker, ένα Data Center με δυο Hosts, καθένας από τους οποίους τρέχει μια Virtual Machine (VM). Αυτή η εικονική υποδομή αποτελεί το τεστ περιβάλλον της πειραματικού μέρους της εφαρμογής. Η επικοινωνία της android εφαρμογής με την υποδομή του Cloudlet γίνεται με χρήση client/server java sockets.

Ο χρήστης μπορεί μέσα από την εφαρμογή να ενημερωθεί για το ποσοστό της CPU της συσκευής, το οποίο χρησιμοποιείται μια δεδομένη στιγμή από άλλες εφαρμογές, καθώς και εργασίες του λειτουργικού συστήματος. Με αυτόν τρόπο ο τελικός χρήστης είναι σε θέση να αποφασίσει τα εξής δύο:

- Αν η συσκευή του έχει αρκετούς ελεύθερους υπολογιστικούς πόρους, ώστε να υποστηρίξει επιπλέον εργασίες από το Cloudlet.
- Το ποσοστό από τους διαθέσιμους πόρους της συσκευής του, που θα δεσμεύσει για τον παραπάνω σκοπό.

Για την προβολή στη οθόνη του ποσοστού της CPU το οποίο χρησιμοποιείται, υλοποιήθηκε ειδική μέθοδος για το «διάβασμα» της χρήσης της.

Ο χρήστης, στην περίπτωση που αποφασίσει ότι επιθυμεί η συσκευή του να προσφέρει πόρους στο Cloudlet, πρέπει να επιλέξει ο ίδιος με χρήση μια μπάρας πρόοδο ένα ποσοστό από τη διαθέσιμη, το οποίο θα παραχωρήσει στο Cloudlet. Στη συνέχεια, το Cloudlet ενημερώνεται άμεσα για την κινητή συσκευή, που συνδράμει στην υποδομή, καθώς και για τους υπολογιστικούς πόρους που αυτή προσφέρει. Σε αυτό το σημείο πρέπει να επισημανθεί ότι η κινητή συσκευή δεν στέλνει στο Cloudlet ένα ποσοστό (επί τοις εκατό)

ως ενημέρωση για τους πόρους, που του προσφέρει, αλλά έναν αριθμό Million Floating Operations Per Second - (MFLOP).

Τα MFLOPS αποτελούν μονάδα μέτρησης της απόδοσης οποιουδήποτε υπολογιστικού συστήματος και παρουσιάζουν τον ρυθμό εκτέλεσης εκατομμύρια εντολών κινητής υποδιαστολής ανά δευτερόλεπτο από ένα υπολογιστικό σύστημα. Για τον υπολογισμό των MFLOPS δεν υλοποιήθηκε κάποια ειδική μέθοδος στην εφαρμογή, αλλά χρησιμοποιήθηκε η εφαρμογή Linpack for Android, η οποία εγκαταστάθηκε σε συσκευή Samsung Galaxy TrendTM, προκειμένου να βρεθεί ο συνολικός αριθμός των MFLOPS, που μπορεί αυτή να διαχειριστεί. Ο αριθμός των MFLOPS, που προέκυψε, χρησιμοποιήθηκε για τις ανάγκες της εργασίας στον κώδικα της εφαρμογής. Το αποτέλεσμα του benchmark test έδειξε ότι, η παραπάνω συσκευή έχει απόδοση περίπου 50 MFLOPS. Με βάση όσα αναφέρθηκαν παραπάνω, όταν ο χρήστης επιλέξει για παράδειγμα να παραχωρήσει το 50% της CPU της συσκευής του στο Cloudlet, τότε αυτή στέλνει τον αριθμό 25 (MFLOPS).

Μόλις το Cloudlet ενημερωθεί για μια συσκευή, η οποία του προσφέρει πόρους, τότε της αναθέτει ένα task, το οποίο όμως να μπορεί να διαχειριστεί. Κάθε εφαρμογή ή task απαιτεί την χρήση συγκεκριμένου αριθμού MFLOPS, προκειμένου να εκτελεστεί. Αυτός ο αριθμός θεωρείται για τις ανάγκες της παρούσας διπλωματικής διατριβής δεδομένος και γνωστός για την Cloudlet υποδομή και δεν υπολογίζεται με κάποια μέθοδο. Σε περίπτωση όπου το Cloudlet έχει κάποιο task, το οποίο μπορεί να εκτελεστεί από την κινητή συσκευή, της το αναθέτει. Η κινητή συσκευή με την σειρά μόλις της ανατεθεί ένα task ελέγχει ξανά τη διαθεσιμότητα των πόρων της. Εάν έχει τους απαραίτητους υπολογιστικούς πόρους για να εκτελέσει το task, τότε ενημερώνει το Cloudlet με ένα success code, αλλιώς στέλνει ένα error code δηλώνοντας την αδυναμία της να το διαχειριστεί πλέον. Η εφαρμογή εμφανίζει τα ανάλογα αποτελέσματα και στην οθόνη της συσκευής προς ενημέρωση του χρήστη. Σε μια ακραία περίπτωση ο στόχος είναι η κινητή συσκευή να μη «καταρρεύσει» λόγω μεγάλου του φόρτου εργασιών.

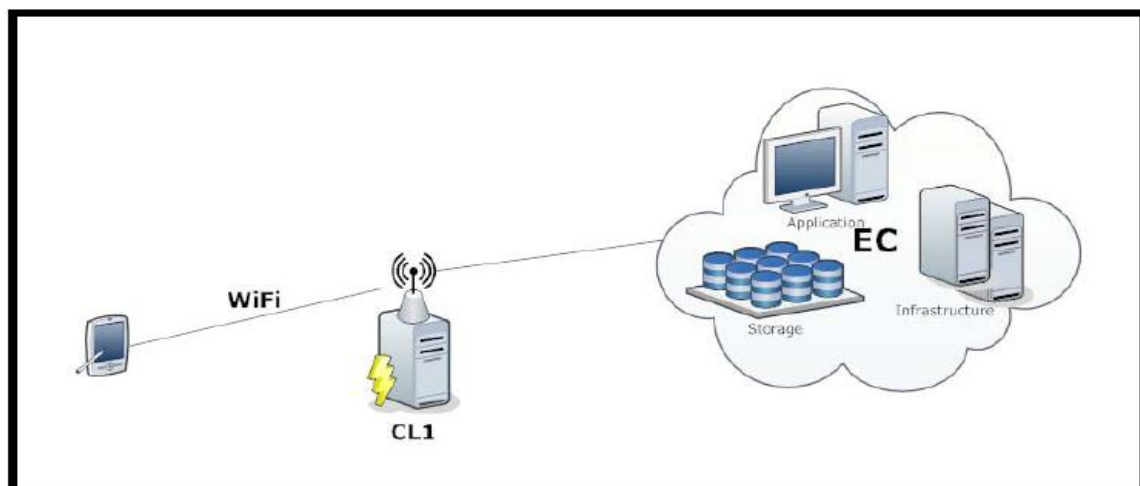
Το επόμενο ζήτημα, το οποίο καλείται επίσης να επιλύσει η παρούσα μεταπτυχιακή διατριβή, αφορά τον εντοπισμό θέσης του χρήστη σε κλειστό χώρο, όπως είναι αυτός ενός πανεπιστημίου. Ο εντοπισμός της θέσης του χρήστη γίνεται με χρήση πληροφοριών από το ασύρματο δίκτυο στο οποίο είναι συνδεδεμένη η συσκευή. Το Android API δίνει τη

δυνατότητα στους προγραμματιστές με χρήση ήδη υλοποιημένων μεθόδων, να βρουν το BSSID - Basic Service Set Identification του Access Point (AP) με το οποίο είναι συνδεδεμένη η συσκευή. Το BSSID αντιστοιχεί στη MAC διεύθυνση του AP και είναι μοναδική για κάθε συσκευή. Με αυτόν τον τρόπο, γνωρίζοντας το AP με το οποίο είναι συνδεδεμένη η συσκευή, καθώς και τη θέση του στο χώρο, είναι εφικτός ο εντοπισμός του χρήστη. Προϋπόθεση για την παραπάνω λύση είναι, η εφαρμογή να είναι ενημερωμένη για τη θέση και τις MAC διευθύνσεις των AP.

Η LBS υπηρεσία, που υλοποιήθηκε, ως στόχο έχει τον εντοπισμό της θέσης του φοιτητή στο χώρο εντός του πανεπιστημίου και την παροχή πληροφοριών ανάλογα με αυτή. Ο φοιτητής μπορεί να λαμβάνει πληροφορίες σχετικά με τα μαθήματα, που διεξάγονται στον όροφο, τον οποίο βρίσκεται εκείνη τη στιγμή.

Θα πρέπει να σημειωθεί ότι, η LBS υπηρεσία δεν είναι ολοκληρωμένη, αλλά αποτελεί το αρχικό στάδιο και τη βάση για μελλοντική ανάπτυξη και επέκταση της παρούσας διπλωματικής διατριβής, όπως θα αναλυθεί και στα επόμενα κεφάλαια.

Η παρακάτω εικόνα αναπαριστά μια Cloudlet-Based υποδομή υπολογιστικού νέφους, όπως αυτή που μελετάται σε αυτή την εργασία. Όπως φαίνεται στην εικόνα το Cloudlet μπορεί να είναι παράλληλα και AP.



Εικόνα 3.1: MCC Cloudlet-Based Μοντέλο [11]

3.3 Περιγραφή των εμπλεκόμενων ρόλων

Φοιτητής: Ο φοιτητής αποτελεί τον βασικό χρήστη του συστήματος μας. Ο φοιτητής του τμήματος πληροφορικής, μπορεί να χρησιμοποιήσει την εφαρμογή ώστε να δει το ποσοστό της CPU της συσκευής που είναι δεσμευμένο και να αποφασίσει, αν θα επιτρέψει στην συσκευή του να προσφέρει πόρους επεξεργαστικής ισχύος στο Cloudlet. Επίσης, ο φοιτητής μπορεί να χρησιμοποιήσει την LBS υπηρεσία, που παρέχεται από την εφαρμογή προκειμένου, να ενημερωθεί για τα μαθήματα, τα οποία διεξάγονται στον όροφο που βρίσκεται καθώς και για αποκτήσει πρόσβαση στο υλικό αυτών.

Cloudlet: Το Cloudlet, όπως ήδη έχουμε αναφέρει, είναι ένα σύστημα το οποίο αποτελεί τμήμα μιας Cloud υποδομής. Το Cloudlet αποτελεί τον δεύτερο ρόλο της εφαρμογής, που έχουμε υλοποιήσει, αφού ζητά πόρους από τις κινητές συσκευές που είναι συνδεδεμένες μαζί του και παράλληλα εκτελούν την εφαρμογή. Το Cloudlet αναθέτει tasks στις κινητές συσκευές, οι οποίες του παρέχουν πόρους, εφόσον αυτές μπορούν να τα διαχειριστούν.

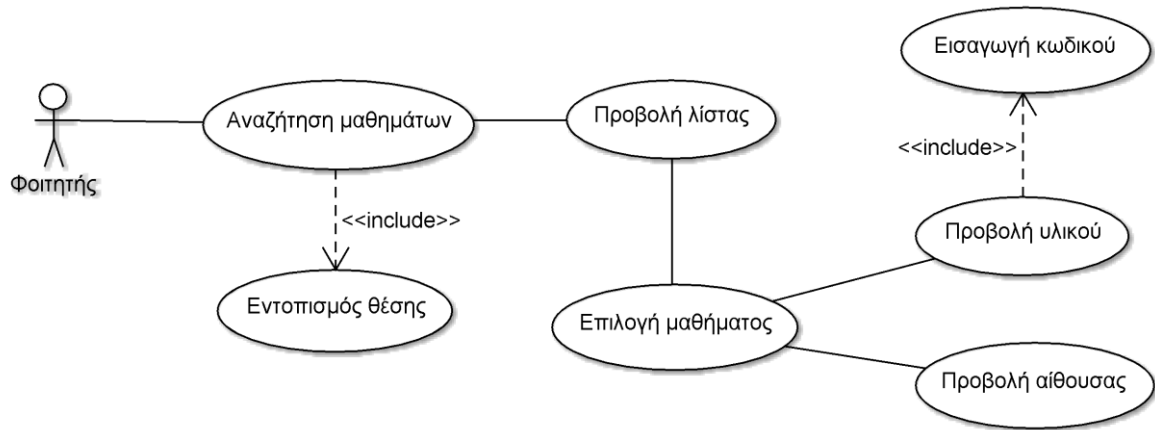
3.4 Περιγραφή Περιπτώσεων Χρήσης

3.4.1 Διαγράμματα Περιπτώσεων Χρήσης

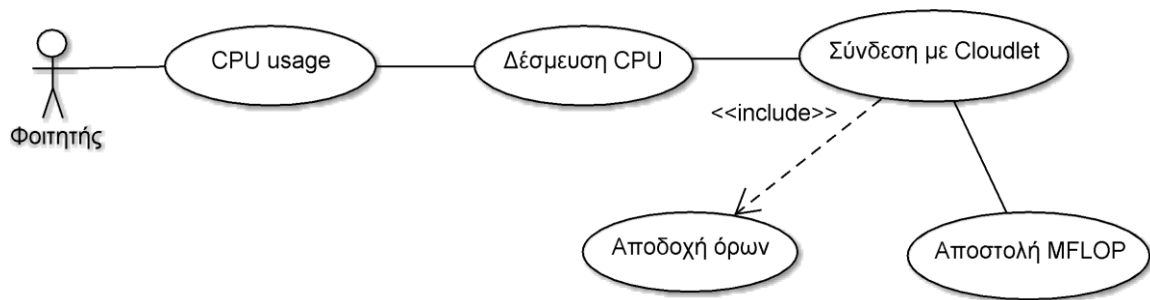
Σε αυτήν την ενότητα παρατίθενται πέντε διαγράμματα περιπτώσεων χρήσης της εφαρμογής. Το πρώτο διάγραμμα παρουσιάζει το βασικό μενού επιλογών της εφαρμογής. Το δεύτερο διάγραμμα αναπαριστά την περίπτωση χρήσης, όπου ο χρήστης αναζητά πληροφορίες για τα μαθήματα του προγράμματος σπουδών του. Το τρίτο διάγραμμα παρουσιάζει τον τρόπο με τον οποίο ο χρήστης δεσμεύει υπολογιστικούς από την συσκευή του και ενημερώνει το Cloudlet. Το τέταρτο διάγραμμα περίπτωσης χρήσης αναπαριστά την ανάθεση εργασιών από το Cloudlet σε μια κινητή συσκευή και την συλλογή των αποτελεσμάτων. Τέλος, το πέμπτο διάγραμμα παρουσιάζει την περίπτωση χρήσης: αναζήτηση βοήθειας για την χρήση της εφαρμογής.



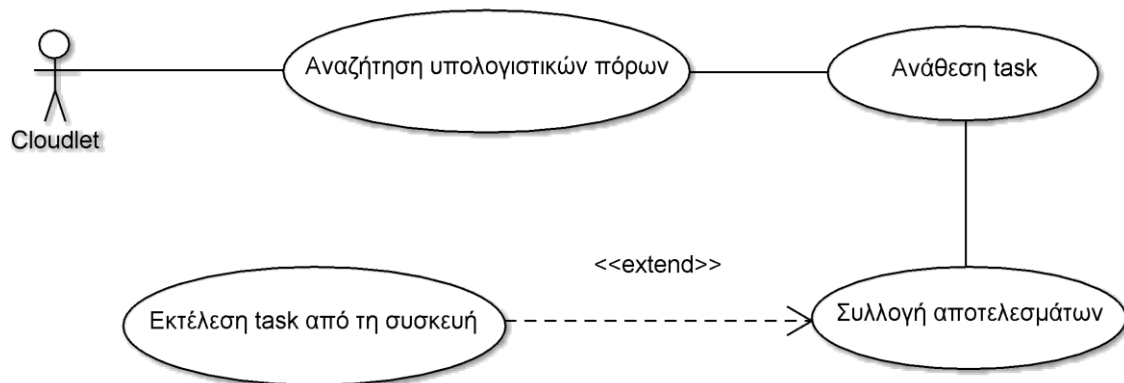
Διάγραμμα 3.1: Κύριο μενού επιλογών



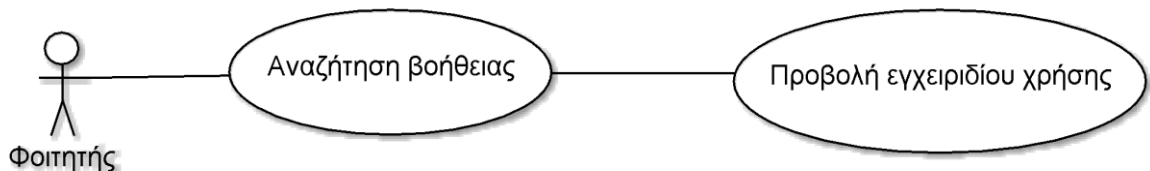
Διάγραμμα 3.2: Αναζήτηση πληροφοριών για τα μαθήματα



