ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΉΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

-ΕΙΣΑΓΕΤΕ ΤΟΝ ΤΟΜΕΑ-
-ΕΙΣΑΓΕΤΕ ΤΟ ΕΡΓΑΣΤΗΡΙΟ-

-Εισάγετε τον τίτλο της διπλωματικής εργασίας-

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

-Εισάγετε το σωστό άρθρο (της/του/των)-

-Εισάγετε το όνομα, αρχικό πατρώνυμου και επώνυμο των συγγραφέων-

-another author, whose name is here-

Επιβλέπων: -Εισάγετε το όνομα, αρχικό πατρώνυμου και επίθετο-
-Εισάγετε τον τίτλο του επιβλέποντα-

Αθήνα, -Εισάγετε το μήνα και το έτος κατάθεσης της εργασίας-

ΕΘΝΙΚΌ ΜΕΤΣΌΒΙΟ ΠΟΛΥΤΕΧΝΕΊΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
-ΕΙΣΑΓΕΤΕ ΤΟΝ ΤΟΜΕΑ-
-ΕΙΣΑΓΕΤΕ ΤΟ ΕΡΓΑΣΤΗΡΙΟ-

-Εισάγετε τον τίτλο της διπλωματικής εργασίας-

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

-Εισάγετε το σωστό άρθρο (της/του/των)-

-Εισάγετε το όνομα, αρχικό πατρώνυμου και επώνυμο των συγγραφέων-

-another author, whose name is here-

Επιβλέπων: -Εισάγετε το όνομα, αρχικό πατρώνυμου και επίθετο-
-Εισάγετε τον τίτλο του επιβλέποντα-

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την -εισάγετε ημερομηνία-.

.....	.....	.....
-Εισάγετε ονοματεπώνυμο-	-Εισάγετε ονοματεπώνυμο-	-Εισάγετε ονοματεπώνυμο-
-Εισάγετε τίτλο-	-Εισάγετε τίτλο-	-Εισάγετε τίτλο-

Αθήνα, (Εισάγετε το μήνα και το έτος κατάθεσης της εργασίας).

.....
(Εισάγετε όνομα, αρχικό πατρώνυμου και επίθετο συγγραφέα)
(Εισάγετε τον απονεμηθέντα τίτλο του συγγραφέα)

.....
(Εισάγετε όνομα, αρχικό πατρώνυμου και επίθετο συγγραφέα)
(Εισάγετε τον απονεμηθέντα τίτλο του συγγραφέα)

.....
(Εισάγετε όνομα, αρχικό πατρώνυμου και επίθετο συγγραφέα)
(Εισάγετε τον απονεμηθέντα τίτλο του συγγραφέα)

Περίληψη

Τα τελευταία χρόνια το cloud computing αποτελεί ένα σημαντικό κεφάλαιο στη σύγχρονη επιστήμη υπολογιστών. Η κύρια τεχνολογία που χρησιμοποιείται, προκειμένου να μπορεί να υποστηριχθεί το cloud computing είναι αυτή της εικονικοποίησης. Με αυτό τον τρόπο ένα φυσικό μηχάνημα, μπορεί να φιλοξενήσει πολλά εικονικά μηχανήματα, κάθε ένα από τα οποία αποτελεί έναν αυτοδύναμο υπολογιστή. Ωστόσο, αποτελεί συχνό φαινόμενο οι εικονικές αυτές μηχανές να χρησιμοποιούνται για την εκτέλεση μίας και μόνο εφαρμογής. Αυτό έχει ως αποτέλεσμα, να χαραμείζονται πόροι σε ενέργειες που δε χρειάζονται από την εφαρμογή, αλλά είναι απαραίτητες για το λειτουργικό σύστημα στο οποίο τρέχουν αυτές οι εφαρμογές.

Μία νεότερη τάση για την υποστήριξη του cloud computing είναι τα containers, που προσφέρουν ελαφρύτερη εικονικοποίηση με γρηγορους χρόνους εκτέλεσης, μικρή κατανάλωση μνήμης και άλλα πλεονεκτήματα. Από την άλλη, παρουσιάζουν αρκετά σημαντικά ζητήματα που αφορούν την ασφάλεια. Ένα από τα ζητήματα αυτά είναι, εκείνο της απομόνωσης το οποίο αναγκάζει σε αρκετές περιπτώσεις να οδηγεί στη χρήση εικονικών μηχανών για τη φιλοξενία των containers, χάνοντας αρκετά από τα πλεονεκτήματα τους.

