

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση Open Solaris για την παροχή οπτικοακουστικού υλικού με ποιότητα υπηρεσίας.

ΜΑΝΟΣ ΚΑΡΑΠΠΕΡΑΚΗΣ
ΓΙΑΝΝΗΣ ΣΦΑΚΙΑΝΑΚΗΣ

Επιβλέπων καθηγητής: Γεώργιος Ξυλούρης.

ΗΡΑΚΛΕΙΟ 2011

Μάνος Καραπιπεράκης, Γιάννης Σφακιανάκης

1

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τον επόπτη καθηγητή της εργασίας μας κύριο Ξυλούρη Γιώργο, για την εμπιστοσύνη που μας έδειξε αναθέτοντάς μας αυτή την εργασία, για την πολύτιμη βοήθεια και καθοδήγησή του καθ' όλη τη διάρκειά της και κυρίως για την ευκαιρία που μας έδωσε να ασχοληθούμε με ένα πολύ ενδιαφέρον αντικείμενο.

Σύνοψη

Το OpenSolaris, παρέχει νέα δυναμικά δικτυακά χαρακτηριστικά για διαχείριση δικτυακών πόρων και ιδεατών δικτύων. Οι δυνατότητες Network Virtualization δίνουν τις επιλογές να μπορεί κάποιος να ιδεατοποιήσει ολόκληρες δικτυακές τοπολογίες με εξυπηρετητές, δρομολογητές, μεταγωγείς, τείχη προστασίας που θα τρέχουν όλα στην ίδια πλατφόρμα χωρίς να απαιτείται περαιτέρω επένδυση σε δικτυακό λογισμικό. Η Διαχείριση Δικτυακών Πόρων, δίνει την δυνατότητα για κάθε δικτυακή διεπαφή να προσδιοριστούν τα όρια στην χρήση εύρους ζώνης, προτεραιότητες ανά κίνηση και οι πόροι CPU για διαχείριση δικτυακής κίνησης. Οι δυνατότητες αυτές μπορούν να εφαρμοστούν σε κάθε πραγματική ή ιδεατή δικτυακή διεπαφή. Στα πλαίσια της διπλωματικής μας εργασίας σχεδιάσαμε μια δικτυακή υποδομή για παροχή πολυμεσικών υπηρεσιών με ποιότητα υπηρεσίας, μέσω της χρήσης του network virtualization. Για την υλοποίηση αυτής της δικτυακής υποδομής έγινε χρήση του λειτουργικού περιβάλλοντος της SUN OpenSolaris. Πέραν της υλοποίησης εκτελέσαμε τα απαραίτητα εργαστηριακά πειράματα προκειμένου να επικυρώσουμε την δικτυακή υποδομή. Τέλος δοκιμάσαμε τις επιδόσεις της δικτυακής υποδομής που υλοποιήσαμε με συγκεκριμένα σενάρια λειτουργίας δείχνουν την βελτίωση των επιδόσεων των υπηρεσιών που παρέχονται.

Abstract

Open Solaris provides a dynamic web feature to manage network resources and virtual networks. Network Virtualization capabilities give the options to be able to virtualize whole network topologies with servers, routers, switches and firewalls will all run on the same platform without the need for further investment in network hardware. Network Resource Manager enables each network interface to determine Limits on bandwidth usage Priorities, per move CPU resources and manage network traffic. These capabilities can be applied to any real or virtual network interface. In this thesis, we designed a network infrastructure for providing multimedia services with quality service, using network virtualization. To implement this network infrastructure we make use of the operating environment of Open Solaris. In addition to implementing we performed the necessary tests to validate the network infrastructure. Finally, we tested the performance of network infrastructure implementation with specific scenarios to illustrate the improved performance of the services which are provided.

Πινάκας περιεχομένων

Πίνακας περιεχομένων

ΕΥΧΑΡΙΣΤΙΕΣ	2
Σύνοψη	2
ABSTRACT.....	2
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	3
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ.....	5
ΚΕΦΑΛΑΙΟ 1	6
ΕΙΣΑΓΩΓΗ.....	6
1.1 Σκοπός και στόχοι εργασίας.....	6
1.2 Δομή εργασίας.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
ΚΕΦΑΛΑΙΟ 2 VIRTUALIZATION	7
2.1 Ορισμός.....	7
2.1.1 Πλεονεκτήματα	7
2.1.2 Αναδρομή στο virtualization	8
2.1.3 Εφαρμογές.....	8
2.1.4 Εφαρμογές που υπάρχουν στην αγορά.....	10
2.2 Εικονικοποίηση πλατφόρμας	14
2.2.1 Είδη εικονικοποίησης πλατφόρμας	15
ΚΕΦΑΛΑΙΟ 3	17
ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ OPEN SOLARIS.....	17
3.1 Οδηγός δικτύωσης.....	17
3.2 Λειτουργικά Παραδείγματα	24
3.2.1 Παράδειγμα 1 (VNICs και ροές σε φυσική διασύνδεση).....	24
3.2.2 Παράδειγμα 2 (Virtualization με χρήση ζωνών).....	26
ΚΕΦΑΛΑΙΟ 4	31
ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ CROSSBOW	31
4.1 Εισαγωγή.....	31
4.2 ΤΕΧΝΟΛΟΓΙΕΣ VIRTUALIZATION.....	32
4.3 Αρχιτεκτονική	33
4.4 Εικονικά σενάρια δικτύωσης.....	36
4.4.1 VWireBuilder	37
ΚΕΦΑΛΑΙΟ 5	38
ΕΠΙΚΑΙΡΟΠΟΙΗΣΗ ΤΟΥ CROSSBOW.....	38
5.1 Project (Network Virtualization and resource control)	38
5.1.1 Σκοπός project.....	38
5.1.2 Υλοποίηση project.....	38
5.2 Project (Network Virtualization and resource control with return link).....	43
Μάνος Καραπιπεράκης, Γιάννης Σφακιανάκης	3

5.2.1 Σκοπός project.....	43
5.2.2 Υλοποίηση project.....	44

ΚΕΦΑΛΑΙΟ 6 49
ΣΕΝΑΡΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ ΕΙΚΟΝΙΚΗΣ ΠΛΑΤΦΟΡΜΑΣ 49

6.1 Σενάριο υλοποίησης συστήματος δυναμικής διαχείρισης εύρους ζώνης.....	49
6.1.1 Σκοπός της υλοποίησης του σεναρίου :.....	49
6.1.2 Υλοποίηση Σεναρίου.....	49
6.1.3 Παρατηρήσεις σεναρίου.....	52
6.2 Σενάριο υλοποίησης συστήματος προσομοίωσης επίγειας ψηφιακής τηλεόρασης.....	52
6.2.1 Σκοπός Σεναρίου.....	52
6.2.2 Υλοποίηση Σεναρίου.....	52
6.2.3 Παρατηρήσεις σεναρίου.....	55
6.3 Συμπεράσματα.....	56

ΠΑΡΑΡΤΗΜΑ..... 57
7.1 ΠΙΝΑΚΕΣ ΡΟΩΝ ΓΙΑ ΤΑ ΣΕΝΑΡΙΑ ΤΗΣ ΕΝΟΤΗΤΑΣ 6..... 69

7.2 Αναφορές.....	74
-------------------	----

Πίνακας εικόνων

Εικόνα 1 Πραγματική και εικονική αρχιτεκτονική.....	9
Εικόνα 2 Desktop/client virtualization.....	10
Εικόνα 3 Εικονικοποίηση πλατφόρμας (α).....	14
Εικόνα 4 Εικονικοποίηση πλατφόρμας (β).....	15
Εικόνα 5 Solaris Core Network Functionality	17
Εικόνα 6 Εικονικό δίκτυο	18
Εικόνα 7 VNIC.....	19
Εικόνα 8 Etherstub	20
Εικόνα 9 OpenSolaris Terminal	21
Εικόνα 10 Εικονικοποίηση ροών.....	21
Εικόνα 11 Παράδειγμα ρυθμίσεις ροών.....	22
Εικόνα 12 Network Virtualization	22
Εικόνα 13 Ζώνες.....	23
Εικόνα 14 Διαδικτύωση σε ζώνες.....	23
Εικόνα 15 VNIC \$ Ροές.....	24
Εικόνα 16 Απενεργοποίηση nwmad	25
Εικόνα 17 Σχέδιο για υλοποίηση στο OpenSolaris.....	27
Εικόνα 18 Αρχιτεκτονική Crossbow.....	34
Εικόνα 19 Λωρίδες κίνησης ταξινομημένες από το OS.....	35
Εικόνα 20 vWireBuilder	37
Εικόνα 21 Crossbow virtual network.....	38
Εικόνα 22 Crossbow virtual network with return link	43
Εικόνα 23 Γραφική παράσταση throughput.....	48
Εικόνα 24 Σχηματική Αναπαράσταση του Σεναρίου.....	49
Εικόνα 25 Γραφική παράσταση πρώτης ροής UDP προς User1.....	50
Εικόνα 26 Γραφική παράσταση των τιμών του Netperf User2(Mbps).....	51
Εικόνα 27 Γραφική παράσταση δεύτερης ροής UDP προς User2.....	51
Εικόνα 28 σχηματική αναπαράσταση του σεναρίου.....	52
Εικόνα 29 Γραφική παράσταση των τιμών του Netperf User2 με την low priority TCP ροη.....	53
Εικόνα 30 Γραφική παράσταση των τιμών του Netperf User1 με την high priority tcp ροη.....	54
Εικόνα 31 Γραφική παράσταση των τιμών του Netperf User1 με την 2Mbps UDP ροη.....	54
Εικόνα 32 Γραφική παράσταση των τιμών του Netperf TCP ροης προς User1 καθώς εισέρχεται μια UDP 2 Mbps ροη στο ίδιο κανάλι.....	55

Κεφάλαιο 1

Εισαγωγή

Η ανάγκη για καλύτερη παροχή υπηρεσιών στους χρηστές έρχεται σε αντίθεση με την αύξηση του κόστους αγοράς λειτουργίας διαχείρισης αλλά και συντήρησης της υποδομής η όποια παρέχει τις πολυμεσικές υπηρεσίες. Αυτό μας οδήγησε στο να ασχοληθούμε με την μελέτη του virtualization και πιο συγκεκριμένα στην χρήση του Open Solaris για την διαχείριση δικτυακών πόρων το οποίο μας δίνει την δυνατότητα να προσδιορίσουμε τα όρια στην χρήση εύρους ζώνης καθώς επίσης και τον καθορισμό της κατανομής της δικτυακής κίνησης. Μέσω του Network Virtualization υλοποιήσαμε κάποιες δικτυακές τοπολογίες οι οποίες δουλεύουν πάνω στην ίδια δικτυακή πλατφόρμα χωρίς να δημιουργείται κάποια περαιτέρω ανάγκη σε δικτυακό λογισμικό. Το συμπέρασμα αυτής της εργασίας από την εννοιολογική της άποψη είναι ότι μας δόθηκε η δυνατότητα να ερευνήσουμε διάφορους τρόπους δημιουργίας μια δικτυακής υποδομής με αρκετές ικανότητες η οποία μπορεί να είναι εξ ολοκλήρου υλοποιήσιμη εικονικά σε έναν server πράγμα το οποίο κάνει αρκετά μεγάλη εξοικονόμηση πόρων σε σχέση με μια απτή δικτυακή δομή αφενός και αφετέρου είναι πιο χρηστική και πιο οικονομική όσον αφορά την όλη διαχείριση της.

Στα πλαίσια της διπλωματικής μας εργασίας σχεδιάσαμε μια δικτυακή υποδομή για παροχή πολυμεσικών υπηρεσιών με ποιότητα υπηρεσίας, μέσω της χρήσης network virtualization. Για την υλοποίηση αυτής της δικτυακής υποδομής έγινε χρήση του λειτουργικού περιβάλλοντος του OpenSolaris. Πέραν της υλοποίησης εκτελέσαμε τα απαραίτητα εργαστηριακά πειράματα προκειμένου να επικυρώσουμε την δικτυακή υποδομή που σχεδιάσαμε. Τέλος δοκιμάσαμε τις επιδόσεις των δικτυακών υποδομών που υλοποιήσαμε με συγκεκριμένα σενάρια λειτουργίας τα όποια δείχνουν την βελτίωση των επιδόσεων των υπηρεσιών που παρέχονται.

1.1 Σκοπός και στόχοι εργασίας

Η πτυχιακή αυτή εργασία επικεντρώνεται στην μελέτη, σχεδίαση, υλοποίηση και αξιολόγηση της δικτυακής υποδομής για παροχή πολυμεσικών υπηρεσιών με ποιότητα υπηρεσίας, μέσω της χρήσης Network Virtualization. Έχοντας ένα δίκτυο με δυναμική διαχείριση εύρους ζώνης θα πρέπει να έχουμε την δυνατότητα να διασφαλίσουμε την QoS που παρέχεται και να γίνει βελτιστοποίηση των επιδόσεων του δικτύου μας καθώς με την όλο και αυξανόμενη ζήτηση για μεγαλύτερη παροχή πολυμεσικών εφαρμογών η εξασφάλιση ποιότητας υπηρεσίας γίνεται όλο και πιο αναγκαία. Τα δίκτυα σχεδιαστήκαν για διασύνδεση ενδιάμεσων κόμβων καθώς επίσης δοκιμάστηκαν και ελέγχτηκαν για την ορθότητα τους με την χρήση του λειτουργικού περιβάλλοντος του Open Solaris.