Διάγραμμα 3.3: Δέσμευση υπολογιστικών πόρων



Διάγραμμα 3.4: Ανάθεση task από το Cloudlet

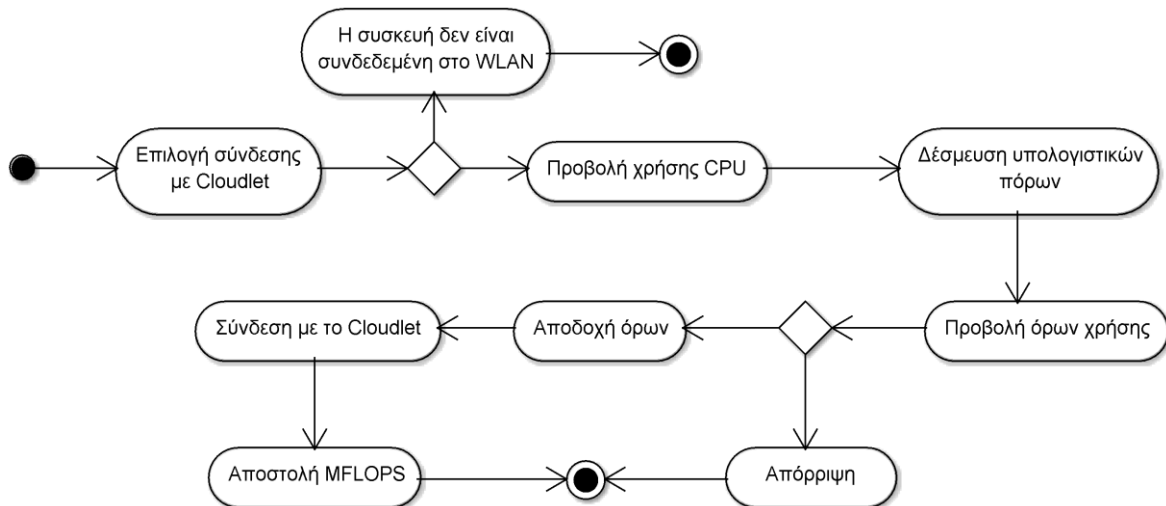


Διάγραμμα 3.5: Αναζήτηση Βοήθειας

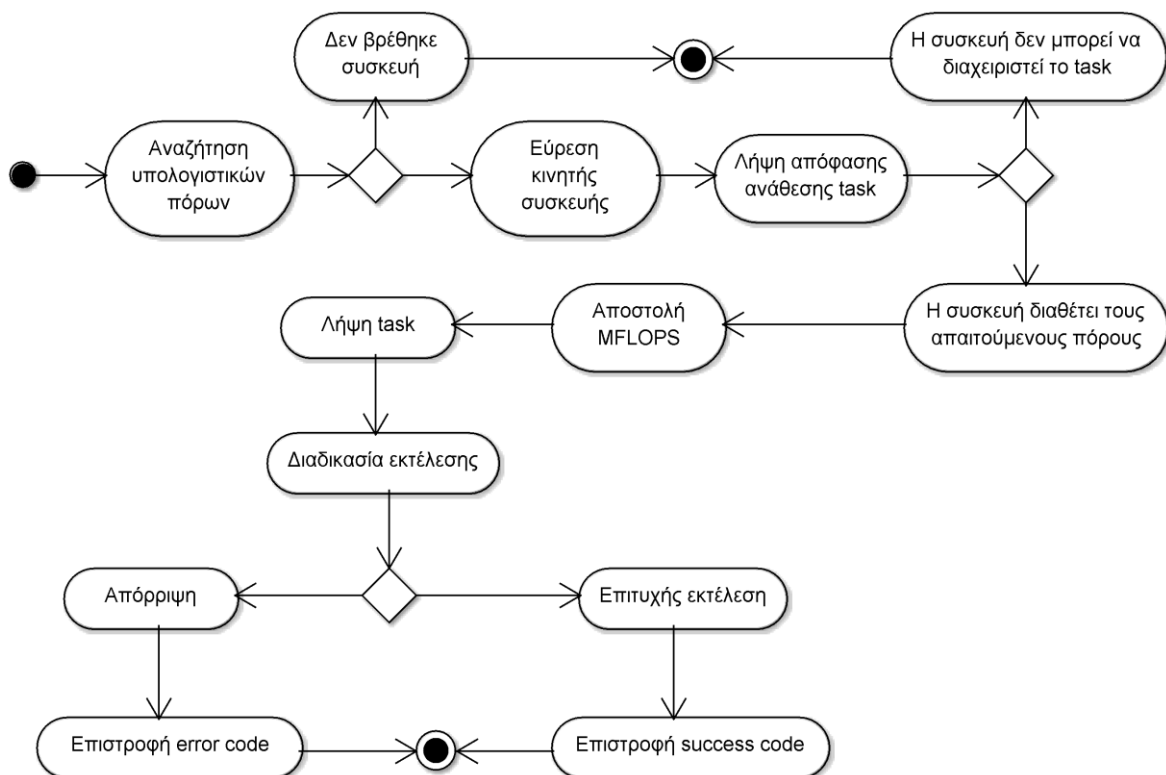
3.4.2 Διαγράμματα Δραστηριοτήτων

Σε αυτήν την ενότητα παρατίθενται δυο διαγράμματα δραστηριοτήτων για την εφαρμογή. Το πρώτο διάγραμμα παρουσιάζει αναλυτικά όλη τη ροή (flow) της διαδικασίας δέσμευσης υπολογιστικών πόρων από τον χρήστη της συσκευής, καθώς και της διαδικασίας ενημέρωσης του Cloudlet με το σύνολο των πόρων, που του παραχωρεί μια συσκευή. Το δεύτερο διάγραμμα δραστηριοτήτων παρουσιάζει συνολικά τη ροή (flow) από την πλευρά του Cloudlet, για τη διαδικασία αναζήτησης επιπλέον υπολογιστικών

πόρων σε κινητές συσκευές, οι οποίες επικοινωνούν μαζί του. Το Cloudlet αναθέτει εργασίες (tasks) σε κινητές συσκευές, που του προσφέρουν πόρους και οι οποίες παράλληλα έχουν τη δυνατότητα να διαχειριστούν τα tasks, που πρόκειται να τους αναθέσει. Η κινητή συσκευή από την πλευρά της εκτελεί ή απορρίπτει ένα task ανάλογα με τη διαθεσιμότητα των πόρων της και ενημερώνει κατάλληλα το Cloudlet για τα αποτελέσματα.



Διάγραμμα 3.6: Δέσμευση πόρων συσκευής και ενημέρωση του Cloudlet



Διάγραμμα 3.7: Αναζήτηση Πόρων σε συσκευές από το Cloudlet και ανάθεση tasks

3.4.3 Περίπτωση χρήσης: CPU Usage

Περιγραφή:

Σε αυτή την περίπτωση ο χρήστης της εφαρμογής πατά το κουμπί «*Cloudlet*» στην αρχική οθόνη. Έπειτα, του εμφανίζεται μια νέα οθόνη όπου στο επάνω αριστερά μέρος μπορεί να δει το ποσοστό της CPU της συσκευής του, το οποίο είναι δεσμευμένο εκείνη τη στιγμή από άλλες εφαρμογές και processes.

Παρουσίαση βήμα προς βήμα:

1. Ο χρήστης πατά το κουμπί «*Cloudlet*».
2. Στη νέα οθόνη βλέπει το ποσοστό επί τοις εκατό της CPU που είναι δεσμευμένη.

3.4.4 Περίπτωση χρήσης: Παραχώρηση πόρων στο Cloudlet

Περιγραφή:

Ο χρήστης αφού έχει πατήσει το κουμπί «*Cloudlet*» και δει την διαθεσιμότητα της CPU της συσκευής του, μπορεί να αποφασίσει, αν θα δώσει πόρους στο *Cloudlet*. Στην περίπτωση όπου, ο χρήστης αποφασίσει να συμμετέχει η συσκευή του στη *Cloudlet* υποδομή υπολογιστικού νέφους, πρέπει να επιλέξει το ποσοστό της CPU, το οποίο θα διαθέσει Αυτό γίνεται χρησιμοποιώντας την μπάρα προόδου που βρίσκεται στην ίδια οθόνη. Στην συνέχεια ο χρήστης πρέπει να πατήσει το κουμπί «*Σύνδεση*» της οθόνης.

Παρουσίαση βήμα προς βήμα:

1. Σε συνέχεια της περίπτωσης χρήστης CPU Usage ο χρήστης επιλέγει με χρήση της «*seek bar*» το ποσοστό από την διαθέσιμη CPU, το οποίο επιθυμεί να παραχωρήσει στο *Cloudlet*.
2. Πατά το κουμπί «*Σύνδεση*» προκειμένου να συνδεθεί και να ενημερώσει το *Cloudlet*.

3.4.5 Περίπτωση χρήσης: Ανάθεση εργασιών από το Cloudlet

Περιγραφή:

Σε αυτή την περίπτωση χρήσης αποκλειστικό ρόλο έχει το *Cloudlet*. Το *Cloudlet* περίπτωση που χρειαστεί επιπλέον πόρους, προκειμένου να εκτελέσει τα διάφορα tasks

που του έχουν ανατεθεί, τους αναζητά στις κινητές συσκευές, οι οποίες είναι συνδεδεμένες μαζί του και του έχουν παραχωρήσει ένα ποσοστό από την CPU τους. Το Cloudlet μοιράζει τα tasks σε συσκευές, οι οποίες μπορούν να τα εκτελέσουν. Το Cloudlet γνωρίζει τις απαιτήσεις των tasks σε επεξεργαστική ισχύ καθώς και τις δυνατότητες της κάθε συσκευής. Η συσκευή από την πλευρά της μπορεί να απορρίψει ένα task σε περίπτωση όπου, δε διαθέτει πλέον του απαιτούμενους πόρους.

Παρουσίαση βήμα προς βήμα:

1. Το Cloudlet αναζητά επιπλέον πόρους για την εκτέλεση εργασιών από τις κινητές συσκευές.
2. Το Cloudlet αναθέτει tasks στις κινητές συσκευές, ανάλογα με το πόρους που του έχουν παραχωρήσει και τις απαιτήσεις των tasks.
3. Εκτέλεση ή απόρριψη του task από την κινητή συσκευή.

3.4.6 Περίπτωση χρήσης: Αναζήτηση Μαθημάτων

Περιγραφή:

Ο φοιτητής μόλις ενεργοποιήσει την εφαρμογή του και συνδεθεί με το cloudlet μέσω WLAN, το σύστημα εντοπίζει αυτόματα την θέση του μέσα στο κτίριο και του παραθέτει μια λίστα με τα μαθήματα, που διεξάγονται στον όροφο, που βρίσκεται εκείνη τη στιγμή.

Παρουσίαση βήμα προς βήμα:

1. Ο χρήστης ενεργοποιεί το wi-fi της συσκευής του και συνδέεται με το WLAN στο οποίο βρίσκεται το cloudlet.
2. Στη συνέχεια ο χρήστης ενεργοποιεί την mobile εφαρμογή.
3. Πατάει στο κουμπί της αρχικής οθόνης «Αναζήτηση»
4. Το σύστημα εντοπίζει την θέση του και του επιστρέφει μια λίστα με τα μαθήματα, που γίνονται στον συγκεκριμένο όροφο.

3.4.7 Περίπτωση χρήσης: Αναζήτηση Αίθουσας Διδασκαλίας

Περιγραφή:

Στην περίπτωση αυτή και σε συνέχεια της προηγούμενης, ο χρήστης αφού λάβει τη λίστα με τα μαθήματα έχει τη δυνατότητα να επιλέξει το μάθημα που τον ενδιαφέρει και να ενημερωθεί για την αίθουσα, στην οποία αυτό διεξάγεται.

Παρουσίαση βήμα προς βήμα:

1. Ο χρήστης επιλέγει από τη λίστα το μάθημα, που τον ενδιαφέρει πατώντας επάνω του.
2. Η εφαρμογή στη συνέχεια του εμφανίζει τις δυο επιλογές: «Αίθουσα» και «Υλικό».
3. Ο χρήστης επιλέγει και πατάει στο κουμπί «Αίθουσα».
4. Η εφαρμογή εμφανίζει τον αριθμό της αίθουσας του μαθήματος.

3.4.8 Περίπτωση χρήσης: Αναζήτηση Υλικού Μαθήματος

Περιγραφή:

Ο φοιτητής σε αυτήν την περίπτωση, αφού λάβει την λίστα με τα μαθήματα, που γίνονται στον όροφο στον οποίο βρίσκεται, επιλέγει το μάθημα, το οποίο τον ενδιαφέρει προκειμένου να λάβει υλικό (όπως έγγραφα παρουσιάσεων) σχετικά με αυτό.

Παρουσίαση βήμα – βήμα:

1. Ο χρήστης επιλέγει το μάθημα, που τον ενδιαφέρει από τη λίστα που εμφανίζεται (use case 2.3.1)
2. Το σύστημα του εμφανίζει όπως και προηγουμένως τις δυο επιλογές: «Αίθουσα» και «Υλικό».
3. Ο χρήστης επιλέγει και πατάει στο κουμπί «Υλικό».
4. Το σύστημα σε νέα οθόνη εμφανίζει τις εξής τρεις επιλογές: 1) «Σημειώσεις», 2) «Εργασίες» και 3) «Παρουσιάσεις.»
5. Ο χρήστης στη συνέχεια επιλέγει τι υλικό θέλει να κατεβάσει στην συσκευή του, πατώντας στο αντίστοιχο κουμπί.

3.4.9 Περίπτωση χρήσης: Βοήθεια

Περιγραφή:

Σε αυτό το σενάριο χρήσης του συστήματος ο φοιτητής αφού ανοίξει την εφαρμογή, επιλέγει το κουμπί βοήθειας από την αρχική οθόνη. Αμέσως, του εμφανίζεται ένα πλαίσιο κειμένου με μια αναλυτική περιγραφή της λειτουργίας της εφαρμογής.

Παρουσίαση βήμα – βήμα:

1. Άνοιγμα mobile εφαρμογής.
2. Επιλογή κουμπιού «Βοήθεια».
3. Εμφάνιση πλαισίου κειμένου με την περιγραφή της εφαρμογής.

3.5 Λειτουργικές απαιτήσεις

3.5.1 CPU Usage

Κωδικός Περίπτωσης:	3.4.3.
Ονομασία:	CPU Usage

Εμπλεκόμενοι Ρόλοι:	Φοιτητής
Περιγραφή:	Σε αυτή την περίπτωση χρήσης ο φοιτητής μπορεί να δει το ποσοστό της CPU της κινητής του συσκευής, το οποίο χρησιμοποιείται από τις εφαρμογές και τα διάφορα processes που τρέχουν σε αυτή.
Γεγονός Εκκίνησης:	Πάτημα του κουμπιού «Cloudlet» από την βασική οθόνη της εφαρμογής.
Προϋποθέσεις:	Η κινητή συσκευή του φοιτητή θα πρέπει να είναι συνδεδεμένη στο ασύρματο δίκτυο.
Τελική Κατάσταση:	Ο φοιτητής βλέπει στην οθόνη του το ποσοστό επί τοις εκατό της CPU που χρησιμοποιείται.
Φυσιολογική Ροή:	Ο χρήστης ανοίγει την εφαρμογή και πατά το κουμπί Cloudlet. Σε νέα οθόνη βλέπει το ποσοστό της δεσμευμένης CPU
Εξαιρέσεις:	Δεν υπάρχουν εξαιρέσεις για αυτήν την περίπτωση χρήσης.
Προτεραιότητα:	Υψηλής, είναι μια από τις περιπτώσεις χρήσης που εκτελείται πάντα πρώτη.

3.5.2 Παραχώρηση πόρων στο Cloudlet

Κωδικός Περίπτωσης:	3.4.4
Ονομασία:	Παραχώρηση πόρων στο Cloudlet

Εμπλεκόμενοι Ρόλοι:	Φοιτητής
Περιγραφή:	Ο χρήστης της εφαρμογής αφού έχει ενημερωθεί για την

	διαθεσιμότητα της CPU της συσκευής του, αποφασίζει πως ένα μέρος από αυτή θα παραχωρηθεί για την εκτέλεση εργασιών από το Cloudlet.
Γεγονός Εκκίνησης:	Ο χρήστης επιλέγει τι ποσοστό από την διαθέσιμη CPU θα παραχωρήσει στο Cloudlet με χρήση της μπάρας προόδου και πατά το κουμπί «σύνδεση» από την οθόνη σύνδεσης με το Cloudlet.
Προϋποθέσεις:	<ol style="list-style-type: none"> 1. Η κινητή συσκευή πρέπει να είναι συνδεδεμένη στο ασύρματο δίκτυο 2. Ο χρήστης πρέπει να επιλέξει ο ίδιος το ποσοστό της CPU που θα προσφέρει 3. Ο χρήστης επίσης πρέπει, να αποδεχθεί ότι ένα μέρος της CPU της συσκευής δεν θα μπορεί να χρησιμοποιηθεί από άλλες εφαρμογές, που εκτελούνται τοπικά σε αυτή
Τελική Κατάσταση:	Η εφαρμογή συνδέεται με το server του Cloudlet και τον ενημερώνει για τους πόρους, που προσφέρει η συσκευή, σε περίπτωση που το Cloudlet τους χρειαστεί.
Φυσιολογική Ροή:	Ο χρήστης επιλέγει από την οθόνη σύνδεσης το ποσοστό της CPU, πατά το κουμπί σύνδεση και στη συνέχεια αποδέχεται τις όρους, που του εμφανίζονται στο alert μήνυμα στην ίδια οθόνη. Στην συνέχεια ανοίγει μια νέα οθόνη στην οποία ενημερώνεται για το εάν έχει ανατεθεί στην συσκευή του κάποιο task.
Εξαιρέσεις:	<ol style="list-style-type: none"> 1. Ο χρήστης σε περίπτωση που δεν επιλέξει ένα ποσοστό από την μπάρα προόδου και πατήσει το κουμπί σύνδεση, τότε εμφανίζεται ένα ενημερωτικό μήνυμα που τον προτρέπει να επιλέξει ένα ποσοστό. 2. Ο χρήστης σε περίπτωση που πατήσει το κουμπί σύνδεση και δεν αποδεχτεί του όρους, τότε επιστρέφει αυτόματα στην αρχική οθόνη του κύριου μενού 3. Σε περίπτωση σφάλματος στην σύνδεση η επικοινωνία διακόπτεται και ο χρήστης επιστρέφει στην προηγούμενη οθόνη.
Ενσωματώνει:	Την περίπτωση χρήσης 2.3.4
Προτεραιότητα:	Εκτελείται πάντα μετά την περίπτωση χρήσης 2.3.4

3.5.3 Ανάθεση task από το Cloudlet

Κωδικός Περίπτωσης:	3.4.5
Ονομασία:	Ανάθεση task από το Cloudlet

Εμπλεκόμενοι Ρόλοι:	Cloudlet
Περιγραφή:	Το Cloudlet ενημερώνεται από τις ίδιες τις συσκευές που είναι συνδεδεμένες στο ίδιο δίκτυο με αυτό για τους πόρους επεξεργαστικής ισχύος που του παραχωρούν. Το Cloudlet όταν χρειαστεί επιπλέον πόρους για την εκτέλεση εργασιών θα τους αναζητήσει στις συσκευές που συνδράμουν σε αυτό. Το Cloudlet γνωρίζει τις απαιτήσεις σε επεξεργαστική ισχύ των tasks που έχει, καθώς και τις δυνατότητες των συνδεδεμένων συσκευών. Έτσι, μπορεί να προχωρήσει αποδοτικότερο διαμοιρασμό των εργασιών.
Γεγονός Εκκίνησης:	Το Cloudlet χρειάζεται επιπλέον για τη εκτέλεση ενός task, τους