Μία ακόμη προσέγγιση στο θέμα είναι οι unikernels. Πρόκειται για μία εικόνα εικονικής μηχανής, με ένα μόνο address space το οποίο κατασκευάζεται από library operating systems και είναι ειδικευμένο για μία συγκεκριμένη εφαρμογή. Πιο απλά, περιέχει τον κώδικα της εφαρμογής και ακριβώς ό,τι κομμάτι του λειτουργικού συστήματος χρειάζεται η εφαρμογή για να λειτουργήσει η διεργασία (drivers, βιβλιοθήκες, κ.λ.π.), ενοποιημένα σαν ένα αυτόνομο πρόγραμμα που μπορεί να τρέξει ως εικονική μηχανή. Οι unikernels καταφέρνουν να έχουν γρήγορους χρόνους εκκίνησης και μικρή κατανάλωση μνήμης, χωρίς να θυσιάζεται η ασφάλεια Εν τούτοις, ένα πρόβλημα είναι ότι οι unikernels υποστηρίζουν μία και μόνο διεργασία, με αποτέλεσμα να μην μπορούν εφαρμογές με παραπάνω από μία διεργασίες να εκτελεστούν σε unikernels.

Σκοπός, λοιπόν, αυτής της εργασίας είναι η υλοποίηση ενός μηχανισμού που θα επιτρέπει σε εφαρμογές με περισσότερες από μία διεργασίες να μπορούν να εκτελεστούν και σε unikernels. Επιπλέον, υλοποιείται και ένας απλός μηχανισμός για επικοινωνία μεταξύ των εικονικών μηχανών, στα πρότυπα του pipe.

Λέξεις-Κλειδιά: εικονικοποίηση, εικονικές μηχανές, ενδοεπικοινωνία εικονικών μηχανών, unikernel, kvm, QEMU

Abstract

In recent years cloud computing is one important chapter of modern commputer science. The main technology used in order to support cloud computing is virtualization. Virtualization makes possible for a physical machine to host many virtual machines, each one of which is a self-sufficient computer. However, virtual machines are often used to execute a single application. As a result, resources are devoted to actions that are not needed by the application, but they are necessary for the operating system in which the applications run.

An another technology to support cloud computing is containers which offer lightweight virtualization, fast instantiation times and small per-instance memory footprints among other features. On the other hand, containers have several important security issues. Isolation is one of these issues , which in several cases leads to the use of virtual machines to host the containers, losing several of their advantages.

A further approach in cloud computing is unikernerls. Unikernels are specialised, single-address-space machine images constructed by using library operating systems and are specialised for one application. In somewhat simplified terms, unikernels consist of the application's source code and the parts of an opperating system that are necessary for the proccess to run (drivers, libraries, etc.) consolidated as a stand-alone virtual machine. The Unikernels manage to have fast instantiation times, small memory footprints, without sacrificing security. However, one of the problems of unikernels is that they are single-proccess and as a result multi-proccess applcations are not able to run on unikernels.

The purpose of this thesis is to implement a mechanism that will enable the execution of multi-proccess applications on unikernels. Furthermore, a pipe-like mechanism for inter-vm communication is impemented.

Keywords: virtualization, virtual machines, inter-vm communication, unikernel, kvm, QEMU

Περιεχόμενα

Κατάλογος σχημάτων

Κεφάλαιο 1

Εισαγωγή

Στις μέρες μας το cloud computing έχει γίνει ένα από τα ταχύτερα αναπτυσσόμενα και ενδιαφέροντα θέματα στην επιστήμη των υπολογιστών. Το cloud computing δίνει τη δυνατότητα σε απομακρυσμένους χρήστες να αποκτάνε πρόσβαση σε υπολογιστικούς πόρους (αποθηκευτικός χώρος, εφαρμογές, υπηρεσίες κ.λ.π.) όταν χρειαστούν από τους χρήστες. Ιδιαίτερα, τα τελευταία χρόνια το cloud έχει μπει και στις ζωές των απλών χρηστών. Η χρήση των προσωπικών υπολογιστών αρχίζει να αλλάζει σημαντικά, καθώς πλέον προγράμματα και δεδομένα απομακρύνονται από τους προσωπικούς υπολογιστές για να εκτελεστούν και να αποθηκευτούν στο λεγόμενο cloud.