Αρχικά, στην ενότητα 2 παρουσιάζουμε το Virtualization με αναφορά στα πλεονεκτήματα του και κάνουμε μια ανάδρομη στην όλη ιστορία του καθώς και στις εφαρμογές που υπάρχουν έως τώρα στην αγορά. Έπειτα αναφερόμαστε στην εικονικοποίηση πλατφόρμας και παρουσιάζουμε τα είδη της. Αντικείμενο της ενότητας 3 είναι η εκτενής αναφορά στο Open Solaris. Ποιο συγκεκριμένα υλοποιούμε 2 παραδείγματα όπου αναλύουμε την σχεδίαση της δικτυακής υποδομής. Η ενότητα 4 ασχολείται με την παρουσίαση του Project Crossbow. Αρχικά γίνεται μια εισαγωγή στην λειτουργία καθώς επίσης μια ανάλυση στην αρχιτεκτονική του Crossbow. Στην συνέχεια αναφερόμαστε σε εικονικά σενάρια δικτύωσης και πιο συγκεκριμένα στο πρόγραμμα vWire Builder. Η ολοκλήρωση της εργασίας γίνεται στην ενότητα 5 όπου γίνεται η επικαιροποίηση του Crossbow. Πιο αναλυτικά υλοποιούμε δυο Projects όπου σχεδιάσαμε δυο δικτυακές υποδομές προκειμένου να επικυρώσουμε την λειτουργία του Crossbow. Στην συνέχεια, στο κεφάλαιο 6 υλοποιούμε δυο σενάρια όπου το πρώτο αναφέρεται στην υλοποίηση συστήματος δυναμικής διαχείρισης εύρους ζώνης και στο δεύτερο εξετάζουμε ένα σενάριο υλοποίησης συστήματος επίγειας ψηφιακής τηλεόρασης. Έπειτα ασχολούμαστε με την στοιχειοθέτηση των πειραματικών μετρήσεων που πραγματοποιήθηκαν στα πλαίσια της πτυχιακής εργασίας. Τα αποτελέσματα των μετρήσεων που καταγράφηκαν αναλύθηκαν με στόχο την εξαγωγή όσο το δυνατόν χρησιμότερων συμπερασμάτων σε μια προσπάθεια βελτιστοποίησης των επιδόσεων αυτών.

Κεφάλαιο 2 Virtualization

2.1 Ορισμός

Η εικονικοποίηση (virtualization) είναι ένας ευρύς όρος των υπολογιστικών συστημάτων που αναφέρεται σε έναν μηχανισμό αφαίρεσης ο οποίος αποκρύπτει λεπτομέρειες της υλοποίησης και της κατάστασης ορισμένων υπολογιστικών πόρων από χρήστες των πόρων αυτών (π.χ. εφαρμογές, άλλα συστήματα, κλπ). Η αφαίρεση αυτή μπορεί είτε να αναγκάζει έναν πόρο να συμπεριφέρεται ως πλειάδα πόρων (π.χ. μία συσκευή αποθήκευσης σε διακομιστή τοπικού δικτύου), είτε πολλαπλούς πόρους να συμπεριφέρονται ως ένας (π.χ. συσκευές αποθήκευσης σε καταναμημένα συστήματα). Η εικονικοποίηση δημιουργεί μία εξωτερική διασύνδεση η οποία αποκρύπτει την υποκείμενη υλοποίηση (π.χ. πολυπλέκοντας την πρόσβαση από διαφορετικούς χρήστες). Αυτή η προσέγγιση στην εικονικοποίηση αναφέρεται ως εικονικοποίηση πόρων. Μία άλλη προσέγγιση, ίδιας όμως νοοτροπίας, είναι η εικονικοποίηση πλατφόρμας, όπου η αφαίρεση που επιτελείται προσομοιώνει ολόκληρους υπολογιστές. Η εικονικοποίηση διαχωρίζει τις εφαρμογές και τους χρήστες από το υλικό. Οι servers (Virtual Machines) λαμβάνουν τους πόρους που πραγματικά χρειάζονται είτε όπως έχουμε προκαθορίσει ανάλογα με τις ανάγκες, είτε δυναμικά ανάλογα με τις απαιτήσεις τις κάθε στιγμής. Οι χρήστες δουλεύουν σε “virtual desktops” με ασφάλεια και οι απλοί διαχειριστές ελέγχουν πιο απλά και αποδοτικά όλη την υποδομή. Το αντίθετο της εικονικοποίησης είναι η διαφάνεια: ένας εικονικός πόρος είναι ορατός, αντιληπτός, αλλά στην πραγματικότητα ανύπαρκτος, ενώ ένας διαφανής πόρος είναι υπαρκτός αλλά αόρατος.

Σε ένα εικονικοποιημένο σύστημα συμμετέχουν τρεις παράγοντες: ο πελάτης μιας υπηρεσίας, ο παροχέας της υπηρεσίας και ένας ενδιάμεσος. Ο πελάτης και ο παροχέας αλληλεπιδρούν μέσω μίας πρότυπης διασύνδεσης, τις κλήσεις προς την οποία όμως αναχαιτίζει ο ενδιάμεσος. Ο τελευταίος επιτελεί την εικονικοποίηση λειτουργώντας ως παροχέας για τον πελάτη και ως πελάτης για τον παροχέα. Ένα παράδειγμα είναι ο μηχανισμός εικονικής μνήμης των σύγχρονων λειτουργικών συστημάτων, όπου ο διαχειριστής εικονικής μνήμης (ο ενδιάμεσος) παρεμβάλλεται μεταξύ ενός πραγματικού χώρου διευθύνσεων (παροχέας) και ενός εικονικού που γίνεται αντιληπτός από κάθε διεργασία (πελάτης). Ο ενδιάμεσος παρέχει την ψευδαίσθηση πολλών ισομεγεθών χώρων διευθύνσεων (ένας για κάθε διεργασία), ενώ στην πραγματικότητα υπάρχει μόνο ένας συνολικά (η πραγματική μνήμη). Ο πελάτης και ο παροχέας δεν γνωρίζουν τίποτα για την εικονικοποίηση και τη μεσολάβηση του ενδιάμεσου.

2.1.1 Πλεονεκτήματα

Σε γενικές γραμμές, το virtualization μας επιτρέπει να εκτελέσουμε πολλαπλό φόρτο εργασίας για το ίδιο υλικό με ένα προβλέψιμο, ασφαλές, και απομονωμένο τρόπο. Ανάλογα με το ποια λύση εικονικοποίησης θα επιλέξουμε, υπάρχει είναι διαφορετικό όφελος, συμπεριλαμβανομένων των εξής:

- Μείωση του κόστους κτήσης, λειτουργίας, διαχείρισης και συντήρησης της υποδομής πληροφορικής.
- Πλήρη εκμετάλλευση των φυσικών πόρων του Η/Υ.
- Εύκολη διαχείριση και δημιουργία νέων server/clients.
- Περιορισμό του φυσικού χώρου που καταλαμβάνουν τα υπολογιστικά συστήματα.
- Μείωση της κατανάλωσης ενέργειας και αναγκών ψύξης.
- Υψηλή διαθεσιμότητα.
- Η αύξηση της χρησιμοποίησης της μηχανής είναι δυνατή μέσω της κοινής χρήσης του συστήματος.
- Οι εφαρμογές είναι τελείως απομονωμένες.
- Εσφαλμένη συμπεριφορά διαδικασίας ή κακόβουλοι χρήστες σε μία περίπτωση δεν μπορεί να παρέμβουν στις δραστηριότητες.
- Οι βλάβες είναι απομονωμένες. Αυτό μπορεί να περιλαμβάνει απομόνωση από μια πλήρη συντριβή του λειτουργικού συστήματος.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

- Αποφεύγεται η ταυτόχρονη χρήση μια θύρας TCP.
- Η ασφάλεια βελτιώνεται. Αν το λειτουργικό σύστημα τίθεται σε κίνδυνο, τότε μόνο αυτό επηρεάζεται, ενώ το υπόλοιπο σύστημα παραμένει ασφαλές
- Cross-platform εφαρμογές μπορεί να αναπτυχθούν και να δοκιμαστούν με το ίδιο υλικό

2.1.2 Αναδρομή στο virtualization

1960:Ο όρος virtualization γεννιέται .Ο όρος “cloud computing”(υπολογιστικού νέφους) επινοήθηκε από τον John McCarthy.

1967: Πλατφόρμα virtualization καινοτόμησε η IBM με το CP-40 και αργότερα με το CP-67.

1972: Εικονικοποίηση Hardware εισήχθησαν με την IBM System/370. Paravirtualization εισήχθη με το σύστημα της IBM VM λειτουργίας.

1988: VMware, Inc

1999: Κυκλοφόρησε σταθμός εργασίας VMware.Απελευθερώνεται το cPanel 3 . Πρώρη λύση για το Linux βασισμένο σε ASP εικονικού ιδιωτικούς διακομιστές(servers)

2001: Ο Gartner προσδιορίζει ένα χρονοδιάγραμμα για την υιοθέτηση των «διαδικτυακών υπηρεσιών», μια πρόωρη απόπειρα στο πλαίσιο cloud computing. Microsoft, IBM, Sun, Software AG, η Oracle και άλλοι να υποβάλουν εργασία για την υποστήριξη του. SWsoft Virtuozzo για Linux Released εμπορικός βιώσιμη λύση virtualization OS για τις επιχειρήσεις. Κυκλοφορούν VMware ESX 1.0 και GSX 1.0.

2003: Διαθέσιμη εφαρμογή εικονικοποίησης ανοιχτού λογισμικού Xen

2004: SWsoft (Parallels) ανακοινώνει παράλληλα Desktop για Mac και παράλληλους σταθμούς εργασίας (Workstation).

2005: SWsoft (Parallels) Virtuozzo προσθέτουν υποστήριξη για το Microsoft OSS. Το Amazon δημιουργεί Amazon Web Services.

2006: XenSource 3.0 κυκλοφόρησε με την υποστήριξη για Microsoft OSS. VMware Virtual Infrastructure 3 Κυκλοφόρησε με ESX 3.0 και VirtualCenter 2.0.

2007: Parallels Server για Mac. ESXi VMware κυκλοφόρησε. VirtualBox OSE (νυν Sun xVM) διανέμεται ως ελεύθερο λογισμικό. Έρευνα του DesktopLinux.com αναφέρει ότι το VirtualBox είναι το τρίτο πιο δημοφιλές λογισμικό για τη λειτουργία των προγραμμάτων Windows σε Linux. Citrix Αποκτά XenSource και δημιουργεί Xen Server

2008: Η Microsoft ανακοινώνει τη μελλοντική διαθεσιμότητα της πλατφόρμας Windows Azure ως Cloud πλατφόρμα.

2009: Το Xen Source γίνεται ελεύθερο λογισμικό. Το VMware vSphere 4 Κυκλοφόρησε. Hyper-V R2 και Hyper-V Server R2

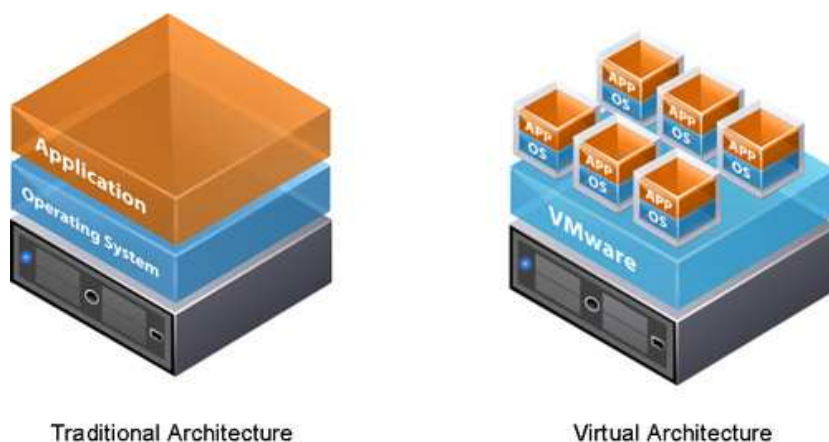
2010: Εξαγορα της Solaris από την Oracle

2.1.3 Εφαρμογές

Server Virtualization

Με τον όρο virtualization αναφερόμαστε στην τεχνολογία που συγκεντρώνει και διαμοιράζει εφαρμογές και resources υπολογιστών με αποτελεσματικό τρόπο. Στην πραγματικότητα, το virtualization επιτρέπει στο software να διαχωριστεί από το φυσικό hardware. Με τη χρήση του VMWare ESX ως πλατφόρμα που εγκαθίσταται σε πολύ χαμηλό επίπεδο (πριν το λειτουργικό σύστημα) σε φυσικά μηχανήματα, είναι εφικτή η δημιουργία πολλών «ιδεατών μηχανών» (Virtual Machines) που λειτουργούν ανεξάρτητα μεταξύ τους και χρησιμοποιούν όλους τους διαθέσιμους πόρους του φυσικού μηχανήματος (CPUs, μνήμη, δίκτυο κλπ.) είτε όπως έχουμε προκαθορίσει ανάλογα με τις ανάγκες, είτε δυναμικά ανάλογα με τις απαιτήσεις κάθε στιγμή.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris



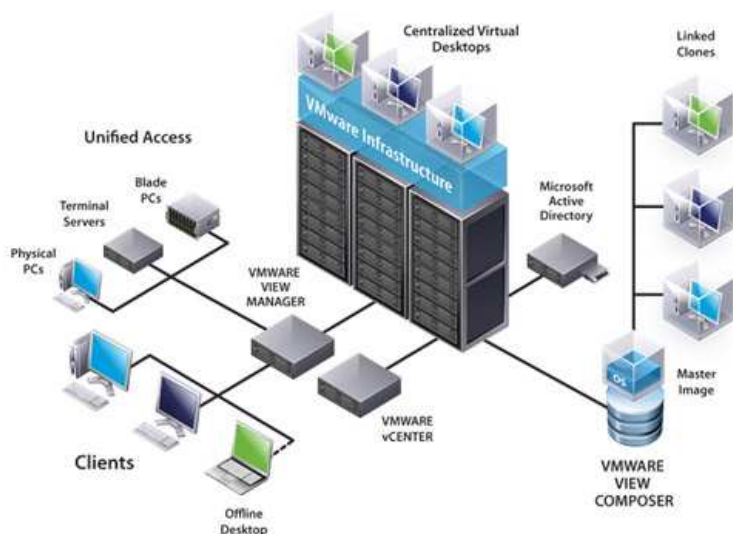
Εικόνα 1 Πραγματική και εικονική αρχιτεκτονική