	οποίους τους βρίσκει σε μια κινητή συσκευή που συνδράμει σε αυτό.
Προϋποθέσεις:	<ol style="list-style-type: none"> 1. Το Cloudlet αναθέτει tasks σε συσκευές που είναι συνδεδεμένες στο ίδιο τοπικό δίκτυο με αυτό και που το έχουν ενημερώσει ότι, συνδράμουν στην υποδομή. 2. Το Cloudlet πρέπει να αναθέτει σε μια συσκευή ένα μόνο task, το οποίο αυτή να μπορεί να το διαχειριστεί. Αυτό εξαρτάται από τον αριθμό των MFLOPS, που στέλνει σε αυτό.
Τελική Κατάσταση:	<ol style="list-style-type: none"> 1. Το task το οποίο ανατίθεται σε μια συσκευή εκτελείται επιτυχώς και τα αποτελέσματα επιστρέφουν στο Cloudlet, 2. Το task απορρίπτεται από την εφαρμογή λόγω έλλειψης πόρων από αυτή και επιστρέφεται στο Cloudlet ένα error code.
Φυσιολογική Ροή:	Το Cloudlet αναθέτει σε μια συσκευή, που συνδράμει σε αυτό, ένα task, το οποίο να μπορεί εκτελέσει σε αυτή, χωρίς να απαιτείται η χρήση επιπλέον πόρων CPU, πέραν από αυτούς που έχει επιλέξει να παραχωρήσει ο χρήστης..
Εξαιρέσεις:	<ol style="list-style-type: none"> 1. Σε περίπτωση όπου καμία κινητή συσκευή δεν συνδράμει στην Cloudlet-based υποδομή, τότε όλα τα tasks θα μουν στην ουρά και θα εκτελεστούν στο Cloudlet. 2. Σε περίπτωση όπου η σύνδεση μεταξύ τις κινητής συσκευής και του Cloudlet διακοπή, η εκτέλεση του task αποτυγχάνει και Cloudlet πρέπει να αναζητήσει πόρους σε άλλη συσκευή.
Ενσωματώνει:	Την περίπτωση χρήσης 2.3.4
Προτεραιότητα:	Εκτελείται μετά την περίπτωση χρήσης 2.3.5

3.5.4 Βοήθεια

Κωδικός Περίπτωσης:	3.4.9
Ονομασία:	Βοήθεια

Εμπλεκόμενοι Ρόλοι:	Φοιτητής
Περιγραφή:	Σε αυτή την περίπτωση ο χρήστης αναζητά ένα εγχειρίδιο χρήσης της εφαρμογής.
Γεγονός Εκκίνησης:	Πάτημα του κουμπιού «Βοήθεια» της αρχικής οθόνης.
Προϋποθέσεις:	Δεν υπάρχουν προϋποθέσεις.
Τελική Κατάσταση:	Ο χρήστης βλέπει μια έναν αναλυτικό οδηγό χρήσης της εφαρμογής.
Φυσιολογική Ροή:	<ol style="list-style-type: none"> 1. Άνοιγμα mobile εφαρμογής. 2. Επιλογή κουμπιού βοήθεια. 3. Εμφάνιση πλαισίου κειμένου με την περιγραφή της εφαρμογής.
Προτεραιότητα:	Χαμηλή.

3.5.5 Αναζήτηση Μαθημάτων

Κωδικός Περίπτωσης:	3.4.6.
Ονομασία:	Αναζήτηση Μαθημάτων

Εμπλεκόμενοι Ρόλοι:	Φοιτητής
Περιγραφή:	Ο χρήστης χρησιμοποιεί την mobile εφαρμογή του, προκειμένου να ενημερωθεί για τα μαθήματα, που διεξάγονται στον όροφο στον οποίο βρίσκεται.
Γεγονός Εκκίνησης:	Επιλογή του κουμπιού «Αναζήτηση» της αρχικής οθόνης της εφαρμογής.
Προϋποθέσεις:	Η κινητή συσκευή του φοιτητή πρέπει να είναι συνδεδεμένη μέσω wi-fi με το τοπικό ασύρματα δίκτυο του πανεπιστημίου, ώστε να επικοινωνεί κάποιο AP.
Τελική Κατάσταση:	Ο χρήστης ενημερώνεται για τα μαθήματα, που λαμβάνουν χώρα στον όροφο τον οποίο κινείται.
Φυσιολογική Ροή:	<ol style="list-style-type: none"> 1. Ο χρήστης ενεργοποιεί το wi-fi στην συσκευή του και συνδέεται στο τοπικό δίκτυο με το cloudlet. 2. Ο χρήστης ανοίγει την mobile εφαρμογή προκειμένου να χρησιμοποιήσει την LBS υπηρεσία που του παρέχει. 3. Ο χρήστης πατάει στο κουμπί αναζήτηση από το κύριο μενού. 4. Το σύστημα εντοπίζει τη θέση του με βάση το BSSID, δηλαδή την φυσική διεύθυνση της κάρτα δικτύου του Access Point με το οποίο έχει συνδεθεί η συσκευή. 5. Το σύστημα έπειτα σε νέα οθόνη επιστρέφει την λίστα με τα μαθήματα του προγράμματος σπουδών του φοιτητή, που διεξάγονται στον συγκεκριμένο όροφο.
Εξαιρέσεις:	<ol style="list-style-type: none"> 1. Σε περίπτωση όπου ο χρήστης ανοίξει την εφαρμογή και πατήσει το κουμπί αναζήτηση χωρίς να έχει συνδεθεί στο δίκτυο, θα του εμφανίζεται ένα alert μήνυμα στην οθόνη το οποίο θα τον ενημερώνει σχετικά με τις ενέργειες τις οποίες θα πρέπει να ακολουθήσει. 2. Επίσης, σε περίπτωση όπου ο φοιτητής βρίσκεται σε κάποιον όροφο στον οποίο δε γίνονται μαθήματα του προγράμματος του, η εφαρμογή τον ενημερώνει με σχετικό μήνυμα.
Προτεραιότητα:	Υψηλής, είναι μια από τις περιπτώσεις χρήσης που εκτελείται πάντα πρώτη.

3.5.6 Αναζήτηση Αίθουσας Διδασκαλίας

Κωδικός Περίπτωσης:	3.4.7
Ονομασία:	Αναζήτηση Αίθουσας Διδασκαλίας Μαθήματος
Εμπλεκόμενοι Ρόλοι:	Φοιτητής
Περιγραφή:	Ο χρήστης σε αυτή την περίπτωση χρησιμοποιεί την εφαρμογή και την LBS υπηρεσία, προκειμένου να ενημερωθεί για τις αίθουσες, στις οποίες διεξάγονται τα μαθήματα του.
Γεγονός Εκκίνησης:	Επιλογή μαθήματος από την λίστα.
Προϋποθέσεις:	Η κινητή συσκευή του φοιτητή πρέπει να είναι συνδεδεμένη μέσω wi-fi με το τοπικό δίκτυο του πανεπιστημίου. Να έχει εκτελεστεί η περίπτωση χρήσης 2.3.1.
Τελική Κατάσταση:	Ο χρήστης ενημερώνεται για την αίθουσα στην οποία διδάσκεται το μάθημα της επιλογής του σε νέα οθόνη.
Φυσιολογική Ροή:	<ol style="list-style-type: none"> 1. Ο χρήστης επιλέγει από την λίστα το μάθημα που τον ενδιαφέρει (use case 2.3.1)

	<ol style="list-style-type: none"> 2. Ο χρήστης στην επόμενη οθόνη πατάει στην επιλογή «Αίθουσα» 3. Η εφαρμογή με ένα pop-up μήνυμα εμφανίζει τον αριθμό της αίθουσας διδασκαλίας του μαθήματος.
Εξαιρέσεις:	Δεν υπάρχουν εξαιρέσεις για αυτή την περίπτωση χρήσης.
Ενσωματώνει:	Την περίπτωση χρήσης 2.3.1.
Προτεραιότητα:	Εκτελείται πάντα μετά την περίπτωση χρήσης 2.3.1.

3.5.7 Αναζήτηση Υλικού Μαθημάτων

Κωδικός Περίπτωσης:	3.4.8
Ονομασία:	Αναζήτηση Υλικού Μαθήματος

Εμπλεκόμενοι Ρόλοι:	Φοιτητής
Περιγραφή:	Σε αυτή την περίπτωση ο φοιτητής αναζητά υλικό σχετικό με το μάθημα που θέλει παρακολουθήσει.
Γεγονός Εκκίνησης:	Επιλογή μαθήματος από την λίστα.
Προϋποθέσεις:	Η κινητή συσκευή του χρήστη πρέπει να είναι συνδεδεμένη μέσω wi-fi με το τοπικό δίκτυο του πανεπιστημίου και να επικοινωνεί με το cloudlet. Να έχει εκτελεστεί η περίπτωση χρήσης 2.3.1.
Τελική Κατάσταση:	Ο χρήστης βλέπει τις επιλογές με το διαθέσιμο υλικό του μαθήματος.
Φυσιολογική Ροή:	<ol style="list-style-type: none"> 1. Ο χρήστης επιλέγει από την λίστα το μάθημα που τον ενδιαφέρει 2. Στην επόμενη οθόνη ο χρήστης επιλέγει «Υλικό» 3. Το σύστημα σε νέα οθόνη εμφανίζει τις επιλογές με το διαθέσιμο υλικό του μαθήματος.
Εξαιρέσεις:	Δεν υπάρχουν εξαιρέσεις για αυτή την περίπτωση χρήσης.
Ενσωματώνει:	Την περίπτωση χρήσης 2.3.1.
Προτεραιότητα:	Εκτελείται πάντα μετά την περίπτωση χρήσης 2.3.1.

3.6 Μη Λειτουργικές απαιτήσεις

3.6.1 Περιβάλλον λειτουργίας

Η mobile εφαρμογή, που αναπτύχθηκε για τους σκοπούς της διπλωματικής διατριβής, προορίζεται για έξυπνες κινητές συσκευές με λειτουργικό σύστημα Android. Οι εκδόσεις του λειτουργικού συστήματος στις οποίες μπορεί να εγκατασταθεί η εφαρμογή είναι από 4.0 έως και 4.4.4. Η εφαρμογή επικοινωνεί μια Cloudlet-Based υποδομή υπολογιστικού νέφους, η οποία υλοποιήθηκε με χρήση του εργαλείου προσημείωσης CloudSim, το οποίο όπως περιγράφεται παρακάτω αναπτύχθηκε με στόχο την υλοποίηση virtual cloud υποδομών. Για την εκτέλεση και τον έλεγχο της εφαρμογής, καθώς και για την

πραγματοποίηση των πειραμάτων, χρησιμοποιήθηκε ο Android Emulator. Ο Android Emulator αποτελεί μια εικονική συσκευή, η οποία συνθέτει τα χαρακτηριστικά μιας πραγματικής συσκευής. Ο Android Emulator συμπεριλαμβάνεται στο Android SDK και αυτόν τον τρόπο δίνεται η δυνατότητα στους προγραμματιστές, να αναπτύσσουν και να ελέγχουν τις εφαρμογές χωρίς απαραίτητα την χρήση μιας κανονικής συσκευής. Όλα τα παραπάνω αποτελούν το περιβάλλον λειτουργίας της εφαρμογής.

Η επικοινωνία της mobile εφαρμογής με την Cloudlet υποδομή στο CloudSim [27] πραγματοποιήθηκε με την χρήση client/server java sockets. Το socket δημιουργία μια αμφίδρομη επικοινωνία μεταξύ δυο προγραμμάτων στο ίδιο δίκτυο με στόχο την ανταλλαγή δεδομένων.

3.6.2 Περιορισμοί στη σχεδίαση και την υλοποίηση

1. Η εφαρμογή, μπορεί να εκτελεστεί αποκλειστικά και μόνο στην πλατφόρμα του λειτουργικού συστήματος Android για εκδόσεις από 4.0 έως και 4.4.2. Οι υπόλοιπες πλατφόρμες mobile λειτουργικών συστημάτων, όπως iOS και Windows Phone δε υποστηρίζονται.
2. Η γλώσσα προγραμματισμού, που χρησιμοποιήθηκε, είναι Java Standard Edition (JSE) v.6⁴. Η συγκεκριμένη έκδοση είναι η καταλληλότερη (και προτεινόμενη) τόσο για την ανάπτυξη Android εφαρμογών, όσο και για εφαρμογών με χρήση του CloudSim.
3. Λόγω έλλειψης μιας android κινητής συσκευής η εφαρμογή εκτελέστηκε και δοκιμάστηκε με χρήση του android emulator.
4. Η Cloudlet υποδομή υλοποιήθηκε με χρήση εργαλείου προσομοίωσης, οπότε δεν ήταν εφικτό πειραματικό μέρος της εργασίας να δοκιμαστεί σε ένα πραγματικό περιβάλλον υπολογιστικού νέφους και συλλεχθούν τα ανάλογα αποτελέσματα.
5. Δεν κατέστη δυνατό να δοκιμαστεί η LBS υπηρεσία σε πραγματικά δεδομένα. Οι Mac διευθύνσεις, που χρησιμοποιούνται στον κώδικά είναι τυχαίες. Για να λειτουργήσει όπως σωστά η LBS υπηρεσία οι Mac διευθύνσεις θα πρέπει να

⁴ <http://www.oracle.com/technetwork/java/javase/overview/index.html>

αντικατασταθούν με τις αντίστοιχες πραγματικές των AP του ασύρματου δικτύου του πανεπιστημίου.

3.6.3 Προϋποθέσεις - Απαιτήσεις

- Η εφαρμογή να είναι απλή και φιλική προς τον χρήστη.
- Να υπάρχει μενού Βοήθειας
- Σύνδεση τις κινητής συσκευής με την cloudlet υποδομή μέσω WLAN.
- Αυτόματος εντοπισμός της θέσης του χρήστη εντός του κτιρίου, ανάλογα με το Access Point του δίκτυο με το οποίο επικοινωνεί.
- Η εφαρμογή να χρησιμοποιείται ελεύθερα από τους φοιτητές ή/και τους καθηγητές της τμήματος πληροφορικής.
- Να απαιτείται η εισαγωγή έγκυρου κωδικού πρόσβασης για την περίπτωση όπου, ο χρήστης επιθυμεί να λάβει υλικό για κάποιο μάθημα.
- Χρήση εργαλείων ελεύθερου λογισμικού.
- Ο χρήστης να μπορεί να δει το ακριβές ποσοστό της CPU της συσκευής του που είναι δεσμευμένο.
- Ο χρήστης να μπορεί να επιλέξει από μόνος το ποσοστό της CPU της συσκευής του, που θα παραχωρήσει στο Cloudlet.
- Η εφαρμογή να απορρίπτει ένα task, το οποίο η συσκευή δεν μπορεί πλέον να το διαχειριστεί έλλειψης πόρων, χωρίς αυτό να αποτελεί σημείο κατάρρευσης του συστήματος.
- Η εφαρμογή να μην απαιτεί πρόσβαση στο δίκτυο με χρήση 3G/4G συνδέσεων, αλλά μόνο Wi-Fi.

3.7 Απαιτήσεις για τις εξωτερικές διεπαφές

3.7.1 Διεπαφή χρήστη

Ο χρήστης να έχει την δυνατότητα να αλληλεπιδρά εύκολα με την εφαρμογή, μέσω ενός απλού γραφικού περιβάλλοντος (GUI).

Για τον παραπάνω σκοπό έχουν υλοποιηθεί τα κατάλληλα alert μηνύματα, τα οποία ενημερώνουν το χρήστη για τις επιλογές του και τον καθοδηγούν. Μηνύματα alert έχουν υλοποιηθεί για τις παρακάτω περιπτώσεις:

- Όταν ο χρήστης πατά το κουμπί «Αναζήτηση» ή «Cloudlet» και δεν έχει συνδεθεί στο Wi-δίκτυο.
- Στην περίπτωση όπου ο χρήστης βρίσκεται σε όροφος στον οποίο δεν διεξάγονται μαθήματα του Μεταπτυχιακού του Προγράμματος σπουδών.
- Αν ο χρήστης εισάγει λάθος κωδικό
- Στην περίπτωση όπου ο χρήστης πατήσει το κουμπί «Σύνδεση» προκειμένου να ενημερώσει το Cloudlet για τους πόρους επεξεργαστικής ισχύος που θα του προσφέρει η συσκευή του.
- Επίσης, στην περίπτωση όπου ο χρήστης επιλέξει το κουμπί «Σύνδεση» χωρίς να έχει επιλέξει τι ποσοστό της CPU θα παραχωρήσει στο Cloudlet.

Επιπλέον, έχουν υλοποιηθεί pop-up μηνύματα τα οποία εμφανίζονται στην οθόνη της συσκευής προκειμένου να δώσουν ένα άμεσο και σύντομο feedback στον χρήστη για τις επιλογές και τις ενέργειες του. Με αυτόν αποφεύγονται περιττά μηνύματα alert και εναλλαγές σε οθόνες. Pop-up μηνύματα έχουν δημιουργηθεί για τις εξής περιπτώσεις:

- Για ενημερωθεί ο φοιτητής για τον αριθμό της αίθουσας που διεξάγεται ένα μάθημα.
- Στην περίπτωση όπου δεν υπάρχει διαθέσιμο υλικό για το μάθημα που έχει επιλέξει ο φοιτητής.
- Για την ενημέρωση του χρήστη για το ένα task οποίο ανατέθηκε στην συσκευή εκτελέστηκε επιτυχώς ή όχι.

Σε κάθε οθόνη της εφαρμογή υπάρχουν σύντομες περιγραφές σχετικά με το τι βλέπει ο χρήστης, καθώς και για το ποία πρέπει να είναι η επόμενη ενέργεια του. Ο χρήστης μπορεί να πλοηγηθεί προς τα πίσω χρησιμοποιώντας το Back Button της Android συσκευής. Τέλος, για να «βγει» από την εφαρμογή θα πρέπει, να πατήσει το κεντρικό κουμπί του Κύριο Μενού της συσκευής του.

3.7.2 Διεπαφές υλικού

Η εφαρμογή χρησιμοποιεί τον αισθητήρα του Wi-Fi της κινητής συσκευής, προκειμένου να συνδεθεί στο τοπικό δίκτυο του πανεπιστημίου και να επικοινωνήσει με την cloudlet υποδομή.