Μία από τις βασικές τεχνολογίες που κρύβονται πίσω από το cloud είναι αυτή της εικονικοποίησης. Χάρις την εικονικοποίηση, μπορούμε σε ένα φυσικό μηχάνημα να φιλοξενήσουμε πολλά εικονικά μηχανήματα κάθε ένα από τα οποία είναι ανεξάρτητο. Με αυτό τον τρόπο, μπορούμε να αξιοποιήσουμε καλύτερα τους φυσικούς πόρους του μηχανήματος και να τους διαμοιράσουμε όπως επιθυμούμε μεταξύ των εικονικών μηχανών. Ακόμα, η εικονικοποίηση δημιούργησε και κάποιες νέες δυνατότητες, όπως αυτή της μεταφοράς εικονικών μηχανών σε διαφορετικό φυσικό μηχάνημα, η δημιουργία αντιγράφων εικονικών μηχανών, αλλά και περισσότερη ασφάλεια, αφού ένα πρόβλημα σε ένα εικονικό μηχάνημα δε θα επηρεάσει ούτε το φυσικό ούτε τα υπόλοιπα εικονικά μηχανήματα.

Ένα συχνό φαινόμενο κατά τη χρήση της εικονικοποίησης, είναι η χρήση μία εικονικής μηχανής με συμβατικά λειτουργικά συστήματα για την υποστήριξη μίας και μόνο υπηρεσίας. Όπως είναι φυσικό το λειτουργικό σύσστημα μέσα στην εικονική μηχανή χρειάζεται κάποιους πόρους για να λειτουργήσει, ενώ συχνά μπορεί να εκτελεί λειτουργίες οι οποίες δε χρειάζονται από την εφαρμογή. Ακόμα, ο κώδικας όλου του λειτουργικού συστήματος αυξάνει αρκετά και το μέγεθος σε μνήμη της εικονικής μηχανής. Γίνεται λοιπόν, κατανοητό ότι σπαταλούνται πόροι, οι οποίοι θα μπορούσα να διατεθούν είτε στην ίδια την υπηρεσία είτε σε κάποια άλλη.

Για την επίλυση του παραπάνω προβλήμαατος αναζητήθηκαν λύσεις για να γίνει η εικονικοποίηση πιο ελαφρυά, αλλά και να μη σπαταλούνται πόροι σε αχρείαστες λειτουργίες. Μία από αυτές τις λύσεις, ήταν η εικονικοποίηση σε επίπεδο λειτουργικού συστήματος ή διαφορετικά containerization. Με τη συγκεκριμένη μέθοδο ο πυρήνας επιτρέπει την ύπαρξη πολλαπλών απομονωμένων user-space instances, που ονομάζονται containers. Τα containers μοιράζονται μεταξύ τους το λειτουργικό σύστημα στα οποία εκτελούνται, ενώ ταυτόχρονα παρέχουν ένα απομονωμένο περιβάλλον για τις διεργασίες μέσα σε αυτό.

Η τεχνολογιά των containers, όπως κάθε άλλη τεχνολογιά δε θα μπορούσε να μην έχει και κάποια μειονεκτήματα. Ένα από τα κύρια μειονεκτήματα, είναι αυτό της ασφάλειας. Τα containers μοιράζονταai τον ίδιο πυρήνα του host, ενώ οι μηχανισμοί απομόνωσης δεν είναι το ίδιο ισχυροί με αυτούς στα εικονικά μηχανήματα. Μάλιστα, έχουν αναφερθεί περιπτώσεις όπου από ένα container μπορούσαν να παρθούν πληροφορίες και δεδομένα τόσο για το host, όσο και για άλλα containers [1]. Όλα αυτά έχουν οδηγήσει σε περιπτώσεις όπου τα containers χρησιμοποιούνται πάνω από ένα πλήρη λειτουργικό σύστημα το οποίο τρέχει μέσα σε μία εικονική μηχανή.

Μία ιδέα που αρχίζει να αποκτά όλο και περισσότερο ενδιαφέρον και προσοχή είναι αυτή των unikernels. Η βάση της ιδέας, είναι η κατασκευή ειδικευμένω εικονικών μηχανών για κάθε ξεχωριστή υπηρεσία. Στην εικονική αυτή μηχανή δε χρειάζεται να υπάρχει ένα πλήρη λειτουργικό σύστημα, αλλά μόνο τα κομμάτια αυτού τα οποία είναι απαραίτητα για την εκτέλεση της υπηρεσίας. Με αυτό τον τρόπο, μειώνεται σημαντικά το μέγεθος των εικονικών μηχανών, ενώ πλέον οι πόροι χρησιμοποιούνται ακριβώς για τις λειτουργίες που χρειάζεται η υπηρεσία. Τέλος, εφόσον αναφερόμαστε σε εικονικές μηχανές, η απομόνωση τους είναι κάτι το οποίο έχουν φροντίσει ήδη οι ελεγκτές.