Με μια λύση virtualization μειώνεται το κόστος κτήσης νέων servers, καθώς δεν είναι απαραίτητο να προμηθευτούμε ένα server για κάθε λειτουργικό σύστημα, εφαρμογή ή περιβάλλον γενικά. Μειώνεται σημαντικά το λειτουργικό κόστος καθώς εκμεταλλευόμαστε το hardware σε πολύ υψηλό βαθμό πριν χρειαστεί να προχωρήσουμε σε αναβαθμίσεις. Δεν υπάρχουν χαμένοι πόροι CPU και μνήμης (δηλαδή δυνατότητες CPU και μνήμης που κάθονται ανεκμετάλλευτες σε κάθε μηχανήμα). Όλα αυτά σημαίνουν μείωση σε φυσικά συστήματα, άρα και μικρότερες ανάγκες διαχείρισης, λιγότερο χώρο και φυσικά μικρότερη κατανάλωση ενέργειας και ανάγκη ψύξης. Τα πλεονεκτήματα όμως δεν σταματούν εκεί, αφού μεταξύ άλλων παρέχεται μεγάλη ευελιξία, η ασφάλεια και η επίτευξη υψηλής διαθεσιμότητας. Είναι δυνατή η εύκολη και πολύ γρήγορη δημιουργία νέων servers ως αντίγραφο των παλαιών, χωρίς να χρειαστεί η χρονοβόρα εγκατάσταση λειτουργικού συστήματος από την αρχή. Αυτοί είναι οι κυριότεροι λόγοι που οδηγούν μια επιχείρηση στην εφαρμογή λύσεων server virtualization.

Desktop/Client Virtualization

Οι λύσεις Desktop Virtualization βοηθούν τις επιχειρήσεις να απλοποιήσουν και να βελτιώσουν τον τρόπο που διαχειρίζονται του υπολογιστές των χρηστών τους (Clients). Οι παραδοσιακοί προσωπικοί υπολογιστές και σταθμοί εργασίας αντικαθίστανται με κεντρικά ιδεατά μηχανήματα που βρίσκονται σε κεντρικά συστήματα και προσπελαύνονται πάνω από το δίκτυο.

Με τον τρόπο αυτό ο κάθε χρήστης παύει πια να διαχειρίζεται τον υπολογιστή του ως φυσικό μηχανήμα. Έτσι η διαχείριση και υποστήριξη όλων των υπολογιστών των χρηστών γίνεται πιο αποτελεσματική. Οι χρήστες απολαμβάνουν αξιοπιστία, προστασία των δεδομένων τους και εύκολη πρόσβαση από παντού (ακόμα και από διαφορετικές τοποθεσίες και συσκευές όπως laptops, handhelds κλπ.). Το προσωπικό πληροφορικής έχει πλέον να διαχειριστεί μια σταθερή υποδομή με υψηλή ασφάλεια και προστασία των δεδομένων. Εργασίες όπως updates των υπολογιστών ή δημιουργία νέου υπολογιστή για νέους χρήστες γίνονται εύκολα, αξιόπιστα και σε ελάχιστο χρόνο



Εικόνα 2 Desktop/client virtualization

Δεν υπάρχει ανησυχία για την καταστροφή ενός υπολογιστή, αφού τίποτα δεν είναι αποθηκευμένο σε αυτόν. Με την αντικατάστασή του σε περίπτωση περίπτωση οποιασδήποτε βλάβης, ο χρήστης βρίσκεται αμέσως στο γνωστό του περιβάλλον όπως το είχε αφήσει, χωρίς να χρειαστεί καμία ανάκτηση ή αντιγραφή εφαρμογών και δεδομένων. Όλες αυτές οι δυνατότητες ήταν παλαιότερα συνυφασμένες μόνο με server-based περιβάλλοντα, αλλά είναι τώρα εύκολα υλοποιήσιμες με την τεχνολογία της VMware. Μειώνεται το κόστος της επένδυσης, αφού οι υπολογιστές των χρηστών μπορούν να είναι φτηνά μηχανήματα χωρίς ιδιαίτερες απαιτήσεις, ακόμα και χωρίς σκληρό δίσκο (Thin Clients). Η οικονομία σε κατανάλωση ενέργειας ή ακόμα και του θορύβου σε πολυπληθή περιβάλλοντα είναι εντυπωσιακά.

Storage Virtualization

Η επένδυση στην καρδιά της πληροφορίας, το σύστημα Data Storage είναι η σημαντικότερη απόφαση για ένα σύγχρονο περιβάλλον πληροφορικής. Τα δεδομένα πρέπει να είναι συνεχώς διαθέσιμα και προστατευμένα. Στις σύγχρονες λύσεις SAN Storage υπάρχουν προηγμένα εργαλεία, όπως low-level snapshots, που επιτρέπουν τη συνεχή προστασία και την άμεση ανάκαμψη σε περίπτωση φυσικού ή λογικού προβλήματος. Το storage διατίθεται ιδεατά και δυναμικά στους servers με δυνατότητα μεταβολών (π.χ. αύξηση του χώρου) χωρίς διακοπή των εφαρμογών, ενώ η τεχνολογία Deduplication επιτρέπει τη σημαντική μείωση του απαιτούμενου χώρου αποθήκευσης, γιατί παρακολουθεί και εξασφαλίζει ότι κοινές εγγραφές αποθηκεύονται μόνο μία φορά.

2.1.4 Εφαρμογές που υπάρχουν στην αγορά

Eucalyptus

Το EUCALYPTUS (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) είναι λογισμικό ανοιχτού κώδικα για την υλοποίηση υπολογιστικής νέφους (cloud computing) σε συστάδες (clusters) υπολογιστών. Η τρέχουσα διεπαφή του EUCALYPTUS είναι συμβατή με τη διεπαφή EC2 της Amazon, όμως έχει σχεδιαστεί με τέτοιο τρόπο ώστε να υποστηρίζει πολλαπλές διεπαφές πελάτη. Το EUCALYPTUS έχει υλοποιηθεί χρησιμοποιώντας διαθέσιμα εργαλεία Linux και βασικές τεχνολογίες διαδικτυακών υπηρεσιών, κάτι που καθιστά εύκολο την εγκατάσταση και συντήρησή του.

KVM

Το KVM (Kernel-based Virtual Machine) είναι εφαρμογή εικονικοποίησης του πυρήνα του Linux. Σήμερα, το KVM υποστηρίζει εγγενή εικονικοποίηση (virtualization) χρησιμοποιώντας είτε την Intel VT είτε την AMD-V, παρέχει σύστημα διαχείρισης εικονικής μνήμης του λειτουργικού συστήματος-επισκέπτη και διαθέτει βελτιστοποίηση της CPU σε περίπτωση που το λειτουργικό σύστημα-επισκέπτης είναι το Linux.

OpenVZ

Το OpenVZ είναι τεχνολογία εικονικοποίησης επιπέδου λειτουργικού συστήματος που βασίζεται στο λειτουργικό σύστημα Linux και στον πυρήνα του. Επιτρέπει σε έναν εξυπηρετητή να εκτελεί πολλαπλά απομονωμένα στιγμιότυπα ενός λειτουργικού συστήματος. Είναι παρόμοιο με το Jails του FreeBSD και το Zones του Solaris. Σε αντίθεση με εικονικές μηχανές, όπως η VMware και η OpenVZ, απαιτεί τόσο το λειτουργικό σύστημα φιλοξενίας όσο και το λειτουργικό σύστημα-επισκέπτης να είναι διανομή του Linux. Η χρήση μίας μόνο εικόνας πυρήνα οδηγεί σε σημαντική μείωση της επιβάρυνσης που προέρχεται από την εικονικοποίηση, επιβάρυνση η οποία συνήθως ανέρχεται σε μικρό ποσοστό. Επίσης, η διαχείριση όλων των εικονικών μηχανών μπορεί να γίνει και από μία κονσόλα.

OpenXVM

Πρόκειται για μηχανή εικονικοποίησης ανοιχτού κώδικα για πληροφοριακά κέντρα, η οποία παρέχει δυνατότητα διαχείρισης εξυπηρετητών σε επίπεδο υπερεπόπτη. Έχει σχεδιαστεί ώστε να μπορεί να εκτελείται σε διάφορες πλατφόρμες με υψηλή αποδοτικότητα και είναι σε θέση να φιλοξενεί πολλαπλά λειτουργικά συστήματα-επισκέπτες (π.χ. Solaris, Windows και Linux), διαθέτοντας προηγμένες δυνατότητες διαχείρισης της μνήμης και της CPU. Ο εξυπηρετητής έχει δημιουργηθεί από το έργο ανοιχτού λογισμικού Xen, καθώς και από το LDOMS της Sun. Ο εξυπηρετητής xVM μετατρέπει τον υπολογιστή σε αφοσιωμένη μηχανή λογισμικού εικονικοποίησης, η οποία διαθέτει ιδιαίτερα εύχρηστη διεπαφή που λειτουργεί μέσω μιας τυπικής σύνδεσης https.

X2Go

Πρόκειται για υπολογιστικό περιβάλλον εξυπηρετητή που συνδυάζει τα πλεονεκτήματα διάφορων υφιστάμενων λύσεων. Το x2go έχει βελτιώσει σε σημαντικό βαθμό τη χρησιμότητα αυτού του είδους των εφαρμογών και είναι κατάλληλο να χρησιμοποιηθεί τόσο σε προσωπικούς υπολογιστές όσο και σε εταιρικά δίκτυα που διαθέτουν πολλαπλούς εξυπηρετητές και δένδρα LDAP. Το x2go παρέχει γρήγορο, ασφαλές και απλό τρόπο σύνδεσης στην επιφάνεια εργασίας του χρήστη είτε μέσω ενός τοπικού δικτύου είτε διαμέσου μιας διαδικτυακής σύνδεσης χαμηλού εύρους ζώνης. Πρόκειται για εφαρμογή ανοιχτού κώδικα που είναι διαθέσιμη σε διάφορες αρχιτεκτονικές CPU. Διαθέτει αρκετά επιπρόσθετα στοιχεία τα οποία βοηθούν το χρήστη να διαχειρίζεται το περιβάλλον της εφαρμογής. Υπάρχει η δυνατότητα χρησιμοποίησης φίλτρου ζωντανής αναζήτησης, καθώς και άλλων καινοτόμων υπηρεσιών οι οποίες βοηθούν το χρήστη να ολοκληρώνει εύκολα τις καθημερινές του εργασίες. Οι σύνοδοι των χρηστών μπορούν να διακοπουν και να ακυρωθούν μέσω γραφικού περιβάλλοντος. Μέσω της υπηρεσίας sryglass μπορεί κανείς να ελέγξει όλους τους συνδεδεμένους στο περιβάλλον x2go πελάτες. Αν κάποιος επιθυμεί να δώσει τη δυνατότητα σε άλλους χρήστες να χρησιμοποιήσουν την εφαρμογή δεν έχει παρά να τους προσθέσει στην κατάλληλη λίστα χρηστών.

Xen

Ο υπερεπόπτης Xen είναι μια εφαρμογή εικονικοποίησης ανοιχτού λογισμικού η οποία παρέχει ισχυρό, αποδοτικό και ασφαλές σύνολο χαρακτηριστικών για την εικονικοποίηση αρχιτεκτονικών CPU, όπως η x86, x86_64, IA64 και PowerPC. Υποστηρίζει πληθώρα λειτουργικών συστημάτων-επισκεπτών, συμπεριλαμβανομένων των Windows, του Linux, του Solaris και διάφορων εκδόσεων των λειτουργικών συστημάτων BSD. Αποτελεί τη βάση αρκετών συστημάτων εικονικοποίησης, όπως το Citrix ή το Oracle.

Windows Server 2008 R2

Οι δυνατότητες συνένωσης και εικονικοποίησης (virtualization) διακομιστή του Windows Server 2008 R2 βοηθάει στη μείωση του κόστους και στη μετακίνηση εικονικών μηχανών χωρίς χρόνο διακοπής λειτουργίας μέσω του Live Migration.

Κορυφαίες δυνατότητες του Windows Server 2008 R2

- Η βελτιωμένη διαχείριση ενέργειας περικόπτει το κόστος υλικού και εγκαταστάσεων
- Η δυνατότητα Live Migration μετεγκαθιστά εικονικές μηχανές χωρίς να τίθενται εκτός λειτουργίας
- Το DirectAccess συνδέει απομακρυσμένους χρήστες χωρίς να απαιτείται η δημιουργία VPN
- Το Hyper-V εικονικοποιεί πόρους συστήματος χωρίς να απαιτείται λογισμικό τρίτων
- Διαλειτουργικότητα με Windows 7

System Center

Το System Center διαχειρίζεται ολόκληρο το σύστημα IT, φυσικό και εικονικό και ο οικείος σχεδιασμός εφαρμογών της Microsoft μπορεί να οδηγήσει σε καμπύλες γρήγορης μάθησης και μείωση του κόστους εκπαίδευσης. Η διαχείριση IT είναι σημαντικό θέμα, επειδή οι επιχειρήσεις χρειάζεται να κάνουν περισσότερα, σε μικρότερο χρονικό διάστημα και με λιγότερα χρήματα. Επίσης μπορεί να διαχειριστεί όλους τους διακομιστές, τους εικονικούς διακομιστές, τους επιτραπέζιους υπολογιστές, τους φορητούς υπολογιστές και τις συσκευές χειρός όλης της επιχείρησης. Το Microsoft System Center διαχειρίζεται ολόκληρο το σύστημα IT. Χειρίζεται εικονικά και φυσικά περιβάλλοντα, σε πολλαπλά λειτουργικά συστήματα, όπως UNIX και Linux και όλα αυτά από μία μόνο οθόνη. Συνδέει δεδομένα με άτομα οπουδήποτε και εάν βρίσκονται, με εικονικοποίηση (virtualization). Η οικεία σχεδίαση εφαρμογών της Microsoft μπορεί να οδηγήσει σε γρήγορες καμπύλες μάθησης και μειωμένο κόστος εκπαίδευσης.