Η εφαρμογή επίσης, για να εντοπίσει την θέση του χρήστη μέσα στο κτίριο, χρησιμοποιεί την Mac διεύθυνση του Access Point του τοπικού δικτύου, στο οποίο είναι συνδεδεμένη η συσκευή. Για τις ανάγκες αυτής της διπλωματικής εργασίας θεωρούμε ότι: Οι φυσικές διευθύνσεις (Mac) των AP μας είναι γνωστές.

3.7.3 Διεπαφές Λογισμικού

Για την υλοποίηση της εφαρμογής είναι απαραίτητη η χρήση της εφαρμογής Linpack for Android. Το Linpack for Android⁵ είναι μια Benchmark εφαρμογή, η οποία μετρά την πραγματική απόδοση της CPU μιας Android κινητής συσκευής σε MFLOPS.

3.7.4 Διεπαφές επικοινωνιών

Η mobile εφαρμογή επικοινωνεί με την cloudlet υποδομή με χρήση της στοίβας πρωτοκόλλων TCP/IP. Για τον σκοπό αυτό υλοποιήθηκε μια client/server σύνδεση μεταξύ των δυο συστημάτων με χρήση java sockets. [30]

3.8 Τεχνολογίες που Χρησιμοποιήθηκαν

3.8.1 Android OS

Το Android είναι ένα λειτουργικό σύστημα, το οποίο σχεδιάστηκε για έξυπνες κινητές συσκευές, οι οποίες διαθέτουν οθόνη αφής και τρέχουν τον πυρήνα του λειτουργικού συστήματος Linux. Το Android είναι mobile λειτουργικό σύστημα ανοικτού κώδικα και διανέμεται με ελεύθερη άδεια. Η Google δημοσιεύει το μεγαλύτερο τμήμα του πηγαίου κώδικα υπό τους όρους της Apache License 2, καθώς επίσης και τις αλλαγές στον πυρήνα του Linux υπό τους όρους της GNU (General Public License) 2. Ένα μεγάλο τμήμα του πηγαίου κώδικα του Android αναπτύσσεται αποκλειστικά από την Google και δημοσιοποιείται με κάθε νέα έκδοση του που διατίθεται στο κοινό. Επιπλέον, το Android OS υποστηρίζεται μια ενεργή κοινότητα προγραμματιστών, οι οποίοι χρησιμοποιώντας

⁵ <https://play.google.com/store/apps/details?id=rs.pedjaapps.Linpack>

τον πηγαίο κώδικα του Android Open Source Project⁶ (AOSP) αναπτύσσουν, παραμετροποιούν και διανέμουν τις δικές τους τροποποιημένες εκδόσεις.

Η ανάπτυξη των εφαρμογών, οι οποίες επεκτείνουν τις δυνατότητες και την λειτουργικότητα των Android συσκευών, γίνεται σε γλώσσα προγραμματισμού Java (SE) χρησιμοποιώντας το Android SDK. Όλες οι εκδόσεις του Android SDK περιλαμβάνουν μια πλήρη σειρά εργαλείων ανάπτυξης, όπως πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες λογισμικού, εξομοιωτή συσκευής, δείγματα κώδικα και διάφορα εγχειρίδια. Επίσης, για ανάπτυξη εφαρμογών σε κώδικα C ή C++ μπορεί να χρησιμοποιηθεί το Android NDK, ενώ πλέον είναι δυνατή και η χρήση HTML5 και Javascript για ανάπτυξη web εφαρμογών τις Android συσκευές. [21][22]

3.8.2 Java

Η Java είναι μια αντικειμενοστραφείς γλώσσα προγραμματισμού, η οποία σχεδιάστηκε από την εταιρία Sun Microsystems. Η επίσημη παρουσίαση της Java από την Sun στη βιομηχανία της πληροφορικής, έγινε το 1995. Η Java διανέμεται δωρεάν, όπως επίσης και όλα τα εργαλεία που απαιτούνται για ανάπτυξη εφαρμογών με αυτή, όπως η JVM, IDEs, web servers και application servers, καθώς και διάφορες προγραμματιστικές βιβλιοθήκες. Πλέον, η Java εξελίσσεται και διανέμεται από την Oracle.

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων προγραμματιστικών γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Οι εφαρμογές, οι οποίες είναι γραμμένες σε Java, εκτελούνται το ίδιο στις διάφορες πλατφόρμες λειτουργικών συστημάτων, όπως Windows, Linux, MacOS, καθώς και σε κονσόλες ηλεκτρονικών παιχνιδιών. Η Java επιπλέον αποτελεί την βασικότερη γλώσσα προγραμματισμού ανάπτυξης εφαρμογών για την πλατφόρμα του λειτουργικού συστήματος Android. [20][24]

3.8.3 CloudSim

Το CloudSim είναι προγραμματιστικό εργαλείο, το οποίο παρέχει στους προγραμματιστές ένα πλαίσιο για την ανάπτυξη, μοντελοποίηση και προσημείωση Cloud υποδομών και υπηρεσιών. Το CloudSim δίνει τη δυνατότητα στους προγραμματιστές να ελέγξουν σε ένα ελεγχόμενο και χωρίς κόστος εικονικό περιβάλλον τις υπηρεσίες τους, πριν αυτές

⁶ <https://source.android.com/source/downloading.html>

εγκατασταθούν και λειτουργήσουν σε μια πραγματική Cloud υποδομή. Το CloudSim παρέχει στους προγραμματιστές κλάσεις για την υλοποίηση data centers, virtual machines, hosts, brokers, εφαρμογών, χρηστών, υπολογιστικών πόρων, για την δημιουργία πολιτικών διαχείρισης πόρων και συστημάτων κ.ά. Με αυτόν τον τρόπο μπορούν να πραγματοποιηθούν και να ελεγχθούν διάφορα σενάρια προσομοίωσης της λειτουργίας των υπηρεσιών για το υπολογιστικό νέφος, μέχρις ότου τα αποτελέσματα των πειραμάτων να καλύπτουν τις προσδοκώμενες απαιτήσεις. Το CloudSim είναι επίσης, ένα εργαλείο το οποίο διανέμεται δωρεάν. Η γλώσσα που χρησιμοποιεί για την ανάπτυξη εφαρμογών για σενάρια προσομοίωσης Cloud υποδομών είναι ή Java. [14]

3.8.4 Eclipse IDE

Το Eclipse αποτελεί ένα από τα δημοφιλέστερα και πιο ολοκληρωμένα περιβάλλοντα ανάπτυξης εφαρμογών οι δυνατότητες επεκτείνονται εύκολα με την ενσωμάτωση στο βασικό του workspace διαφόρων plug-ins. Το Eclipse είναι γραμμένο σε Java και στην αρχή η χρήση του προοριζόταν για την ανάπτυξη Java εφαρμογών. Πλέον, με το Eclipse μπορούν υλοποιηθούν εφαρμογές σε διάφορες άλλες γλώσσες, όπως C, C++, PHP, Ruby κ.ά. [25]. Επίσης, με το Eclipse οι προγραμματιστές μπορούν να αναπτύξουν mobile εφαρμογές για το λειτουργικό σύστημα Android. Το μόνο που απαιτείται για την ανάπτυξη Android εφαρμογών με το Eclipse είναι η εγκατάσταση του Android SDK και η ενσωμάτωση του ADT Plug-in στο Eclipse. Επίσης, με το Eclipse μπορούν να υλοποιηθούν και εφαρμογές για το CloudSim. Το CloudSim παρέχει στους προγραμματιστές, κλάσεις, βιβλιοθήκες και παραδείγματα για τη μοντελοποίηση και προσομοίωση Cloud υποδομών και υπηρεσιών, αλλά δεν παρέχει ένα γραφικό και ολοκληρωμένο περιβάλλον ανάπτυξης. Σε αυτή την περίπτωση, χρειάζεται απλά το CloudSim να γίνει import ως project μέσα στο Eclipse. [22]

3.8.5 Linpack Benchmark

Το Linpack Benchmark είναι ένα μέτρο υπολογισμού του ρυθμού εκτέλεσης εντολών κινητής υποδιαστολής από ένα υπολογιστικό σύστημα. Το Linpack χρησιμοποιείται για τον υπολογισμό της πραγματικής μέγιστης απόδοσης ενός υπολογιστικού συστήματος. Στα περισσότερα συστήματα η απόδοση που αναφέρεται από τους κατασκευαστές αντιστοιχεί στην θεωρητική απόδοση του συστήματος. Η θεωρητική απόδοση ενός επεξεργαστή προκύπτει από το πλήθος των εντολών κινητής υποδιαστολής (FLOPS), που μπορεί να εκτελέσει σε έναν κύκλο ρολογιού επί την ταχύτητα του σε HZ. Για παράδειγμα ένας

μονοπύρηνος επεξεργαστής στα 2 GHz, που εκτελεί 4 FLOPS σε κάθε κύκλο έχει θεωρητική απόδοση: $2 * 4 = 8$ GFLOPS. Τα αποτελέσματα της πραγματικής απόδοσης με χρήση του Linpack Benchmark προκύπτουν από την εκτέλεση ενός προγράμματος, που λύνει ένα σύνθετο σύστημα γραμμικών εξισώσεων. Η απόδοση του επεξεργαστή υπολογίζεται σε MFLOPS. Σε αυτήν την εργασία χρησιμοποιήθηκε το Linpack for Android, το οποίο εγκαταστάθηκε σε μια Android συσκευή προκειμένου να υπολογιστούν τα MFLOPS του επεξεργαστή της. [29]

3.9 Ανάλυση Κώδικα

Σε αυτήν την ενότητα παρουσιάζονται και αναλύονται τα κυριότερα κομμάτια κώδικα, τα οποία συντέλεσαν σημαντικά στην ανάπτυξη της εφαρμογής και αξίζει να σημειωθούν.

```
<!--These permissions should be added in Manifest.xml file -->
<!-- using the below permissions we are able to check and -->
<!-- and manage connectivity of an android device. In addition -->
<!-- we can get information about Wi-Fi network -->
<uses-permission
    android:name="android.permission.ACCESS_WIFI_STATE"
    android:maxSdkVersion="20" />
<uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE"
    android:maxSdkVersion="20" />
<uses-permission
    android:name="android.permission.CHANGE_WIFI_STATE"
    android:maxSdkVersion="20" />
<uses-permission
    android:name="android.permission.INTERNET"
    android:maxSdkVersion="20" />
```

Κώδικας 3.1: Διαχείριση αδειών εφαρμογή για πόρους δικτύου

Για να προσπελαστούν οι υπηρεσίες δικτύου σε μια Android συσκευή, πρέπει να υπάρχουν οι κατάλληλες άδειες. Μια εφαρμογή Android μπορεί να χρησιμοποιεί τις περισσότερες υπηρεσίες δικτύου μόνο εάν της έχουν εκχωρηθεί οι κατάλληλες άδειες <uses-permissions> διαμορφωμένες μέσα στο αρχείο Manifest του Android. Υπάρχουν αρκετές άδειες, που σχετίζονται με τη δικτύωση, οι περιλαμβάνουν και εκείνες που επιτρέπουν την προσπέλαση και τις αλλαγές στην κατάσταση του δικτύου Wi-Fi. Στον παραπάνω κώδικα φαίνονται οι άδειες, που έχουν οριστεί στο αρχείο Manifest είναι απαραίτητες για την εφαρμογή που υλοποιήθηκε.

Στο παρακάτω πλαίσιο παρουσιάζεται ο κώδικας της εφαρμογής, που χρησιμοποιείται για τον έλεγχο της κατάστασης του Wi-Fi δικτύου και για τον εντοπισμό της θέσης του χρήστη. Χρησιμοποιώντας τις κλάσεις WifiManager και ConnectivityManager, είναι

δυνατή η προσπέλαση πληροφοριών κατάστασης του δικτύου της συσκευής. Όπως έχει ήδη αναφερθεί, η εφαρμογή για να εντοπίσει την θέση του χρήστη, καθώς και για να συνδεθεί με το Cloudlet, πρέπει να είναι συνδεδεμένη στο ίδιο Wi-Fi δίκτυο με αυτό. Για να εντοπιστεί η θέση του χρήστη στο χώρο, η εφαρμογή χρησιμοποιεί τη `mac` διεύθυνση του AP. Η ανάκτηση της `mac` διεύθυνσης γίνεται με χρήση της μεθόδου `getBSSID` [26]. Στην συνέχεια ελέγχεται σε ποιόν όροφο βρίσκεται το Access Point με το `bssid` που ανακτήθηκε.

```

/* Create an instance of connectivity manager Class in order to access
   the network status of an android device */
WifiManager wifiMgr = (WifiManager)
getSystemService(Context.WIFI_SERVICE);

//get infos about Wi-Fi network
WifiInfo wifiInfo = wifiMgr.getConnectionInfo();
ConnectivityManager connManager = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo mWifi =
connManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

    //check status of Wi-Fi
    if ((wifiMgr.isWifiEnabled() && mWifi.isConnected())
        {
            /* This flag indicates that the device is connected with
               LAN. This flag variable is used
               in order to display the appropriate
               alert messages about user's actions
            */
            flag = 1;

            /* with the below method we can get the MAC address of an
               Access Point
            */
            mac_addr = wifiInfo.getBSSID(); //null if does not find AP

            /* with the below block we find the floor on which the user of the device
               located */
            if (mac_addr.equals("CC:08:CC:03:47:10"))//match mac address
                myfloor = 0; //ground
            else if (mac_addr.equals("00:BB:C1:04:23:1A"))
                myfloor = 1; //first
            else if (mac_addr.equals("00:10:CC:AA:47:10"))
                myfloor = 2; //second
            else if (mac_addr.equals("11:08:CC:03:41:CC"))
                myfloor = 3; // third
            else
                myfloor = 10; /* This value indicates that on the
                               floor on which the user located, no lessons
                               conduct */
        }

```

Κώδικας 3.2: Έλεγχος σύνδεσης στο Wi-Fi δίκτυο

Ο κώδικας στο επόμενο πλαίσιο παρουσιάζει τη μέθοδο, που υλοποιήθηκε, προκειμένου να ανακτηθεί η συνολική χρήση της CPU της Android κινητής συσκευής. Για να γίνει αυτό εφικτό χρησιμοποιήθηκε η εντολή `top` του λειτουργικού συστήματος Linux με παραμέτρους: `-n 1`. Όπως έχει ήδη αναφερθεί σε προηγούμενη ενότητα το Android είναι ένα Linux-based λειτουργικό σύστημα κινητών συσκευών, οπότε μπορούν να χρησιμοποιηθούν χωρίς πρόβλημα εντολές του Linux OS. Για να εκτελεστεί η παραπάνω εντολή όπως ακριβώς θα μπορούσε να εκτελεστεί και στην κονσόλα του Linux, χρησιμοποιείται η μέθοδος `getRuntime` [24]. Κάθε Java εφαρμογή έχει ένα στιγμιότυπο της κλάσης `Runtime`, το οποίο επιτρέπει στην εφαρμογή να αλληλεπιδρά με το περιβάλλον στο οποίο εκτελείται.

```
private String executeTop() {
    //initialization
    java.lang.Process p = null;
    BufferedReader in = null;
    String returnString = null;

    try {

        /* execute top command of Linux with parameters: n -1.
        Class Runtime allows an application to interface with
        the environment in which it is running */
        p = Runtime.getRuntime().exec("top -n 1"); //runtime execution
        /* The above command retrieves CPU statistics */

        //create new input stream
        in = new BufferedReader(new
        InputStreamReader(p.getInputStream()));

        //read the result data after top command execution
        while (returnString == null ||
        returnString.contentEquals("")) {
            //store data in string var
            returnString = in.readLine();
        }

    } catch (IOException e) { // manage exeptions
        //print logs with the below error message
        Log.e("executeTop", "error in getting first line of top");

        /* The below command prints a stack trace for this Throwable object
        on the error output stream, that is the value of field System.err
        */
        e.printStackTrace();

    } finally {
        //the code in finally block executed in any case
        try {
            in.close(); // close stream
            p.destroy(); //stop top command process
        } catch (IOException e) {

            //print logs with the below error message
            Log.e("executeTop",
```

```

        "error in closing and destroying top process");
        e.printStackTrace();
    }
}

return returnString; //return the first line as result
}

```

Κώδικας 3.3: Εκτέλεση της εντολής Top του Linux

Η παραπάνω μέθοδος επιστρέφει τα δεδομένα σε μια μεταβλητή τύπου String με την μορφή: User x%, System x%, IOW x%, IRQ x%. Για τις ανάγκες της εφαρμογής χρειάζεται να εμφανιστεί στην οθόνη της κινητής συσκευής, ως ενημέρωση προς το χρήστη, ένα ποσοστό επί τοις εκατό, το οποίο θα υποδηλώνει τη συνολική χρήση της CPU της συσκευής από εφαρμογές και εργασίες του λειτουργικού συστήματος. Με αυτόν τον τρόπο ο χρήστη θα γνωρίζει κάθε στιγμή τη διαθεσιμότητα των υπολογιστικών πόρων της συσκευής του, ώστε να μπορεί εύκολα να αποφασίσει εάν θα προσφέρει ένα μέρος από αυτούς στο Cloudlet. Η μέθοδος που παρουσιάζεται παρακάτω καλεί τη μέθοδο executeTop, από τη συμβολοσειρά που της επιστρέφει ως αποτέλεσμα ανακτά μόνο τους αριθμούς των ποσοστών. Στη συνέχεια, προσθέτει αυτούς τους αριθμούς, προκειμένου να επιστρέψει με τη σειρά της τη συνολική χρήση της CPU ως έναν ακέραιο αριθμό.

```

private int getCpuUsageStatistic() {

    String tempString = executeTop();// call method

    tempString = tempString.replaceAll(",", ""); //remove all ","
    tempString = tempString.replaceAll("User", ""); //remove "User"
    tempString = tempString.replaceAll("System", ""); //remove
//"System"
    tempString = tempString.replaceAll("IOW", ""); //remove "IOW"
    tempString = tempString.replaceAll("IRQ", ""); //remove IRQ
    tempString = tempString.replaceAll("%", ""); //remove "%"

    for (int i = 0; i < 10; i++) {
        tempString = tempString.replaceAll(" ", " ");
    }

    tempString = tempString.trim(); //trim string
    //store it in new String array
    String[] myString = tempString.split(" ");
    //create new array with n=mystring.length positions
    int[] cpuUsageAsInt = new int[myString.length];

    int cpuusage = 0; //init new int var
    for (int i = 0; i < myString.length; i++) {
        myString[i] = myString[i].trim();

        /*convert string values to integers and store them
        in cpuUsageAsInt array */