1.1 Σκοπός

Η φιλοσοφία των unikernels είναι ότι κάθε εικονική μηχανή θα έχει ένα συγκεκριμένο σκοπό. Για το λόγο αυτό οι unikernels δεν υποστηρίζουν παραπάνω από μία διεργασία σε κάθε εικονική μηχανή. Από την άλλη, υπάρχουν unikernel frameworks τα οποία επιτρέπουν την υποστήριξη περισσότερων από ένα νήμα. Ωστόσο οι περισσότερες εφαρμογές και υπηρεσίες στο cloud είναι φτιαγμένες για ένα πλήρη λειτουργικό σύστημα. Επομένως, προκειμένου να μπορούν να τρέξουν σε ένα unikernel θα πρέπει να αλλάξει άλλοτε περισσότερο και άλλοτε λιγότερο ο σχεδιασμός τους. Ιδιαίτερα, οι εφαρμογές που χρησιμοποιούν παραπάνω από μία διεργασίες θα πρέπει να αλλάξουν

σε ένα μοντέλο με μία μόνο διεργασία.

Ο σκοπός της συγκεκριμένης εργασίας είναι να υλοποιήσει, κρατώντας το single-proccess χαρακτηριστικό των unikernels, ένα μηχανισμό για την υποστήριξη των εφαρμογών που απαιτούν παραπάνω από μία διεργασία.

Αρχικά έγινε μία μελέτη γύρω από τα υπάρχοντα unikernel frameworks, παρατηρώντας τα χαρακτηριστικά του κάθε ενός. Στη συνέχεια σε ένα από αυτά τα unikernels (rumprun) υλοποιήθηκαν οι δύο παρακάτω λειτουργίες:

- ένας μηχανισμός για επικοινωνία μεταξύ των vms, ο οποίος είναι στα πρότυπα της κλήσης συστήματος pipe (POSIX). Ουσιαστικά πρόκειται για την υλοποίηση της pipe σε επίπεδο εικονικών μηχανών.
- ένας μηχανισμός για την υποστήριξη της κλήσης συστήματος fork από τους unikernels. Όταν μία εφαρμογή θα χρησιμοποιεί τη συγκεκριμένη κλήση συστήματος, θα δημιουργείται μία νέα εικονική μηχανή κλώνος της αρχικής και θα ξεκινά την εκτέλεση της, ακριβώς μετά την κλήση συστήματος fork.

1.2 Οργάνωση Κειμένου

Στη συνέχεια παρουσιάζεται αναλυτικά η περιγραφή του σχεδιασμού και της υλοποίησης των δύο λειτουργιών pipe, fork, γίνεται μία ανασκόπηση στα ήδη υπάρχοντα unikernel frameworks και περιγράφεται το απαραίτητο θεωρητικό υπόβαθρο.

Πιο συγκεκριμένα, στο Κεφάλαιο 2 καλύπτεται το απαραίτητο θεωρητικό υπόβαθρο, κάνοντας μία μικρή αναφορά για το cloud computing, μία εισαγωγή στην εικονικοποίηση και στα λειτουργικά συστήματα.

Στο Κεφάλαιο 3 παρουσιάζονται τα unikernel frameworks που μελετήθηκαν, τα χαρακτηριστικά τους, η φιλοσοφία τους κ.λ.π.

Στο Κεφάλαιο 4 γίνεται αναλυτική περιγραφή του σχεδιασμού και της υλοποίησης των δύο λειτουργιών που αναφέρθηκαν προηγουμένως στο σκοπό της εργασίας (pipe, fork).

Τέλος,στο Κεφάλαιο 6 αναφέρονται τα τελικά συμπεράσματα της παρούσας μελέτης όπως επίσης και πιθανές μελλοντικές επεκτάσεις αυτής της διπλωματικής εργασίας.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

μπλα μπλα μπλα

2.1 Mpla

dsfdsgfdsgdfs

2.2 Mpla

dsfdsgfdsgdfs

2.2.1 Mpla

dsfdsgfdsgdfs

Κεφάλαιο 3

Σχεδιασμός και υλοποίηση

μπλα μπλα μπλα

3.1 Mpla

dsfdsgfdsgdfs

3.2 Mpla

dsfdsgfdsgdfs

3.2.1 Mpla

dsfdsgfdsgdfs

Κεφάλαιο 4

Επίλογος

μπλα μπλα μπλα

4.1 Mpla

dsfdsgfdsgdfs

4.2 Mpla

dsfdsgfdsgdfs

4.2.1 Mpla

dsfdsgfdsgdfs

mpla [1]

Βιβλιογραφία

[1] Xing Gao, Zhongshu Gu, Mehmet Kayaalp, Dimitrios Pendarakis, and Haining Wang. Containerleaks: emerging security threats of information leakages in container clouds. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*, pages 237–248. IEEE, 2017.