Oracle VM VirtualBox 4.0

Το Oracle VM Virtual Box 4.0 είναι λογισμικό εικονικοποίησης (virtualization software). Το Oracle VM VirtualBox 4.0 είναι η 3η σημαντική έκδοση του προϊόντος σε μόλις ένα χρόνο, και προσθέτει στις πολλές νέες κυκλοφορίες προϊόντων σε όλη την σειρά Virtualization προϊόντων της Oracle, απεικονίζοντας την επένδυση και τη σημασία που δίνει η Oracle στην παροχή συνολικής λύση εικονικοποίησης τόσο για desktop όσο και για datacenter. Με το hardware, οι χρήστες θα ανακαλύψουν ότι το Oracle VM VirtualBox 4.0 παρέχει μια πλουσιότερη εμπειρία στο χρήστη. Το Oracle VM VirtualBox επιτρέπει σε desktop ή laptop υπολογιστές να τρέξουν πολλαπλά λειτουργικά συστήματα ταυτόχρονα, επιτρέποντας στους χρήστες να ωφεληθούν τα μέγιστα σε επίπεδο ευελιξίας και χρήσης των συστημάτων τους, και υποστηρίζει μια ποικιλία φιλοξενούμενων λειτουργικών συστημάτων, συμπεριλαμβανομένων Windows, MacOS X, δημοφιλών εκδόσεων του Linux (συμπεριλαμβανομένου του Oracle Linux), και φυσικά Oracle Solaris. Υποστηρίζοντας ένα ευρύ φάσμα λειτουργικών συστημάτων υποδοχής (host) όσο και των φιλοξενούμενων (guest) που προαναφέρθηκαν. Το Oracle VM VirtualBox 4.0 προσφέρει αυξημένη χωρητικότητα και τη δυναμικότητα να χειριστεί μεγαλύτερο φόρτο εργασίας, ενισχυμένες δυνατότητες εικονικών συσκευών, και σημαντικές βελτιώσεις χρηστικότητας. Με υποστήριξη για την τελευταία λέξη της τεχνολογίας εικονικού υλικού, συμπεριλαμβανομένων των chipsets με υποστήριξη PCI Express, επεκτείνει περαιτέρω την αξία που παραδίδει σε πελάτες, συνεργάτες, καθώς και προγραμματιστές και τεχνικούς ανάπτυξης.

A/A	ΟΝΟΜΑΣΙΑ	ΠΕΡΙΓΡΑΦΗ
1	ATL	Μια εικονική μηχανή MTL
2	Bochs	Φορητός εξομοιωτής ανοικτού κώδικα για x86 και AMD64.
3	CHARON-AXP	Παρέχει εξομοίωση του Alpha Server για να επιτρέψει την εκτέλεση των OpenVMS ή Tru64 εφαρμογών σε πλατφόρμες x86.
4	CHARON-VAX	Παρέχει εξομοίωση των PDP-11 ή VAX για να επιτρέψει εκτέλεση OpenVMS ή Tru64 σε πλατφόρμες x86 ή HP integrity.
5	CoLinux	Linux ανοικτού κώδικα μέσα σε Windows.
6	CoWare	Εικονική πλατφόρμα.
7	Denali	Χρησιμοποιεί παραεικονικοποίηση (paravirtualization) των x86 για εκτέλεση Παραεικονικοποιημένων λειτουργικών συστημάτων για προσωπικούς υπολογιστές.
8	Hercules Emulator	Free System/370, ESA/390, z/Mainframe
9	KVM	Εικονική μηχανή που εξομοιώνει πυρήνα Linux.
10	Logical Domains	Εικονικοποίηση διακομιστών και τεχνολογία διαχωρισμού.
11	LynxSecure	Χρησιμοποιεί αρχιτεκτονική MILS για να παρέχει συστήματα x86 υψηλής αξιοπιστίας.
12	Microsoft Virtual PC	Εικονικός υπολογιστής x86 της Microsoft.
13	Microsoft Virtual Server	Εικονικός υπολογιστής x86 της Microsoft
14	Oracle VM	Εικονικός υπολογιστής για διακομιστές.
15	OVPSim	Εξομοιωτής πλατφόρμας πολλών επεξεργαστών για την επίτευξη πολύ υψηλών ταχυτήτων.
16	Parallels Workstation	Εμπορική εφαρμογή εξομοίωσης x86 για εκτέλεση εικονικών υπολογιστών σε Windows ή Mac.
17	QEMY	Εξομοιωτής επεξεργαστή ανοικτού κώδικα.
18	SheepShaver	Εξομοιωτής MAC για Linux.
19	Simics	Εξομοιωτής που επιτρέπει τον ορισμό και την ανάπτυξη ψηφιακών συστημάτων.
20	Sun xVM	Δημοφιλής εικονικός υπολογιστής της SUN.
21	SVISTA	Εικονικός υπολογιστής της Serenity (Serenity Visual Station) για Intel x86.
22	twoOSTwo	Εικονικός υπολογιστής της Parallels για Intel x86.
23	User-mode Linux	Εκτελείτε σε περιβάλλον Linux και επιτρέπει την ταυτόχρονη εκτέλεση πολλών συστημάτων Linux.
24	Virtual Box	Εικονικός υπολογιστής της Oracle που εξομοιώνει και πλατφόρμες 64bit.
25	Virtual Iron	Είναι από τους πρώτους εικονικούς υπολογιστές που εξομοιώνουν Intel VT-x και AMD-V.
26	VM from IBM	Εικονικός υπολογιστής από την IBM.
27	VMLite	Εναλλακτικός τρόπος εκτέλεσης Windows XP σε Windows Vista και Windows 7.
28	VMware	Ολοκληρωμένο σύστημα διαχείρισης εικονικών υπολογιστών.
29	Xen	Επιτρέπει πολλά λειτουργικά συστήματα να εκτελούνται σε έναν φυσικό υπολογιστή.

Πίνακας 1 Λύσεις εικονικών υπολογιστών που υπάρχουν στην αγορά

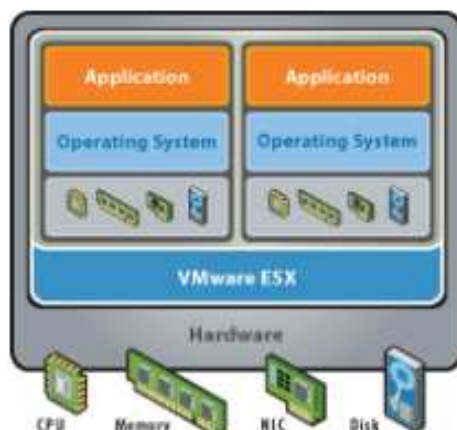
2.2 Εικονικοποίηση πλατφόρμας



Εικόνα 3 Εικονικοποίηση πλατφόρμας (α)

Ιδιαίτερο ενδιαφέρον παρουσιάζει η εικονικοποίηση πλατφόρμας, όπου ένα λογισμικό ελέγχου («επόπτης» ή hypervisor) εκτελούμενο σε πραγματικό υλικό προσομοιώνει ένα υπολογιστικό περιβάλλον, μία «εικονική μηχανή», επάνω από το οποίο μπορεί να τρέξει κάποιο φιλοξενούμενο λογισμικό (συνήθως ένας πλήρης πυρήνας), απομονωμένο από το υπόλοιπο σύστημα. Η θεμελιώδης λογική πίσω από την εικονικοποίηση πλατφόρμας είναι η αρχή πως οποιαδήποτε λειτουργία μπορεί να εκτελεστεί είτε από λογισμικό είτε από εξειδικευμένο υλικό οι μόνες διαφορές αφορούν την ευελιξία και την απόδοση. Είναι δυνατόν να προσομοιώνονται ταυτόχρονα πολλαπλές εικονικές μηχανές, εντελώς απομονωμένες μεταξύ τους, από το ίδιο λογισμικό ελέγχου. Η εικονικοποίηση πλατφόρμας εμφανίστηκε αρχικά τη δεκαετία του 1960, πριν από την επέλαση των μικροϋπολογιστών, σε μεγάλα, συγκεντρωτικά συστήματα (mainframes), αλλά μετά το 2000 και την αλματώδη αύξηση των επιδόσεων του υλικού των PC έχει γίνει πλέον κοινή πρακτική.

2.2.1 Είδη εικονικοποίησης πλατφόρμας



Εικόνα 4 Εικονικοποίηση πλατφόρμας (β)

Υπάρχουν πολλά είδη εικονικοποίησης πλατφόρμας. Ακολουθούν τα σημαντικότερα:

Εξομοίωση: η εικονική μηχανή προσομοιώνει εξολοκλήρου μία αρχιτεκτονική υλικού, πιθανώς διαφορετική από το πραγματικό υποκείμενο υλικό, επιτρέποντας έτσι να εκτελεστεί επάνω της ένα μη τροποποιημένο, φιλοξενούμενο ΛΣ σχεδιασμένο για τον εξομοιούμενο επεξεργαστή (π.χ. QEMU, έκδοση για PowerPC του VirtualPC κλπ). Η εξομοίωση σε χρόνο εκτέλεσης του κώδικα του φιλοξενούμενου ΛΣ, με έναν κύκλο ανάγνωσης-αποκωδικοποίησης-εκτέλεσης όπου κάθε εντολή που ανήκει στο σύνολο εντολών του επεξεργαστή-πηγή μεταφράζεται σε μία εντολή του συνόλου εντολών του επεξεργαστή-στόχου. Παράλληλα η εικονική μηχανή παρέχει μία αφαίρεση της μνήμης, των συσκευών Εισόδου/Εξόδου κλπ, φροντίζοντας ώστε κάθε μεταφρασμένη εντολή που απευθύνεται σε αυτά τα υποσυστήματα να τροποποιεί μόνο τις αφαιρέσεις/λογικές αναπαραστάσεις τους, οι οποίες κατευθύνονται και υλοποιούνται από το λογισμικό ελέγχου, και όχι το πραγματικό υλικό. Προκειμένου να αυξηθούν οι επιδόσεις είναι δυνατόν να χρησιμοποιηθεί δυναμική μετάφραση αντί για απλή εξομοίωση, όπου οι μεταφρασμένες εντολές αποθηκεύονται σε κρυφή μνήμη και μπορούν να επαναχρησιμοποιηθούν αργότερα χωρίς εκ νέου μετάφραση, ή δυναμική επαναμεταγλώττιση, όπου εκτός της χρήσης κρυφής μνήμης γίνεται και βελτιστοποίηση κρίσιμων τμημάτων του κώδικα (παρόμοια με τη μεταγλώττιση JIT της Java, του .NET και άλλων παρόμοιων πλατφορμών υψηλού επιπέδου).

Πλήρης: η εικονική μηχανή προσομοιώνει επαρκές τμήμα του πραγματικού υποκείμενου υλικού ώστε να επιτρέπει την εκτέλεση επάνω της ενός μη τροποποιημένου, φιλοξενούμενου ΛΣ σχεδιασμένου για τον ίδιο τύπο επεξεργαστή με την πραγματική CPU (π.χ. VirtualPC, VMWare, Win4Lin κλπ). Στην πλήρη εικονικοποίηση δεν χρειάζεται εξομοίωση του συνόλου εντολών του επεξεργαστή και μάλιστα ένα τμήμα του κώδικα του φιλοξενούμενου ΛΣ μπορεί να εκτελείται απευθείας από το υλικό, χωρίς μεσολάβηση του επόπτη, αρκεί να μην επηρεάζει υποσυστήματα εκτός του άμεσου ελέγχου του τελευταίου. Τα κρίσιμα σημεία του φιλοξενούμενου κώδικα ωστόσο, όπως αυτόπουπροσπαθούν να αποκτήσουν πρόσβαση στο υλικό (π.χ. κλήσεις συστήματος), συλλαμβάνονται από το λογισμικό ελέγχου και προσομοιώνονται, αφού τα αποτελέσματα κάθε λειτουργίας που επιτελείται σε μία εικονική μηχανή δεν επιτρέπεται να τροποποιούν την κατάσταση άλλων εικονικών μηχανών, του επόπτη ή του υλικού. Αν το πραγματικό υλικό βοηθά και επιταχύνει τη λειτουργία του λογισμικού ελέγχου τότε η πλήρης εικονικοποίηση ονομάζεται εγγενής (native). Η βοήθεια αυτή αφορά κυρίως εύκολη διάκριση μεταξύ εντολών που μπορούν να εκτελεστούν απευθείας και εντολών που πρέπει να προσομοιωθούν από το λογισμικό. Όπως και στην εξομοίωση η εικονική μηχανή παρέχει στο φιλοξενούμενο ΛΣ μία αφαίρεση της μνήμης, των συσκευών Εισόδου/Εξόδου κλπ, ενώ η εγγενής εκτέλεση μεγάλου μέρους του κώδικα παρέχει πολύ καλύτερες επιδόσεις σε σχέση με την εξομοίωση