```

```

        cpuUsageAsInt[i] = Integer.parseInt(myString[i]);
        //add all value of cpuUsageAsInt array
        cpuusage = cpuusage + cpuUsageAsInt[i]; //total
    }

    //return an integer with total CPU usage
    return cpuusage;
}

```

Κώδικας 3.4: Υπολογισμός χρήσης της CPU

Στο επόμενο πλαίσιο κειμένου φαίνεται ο κώδικας, ο οποίος υλοποιήθηκε με στόχο ο χρήστης της εφαρμογής να μπορεί εύκολα και γρήγορά επιλέξει το ποσοστό της CPU της κινητής του συσκευής, το οποίο θέλει να παραχωρήσει για την εκτέλεση εργασιών από τη Cloudlet υποδομή. Ο χρήστης μετακινώντας μια μπάρα προόδου, μπορεί να επιλέξει το ακριβές ποσοστό, χωρίς αυτό να υπερβαίνει σε καμία περίπτωση το ποσοστό διαθεσιμότητας της CPU. Σε μια κινητή συσκευή, όπως και σε οποιοδήποτε ηλεκτρονική συσκευή, τρέχουν συνεχώς διάφορες διεργασίες, οι οποίες δεσμεύουν ένα μέρος από τους υπολογιστικούς πόρους της συσκευής. Επομένως, οι διεργασίες αυτές θα πρέπει να συνεχίσουν να τρέχουν κανονικά, χωρίς να επηρεάζονται τα tasks, τα οποία θα ανατίθενται στην συσκευή από το Cloudlet. Με αυτόν τον τρόπο ο χρήστης δεσμεύει τους υπολογιστικούς πόρους της συσκευής του. Σε αυτό το σημείο, όπως φαίνεται στον κώδικα που ακολουθεί, γίνεται ο υπολογισμός των MFLOPs, που θα στείλει η συσκευή στο Cloudlet.

```

/*the below code display a seek bar with the CPU percent that the user
has allocates for the cloudlet.*/
cputocloud.setText(seekbar1.getProgress() + "/" + (seekbar1.getMax() -
cpu_usage));

        seekbar1.setOnSeekBarChangeListener(new
OnSeekBarChangeListener() {
            //manage seek bar movement
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress,
                boolean fromUser)
            {
                /* progress indicates the position of the pointer
of seek bar */
                progress_value = progress;
                if (progress_value > (100 - cpu_usage))
                {
                    progress_value = 100 - cpu_usage;
                }
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

```

```

        // TODO Auto-generated method stub
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        /* this change the percentage when user move the
pointer of seek bar */

        cputocloud.setText(progress_value + "/" +
(seekbar1.getMax() - cpu_usage));

        /* the maximum CPU percent that the user can allocate is
100 - (the number of CPU usage) which it is returned by
getCpuUsageStatistic method
*/

        /* the below var stores the total MFLOPS that the device
can manage */
        double mflops_local = 50;

        /* the below block calculates the MFLOPS that the
user allocates for tasks from cloudlet. This is
depending from the CPU percent that he wants to offer
*/
        if ((100 - (progress_value + cpu_usage)) >= 40)
        { //MFLOPS calculation
            mflops = mflops_local*(progress_value)/100;
        }
        else
        {
            /*in case that the sum of CPU percent which is used
by other processes which executed by the device and
the CPU percent that the user allocates is greater than
60% the application allocate less computation resources
*/
            mflops = mflops_local*(progress_value)/100;
            mflops = mflops*80/100; //MFLOPS calculation
        }
    }
});

```

Κώδικας 3.5: Δέσμευση υπολογιστικών πόρων

Η επικοινωνία μεταξύ της Android εφαρμογής και της εικονικής Cloudlet υποδομής, που υλοποιήθηκε με το εργαλείο CloudSim, γίνεται με χρήση Java Sockets. Με χρήση των Java Sockets υλοποιείται μια αμφίδρομη επικοινωνία μεταξύ μιας client και μιας server εφαρμογής στο δίκτυο. Ο server όταν τρέχει ακούει σε μια πόρτα και περιμένει για κάποιο αίτημα σύνδεσης από έναν client. Μόλις ο server αποδεχτεί το αίτημα σύνδεσης με τον client, τότε το socket δημιουργείται με επιτυχία και ο client μπορεί πλέον να το χρησιμοποιήσει για να στείλει δεδομένα. Το Socket είναι ένα endpoint μιας αμφίδρομης επικοινωνίας μεταξύ δυο προγραμμάτων, τα οποία τρέχουν σε ένα δίκτυο. Ένα endpoint είναι ο συνδυασμός μιας IP διεύθυνσης και μιας πόρτας. Κάθε TCP σύνδεση μπορεί να

ταχτοποιηθεί μοναδικά από τα δυο endpoints (client/server). Με αυτόν τον τρόπο το επίπεδο μεταφοράς (TCP Layer) μπορεί να αναγνωρίσει την εφαρμογή στην οποία τα δεδομένα προορίζονται για να σταλούν. Για τις ανάγκες της παρούσας μεταπτυχιακής διατριβής η Android εφαρμογή έχει το ρόλο του client και το Cloudlet το ρόλο του server. Αντίστοιχα, έχει υλοποιηθεί ο client και server side κώδικας και παρατίθενται παρακάτω. Ο Server ακούει στην πόρτα 4444.

```
/* Server side code */
try{
    //server listens on port 4444
    s = new ServerSocket(4444); //use ServerSocket Class
    System.out.println("Cloud Server is up and running" + "\n");
} catch (IOException e)
{
    //fail, server is down
    System.out.println("Server cannot start");
}

try{
    //wait for a new connection
    Socket s1 = s.accept(); //connection established
    System.out.println("Connected" + "\n");

    //create a new input stream and read data
    InputStream slin = s1.getInputStream();
    DataInputStream dis = new DataInputStream(slin);
    mflops_str = dis.readUTF(); // get MFLOPS

    //convert this value from string to double
    rec_mflops = Double.valueOf(mflops_str);

    /*check if the device has the available CPU
    resources to execute the task
    */
    if (rec_mflops >= mflops){
        i = 1; //device can manage the task
    }

    dis.close(); //close stream
    s1.close(); //close connection
} //end of try block

//exception, connection fail
catch (IOException e) {
    System.out.println("Connection failed");
}
```

Κώδικας 3.6: Κώδικας από την πλευρά του server


```
//client side code
new Thread(new Runnable() {
//the below code should be in a Thread
@Override
public void run() {

// TODO Auto-generated method stub
try
{
/*create a new connection with the server at port 4444.
IP 10.0.2.2 is specific for android emulator */
Socket s1 = new Socket("10.0.2.2",4444);

//create new output stream and send data
OutputStream os = s1.getOutputStream();
DataOutputStream dos = new DataOutputStream(os);

//display the connection status
conn_status.setText("Connected");
dos.writeUTF(mess);//send MFLOPS to Cloudlet
dos.close(); //close stream
s1.close(); // close connection
//Manage exceptions
}catch (ConnectException connExc) {

//print logs
Log.i(er, "Could not connect with the server.");
//display connection error
assigntask.setText("Connection fail, Server is down ..!!");
}

catch (IOException e) {
//ignore this block
}
}
```

Κώδικας 3.7: Κώδικας από την πλευρά του client

Παρακάτω παρουσιάζεται ο τρόπος με τον οποίο ορίζεται ένα task στο εργαλείο προσομοιώσεις CloudSim. Στο CloudSim δεν εκτελούνται πραγματικές εφαρμογές, οι προγραμματιστές πρέπει να προσημειώσουν τις απαιτήσεις τους θέτοντας τιμές σε κάποιες ιδιότητες. Θα πρέπει σε αυτό το σημείο να τονιστεί ότι, με τον όρο Cloudlet στο CloudSim εννοούμε ένα task [28]. Το CloudSim παρέχει την κλάση Cloudlet για τη δημιουργία tasks. Για τον λόγο αυτό δεν πρέπει να συγχέεται με τη Cloudlet-Based υποδομή υπολογιστικού νέφους.

```
/* Cloudlet properties */
int id = 0; //the id number of a cloudlet (task)
long length = 4000;
long fileSize = 300;
long outputSize = 300;
double mflops = 25; //total number of MFLOPS that the cloudlet consumes
UtilizationModel utilizationModel = new UtilizationModelFull();

//call Cloudlet Class constructor
Cloudlet cloudlet1 = new Cloudlet(id, length, pesNumber, fileSize,
outputSize, utilizationModel, utilizationModel, utilizationModel);
```

Κώδικας 3.8: Δημιουργία cloudlet στο CloudSim

Ο επόμενος κώδικας παρουσιάζει τη διαδικασία της δημιουργία μιας Virtual Machine στο CloudSim. Για τις ανάγκες της εργασίας υλοποιήθηκαν δυο VMs.

```
//VM description
int vmid = 0; //id number of Virtual Machine
int mips = 250; //number of millions instruction per second
//that this VM can manage
long size = 10000; //image size (MB)
int ram = 512; //vm memory (MB)
long bw = 1000; //set bandwidth
int pesNumber = 1; //number of cpus
String vmm = "Xen"; //Virtual Machine Manager name

//create the first VM
Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new
CloudletSchedulerTimeShared());
//execution policy for cloudlets (task): Timeshared on this VM
```

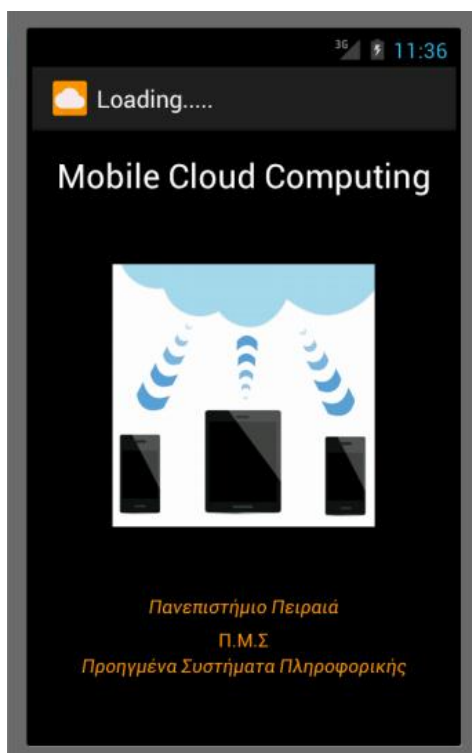
Κώδικας 3.9: Δημιουργία Virtual Machine στο CloudSim

4. Αποτελέσματα

Στις ενότητες που ακολουθούν παρουσιάζεται η εφαρμογή και τα αποτελέσματα του πειραματικού μέρους της μεταπτυχιακής διατριβής. Επιπλέον, γίνεται αξιολόγηση του συστήματος και παρατίθενται θέματα για συζήτηση.

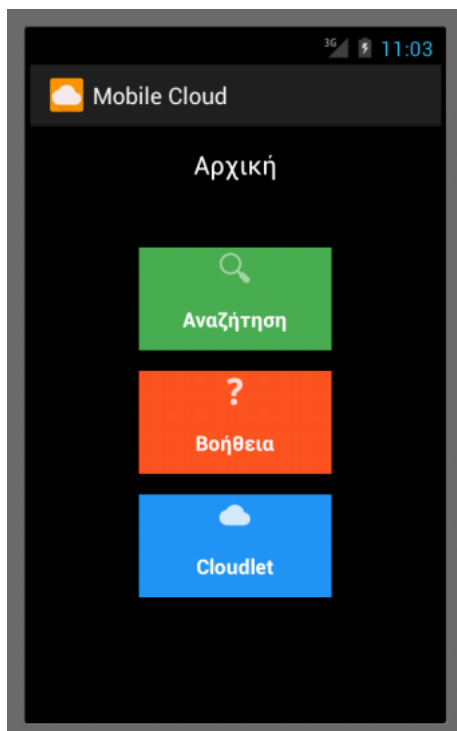
4.1 Παρουσίαση Εφαρμογής

Σε αυτήν την ενότητα παρουσιάζεται η εφαρμογή, που υλοποιήθηκε για την πλατφόρμα του λειτουργικού συστήματος Android, καθώς και τα αποτελέσματα του πειραματικού μέρους της μεταπτυχιακής διατριβής. Τα αποτελέσματα των πειραμάτων αφορούν: α) την επικοινωνία της mobile εφαρμογής με την εικονική υποδομή του Cloudlet, β) την ανάθεση εργασιών από το Cloudlet σε μια κινητή συσκευή, ανάλογα με τους υπολογιστικούς πόρους που αυτή προσφέρει και γ) την εκτέλεση ή απόρριψη ενός task από την κινητή συσκευή ανάλογα με την διαθεσιμότητα των πόρων της.



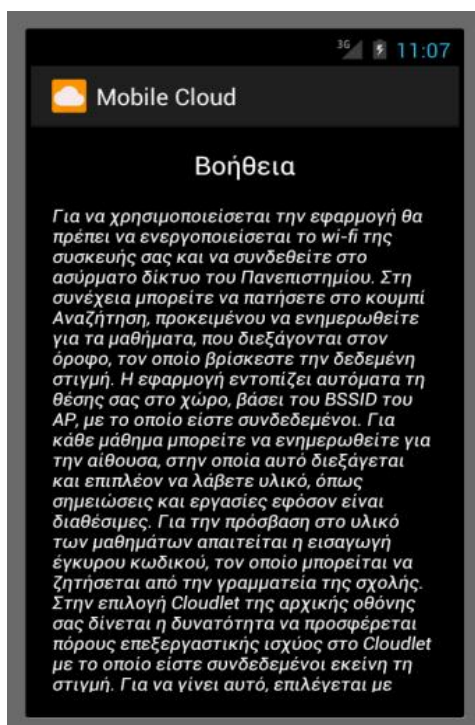
Εικόνα 4.1: Εμβόλιμη Οθόνη

Μόλις ο χρήστης ενεργοποιήσει την εφαρμογή Mobile Cloud Computing, εμφανίζεται μια εμβόλιμη οθόνη. Η Εμβόλιμη Οθόνη ή αλλιώς Splash Screen λειτουργεί ως το αρχικό σημείο εισόδου του χρήστη στην εφαρμογή. Στόχος αυτής της οθόνης είναι η εμφάνιση πληροφοριών ενός έξυπνου λογότυπου για την εφαρμογή, το οποίο να προΐδεάζει τον χρήστη για την λειτουργικότητα της. Από την εμβόλιμη οθόνη ο χρήστης μεταβαίνει αυτόματα στην οθόνη κύριου μενού μετά από μερικά δευτερόλεπτα, όταν πλέον η εφαρμογή φορτώσει πλήρως.



Εικόνα 4.2: Κύρια Οθόνη

Η οθόνη κύριου μενού λειτουργεί ως η κύρια οθόνη περιήγησης μέσα στην εφαρμογή Mobile Cloud. Όπως φαίνεται στην διπλανή εικόνα ο χρήστης σε αυτή την οθόνη έχει συνολικά τρεις επιλογές: 1) «Αναζήτηση». Πατώντας το κουμπί ξεκινά η διαδικασία εντοπισμού της θέσης του χρήστη, με στόχο την προβολή πληροφοριών για τα μαθήματα της σχολής του, ανάλογα με τον όροφο του κτιρίου του πανεπιστημίου στον οποίο βρίσκεται. 2) «Βοήθεια». Πατώντας σε αυτό το κουμπί προβάλλεται το εγχειρίδιο χρήσης της εφαρμογής. 3) «Cloudlet». Πατώντας αυτό το κουμπί ο χρήστης ξεκινά την διαδικασία δέσμευσης υπολογιστικών πόρων και επικοινωνίας με το Cloudlet.



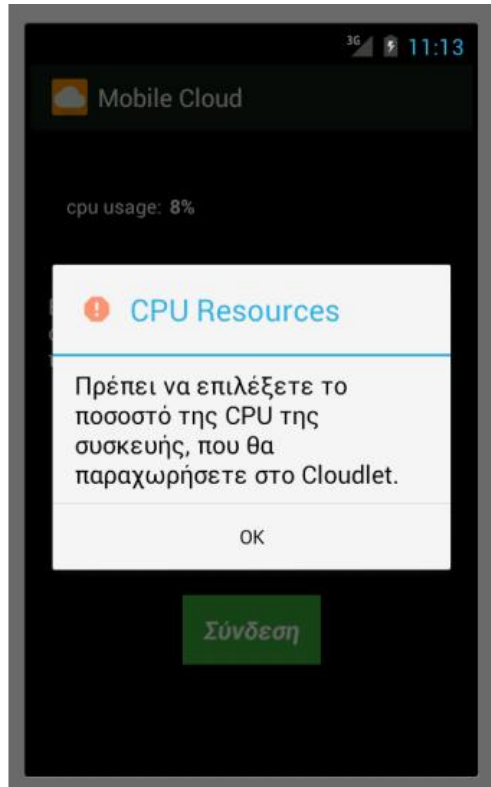
Εικόνα 4.3: Οθόνη Βοήθειας

Η οθόνη βοήθειας καθοδηγεί τον χρήστη για το πώς θα χρησιμοποιήσει την εφαρμογή. Η οθόνη βοήθειας εμφανίζει στον χρήστη ένα κείμενο με αναλυτικές οδηγίες χρήσης, στο οποίο του επιτρέπει να κάνει κύλιση προς τα επάνω και προς τα κάτω. Ο χρήστης μπορεί να επιστρέψει στην προηγούμενη οθόνη χρησιμοποιώντας το «back button» της Android συσκευής του. Το ίδιο ακριβώς ισχύει για την πλοήγηση προς τα πίσω και για τις υπόλοιπες οθόνες της εφαρμογής.



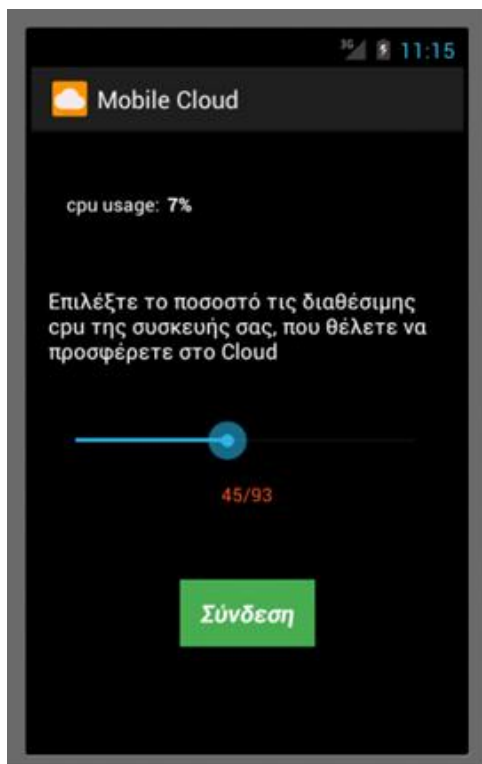
Εικόνα 4.4: Χρήση της CPU

Μόλις ο χρήστης πατήσει το κουμπί Cloudlet της κύρια οθόνης, μεταβαίνει στην δίπλα οθόνη. Σε αυτή την οθόνη πάνω αριστερά φαίνεται το ποσοστό της CPU της κινητής συσκευής, που είναι δεσμευμένο εκείνη την στιγμή από διεργασίες του λειτουργικού συστήματος ή/και από άλλες εφαρμογές. Με χρήση της μπάρας προόδου ο χρήστης μπορεί να επιλέξει το ποσοστό από τη διαθέσιμη CPU, που επιθυμεί να παραχωρήσει για εκτέλεση εργασιών από το Cloudlet. Όπως φαίνεται στην διπλανή εικόνα το 92% της CPU της συσκευής δεν χρησιμοποιείται (idle) και επομένως μπορεί να διατεθεί για άλλους σκοπούς. Ο χρήστης για να συνεχίσει, πρέπει να επιλέξει ένα ποσοστό και έπειτα να πατήσει το κουμπί σύνδεση.



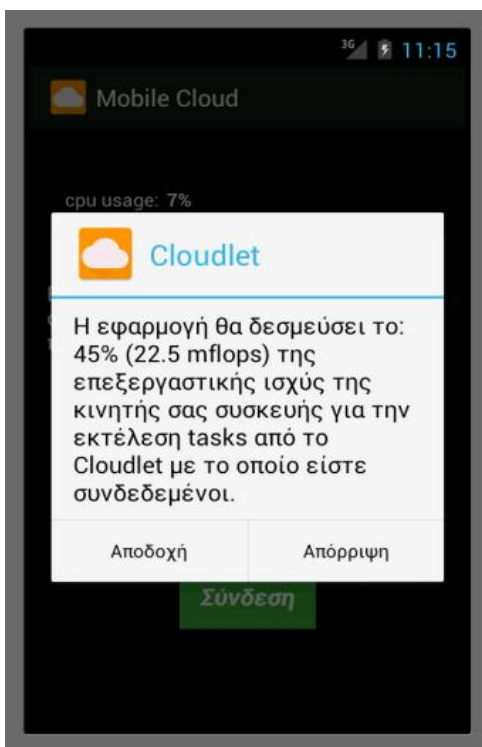
Εικόνα 4.5: Alert Λάθος Επιλογή

Σε περίπτωση όπου, ο χρήστης πατήσει το κουμπί σύνδεση στην αριστερή οθόνη χωρίς να έχει επιλέξει το ένα ποσοστό της CPU διάφορο του μηδενός, τότε εμφανίζεται στην οθόνη του ένα μήνυμα λάθους μέσα σε ένα alert πλαίσιο. Το μήνυμα ενημερώνει το χρήστη για την αποδεκτή ενέργεια στην οποία θα πρέπει να προχωρήσει. Για να κλείσει το alert μήνυμα, ο χρήστης το μόνο που χρειάζεται να κάνει είναι, να πατήσει το κουμπί OK αυτού.



Εικόνα 4.6: Δέσμευση Πόρων Πείραμα 1

Όπως φαίνεται στην οθόνη αριστερά ο χρήστης έχει επιλέξει, να δεσμεύσει το 45% της διαθέσιμης CPU της συσκευής του για να το παραχωρήσει στο Cloudlet. Εάν, ο χρήστης συνεχίσει την διαδικασία και παραχωρήσει πόρους στο Cloudlet, τότε σε περίπτωση όπου αυτό αναθέσει στη συσκευή κάποιο task, το παραπάνω ποσοστό της CPU δεν θα μπορεί πλέον, να χρησιμοποιηθεί από τις native εφαρμογές, που τρέχουν στην συσκευή.



Εικόνα 4.7: Ειδοποίηση Δέσμευσης Πόρων

Μόλις ο χρήστης επιλέξει τους υπολογιστικούς πόρους, που επιθυμεί να δεσμεύσει για το Cloudlet και πατήσει το κουμπί σύνδεσης, εμφανίζεται στην οθόνη ένα πλαίσιο alert, όπως αυτό που φαίνεται στη διπλανή εικόνα. Το μήνυμα σε αυτή την οθόνη ενημερώνει τον χρήστη ότι το 45% της υπολογιστικής ισχύος του συσκευής του, θα μπορεί να χρησιμοποιηθεί από το Cloudlet. Αυτό συνεπάγεται λιγότερους υπολογιστικούς πόρους για τις native εφαρμογές της συσκευής και φυσικά μεγαλύτερη κατανάλωση μπαταρίας λόγω της μεγαλύτερης χρήσης της CPU. Σε αυτό το μήνυμα φαίνεται ο αριθμός των MFLOPS, που αντιστοιχεί στο 45% της CPU. Ο υπολογισμένος αριθμός των MFLOPS είναι αυτός που στέλνεται

ως ενημέρωση για τους επιπλέον υπολογιστικούς επιπλέον πόρους από την mobile εφαρμογή στο Cloudlet. Ο χρήστης σε περίπτωση, που αποδέχεται τους παραπάνω όρους,

πρέπει να πατήσει το κουμπί Αποδοχή για να συνδεθεί και να ενημερώσει το Cloudlet, αλλιώς πρέπει να πατήσει το κουμπί απόρριψη και να επιστρέψει στην Κύρια Οθόνη της εφαρμογής.



Εικόνα 4.8: Αποτυχία Εκτέλεσης task

Η διπλανή οθόνη ενημερώνει το χρήστη της εφαρμογής ότι, το task με id αριθμό 4 ανατέθηκε στη συσκευή από το Cloudlet, αλλά απέτυχε να εκτελεστεί. Ο λόγος για τον οποίο ένα task αποτυγχάνει να εκτελεστεί είναι επειδή η συσκευή δεν έχει πλέον στη διάθεση της τους απαιτούμενους για την εκτέλεση υπολογιστικούς πόρους. Η εφαρμογή ενημερώνει τόσο τον χρήστη, όσο και το για την αποτυχία εκτέλεσης του task. Σε αυτή την περίπτωση το task αυτό θα πρέπει να ανατεθεί σε άλλη συσκευή για εκτέλεση, ή να μπει στην ουρά μαζί με τα υπόλοιπα tasks και να εκτελεστεί στο Cloudlet, ή να προωθηθεί για εκτέλεση στο Cloud.

Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα της εκτέλεσης των tasks στο εργαλείο προσομοίωσης CloudSim, το οποίο χρησιμοποιήθηκε για την υλοποίηση της Cloudlet-Based υποδομής. Όπως φαίνεται το Cloudlet είχε αναλάβει την εκτέλεση πέντε tasks (cloudlets). Σε προηγούμενο κεφάλαιο έχει διευκρινιστεί ότι στο CloudSim με τον όρο cloudlet περιγράφεται ένα task, οπότε δεν πρέπει να συγχέεται με την Cloudlet-Based υποδομή υπολογιστικού νέφους. Στα αποτελέσματα της προσημείωσης φαίνεται ότι cloudlet-task με id 4 είχε ανατεθεί σε κινητή συσκευή, που συνδράμει στο υπολογιστικό νέφος, αλλά η εκτέλεση του απέτυχε. Το cloudlet-task με id 4 μπήκε στην ουρά μαζί με τα υπόλοιπα και εκτελέστηκε στο Cloudlet στη VM με id 0. Σε αυτό το σημείο πρέπει να σημειωθεί ότι για την πραγματοποίηση των πειραμάτων χρησιμοποιήθηκε μόνο μια εικονική κινητή συσκευή (android emulator) και δεν έχει υλοποιηθεί σύνδεση επικοινωνίας του Cloudlet με μια Cloud υποδομή.

```
Simulation completed.
```

```
=====
::: RESULTS :::

Cloudlet  STATUS  DataCenter  VM  Time
  1        SUCCESS    3           1    32
  3        SUCCESS    3           1    32
  0        SUCCESS    2           0   47,99
  2        SUCCESS    2           0   47,99
  4        SUCCESS    2           0   47,99

Cloudlet with id 4 failed to execute by mobile device 1
=====
```

Εικόνα 4.9: Cloudlets Αποτελέσματα Εκτέλεσης Πειράματος 1



Εικόνα 4.10: Δέσμευση Πόρων Πείραμα 2

Στην παραπάνω εικόνα φαίνεται ο χρήστης έχει δεσμεύσει περισσότερους υπολογιστικούς πόρους από την CPU της κινητής του συσκευής. Προσφέρει στο Cloudlet το 85% της CPU της συσκευής του, αφήνοντας ελεύθερο μόλις το 7% για άλλες εφαρμογές. Στην περίπτωση όπου ο χρήστης δεσμεύσει ένα τόσο μεγάλο ποσοστό, η εφαρμογή όπως έχει ήδη αναφερθεί στο κεφάλαιο 3, υπολογίζει και παραχωρεί στο Cloudlet λιγότερους υπολογιστικούς πόρους. Το παραπάνω είναι σκόπιμο ώστε, να αποφευχθεί η πιθανότητα κατάρρευσης του συστήματος, σε περίπτωση που η εφαρμογές, που τρέχουν ήδη στη συσκευή, απαιτήσουν περισσότερους υπολογιστικούς όρους και η CPU να αδυνατεί να τους προσφέρει.



4.11: Επιτυχής Εκτέλεση task

Η οθόνη στη διπλανή εικόνα ενημερώνει τον χρήστη για την επιτυχή εκτέλεση του task με id 4 από τη συσκευή. Επίσης, το popup μήνυμα, που εμφανίζεται στο κάτω μέρος τη οθόνης, ενημερώνει το χρήστη ότι στην συσκευή του δεν έχει ανατεθεί κάποιο άλλο task.

Η παρακάτω εικόνα παρουσιάζει τα αποτελέσματα από το CloudSim, όπου τα cloudlets-tasks με ids 0, 1, 2 και 3 εκτελέστηκαν στις VMs 0 και 1 της εικονικής Cloudlet υποδομής, ενώ το cloudlet-task με id 4 εκτελέστηκε επιτυχώς από μια κινητή συσκευή, που συμμετέχει στην υποδομή.

Simulation completed.

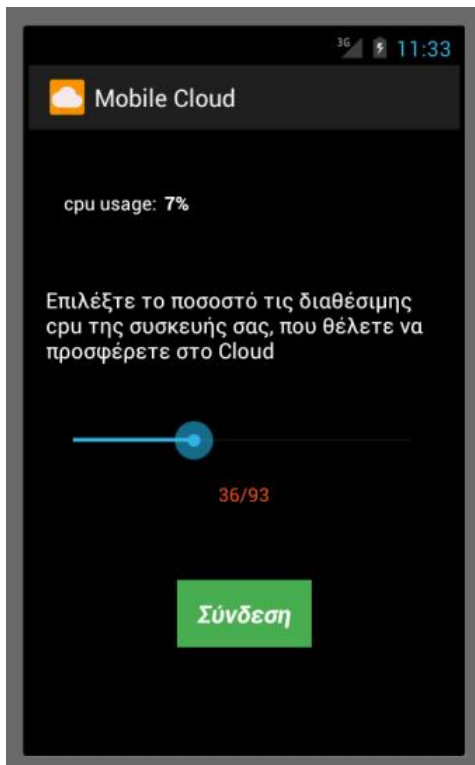
```

=====
::::: RESULTS ::::::

Cloudlet   STATUS   DataCenter   VM   Time
  0         SUCCESS     2           0    32
  2         SUCCESS     2           0    32
  1         SUCCESS     3           1    32
  3         SUCCESS     3           1    32

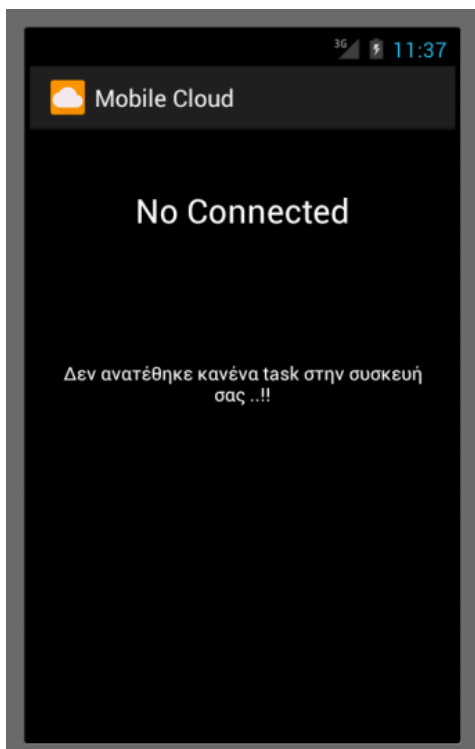
Cloudlet with id 4 was executed successfully by mobile device 1
=====
  
```

Εικόνα 4.12: Cloudlets Αποτελέσματα Εκτέλεσης Πειράματος 2



Εικόνα 4.13: Δέσμευση Πόρων Πείραμα 3

Στην διπλανή οθόνη ο χρήστης έχει δεσμεύσει και πάλι ένα μέρος των υπολογιστικών πόρων της συσκευής για να χρησιμοποιηθεί από το Cloudlet. Σε αυτό το σημείο παρουσιάζεται μια ακόμα εκδοχή, όπου μια κινητή συσκευή προσφέρει πόρους στο Cloudlet, αλλά αυτό δεν τους έχει ανάγκη τη δεδομένη στιγμή.



Εικόνα 4.14: Δεν Ανατέθηκε task

Η διπλανή οθόνη ενημερώνει τον χρήστη ότι δεν έχει ανατεθεί κανένα task στη συσκευή, είτε επειδή δεν υπάρχουν εκείνη τη στιγμή, είτε επειδή οι πόροι της συσκευής δεν επαρκούν.

Simulation completed.

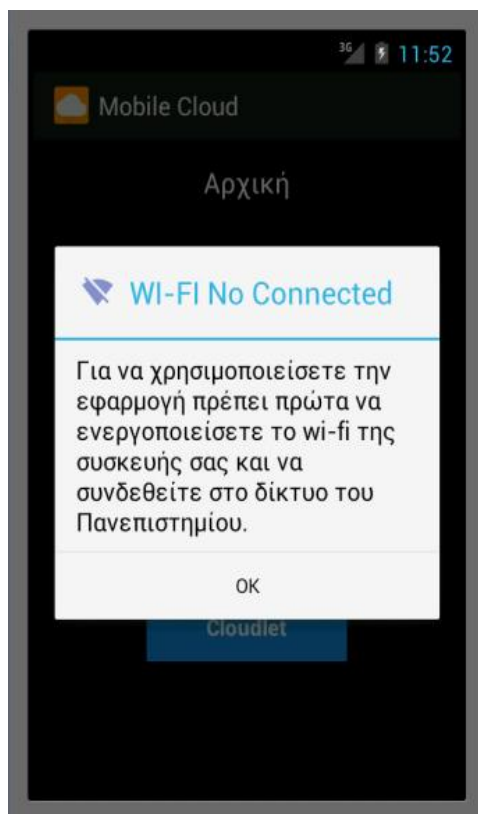
```

=====
::: RESULTS :::
Cloudlet  STATUS  DataCenter  VM  Time
  0        SUCCESS    2           0   32
  2        SUCCESS    2           0   32
  1        SUCCESS    3           1   32
  3        SUCCESS    3           1   32
=====

```

Εικόνα 4.15: Cloudlets Αποτελέσματα Εκτέλεσης Πειράματος 3

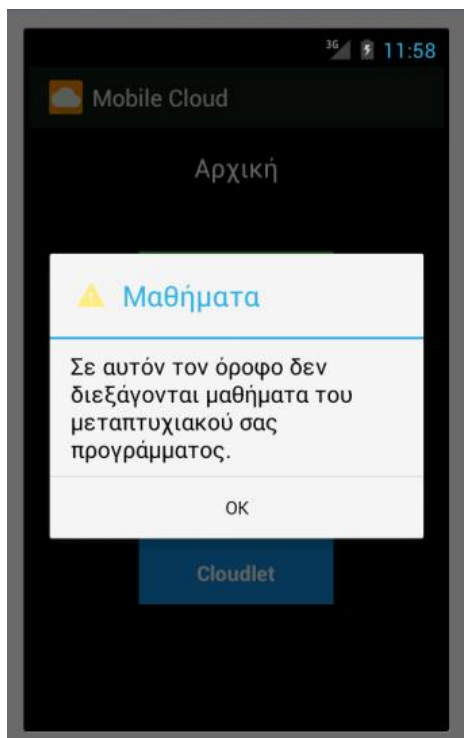
Τα αποτελέσματα από την υλοποίηση στο CloudSim δείχνουν, ότι όλα τα cloudlets-tasks εκτελέστηκαν επιτυχώς στο Cloudlet. Το Cloudlet ζητά πόρους από κινητές συσκευές όταν του έχουν ανατεθεί περισσότερα από 4 tasks για εκτέλεση. Ουσιαστικό αυτό αποτελεί και ένα threshold για το πειραματικό στάδιο της εργασίας.