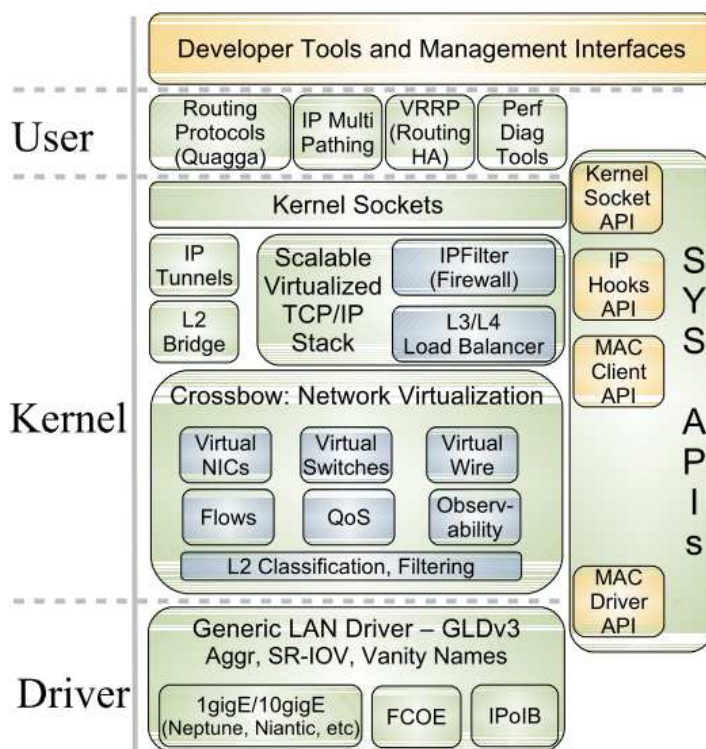
Παραεικονικοποίηση: η εικονική μηχανή δεν προσομοιώνει επακριβώς το υλικό αλλά παρέχει στις εικονικές μηχανές ένα API, μία προγραμματιστική διασύνδεση, ώστε να επιτρέπει την εκτέλεση επάνω της ενός τροποποιημένου, φιλοξενούμενου ΛΣ σχεδιασμένου για εκτέλεση από τον συγκεκριμένο επόπτη (π.χ. Denali, XEN). Το προαναφερθέν API ονομάζεται διασύνδεση υπερκλήσεων και ένα λειτουργικό σύστημα πρέπει να μεταφερθεί ρητά σε έκδοση κατάλληλη για εκτέλεση από ένα σύστημα παραεικονικοποίησης, ώστε ο φιλοξενούμενος πυρήνας αντί να προσπελαύνει το υλικό άμεσα να εκτελεί υπερκλήσεις και να αναμένει απαντήσεις ή ασύγχρονες ειδοποιήσεις από τον επόπτη. Το όφελος από τη βελτίωση των επιδόσεων και την απλοποίηση της γραφής του επόπτη είναι μεγάλο.

Μία άλλου είδους κατηγοριοποίηση των εποπτών εικονικοποίησης είναι σε αυτούς που εκτελούνται ως εφαρμογές πάνω από ένα υποκείμενο λειτουργικό σύστημα-ξενιστή (π.χ. VirtualPC, VMWare κλπ) και σε αυτούς που λειτουργούν οι ίδιοι ως λιτά λειτουργικά συστήματα και άρα εκτελούνται απευθείας επάνω από το υποκείμενο υλικό (π.χ. Xen).

Κεφάλαιο 3 Παρουσίαση του Open Solaris

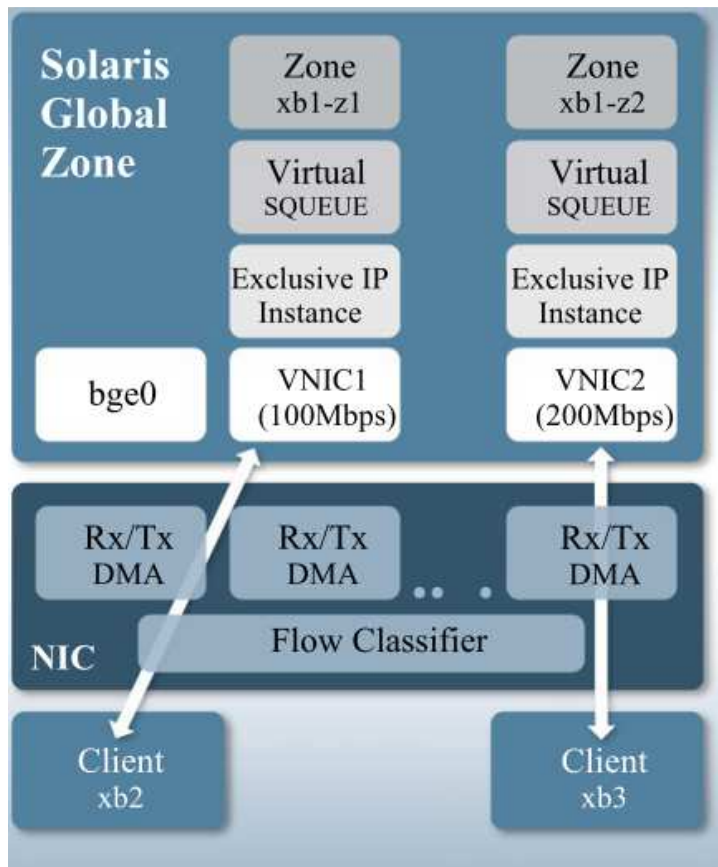
3.1 Οδηγός δικτύωσης

Το OpenSolaris προσφέρει ισχυρά χαρακτηριστικά δικτύωσης για τη διαχείριση των πόρων του δικτύου και του δικτύου virtualization. Το virtualization δίκτυου έχει την ικανότητα να εικονικοποιήσει ολόκληρη τη τοπολογία των server, switches, firewalls και όλα να λειτουργούν σε μια ενιαία πλατφόρμα που δεν απαιτεί πρόσθετες επενδύσεις σε δικτύωση υλικού. Το δίκτυο virtualization μπορεί να χρησιμοποιηθεί για ποικίλους λόγους, από τη δημιουργία πρωτοτύπων, την ανάπτυξη και τον έλεγχο, στην υπηρεσία ανάπτυξης. Το σύστημα διαχείρισης των πόρων προσθέτει την ικανότητα να καθοριστούν: (i) οι διεπαφές του δικτύου, (ii) τα όρια bandwidth και (iii) προτεραιότητες των ροών κίνησης. Η cpu χειρίζεται τη κίνηση και την διαχείριση των πόρων του δικτύου που μπορεί να ανατεθεί είτε σε φυσικές ή εικονικές διεπαφές χωρίς να απαιτούνται μεγαλύτερες ανάγκες στις επιδόσεις του δικτύου.



Εικόνα 5 Solaris Core Network Functionality

Στην **Σφάλμα!** Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. φαίνεται η αρχιτεκτονική του πυρήνα του συστήματος διαχείρισης διαδικτυακών λειτουργιών του Open Solaris. Το σύστημα αυτό υποστηρίζει όλες τις κλασσικές λειτουργίες που επιτούνται σε ένα δρομολογητή (π.χ. δρομολόγηση, DNS, DHCP, VoIP, tunneling, multipathing etc), επιπλέον μέσω του συστήματος crossbow υποστηρίζει υπηρεσίες εικονικοποίησης όπως: εικονικές διεπαφές δικτύου, εικονικούς μεταγωγείς, εικονικές συνδέσεις και τέλος υπηρεσίες που ανήκουν στο δεύτερο επίπεδο (L2).



Εικόνα 6 Εικονικό δίκτυο

Όπως φαίνεται και στο παράδειγμα (Εικόνα 6), η εικονικοποίηση δικτύου (Network Virtualization) που υλοποιείται από το Crossbow, δίνει την δυνατότητα ελέγχου των παρακάτω λειτουργιών/χαρακτηριστικών:

- Ροές
- Εικονικά NICs και εικονικούς διακόπτες (etherstubs)
- Virtual wire
- Έλεγχος πόρων
- Διαμερισμός εύρους ζώνης
- Διαμερισμός NIC H/W
- Εκχώρηση CPUs
- Πραγματικό χρόνο χρήσης για κάθε link/ροή
- Λεπτομερή στατιστικά για κάθε link/ροή
- Ιστορικό

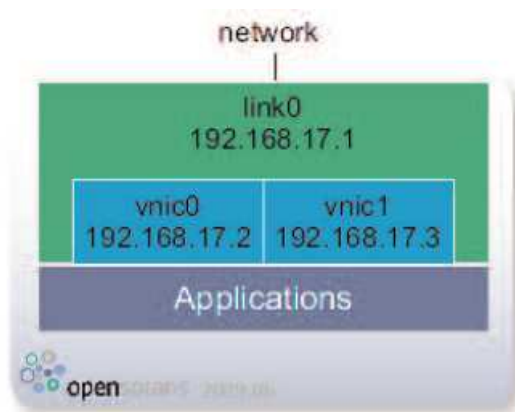
Θα ξεκινήσουμε χρησιμοποιώντας virtualization δικτύου και διαχείριση των πόρων του δικτύου στο Open Solaris. Συγκεκριμένα θα εξετάσουμε:

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

- i. **VNICs:** Virtual Network Interface Ελεγκτές (VNICs), το θεμελιώδες δομικό στοιχείο της εικονικοποίησης δικτύου
- ii. **Etherstubs:** Εικονικοί διακόπτες που επιτρέπουν τη μετάβαση της πληροφορίας εντός ενός συστήματος
- iii. **Flows:** Ανά σύνδεση (εικονικής ή φυσικής) διαχείριση για την επίτευξη Quality of Service (QoS) με τον καθορισμό ορίων του εύρους ζώνης και προτεραιότητες της κυκλοφορίας.

VNIC: Εικονική διαχείριση δικτύου

VNICs δημιουργούνται πάνω από φυσικές διασυνδέσεις ή πάνω από etherstubs ,καθώς και από το σημείο του συστήματος που είναι ακριβώς σαν φυσική διασύνδεση. VNICs μπορεί να συνδέσει, να ρυθμιστεί με τις διευθύνσεις ip και λαμβάνοντας υπόψη τις δικές του επιλογές TCP .Εργαλεία όπως ipconfig,dladm και snoop δουλεύουν σε VNICs.



Εικόνα 7 VNIC

Η Εικόνα 7 δείχνει δύο VNICs σε μια μόνο φυσική διασύνδεση. Κάθε διεπαφή (interface) φυσικά η εικονικά έχει τη δική της διεύθυνση ip. Το link δε πρέπει να συνδεθεί ή να ρυθμιστεί.

Χαρακτηρηστικά Εικονικών NICs

- Η διεύθυνση IP αποδίδεται στατικά ή μέσω του DHCP
- Μπορούν να δημιουργηθούν πάνω από φυσική NIC (χωρίς να χρειάζονται Vswitch) για να παρέχει εξωτερική συνδεσιμότητα με μεταγωγή
- Τα VNICs έχουν διαμορφώσιμη ταχύτητα σύνδεσης, CPU και την εκχώρηση προτεραιότητας
- Δεν υπάρχουν αλλαγές ρυθμίσεων που απαιτούνται για να μεταβείτε σε υποστήριξη virtualization

Για τη δημιουργία ενός vnic στη global zone(gz) χρησιμοποιούμε τη παρακάτω εντολή:

```
#dladm create-vnic -l data-link vnic
```

Data-link είναι το όνομα της διεπαφής όπου το vnic πρόκειται να ρυθμιστεί, **Vnic** είναι το όνομα που θέλουμε να δώσουμε στο vnic

Για να επαληθεύσουμε ότι το vnic που φτιάξαμε έχει δημιουργηθεί γράφουμε τη παρακάτω εντολή:#dladm show-vnic

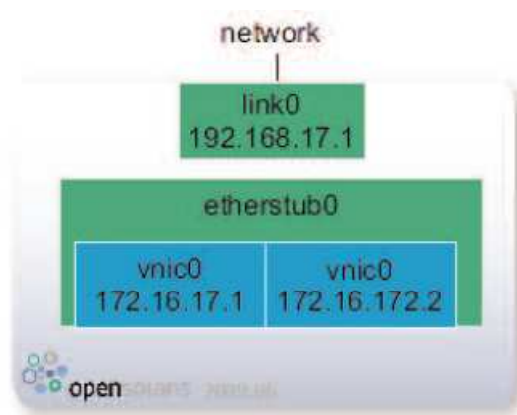
Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Με τη παρακάτω εντολή δίνουμε μια ip address στο vnic που έχουμε ανεβάσει στη global zone

```
#ifconfig vnic plumb IP-address up
```