Εικόνα 4.16: Δεν Υπάρχει Σύνδεση

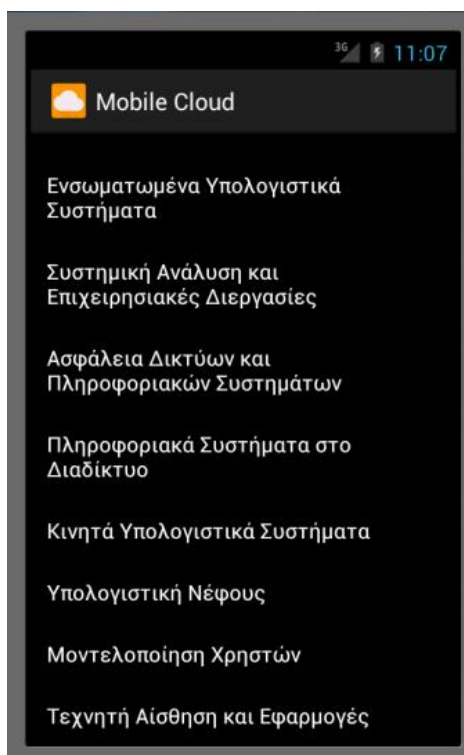
Η εφαρμογή εκτός από την δυνατότητα της δυναμικής δέσμευσης υπολογιστικών πόρων από τον χρήστη πόρων, παρέχει παράλληλα και μιας LBS υπηρεσία, όπως έχει ήδη αναφερθεί και αναλυθεί στα προηγούμενα κεφάλαια. Για να μπορέσει να χρησιμοποιηθεί η υπηρεσία θα πρέπει η συσκευή να είναι συνδεδεμένη στο Wi-Fi δίκτυο του πανεπιστημίου. Για να ενεργοποιηθεί η υπηρεσία και η υπηρεσία και να εντοπιστεί η θέση του χρήστη, πρέπει ο χρήστης να πατήσει το κουμπί Αναζήτηση. Σε περίπτωση που η συσκευή δεν είναι συνδεδεμένη στο δίκτυο, εμφανίζεται στην οθόνη του χρήστη το μήνυμα ειδοποίησης, το οποίο φαίνεται στην παραπάνω εικόνα. Το ίδιο μήνυμα εμφανίζεται όταν ο χρήστης πατήσει το κουμπί Cloudlet, χωρίς να έχει συνδεθεί στο τοπικό

θέσης του χρήστη γίνεται βάσει της MAC διεύθυνσης του AP με το οποίο είναι συνδεδεμένη η συσκευή. Θα πρέπει επίσης να σημειωθεί ότι ένα Cloudlet μπορεί να είναι παράλληλα και AP.



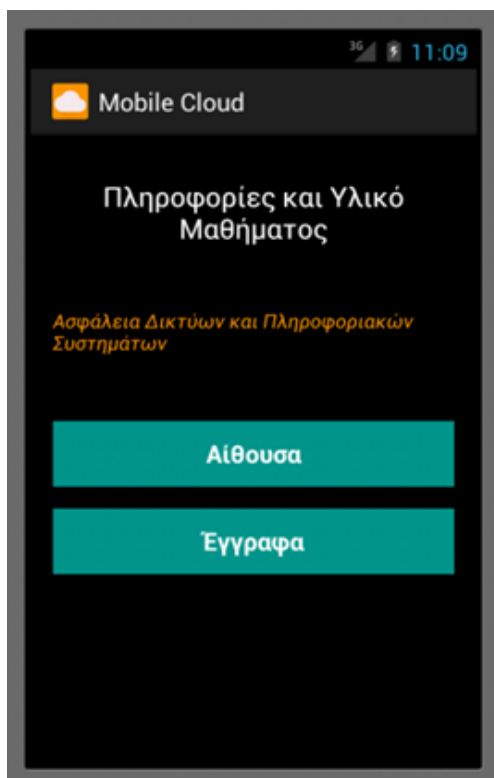
Εικόνα 4.17: Δεν βρέθηκαν Μαθήματα

Εάν ο φοιτητής – χρήστης βρίσκεται σε όροφο στον οποίο δεν διεξάγονται μαθήματα της σχολής του, τότε εμφανίζεται στην οθόνη του ένα ενημερωτικό μήνυμα, όπως αυτό που φαίνεται στη δίπλα εικόνα. Ο χρήστης μπορεί να κλίσει το παράθυρο πατώντας το κουμπί OK.



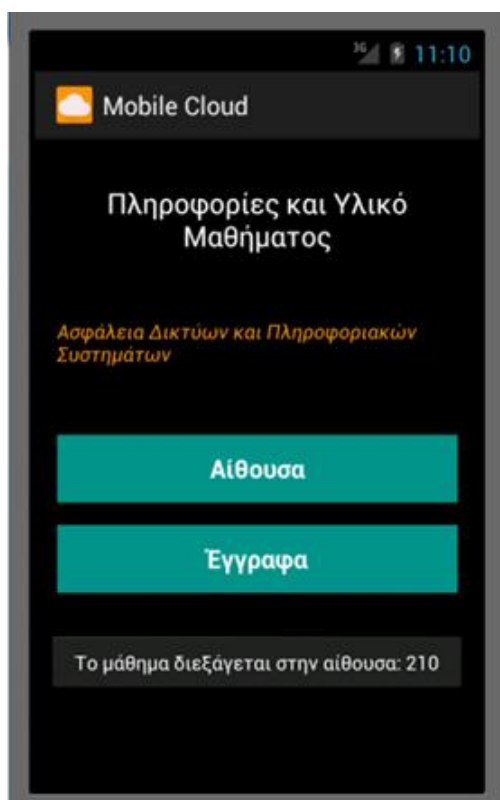
Εικόνα 4.18: Λίστα Μαθημάτων

Σε διαφορετική περίπτωση σε σχέση με την παραπάνω στην οθόνη του χρήστη εμφανίζεται μια λίστα με τα μαθήματα της σχολής του φοιτητή, τα οποία διεξάγονται στον όροφο που βρίσκεται. Στη διπλανή εικόνα φαίνονται τα μαθήματα του δευτέρου ορόφου. Ο φοιτητής – χρήστης μπορεί να επιλέξει ένα μάθημα και μεταβεί σε νέα οθόνη, προκειμένου βρει περισσότερες πληροφορίες για αυτό.



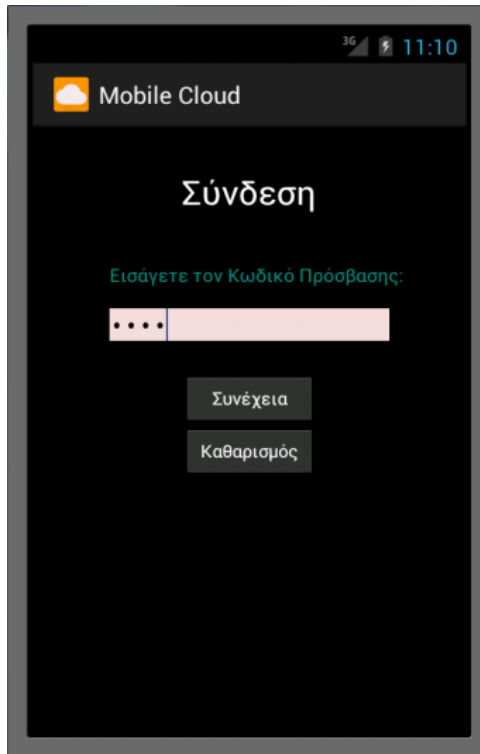
Εικόνα 4.19: Επιλογές Μαθήματος

Όπως φαίνεται στην διπλανή εικόνα ο φοιτητής – χρήστης έχει επιλέξει το μάθημα «Ασφάλεια Δικτύων και Πληροφοριακών Συστημάτων». Οι πληροφορίες, που παρέχονται από την εφαρμογή για κάθε μάθημα είναι, α) η αίθουσα στην οποία διεξάγονται, καθώς β) και η πρόσβαση σε υλικό αυτών π.χ. έγγραφα με σημειώσεις.



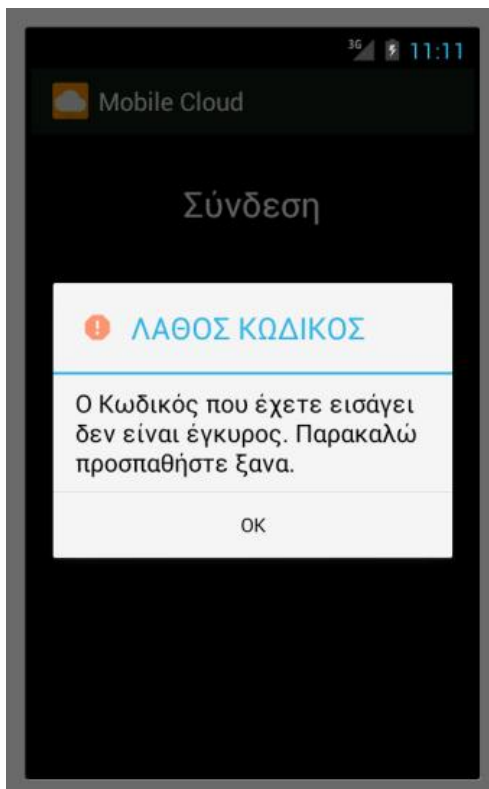
Εικόνα 4.20: Προβολή Αίθουσας

Εάν, ο χρήστης πατήσει το κουμπί «Αίθουσα» της διπλανής οθόνης, τότε εμφανίζεται ένα popup μήνυμα για μερικά δευτερόλεπτα, το οποίο τον ενημερώνει για τον αριθμό της αίθουσας του μαθήματος.



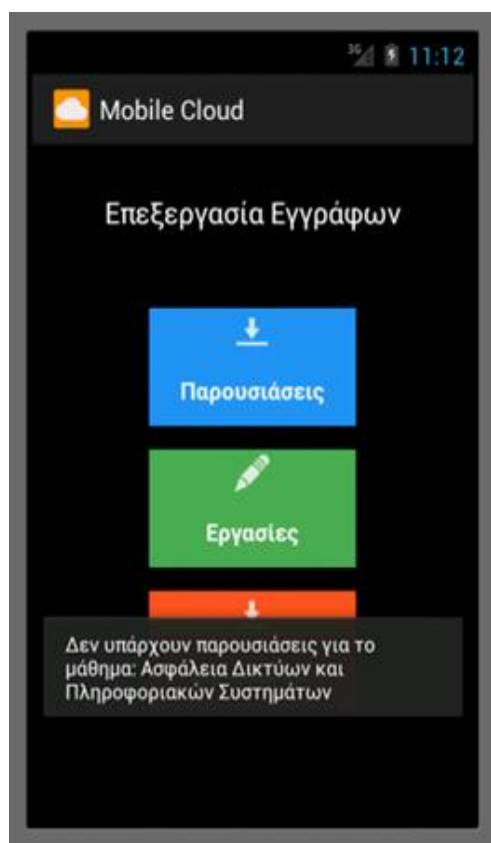
Εικόνα 4.21: Εισαγωγή Κωδικού

Εάν, ο χρήστης πατήσει στην επιλογή «Έγγραφα», τότε μεταβαίνει σε νέα οθόνη, όπου του ζητείται η εισαγωγή έγκυρου κωδικού για την πρόσβαση του σε υλικό του μαθήματος. Το υλικό των μαθημάτων είναι αποθηκευμένο σε Cloud υποδομή.



Εικόνα 4.22: Λάθος Κωδικός

Στην διπλανή εικόνα φαίνεται το μήνυμα, το οποίο θα εμφανιστεί στην οθόνη του χρήστη, σε περίπτωση όπου αυτός εισάγει έναν μη έγκυρο κωδικό. Η εφαρμογή ενημερώνει το χρήστη για το λάθος του και τον προτρέπει να προσπαθήσει ξανά.



Εικόνα 4.23: Υλικό Μαθήματος

Μόλις ο χρήστης εισάγει τον έγκυρο κωδικό μεταβαίνει στην οθόνη, που απεικονίζεται παραπάνω. Σε αυτό το σημείο η υλοποίηση της LBS υπηρεσία της εφαρμογής σταματά στα πλαίσια εκπόνησης της παρούσας διπλωματικής διατριβής. Η επιπλέον ανάπτυξη, που απαιτείται για την ολοκλήρωση αυτής της υπηρεσίας, ανήκει στις προτάσεις για περαιτέρω ανάπτυξη και επέκταση της εφαρμογής και σε άλλους τομείς του resource allocation. Όλα αυτά αναλύονται στο επόμενο κεφάλαιο. Όταν ο χρήστης πατήσει κάποιο από τα κουμπιά της οθόνης, το μόνο που βλέπει είναι ένα popup μήνυμα, το οποίο τον ενημερώνει ότι, δεν υπάρχει διαθέσιμο υλικό για το μάθημα.

4.2 Αξιολόγηση Συστήματος

Η εφαρμογή, η οποία παρουσιάστηκε στην προηγούμενη ενότητα, αποτελεί μια εναλλακτική προσέγγιση της Cloudlet-Based υποδομής υπολογιστικού νέφους, στην οποία οι κινητές συσκευές της προσφέρουν υπολογιστικούς πόρους. Η εφαρμογή παρέχει στον χρήστη της έναν εύκολο και αποτελεσματικό τρόπο δέσμευσης υπολογιστικών δυναμικά κατά την ώρα εκτέλεσής της (runtime). Ο χρήστης μπορεί εύκολα και γρήγορα να αποφασίσει εάν η συσκευή του θα προσφέρει πόρους σε ένα Cloudlet, αφού ενημερώνεται από την ίδια την εφαρμογή για την διαθεσιμότητα της CPU της συσκευής του οποιαδήποτε στιγμή. Η εφαρμογή υπολογίζει αυτόματα τον αριθμό των MFLOPS, ο οποίος θα σταλεί στο Cloudlet, προκειμένου να ενημερωθεί για την διαθεσιμότητα των πόρων της συγκεκριμένης συσκευής, πριν της αναθέσει κάποιο task για εκτέλεση. Ο χρήστης το μόνο που έχει να κάνει είναι, να σύρει τη μπάρα προόδου και να σταματήσει στο επιθυμητό για αυτόν ποσοστό. Η εφαρμογή διαθέτει κατάλληλο αλγόριθμο διαχείρισης των υπολογιστικών πόρων της συσκευής, ώστε να αποφευχθεί η πιθανότητα οι υπολογιστικοί πόροι να εξαντληθούν, με αποτέλεσμα την κατάρρευση του συστήματος. Η εφαρμογή πριν αναλάβει την εκτέλεση ενός task ελέγχει τη διαθεσιμότητα των πόρων ξανά και επιπλέον

απορρίπτει όσα task απαιτούν περισσότερους υπολογιστικούς πόρους από αυτούς που προσφέρει. Ακόμη, στην περίπτωση όπου ο χρήστης επιλέξει να δώσει στο ένα πολύ μεγάλο μέρος της CPU της συσκευής, η εφαρμογή αυτόματα και χωρίς να το γνωρίζει ο χρήστης δεσμεύει μικρότερο ποσοστό της CPU, αφήνοντας περιθώριο για την ομαλή ολοκλήρωση της εκτέλεσης διεργασιών που τρέχουν ήδη τοπικά στην συσκευή.

Η εφαρμογή η οποία παρουσιάστηκε, αποτελεί μια λύση στο τομέα της δέσμευσης και διαμοιρασμού υπολογιστικών πόρων από κινητές συσκευές. Η εφαρμογή υλοποιεί ουσιαστικά μια νέα προσέγγιση της Cloudlet-Based υποδομής υπολογιστικού νέφους και παράλληλα αποτελεί μια βάση για μελλοντικές επεκτάσεις και επιπλέον ανάπτυξη.

Στην ενότητα που ακολουθεί, αναλύονται οι περιορισμοί του συστήματος, που παρουσιάστηκε και αξιολογήθηκε στις παραπάνω ενότητες, καθώς επίσης περιγράφονται και τα σημεία, τα οποία δεν μελετήθηκαν, διότι ήταν πέρα από τα πλαίσια εκπόνησης της παρούσας μεταπτυχιακής διατριβής.

4.3 Περιορισμοί Συστήματος

Η εφαρμογή, η οποία υλοποιήθηκε στα πλαίσια εκπόνησης της παρούσας μεταπτυχιακής διατριβής, δεν εκτελέστηκε σε μια πραγματική Cloudlet-Based αρχιτεκτονική υπολογιστικού νέφους, στην οποία το Cloudlet στέλνει μέρος της εφαρμογής σε μια κινητή συσκευή για επεξεργασία. Για τις ανάγκες της παρούσας εργασίας, χρησιμοποιήθηκαν, όπως ήδη αναφέρθηκε σε προηγούμενο κεφάλαιο, εργαλεία προσομοίωσης. Αυτά τα εργαλεία με την σειρά τους όμως, συμβάλουν σε μια ολοκληρωμένη ανάπτυξη και παρουσίαση της υπό μελέτη προσέγγισης.

Επίσης, για τον υπολογισμό της απόδοσης της CPU της κινητής συσκευής χρησιμοποιήθηκε μια ήδη υλοποιημένη εφαρμογή για Android smartphones, το Linpack Benchmarking for Android. Ο αριθμός των MFLOPS, που προέκυψε από την εκτέλεση της παραπάνω εφαρμογής σε Android συσκευή, χρησιμοποιήθηκε ως ένας δεδομένος και σταθερός αριθμός. Η εφαρμογή δεν υλοποιεί στον κώδικα της έναν ανάλογο benchmark αλγόριθμο. Αυτό έχει ως αποτέλεσμα η λειτουργία της εφαρμογής να μην είναι αντιπροσωπευτική για κάθε συσκευή. Επιπλέον, θα πρέπει να σημειωθεί ότι για την πραγματοποίηση του πειραματικού μέρους, δηλαδή της ανάθεσης εργασιών από το Cloudlet στις κινητές συσκευές, χρησιμοποιήθηκε μόνο μια συσκευή. Η λογική της

λειτουργίας του συστήματος είτε χρησιμοποιείται μόνο μια συσκευή, είτε περισσότερες είναι ανάλογη. Στη δεύτερη περίπτωση όμως απαιτείται επιπλέον υλοποίηση στην πλευρά του Cloudlet, ώστε να μπορεί να διαχειρίζεται πολλαπλές συνδέσεις.

Τέλος, η υπηρεσία παροχής πληροφοριών θέσης της εφαρμογής παρέμεινε υλοποιημένη ως ένα βαθμό, καθώς αποτελεί πρόταση για εξέλιξη της εφαρμογής, έτσι ώστε να προσφέρει πόρους αποθήκευσης δεδομένων και διαμοιρασμού αρχείων και επομένως μένει εκτός των πλαισίων αυτής της μεταπτυχιακής διατριβής.