Etherstub: Virtual Network Switch

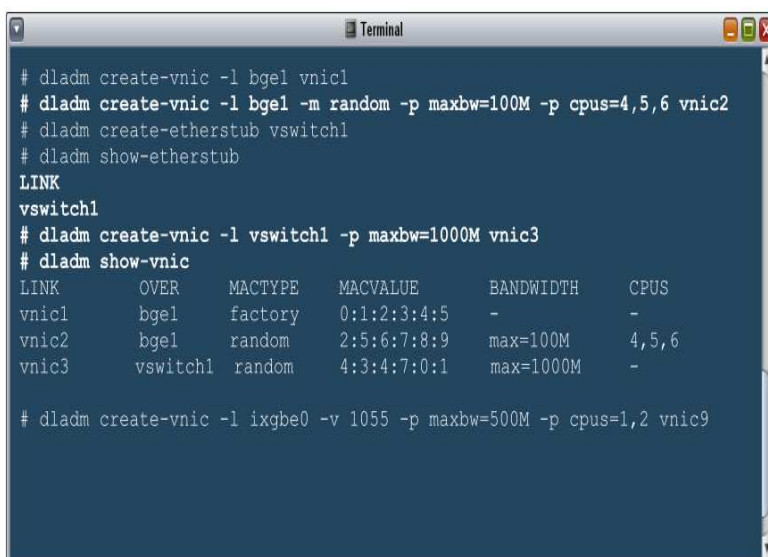
Τα Etherstubs είναι μεταγωγείς λογισμικού που συνδέονται VNICs (Όπως φαίνεται στο ακόλουθο διάγραμμα ως το πλαίσιο με την ένδειξη etherstub0). Τα VNICs που δημιουργούνται πάνω σε ίδια etherstub μπορούν να στείλουν πακέτα το ένα στο άλλο. Επίσης τα Etherstub δε μπορούν να στείλουν πακέτα απευθείας στο φυσικό δίκτυο ούτε σε VNIC σε άλλο etherstub, αλλά τα πακέτα μπορούν να δρομολογηθούν. Έτσι σε αυτό το παράδειγμα το λειτουργικό σύστημα **δρομολογεί πακέτα μεταξύ του εξωτερικού συνδέσμου, link0 και του εσωτερικού διακόπτη, etherstub0**



Εικόνα 8 Etherstub

Η δρομολόγηση πακέτων μεταξύ του εξωτερικού link0 και του εσωτερικού διακόπτη etherstub0 γίνεται από το λειτουργικό σύστημα. Σε ένα επόμενο παράδειγμα θα δούμε πως γίνεται αυτό. Οι Εικονικοί Μεταγωγείς (Virtual Switches) μπορούν να δημιουργηθούν για να παρέχουν ιδιωτική συνδεσιμότητα μεταξύ εικονικών μηχανών (virtual machines)

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris



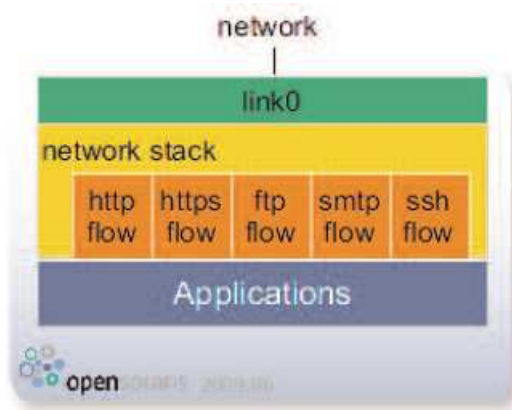
```
# dladm create-vnic -l bge1 vnic1
# dladm create-vnic -l bge1 -m random -p maxbw=100M -p cpus=4,5,6 vnic2
# dladm create-etherstub vswitch1
# dladm show-etherstub
LINK
vswitch1
# dladm create-vnic -l vswitch1 -p maxbw=1000M vnic3
# dladm show-vnic
LINK      OVER      MACTYPE   MACVALUE   BANDWIDTH  CPUS
vnic1     bge1     factory   0:1:2:3:4:5   -           -
vnic2     bge1     random    2:5:6:7:8:9   max=100M   4,5,6
vnic3     vswitch1 random    4:3:4:7:0:1   max=1000M  -

# dladm create-vnic -l ixgbe0 -v 1055 -p maxbw=500M -p cpus=1,2 vnic9
```

Εικόνα 9 OpenSolaris Terminal

POH - Υπεύθυνη για τη διαχείριση της κίνησης του δικτύου

Μια ροή δημιουργείται πάνω από μια διεπαφή (interface) φυσικά ή εικονικά και περιγράφει την κυκλοφορία του δικτύου από την άποψη της εφαρμογής (port), πηγή και τον προορισμό, και παρέχει μια λαβή που μπορεί να χρησιμοποιηθεί για την εφαρμογή των πολιτικών των πόρων για εύρος ζώνης και προτεραιότητα.



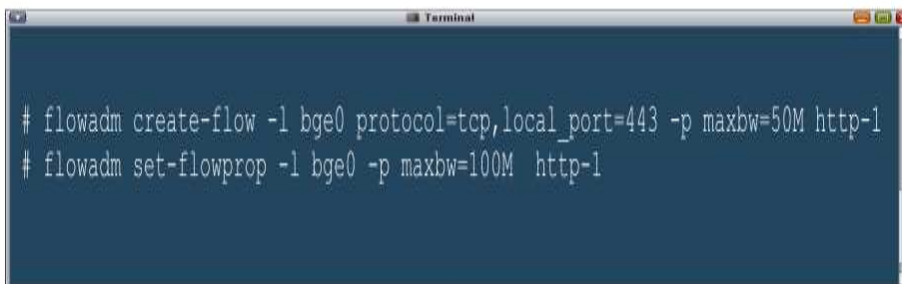
Εικόνα 10 Εικονικοποίηση ροών

Το διάγραμμα δείχνει τη διασύνδεση με διάφορες ροές TCP: web, ασφαλές διαδίκτυο, μεταφορά αρχείων, email και ασφαλή σύνδεση. Κάθε κίνηση που δε ταιριάζει με καμία από τις ροές περνά μέσα από το δίκτυο της στοίβας χωρίς να επιβληθεί οποιαδήποτε πολιτική (εκτός από τις πολιτικές που ζήτησε από το ίδιο το interface). Κίνηση που ταιριάζει με μία από τις ροές θα περιορίζεται από το εύρος ζώνης και την προτεραιότητα που έχει ανατεθεί στη συγκεκριμένη ροή.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Τα ακόλουθα χαρακτηριστικά μπορούν να οριστούν σε μια ροή

- B/W όρια
- Εγγυήσεις (μελλοντικά)
- Προτεραιότητες
- CPUs

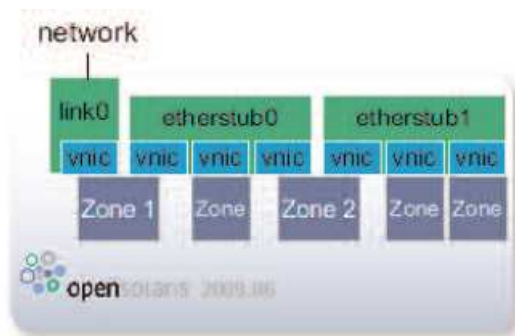


```
# flowadm create-flow -l bge0 protocol=tcp,local_port=443 -p maxbw=50M http-1
# flowadm set-flowprop -l bge0 -p maxbw=100M http-1
```

Εικόνα 11 Παράδειγμα ρυθμίσεις ροών

Virtualization Δικτύου και virtualization εξυπηρετητή

VNICs μπορεί να αποδοθεί σε virtual servers. VNICs μπορούν να δουλέψουν με το Xvm Hypervisor.



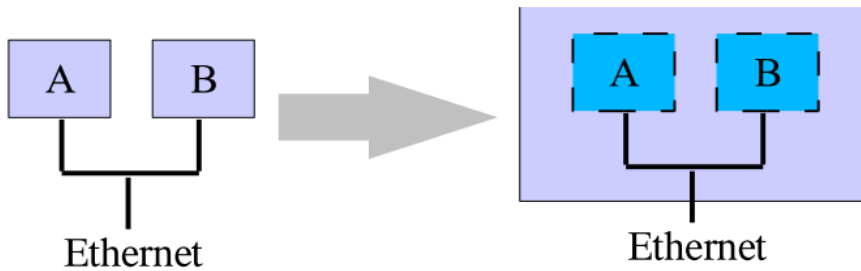
Εικόνα 12 Network Virtualization

Στο παραπάνω διάγραμμα ζώνες με VNICs με την ίδια etherstub μπορούν να στείλουν πακέτα άμεσα ή μια στην άλλη. Zone1 μπορεί να δρομολογήσει πακέτα μεταξύ etherstub0 και του φυσικού δικτύου. Η Zone2 μπορεί να δρομολογήσει πακέτα μεταξύ etherstub0 και etherstub1.

Zones – Application Containers

- Κάθε ζώνη είναι διαχωρισμένη από τις υπόλοιπες ζώνες
- Μερικά σύνολα από πόρους που διατίθενται για τις ζώνες (CPU, file systems)
- Η ζώνη έχει μία ή περισσότερες διευθύνσεις ip
- Επιτρέπει την εδραίωση των Server που βρίσκονται στο ίδιο δίκτυο

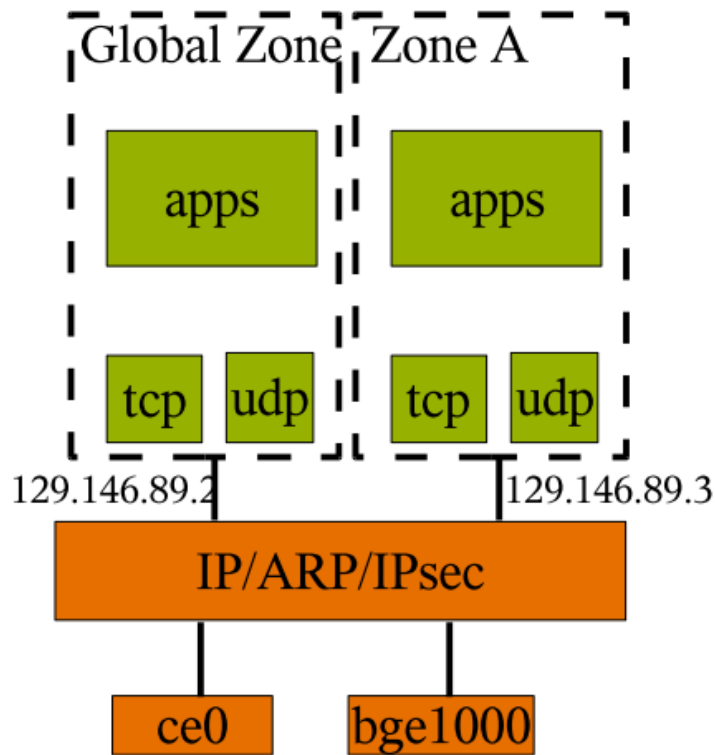
Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris



Εικόνα 13 Ζώνες

Δικτύωση σε ζώνες (networking in zones)

- Η ζώνη χρησιμοποιεί περιορισμούς στη χρήση των ip διευθύνσεων
- IP / ARP / IPsec κοινά για όλες τις ζώνες (Κοινή δρομολόγηση, ARP διαμόρφωση)
- TCP, UDP, SCTP χωριστά για κάθε ζώνη



Εικόνα 14 Διαδικτύωση σε ζώνες

Δικτύωση στις ζώνες (networking in zones)

- Οι διευθύνσεις ip που αποδίδονται στις ζώνες
- Μια ζώνη θα λάβει μόνο πακέτα που αποστέλλονται στη διεύθυνση ip
- Μια ζώνη δε μπορεί να στείλει πακέτα χρησιμοποιώντας κάποιου άλλου τη διεύθυνση ip
- Ξεχωριστό αριθμό θύρας (port) ανά ζώνη

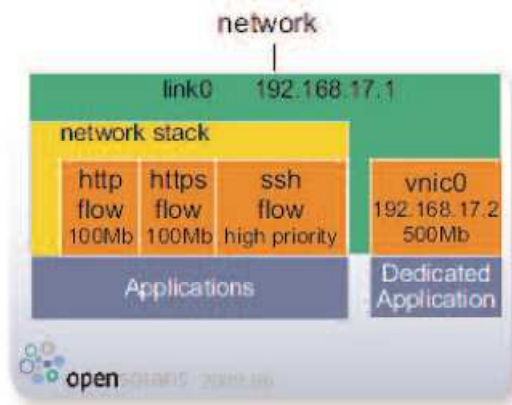
Περιπτώσεις ip για τις ζώνες – τα οφέλη του διαχωρισμού

- Χωριστά LANs ή VLANs μπορούν να συνδεθούν σε διαφορετικές ζώνες και δεν υπάρχουν διαρροές IP μεταξύ τους.(Για παράδειγμα, είναι δυνατόν να έχουμε ένα δίκτυο διαχείρισης χωριστό από το δίκτυο δεδομένων)
- Επιτρέπει τη χρήση ip στις ζώνες (DHCP,IPsec,IPfilter)
- Ανάδιαμόρφωση δικτύου περίπτωσης (πίνακες δρομολόγησης,συντονισμό μεταφορών, κλπ)

3.2 Λειτουργικά Παραδείγματα

3.2.1 Παράδειγμα 1(VNICs και ροές σε φυσική διασύνδεση)

Αυτό το παράδειγμα δημιουργεί ένα VNIC με τη δική του διεύθυνση ip για μια ειδική εφαρμογή και στήνει διάφορες ροές στη κύρια διεπαφή για τη διαχείριση της κίνησης. Η διαμόρφωση του δικτύου μοιάζει με αυτή που φαίνεται στην παρακάτω εικόνα (Εικόνα 15):



Εικόνα 15 VNIC \$ Ροές

Το παρακάτω παράδειγμα δείχνει πως να:

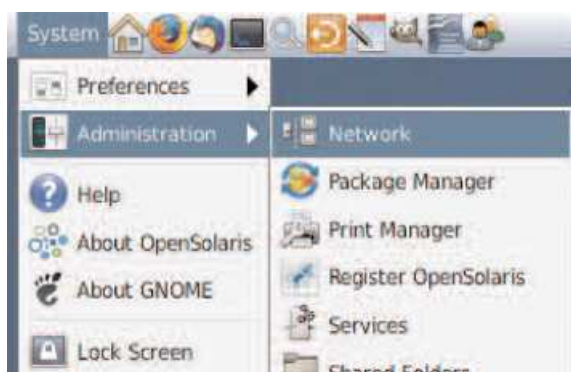
- Δημιουργήσουμε ένα VNIC
- Δημιουργήσουμε μια ροή
- Πως εφαρμόζονται τα όρια εύρους ζώνης για VNICs
- Και ροές
- Βάλουμε προτεραιότητες σε μια ροή
- Δύο ροές και ένα VNIC έχουν όρια εύρους ζώνης. Η μια ροή έχει υψηλή προτεραιότητα.

Βήμα 1:Απενεργοποίηση nwamd

Από προεπιλογή, το OpenSolaris χρησιμοποιεί nwamd,το Δίκτυο Automagic διαμόρφωσης daemon,για να ρυθμιστεί αυτόματα το δίκτυο .Για να πειραματιζόμαστε με το virtualization του δικτύου και του δικτύου διαχείρισης των πόρων , είναι καλύτερα να απενεργοποιήσουμε το nwamd.

Αυτό μπορεί να επιτευχθεί με δύο τρόπους:

- i) Επιλέγουμε απ το μενού :System->Administration->Network.Αυτό μας εμφανίζει ένα παράθυρο που μας ρωτάει εάν θα θέλαμε να χειριστούμε εμείς τη διαχείριση του δικτύου μας .Επιλέγουμε 'Manual' και στη συνέχεια χρησιμοποιώντας το εργαλείο διαχείρισης δικτύου δημιουργούμε την ip,μάσκα ,τη δρομολόγηση κ.τ.λ



Εικόνα 16 Απενεργοποίηση nwmad

- ii) Στη γραμμή εντολών τρέχουμε αυτές τις εντολές για να μεταβούμε από nwmad σε χειροκίνητη ρύθμιση:

```
svcadm disable network/physical:nwam
svcadm enable network/physical:default
Διαμόρφωση του δικτύου μέσω της επεξεργασίας αυτών των αρχείων
/etc/hostname.<interface>
/etc/defaultrouter
/etc/nsswitch.conf
/etc/resolv.conf
```

```
svcadm disable network/physical:nwam
svcadm enable network/physical:default
```

Κάνουμε επανενκίνηση την υπηρεσία δικτύου ώστε η διαμόρφωση να αρχίζει να παράγει αποτελέσματα.

```
svcadm restart network/physical:default
```

Ο λόγος για την εκτέλεση αυτών των εντολών από τη κονσόλα και όχι μέσω του δικτύου (για παράδειγμα, πάνω από μια ssh σύνδεση) είναι για να αποφεύγεται η απώλεια σύνδεσης με το σύστημα κατά τη διάρκεια της αναδιαμόρφωσης.

ΒΗΜΑ 2 :Δημιουργία VNIC

Η δημιουργία της εικονικής κάρτας δικτύου(vnic0) γίνεται πάνω στη φυσική κάρτα δικτύου που στη περίπτωση μας είναι η bge0. Η ip που θα εκχωρήσουμε στη vnic0 είναι η 192.168.17.2

```
# dladm show-phys
```

```
LINK MEDIA STATE SPEED DUPLEX DEVICE
bge0 Ethernet up 1000 full bge0
```

```
# dladm create-vnic -l bge0 vnic0
# dladm show-vnic vnic0
```

```
LINK OVER SPEED MACADDRESS MACADDRTYPE VID
vnic0 bge0 1000 2:8:20:5e:69:f2 random 0
```

```
# ifconfig vnic0 plumb
# ifconfig vnic0 192.168.17.2/24 up
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

ΒΗΜΑ 3: Δημιουργούμε τις ροές

Δημιουργούμε 3 ροές που χρησιμοποιούν πρωτόκολλο μεταφοράς το TCP. Η http-flow ακούει στη πόρτα 80, Η https-flow ακούει στη πόρτα 43 και η Η ssh-flow ακούει στη πόρτα 22

```
# flowadm add-flow -l bge0 -a transport=TCP,local_port=80 http-flow
# flowadm add-flow -l bge0 -a transport=TCP,local_port=443 https-flow
# flowadm add-flow -l bge0 -a transport=TCP,local_port=22 ssh-flow
```

ΒΗΜΑ 4: Όρια κίνησης σε VNIC. Όριο web(HTTP και HTTPS) κίνησης .Φτιάχνουμε ssh κίνησης υψηλής προτεραιότητας.

```
# dladm set-linkprop -p maxbw=500M vnic0
# flowadm set-flowprop -p maxbw=100M http-flow
# flowadm set-flowprop -p maxbw=100M https-flow
# flowadm set-flowprop -p priority=high ssh-flow
```

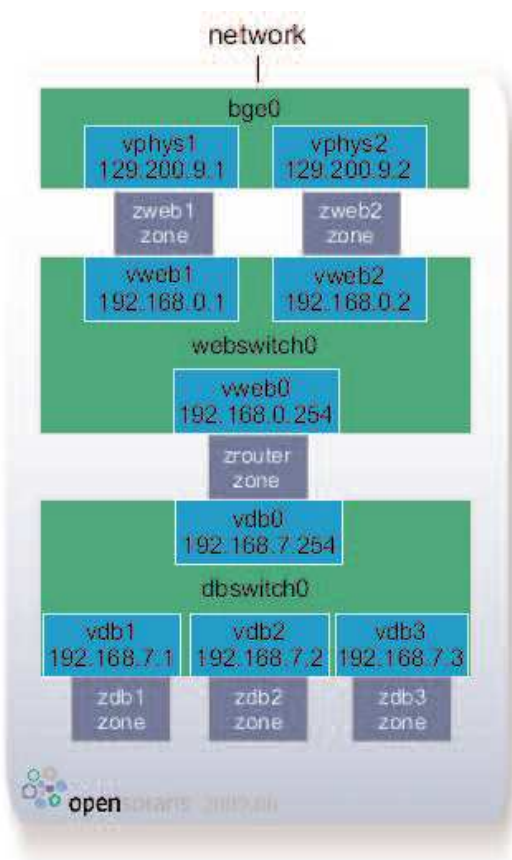
ΒΗΜΑ 5: Αναθεώρηση της διαμόρφωσης

```
# dladm show-linkprop -p maxbw vnic0
LINK PROPERTY PERM VALUE DEFAULT POSSIBLE
vnic0 maxbw rw 500 -- --
# flowadm show-flowprop http-flow
FLOW PROPERTY VALUE DEFAULT POSSIBLE
http-flow maxbw 100 -- 100M
http-flow priority -- --
# flowadm show-flowprop https-flow
FLOW PROPERTY VALUE DEFAULT POSSIBLE
https-flow maxbw 100 -- 100M
https-flow priority -- --
# flowadm show-flowprop ssh-flow
FLOW PROPERTY VALUE DEFAULT POSSIBLE
ssh-flow maxbw -- --
ssh-flow priority high -- high
```

3.2.2 Παράδειγμα 2 (Virtualization με χρήση ζωνών)

Αυτό το παράδειγμα είναι μια πολυεπίπεδη εφαρμογή. Υπάρχουν αρκετοί web servers που τρέχουν σε SOLARIS CONTAINERS που είναι εκτεθειμένοι στο φυσικό δίκτυο. Υπάρχουν αρκετοί database servers που τρέχουν σε SOLARIS CONTAINERS που δεν είναι εκτεθειμένοι στο φυσικό δίκτυο. Οι web servers μοιράζονται ένα back-end etherstub (εικονικό διακόπτη) που ονομάζεται webswitch0 στο ακόλουθο διάγραμμα .Οι database servers μοιράζονται επίσης ένα back-end etherstub που ονομάζεται dbswitch0 στο ακόλουθο διάγραμμα .Το δίκτυο θα διαμορφωθεί έτσι ώστε τα database containers δε θα έχουν άμεση πρόσβαση στο φυσικό δίκτυο το οποίο με ασφάλεια τους απομονώνει από την εξωτερική επικοινωνία. Το δίκτυο και η ζώνη διαμόρφωσης μοιάζει με αυτό :

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris



Εικόνα 17 Σχέδιο για υλοποίηση στο OpenSolaris

ΒΗΜΑ 1: Απενεργοποίηση nwam

Το έχουμε δείξει στο βήμα 1 του προηγούμενου παραδείγματος.

ΒΗΜΑ 2: Δημιουργία etherstubs και VNICs

Σε αυτό το βήμα δημιουργούμε τους δύο εικονικούς διακόπτες (etherstubs) και τις VNICs που χρησιμοποιείται στις έξι ζώνες, όπως απεικονίζεται στο παραπάνω διάγραμμα. Πρώτα μαθαίνουμε την ονομασία του φυσικού NIC:

```
# dladm show-phys
LINK MEDIA STATE SPEED DUPLEX DEVICE
bge0 Ethernet up 1000 full bge0
```

Γι αυτό το σύστημα υπάρχει μόνο ένα φυσικό interface, "bge0" θα ξεκινήσουμε δημιουργώντας δύο εικονικές κάρτες δικτύου στο φυσικό interface, "bge0".

```
# dladm create-vnic -l bge0 vphys1
# dladm create-vnic -l bge0 vphys2
```

Στη συνέχεια δημιουργούμε τους εικονικούς διακόπτες (etherstubs) και VNICs για το υπόλοιπο των διασυνδέσεων όπως φαίνεται στο προηγούμενο διάγραμμα.

```
# dladm create-etherstub webswitch0
# dladm create-etherstub dbswitch0
# dladm create-vnic -l webswitch0 vweb0
# dladm create-vnic -l webswitch0 vweb1
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
# dladm create-vnic -l webswitch0 vweb2
# dladm create-vnic -l dbswitch0 vdb0
# dladm create-vnic -l dbswitch0 vdb1
# dladm create-vnic -l dbswitch0 vdb2
# dladm create-vnic -l dbswitch0 vdb3
```

Ελέγχουμε το setup

```
# dladm show-vnic
LINK OVER SPEED MACADDRESS MACADDRTYPE VID
vphys1 bge0 100 2:8:20:d1:b6:26 random 0
vphys2 bge0 100 2:8:20:ae:69:f7 random 0
vweb0 webswitch0 0 2:8:20:ae:89:e0 random 0
vweb1 webswitch0 0 2:8:20:77:c6:1 random 0
vweb2 webswitch0 0 2:8:20:51:35:ec random 0
vdb0 dbswitch0 0 2:8:20:b1:85:b1 random 0
vdb1 dbswitch0 0 2:8:20:d7:6f:df random 0
vdb2 dbswitch0 0 2:8:20:22:38:2e random 0
vdb3 dbswitch0 0 2:8:20:f5:b5:ce random 0
```

ΒΗΜΑ 3: Δημιουργία το ZFS Pool για την αποθήκευση των ζωνών

Πρόκειται να δημιουργηθεί ένα ενιαίο σύστημα Solaris Zone και στη συνέχεια να μειωθεί ο χρόνος για τη δημιουργία επιπρόσθετων στοιχείων. Cloned Zones απαιτούν να μοιράζονται ένα σύνολο δεδομένων ZFS, δημιουργώντας έτσι ένα σύνολο δεδομένων ZFS είναι το πρώτο βήμα.

```
# zfs create -o mountpoint=/zonefs rpool/zonefs
# chmod 700 /zonefs
```

ΒΗΜΑ 4: Δημιουργία ζώνης για cloning

Δεδομένου ότι θα δημιουργήσουμε έξι ζώνες η πιο αποτελεσματική προσέγγιση είναι να δημιουργήσουμε μια από την αρχή και στη συνέχεια να κάνουμε έξι κλώνους. Η κλωνοποίηση χρειάζεται μόνο λίγα δευτερόλεπτα.