5. Επίλογος

5.1 Σύνοψη

Στο πλαίσιο της παρούσας μεταπτυχιακής διατριβής μελετήθηκε μια διαφορετική προσέγγιση στον τομέα της δέσμευσης και του διαμοιρασμού πόρων σε μια Cloudlet-Based υποδομή υπολογιστικού νέφους. Επιπλέον, σχεδιάστηκε και υλοποιήθηκε μια client εφαρμογή για τη δημοφιλέστερη πλατφόρμα λειτουργικού συστήματος έξυπνων κινητών συσκευών, Android. Μέσα από αυτή την εφαρμογή ο χρήστης μπορεί να δεσμεύσει δυναμικά ένα ποσοστό από την διαθέσιμη CPU της Android συσκευής του, προκειμένου αυτή να συνδράμει στην Cloud υποδομή και να αποτελεί ενεργό μέλος αυτής. Όπως αναλύθηκε στα προηγούμενα κεφάλαια, οι κινητές συσκευές πλέον είναι σε θέση να αναλάβουν την εκτέλεση μικρών tasks, τα οποία τους αποστέλλονται από το Cloud. Θα πρέπει βέβαια, να επισημάνουμε ότι σε αυτό συμβάλλει η συνεχής εξέλιξη των έξυπνων κινητών συσκευών, οι οποίες στις μέρες μας αποτελούν πραγματικούς υπολογιστές χειρός. Στην παρούσα εργασία η μελέτη και τα πειράματα περιορίστηκαν μόνο σε ότι αφορά την επικοινωνία και την διάθεση πόρων μεταξύ των κινητών συσκευών και των Cloudlets. Δεν ερευνήθηκε το κομμάτι της επικοινωνίας του Cloudlet ή/και των κινητών συσκευών με το Cloud.

Επιπλέον, αναπτύχθηκε ένα σύστημα, με χρήση εργαλείων προσομοίωσης, το οποίο παρουσιάζει μια εναλλακτική προσέγγιση στον τομέα της δέσμευσης και του διαμοιρασμού υπολογιστικών πόρων. Στο σύστημα, που αναπτύχθηκε και παρουσιάστηκε, το Cloudlet είναι αυτό που αναζητά επιπλέον υπολογιστικούς, ενώ οι κινητές συσκευές είναι αυτές που τους προσφέρουν. Το Cloudlet είναι υπεύθυνο για το πότε και σε ποια συσκευή, θα αναθέσει μια εργασία, καθώς επίσης αναλαμβάνει και την συλλογή των αποτελεσμάτων.

Επιπλέον, χρησιμοποιώντας τα χαρακτηριστικά ενός τοπικού ασύρματου δικτύου καθώς και τις δυνατότητες του Android API υλοποιήθηκε μια LBS υπηρεσία, η οποία παρέχεται μέσω της ίδιας εφαρμογής. Στόχος, σε αυτό το κομμάτι της έρευνας και ανάπτυξης ήταν η παρουσίαση μιας μεθοδολογία εντοπισμού της θέσης ενός κινητού χρήστη μέσα σε ένα κλειστό χώρο, όπως είναι αυτός του κτιρίου ενός Πανεπιστημίου, καθώς και η παροχή πληροφοριών σχετικά με τους χώρους που αυτός κινείται.

Σε αυτό το σημείο είναι σκόπιμο, να τονιστεί ότι, η εφαρμογή που παρουσιάστηκε μπορεί να επεκταθεί και να αποτελέσει τη βάση για την ανάπτυξη άλλων εργασιών στο τομέα του MCC και του resource allocation.

5.2 Συμπεράσματα

Η συγκεκριμένη μεταπτυχιακή διατριβή συνέβαλε στην εκπόνηση των παρακάτω συμπερασμάτων:

- Είναι εφικτή η ανάπτυξη mobile εφαρμογών, στις οποίες ο χρήστης να έχει τη δυνατότητα δέσμευσης και διαμοιρασμού υπολογιστικών πόρων μιας κινητής συσκευής.
- Οι κινητές συσκευές πλέον, μπορούν να παραχωρούν πόρους τόσο σε άλλες συσκευές όσο και σε Cloud υποδομές.
- Οι Cloudlet-Based αρχιτεκτονικές υπολογιστικού νέφους αποτελούν το μέλλον για το Mobile Cloud Computing.
- Ο εντοπισμός της θέσης ενός κινητού χρήστη εντός κλειστού χώρου είναι εφικτός με χρήση δεδομένων από το Wi-Fi δίκτυο στο οποίο είναι συνδεδεμένη η συσκευή του.

Στην ενότητα που ακολουθεί αναφέρονται προτάσεις για περαιτέρω ανάπτυξη και εξέλιξη της εργασίας – εφαρμογής, καθώς και ιδέες για νέες εργασίες βασισμένες σε αυτή. Οι προτάσεις βασίζονται στα αποτελέσματα του πειραματικού σταδίου και τα συμπεράσματα που προέκυψαν.

5.3 Μελλοντική Εργασία

Προτείνεται η υλοποίηση μια πραγματικής Cloudlet-Based υποδομής υπολογιστικού νέφους, στην οποία ένας υπολογιστής θα έχει τον ρόλο του Cloudlet και θα μοιράζει τα εργασίες σε κινητές συσκευές, οι οποίες επικοινωνούν με αυτό, ανάλογα με τους υπολογιστικούς πόρους που του προσφέρουν. Το Cloudlet θα πρέπει να ενημερώνεται συνεχώς, για τους υπολογιστικούς πόρους τους οποίους του παραχωρεί κάθε συσκευή και επιπλέον θα πρέπει να είναι υπεύθυνο για την συλλογή και αποθήκευση των αποτελεσμάτων. Σε αυτήν την περίπτωση κρίνεται απαραίτητη η ανάπτυξη κατάλληλου

αλγορίθμου κατανομής και διαχείρισης πόρων. Για την πραγματοποίηση των πειραμάτων σε πραγματική υποδομή προτείνεται η χρήση περισσότερων εκ των μια Android κινητών συσκευών. Σε αυτό το σύστημα κάθε συσκευή θα αναλαμβάνει την εκτέλεση ενός task και θα επιστρέφει τα αποτελέσματα στο Cloudlet. Για το παραπάνω απαιτείται επιπλέον υλοποίηση ώστε το σύστημα να υποστηρίζει πολλαπλές συνδέσεις.

Επίσης, στο τρίτο και στο τέταρτο κεφάλαιο αναφέρθηκε ότι, για τον υπολογισμό της μέγιστης απόδοσης μιας κινητής συσκευής σε MFLOPS, χρησιμοποιήθηκε η εφαρμογή Linpack for Android. Η εφαρμογή, η οποία υλοποιήθηκε στο πλαίσιο της παρούσας μεταπτυχιακής διατριβής, δεν έχει την δυνατότητα κατά την ώρα εκτέλεσής της να υπολογίζει την απόδοση της CPU της συσκευής. Ο αριθμός των MFLOPS, που εξήγαγε το Linpack αντιστοιχεί σε μια συγκεκριμένη έκδοση συσκευής. Αυτό έχει ως αποτέλεσμα η εφαρμογή να μην είναι κατάλληλη για εκτέλεση σε οποιαδήποτε Android κινητή συσκευή. Για τον παραπάνω λόγο προτείνεται η ενσωμάτωση στον κώδικα της εφαρμογής ενός benchmarking αλγορίθμου, ο οποίος θα υπολογίζει την απόδοση της συσκευής σε MFLOPS, μόλις ο χρήστης ενεργοποιήσει την εφαρμογή.

Ακόμη, προτείνεται η επέκταση της εφαρμογής ώστε οι κινητές συσκευές που την εκτελούν, να μπορούν να προσφέρουν εκτός από υπολογιστικούς πόρους και πόρους για την αποθήκευση δεδομένων. Ακολουθώντας μια ανάλογη μεθοδολογία, με αυτή που πραγματοποιήθηκε για την καταγραφή και την δέσμευση της διαθέσιμης CPU, προτείνεται η ανάπτυξη μιας εργασίας με στόχο την αναζήτηση λύσης και στον παραπάνω τομέα. Για την αποθήκευση δεδομένων θα μπορούσε να χρησιμοποιηθεί η κάρτα SD της Android κινητής συσκευής. Η εφαρμογή θα πρέπει να καταγράφει αυτόματα τον ελεύθερο χώρο της κάρτας SD της συσκευής και να ενημερώνει τον χρήστη. Ο χρήστης προτείνεται και σε αυτή την περίπτωση να έχει τη δυνατότητα να επιλέξει ο ίδιος το ποσοστό των πόρων που θα προσφέρει στην Cloudlet-Based υποδομή υπολογιστικού νέφους. Η εφαρμογή θα στέλνει στο Cloudlet έναν αριθμό από MBs ή GBs. Μια τέτοια λύση θα μπορούσε να χρησιμοποιηθεί για παράδειγμα για το διαμοιρασμό αρχείων, όπως εργασίες, αποτελέσματα πειραμάτων κ.ά. μεταξύ των φοιτητών στα εργαστήρια της σχολής.

Επίσης, προτείνεται η δημιουργία ενός repository σε μια Cloud υποδομή, όπως αυτή του Ωκεανού, η οποία θα περιέχει εργασίες, σημειώσεις και οποιοδήποτε άλλο υλικό

μαθημάτων. Οι χρήστες των κινητών συσκευών θα έχουν την δυνατότητα πρόσβασης στο συγκεκριμένο repository για λήψη υλικού που τους ενδιαφέρει, επικοινωνώντας πρώτα με ένα Cloudlet. Το Cloudlet θα υπεύθυνο για τη διαχείριση των αιτημάτων (requests) των κινητών συσκευών. Επιπλέον, το Cloudlet θα πρέπει να είναι σε θέση, να αποθηκεύει προσωρινά στη μνήμη του έγγραφα για τα μαθήματα με την μεγαλύτερη ζήτηση από τους φοιτητές. Με τον παραπάνω τρόπο θα ελαχιστοποιείται η επικοινωνία του Cloudlet με το Cloud. Η παραπάνω υλοποίηση μπορεί να στηριχθεί στο ήδη υλοποιημένο κομμάτι της LBS υπηρεσίας, που προσφέρει η εφαρμογή. Η εφαρμογή στο παρόν στάδιο υλοποίησης της εντοπίζει αυτόματα τη θέση του χρήστη στο χώρο του πανεπιστημίου και του παραθέτει την αντίστοιχη λίστα μαθημάτων και πληροφορίες για σχετικά με την αίθουσα, που αυτά διδάσκονται.

Τέλος, η εφαρμογή που υλοποιήθηκε μπορεί να εγκατασταθεί και να εκτελεστεί μόνο σε Android φορητές συσκευές. Ανάλογη όμως υλοποίηση μπορεί να πραγματοποιηθεί και για τις υπόλοιπες πλατφόρμες mobile λειτουργικών συστημάτων iOS και Windows Phone. Η διαφορά εντοπίζεται στην χρήση διαφορετικών τεχνολογιών ανάπτυξης κώδικα και γλώσσας προγραμματισμού. Επίσης, πρέπει να χρησιμοποιηθεί ένα αντίστοιχο benchmarking εργαλείο και επιπλέον να αναπτυχθεί η κατάλληλη μέθοδος για τον υπολογισμό των υπολογιστικών πόρων της κινητής συσκευής, που είναι δεσμευμένοι.

Βιβλιογραφία

- [1] Huerta G. and Lee D., (2010). *A Virtual Cloud Computing Provider for Mobile Devices*, 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, pp. 1 – 5, 2010.
- [2] Brandicc I., Broger J., Buyya R., Yea C. and Venugopala, S., (2009). *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility*, International Journal Future Generation Computer Systems, 25(6), pp 599-616.
- [3] Soyata T., Heinzelman W., Ba H., Kwon M. and Shi J., (2013). *Accelerating Mobile – Cloud Computing: A Survey*, Communication Infrastructures for Cloud Computing, pp. 175 – 197.
- [4] Fernando N., Loke S., W. and Rahayu W., (2013). *Mobile cloud computing: A survey*, International Journal of Future Generation Computer Systems, 29 (1), pp. 84 – 106.
- [5] Divya V., (2012). *Mobile Application with Cloud Computing*, International Journal of Scientific and Research Publications, 2 (4).
- [6] Bohn R. B., Messina J., Liu F., Tong J. and Mao J. (2011). *NIST Cloud Computing Reference Architecture*, 7th IEEE World Congress on Services, pp. 594 – 596, 2011.
- [7] Wang H., (2013). *Accelerating Mobile – Cloud Computing Using a Cloudlet*, Thesis degree, Department of Electrical and Computer Engineering Arts, University of Rochester.
- [8] Bhatnagar P., (2013). *Implementation of Mobile – Cloudlet – Cloud Architecture for Face Recognition in Cloud Computing using Android Mobile*, International Journal of Computer Applications Technology and Research, 2 (6), pp. 671 – 675.
- [9] Satyanarayanan M., Lewis G., Morris E., Simanta S., Boleng J. and Ha K., (2013). *The Role of Cloudlets in Hostile Environments*, IEEE Pervasive Computing and Communications, 12 (4), pp. 40-49.
- [10] Satyanarayanan M., Bahl, P., Caceres R. and Davies N., (2013). *The Case for VM-based Cloudlets in Mobile Computing*, IEEE Pervasive Computing and Communications, 8 (4), pp. 14 – 23.
- [11] Jararweh Y., Tawalbeh L., Ababneh F. & Dosari F., (2013). *Resource Efficient Mobile Computing using Cloudlet Infrastructure*, 9th IEEE International Conference on Mobile Ad-hoc and Sensor Networks, pp. 373 – 377, 2013.
- [12] Chen G. and Kotz D., (2000). *A Survey of Context-Aware Mobile Computing Research*, Thesis degree, Department of Computer Science, Dartmouth College.
- [13] Li Y. and Wang W., (2014). *Can Mobile Cloudlets Support Mobile Application?*, 33rd IEEE Annual International Conference on Computer Communications, pp 1 – 9, 2014.

- [14] Beloglazov A., Buyya R., Calheiros R. N., César A. F., De Rose and Ranjan R., (2011). *CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms*, Wiley Press: Journal Software – Practice & Experience, 41 (1), pp. 23 – 50.
- [15] Goyal M. and Singh S., (2014). *Mobile Cloud Computing*, International Journal of Enhanced Research in Science Technology & Engineering, 2 (4), pp. 517 – 521.
- [16] Singhal M. and Shukla A., (2012). *Implementation of Location Based Services in Android using GPS and Web Service*, International Journal of Computer Science Issues, 9 (1), pp. 237 – 242.
- [17] Kushwaha A. and Kushwaha V., (2011). *Location Based Services using Android Mobile Operating System*, International Journal of Advances in Engineering and Technology, 1 (1), pp. 14 – 20.
- [18] Olsson M. and Skanberg S. (2011). *Resource reservation and power management in Android*. Thesis Degree, Department of Automatic Control, university of Lund.
- [19] Lal M. and Pandey R. A. (2012). *Scheduling schema for controlling allocation of CPU resources for mobile programs*, Thesis Degree, Computer Science Department, University of California.
- [20] Cadenhead R. (2013). *Πλήρες Εγχειρίδιο της Java 7*. Αθήνα: Εκδόσεις Γκιούρδας Μ.
- [21] Darcey L. & Shane C. (2012). *Μάθετε την Ανάπτυξη Εφαρμογών για το Android σε 24 ώρες*. Αθήνα: Εκδόσεις Γκιούρδας Μ.
- [22] Deitel A., Deitel H., Deitel, P. & Morgano M. (2012). *Android για Προγραμματιστές Μάθετε μέσα από Εφαρμογές*. Αθήνα: Εκδόσεις Γκιούρδας Μ.
- [23] Afnan F., Abderrahmen M. and Khaled A. (2013). *Making the Case for Computational Offloading in Mobile Device Clouds*, Thesis Degree, School of Computer Science of Carnegie Mellon University.
- [24] Oracle. *Java Documentation*. Διαθέσιμο: <https://www.oracle.com/java/index.html>. Προσπελάστηκε: 9η Νοεμβρίου 2014.
- [25] Eclipse. *Eclipse IDE Documentation*. Διαθέσιμο: <https://eclipse.org/>. Προσπελάστηκε: 14η Νοεμβρίου 2014.
- [26] Android Developers. *Android SDK Documentation*. Διαθέσιμο: <http://developer.android.com/index.html>. Προσπελάστηκε: 22η Δεκεμβρίου 2014.
- [27] Ladha, A. (2013). *CloudSim: A Framework for modeling and simulation*. Διαθέσιμο: <http://www.academia.edu/3755846/cloudsim>. Προσπελάστηκε: 20η Ιουνίου 2015.
- [28] CloudSim API. *API Documentation*. Διαθέσιμο: <http://www.cloudbus.org/cloudsim/doc/api/overview-tree.html>. Προσπελάστηκε: 2η Απριλίου 2015.

- [29] Netlib. *Linpack Benchmark*. Διαθέσιμο: <http://www.netlib.org/utk/people/JackDongarra/faq-linpack.html>. Προσπελάστηκε: 11η Μαΐου 2015.
- [30] Oracle. *Java Sockets*. Διαθέσιμο: <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>. Προσπελάστηκε: 20η Ιουλίου. 2015.
- [31] Mukesh G., Sukhwinder S., (2014). *Mobile Cloud Computing*, International Journal of Enhanced Research in Science Technology & Engineering, 3 (4), pp.517 – 521.

Παράρτημα

Στο συνοδευτικό CD περιέχει όλα τα εκτελέσιμα αρχεία των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση του πρακτικού μέρους της παρούσας μεταπτυχιακής διατριβής, καθώς επίσης και όλες οι προγραμματιστικές βιβλιοθήκες και plug-ins:

- ❖ **Java:** JDK 8u5 & JRE 8
- ❖ **Android:** Android SDK 23.0.2 & ADT-Plugin
- ❖ **IDE:** Eclipse Luna 64-bit
- ❖ **CloudSim:** CloudSim API 3.0.3 & Commons Math 3 Library
- ❖ **Linpack Benchmark:** Linpack for Android App

Επιπλέον, στο αρχείο Instruction υπάρχουν οδηγίες για τη διαδικασία εγκατάστασης των παραπάνω εργαλείων, καθώς επίσης και για τον τρόπο εκτέλεσης της εφαρμογής. Στο CD επίσης, είναι αποθηκευμένα τα Projects: CloudSharingDocs (Mobile Application) και MyCloudlet (CloudSim Application) μαζί με τον πηγαίο κώδικα τους. Τέλος, το CD περιέχει το βιβλίο της μεταπτυχιακής διατριβής σε ηλεκτρονική μορφή, καθώς επίσης και με την παρουσίαση της.