```
# zonecfg -z vnmbase
zclone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zclone> create
zonecfg:zclone> set zonepath=/vnm/vnmbase
zonecfg:zclone> set ip-type=exclusive
zonecfg:zclone> verify
zonecfg:zclone> commit
zonecfg:zclone> exit
# zoneadm -z vnmbase install
```

Η εγκατάσταση μπορεί να πάρει αρκετά λεπτά ενώ τα πακέτα είναι εγκατεστημένα. Μπορεί να δούμε μερικά λάθη από sys-unconfig όπως:

```
rm: /zonefs/zweb1/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /zonefs/zweb1/root/etc/dumpadm.conf
που μπορεί να αγνοηθούν.
```

Τώρα κάνουμε boot το clone zone:

```
# zoneadm -z vnmbase boot ; zlogin -C vnmbase
```

Θέλουμε αυτό το βήμα να ολοκληρωθεί έτσι ώστε όταν κάνουμε clone ζώνης δεν θα πρέπει να περιμένουμε για κάθε κλώνο να ρυθμίσει το SMF. Ξέρουμε ότι είναι ασφαλές να σταματήσουμε το clone, όταν μας ζητηθεί να επιλέξουμε έναν τύπο τερματικού

Type the number of your choice and press Return:

Και μετά παύση zclone

```
# zoneadm -z vnmbase halt
```

Δημιουργία του zweb1

Ο κώδικας υλοποίησης του zweb1 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 129.200.9.1 ως ip για το vphys1 και 192.168.0.1 ως ip για το vweb1 .Βάζουμε none για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ. Όταν κάνουμε log in στη ζώνη κάνουμε τις παρακάτω ρυθμίσεις:

```
root@zweb1:~# route -p add net 192.168.7.0/24 192.168.0.254
add net 192.168.7.0/24: gateway 192.168.0.254
add persistent net 192.168.7.0/24: gateway 192.168.0.254
root@zweb1:~# ping 192.168.7.254
192.168.7.254 is alive
root@zweb1:~# ping 192.168.7.1
192.168.7.1 is alive
```

Δημιουργία του zweb2

Ο κώδικας υλοποίησης του zweb2 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 129.200.9.2 ως ip για το vphys2 και 192.168.0.2 ως ip για το vweb2 .Βάζουμε none για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ. Όταν κάνουμε log in στη ζώνη κάνουμε τις παρακάτω ρυθμίσεις:

```
root@zweb2:~# route -p add net 192.168.7.0/24 192.168.0.254
add net 192.168.7.0/24: gateway 192.168.0.254
add persistent net 192.168.7.0/24: gateway 192.168.0.254
```

Δημιουργία του zdb1

Ο κώδικας υλοποίησης του zdb1 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 192.168.7.1 ως ip για το vdb1 .Βάζουμε none για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ. Όταν κάνουμε log in στη ζώνη κάνουμε τις παρακάτω ρυθμίσεις:

```
root@zdb1:~# route -p add net 192.168.0.0/24 192.168.7.254
add net 192.168.0.0/24: gateway 192.168.7.254
add persistent net 192.168.0.0/24: gateway 192.168.7.254
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Δμιουργια το zdb2

Ο κώδικας υλοποίησης του zdb2 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 192.168.7.2 ως ip για το vdb2 .Βάζουμε none για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ .Όταν κάνουμε log in στη ζώνη κάνουμε τις παρακάτω ρυθμίσεις

```
root@zdb2:~# route -p add net 192.168.0.0/24 192.168.7.254
add net 192.168.0.0/24: gateway 192.168.7.254
add persistent net 192.168.0.0/24: gateway 192.168.7.254
```

Δημιουργια του zdb3

Ο κώδικας υλοποίησης του zdb3 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 192.168.7.3 ως ip για το vdb3 .Βάζουμε none για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ.Όταν κάνουμε log in στη ζώνη κάνουμε τις παρακάτω ρυθμίσεις

```
root@zdb3:~# route -p add net 192.168.0.0/24 192.168.7.254
add net 192.168.0.0/24: gateway 192.168.7.254
```

Δμιουργια του zrouter

Ο κώδικας υλοποίησης του zrouter βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 192.168.0.254 ως ip για το vweb0 και 192.168.7.254 ως ip για το vdb0 .Βάζουμε none για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ.Όταν κάνουμε log in στη ζώνη κάνουμε τις παρακάτω ρυθμίσεις.

```
root@zrouter:~# svcadm enable network/ipv4-forwarding:
```

VNICs:Virtual Network Interface Ελεγκτές (VNICs),το θεμελιώδες δομικό στοιχείο της εικονικοποίησης δικτύου.

Etherstubs:Εικονικοί διακόπτες που επιτρέπουν τη μετάβαση της πληροφορίας εντός ενός συστήματος.

Κεφάλαιο 4

Παρουσίαση του Crossbow

4.1 Εισαγωγή

Το Project Crossbow είναι μια φιλόδοξη προσπάθεια επανασχεδίασης του open Solaris στην δικτύωση ώστε να μπορεί να εκπληρώσει τρεις βασικούς σκοπούς

Το project Crossbow ενισχύει την ικανότητα να εδραιώσει φόρτο εργασίας σε servers. Στην αγορά ο Όρος ιδεατή δικτύωση τείνει να έχει μια κάπως περιορισμένη έκταση, εστιάζοντας σε συγκεκριμένα θέματα όπως εικονικά LANs και τεχνικές συσσώρευσης. Η δημιουργία ιδεατού δικτύου στο πλαίσιο αυτής της εργασίας είναι πολυάφηρημένη. Η έννοια της εικονικοποίησης σε όλες τις πτυχές μιας τοπολογίας δικτύου μέσα σε ένα εικονικό server. Υπάρχουν πολλές όψεις στην δημιουργία του project Crossbow: Με την δημιουργία του εικονικού υλικού διασύνδεσης δικτύου (NIC) σε Virtual NICs (VNICs) το Crossbow προωθεί την αποτελεσματικότερη κατανομή των πόρων δικτύωσης. Η δομή του VNIC επιτρέπει τη διαίρεση της φυσικής θύρας NIC σε πολλαπλές εικονικές για τη δημιουργία του πυρήνα που περιβάλλεται από απομονωμένες και ειδικευμένες σε αυτό στοιβές από τη φυσική διασύνδεση μέχρι τις εφαρμογές.

Το Crossbow δίνει την δυνατότητα κατασκευής ενός υλικού διακόπτη LAN μέσω της δικτυακής στοιβας επιτρέποντας έτσι την διαδικασία του switching να λάβει χώρα μέσα σε ένα εικονικό περιβάλλον. Το Virtualization δεν είναι κατάλληλο μόνο για τη διαίρεση ενός φυσικού NIC, αλλά και για άθροιση άλλων σκοπών. Αθροίζοντας δύο φυσικά NICs σε ένα και διαχωρίζοντας το αποτέλεσμα από ένα σε (για παράδειγμα) τρία VNICs μας δίνεται η δυνατότητα να έχουμε καλύτερη κατανομή των πόρων του δικτύου και μας παρέχει ένα πλεονέκτημα σε περίπτωση που μια από τις φυσικές συνδέσεις αποτύχει. Το πρότυπο Virtual LAN (VLAN) υποστηρίζεται, επιτρέποντας NIC και / ή VNICs να ανατεθούν σε VLAN. Σε ένα περιβάλλον με switches και routers που υποστηρίζουν επίσης VLANs, αυτό επιτρέπει end-to-end απομόνωση της κίνησης, ακόμη και αν η κίνηση μπορεί να τρέχει σε μια κοινόχρηστη φυσική σύνδεση. Επιπλέον, άλλα δικτυακά στοιχεία του λειτουργικού συστήματος μπορεί να μπουν στο παιχνίδι, ιδίως το router και firewall που περιλαμβάνεται στο λειτουργικό σύστημα των OpenSolaris. Όλα αυτά τα στοιχεία που καθιστούν δυνατή την κατασκευή μιας ολόκληρης τοπολογίας δικτύων εντός ενός υπολογιστικού συστήματος - χρήσιμο για αρχιτεκτονική και προτυποποίηση.

Η δικτυακή διαχείριση που έχει το Crossbow επιτρέπει σε διαφόρους οργανισμούς να επιτύχουν QoS στόχους όσον αφορά την δικτύωση με το να ειδικεύει με ασφαλή τρόπο τους δικτυακούς πόρους όπως επίσης και να καθορίζει το μέγιστο της χρήσης των πόρων στις διεργασίες. Πάντοτε αυτή η δυνατότητα την είχαν οι administrators όσο αφορά την διαχείριση της CPU και της μνήμης ώστε να έχουν την ικανότητα να ρυθμίζουν την ελάχιστη παροχή πόρων στις διεργασίες ανεξάρτητα από τις ανάγκες που μπορεί αυτές να απαιτούσαν. Η διαχείριση των πόρων μπορεί να βοηθήσει στην αποφυγή υπερβολικής χρήσης των πόρων αυτών.

Το Crossbow επεκτείνεται επίσης στην διαχείριση της δικτύωσης με τρεις τρόπους:

- 1) Με το να επιτρέπει την ειδικευση των πόρων στις CPU αναθέτοντας σε ένα NIC ή σε ένα VNIC. Αυτό μπορεί να χρησιμοποιηθεί στο να υπάρχει ένα όριο στους πόρους ή στο να διασφαλίσει ότι οι πόροι έχουν κατανεμηθεί σε αυξημένες ανάγκες (πχ bandwidth) του συστήματος.
- 2) Με το να δίνει την δυνατότητα στον καθορισμό των ορίων του bandwidth για ένα NIC ή ένα VNIC. Αυτό επίσης θα μπορούσε να βοηθήσει στην καλύτερη κατανομή των πόρων η στην διασφάλιση της ελάχιστης χρήσης τους.
- 3) Με το να καθορίζει τις προτεραιότητες για ειδή δικτυακής κίνησης για NIC/VNIC βασισμένο στην πηγή ή στην IP διεύθυνση του προορισμού την διήθηση MAC, την θύρα ή ακόμα και το πρωτόκολλο μεταφοράς.

Αυτές οι δυνατότητες διαχείρισης των πόρων επιτρέπουν πολλές δυναμικές πολιτικές διαχείρισης της δικτύωσης .

Τέλος το Crossbow μπορεί να αυξήσει το throughput του δικτύου με πιο αποτελεσματικό προγραμματισμό και με διαχείριση των πακέτων. Η μεγαλύτερη απόδοση επιτυγχάνεται με τα τελευταία γένια έξυπνα NICs με ταξινόμηση των πακέτων και διάφορα buffers λήψης και αποστολής τα όποια το Crossbow είναι σε θέση να διαχειριστεί. Υπάρχουν πολλές πτυχές του σχεδιασμού του Crossbow που διευκολύνουν την αύξηση της αποδοτικότητας, αλλά ένας από τους μεγαλύτερους είναι ότι ασχολείται πιο αποτελεσματικά και με την ροή των εισερχόμενων πακέτων

4.2 ΤΕΧΝΟΛΟΓΙΕΣ VIRTUALIZATION

Το Crossbow επίσης ενισχύει την αξία των τεχνολογιών για virtualization της Sun για την ενοποίηση server χρησιμοποιώντας Open Solaris είτε ως OS είτε ως υποκείμενο σύστημα υποδοχής για εικονικά λειτουργικά συστήματα μηχανών clients. Το Λειτουργικό Σύστημα Solaris Containers είναι μια τεχνολογία Virtualization που επιτρέπει σε ένα λειτουργικό σύστημα να εμφανίζεται σαν πολλαπλά εικονικά ατομικά OS. Το πλεονέκτημα σε αυτή την προσέγγιση είναι ότι ενώ οι εφαρμογές βλέπουν ένα λειτουργικό σύστημα το οποίο μπορούν να διαχειρίζονται αποκλειστικά σαν δικό τους στην πραγματικότητα όμως αυτά τα εικονικά OS τρέχουν πάνω σε ένα. Τα Solaris Containers όταν μπαίνουν σε θέση να συγκριθούν με άλλες τεχνολογίες virtualization, μπορούν να κάνουν πιο καλή διαχείριση των πόρων του συστήματος και αυτό γιατί το OS εποπτεύει την χρήση της CPU και της μνήμης καθώς επίσης και τους διαθέσιμους πόρους που υπάρχουν στο δίκτυο. Επίσης έχουν και πολλές δυνατότητες κλιμάκωσης εξαιτίας των μικρών αναγκών που χρειάζονται από πόρους του OS. Τέλος η δημιουργία και η καταστροφή των containers είναι μια πολύ "ελαφριά" διεργασία που διευκολύνει την χρήση τους σε πολύ δυναμικά περιβάλλοντα. Πριν από το crossbow, οι εφαρμογές που έτρεχαν πάνω σε Solaris Containers μπορούν να έχουν πρόσβαση NICS αλλά, εάν μια δικτυακή στοίβα που ήταν αποκλειστικά για ένα container διαμεριζόταν τότε ένα αποκλειστικό NIC θα χρειαζόταν ή ένα αποκλειστικό VLAN. Με το Crossbow στον Container μπορούν να εκχωρηθούν όλα VNICS χρειάζονται και κάθε ένα θα έχει την δίκια του αποκλειστική στοίβα της όποια το bandwidth, η κατανομή της CPU και όλες οι προτεραιότητες του συστήματος να μπορούν να διαχειριστούν .

Λειτουργικό Σύστημα Virtualization

Το Solaris Containers είναι μια τεχνολογία Virtualization που επιτρέπει σε μία εμφάνιση του λειτουργικού συστήματος να προσφέρει πολλαπλά εικονικά λειτουργώντας σαν απομονωμένα εικονικά λειτουργικά συστήματα. Το πλεονέκτημα σε αυτήν την προσέγγιση είναι ότι καθώς οι διεργασίες βλέπουν ένα εξ ολοκλήρου δικό τους OS, στην πραγματικότητα υπάρχουν αλλά εικονικά "περιβάλλοντα" τα όποια τρέχουν σε ένα OS παράλληλα. Το Solaris Containers όταν συγκρίνεται με άλλες τεχνολογίες Virtualization μπορεί να κάνει καλύτερη χρήση των πόρων του συστήματος και αυτό γιατί το OS μπορεί να ελέγχει τον επεξεργαστή, την μνήμη και την κατανομή των δικτυακών πόρων που υπάρχουν. Οι Containers έχουν επίσης πολύ καλές ιδιότητες κλιμάκωσης εξαιτίας του μικρού φόρτου που χρησιμοποιούν στο OS.

Τέλος, η κατασκευή και η δημιουργία των Containers είναι μια πολύ μικρή διεργασία η όποια διευκολύνει την χρήση τους σε πιο δυναμικά περιβάλλοντα.

Με το Open Solaris Containers και το virtualization δικτύου, μπορούν πολλοί servers (καθώς και υπηρεσίες) να ενοποιούνται σε μια εμφάνιση του Open Solaris.

Το xVM είναι μια open source τεχνολογία virtualization βασισμένη στο Xen η όποια επιτρέπει να υπάρχουν windows, Linux και Solaris σαν guest λειτουργικά συστήματα πάνω σε ένα Open Solaris ως διαχειριστή. Αυτή η δυνατότητα θα τροφοδοτηθεί μέσω του Sun TM xVM Ops Center, το όποιο κυκλοφορεί σύντομα.

Το logical domains είναι μια τεχνολογία virtualization η όποια είναι ενσωματωμένη στο SPARC και είναι βασισμένη πάνω σε μια επίσης διαχειριστική αρχιτεκτονική. Κάθε guest domain έχει μια εικονική δικτυακή συσκευή η όποια θα χρησιμοποιείται για να έχει πρόσβαση σε μια υποκείμενη

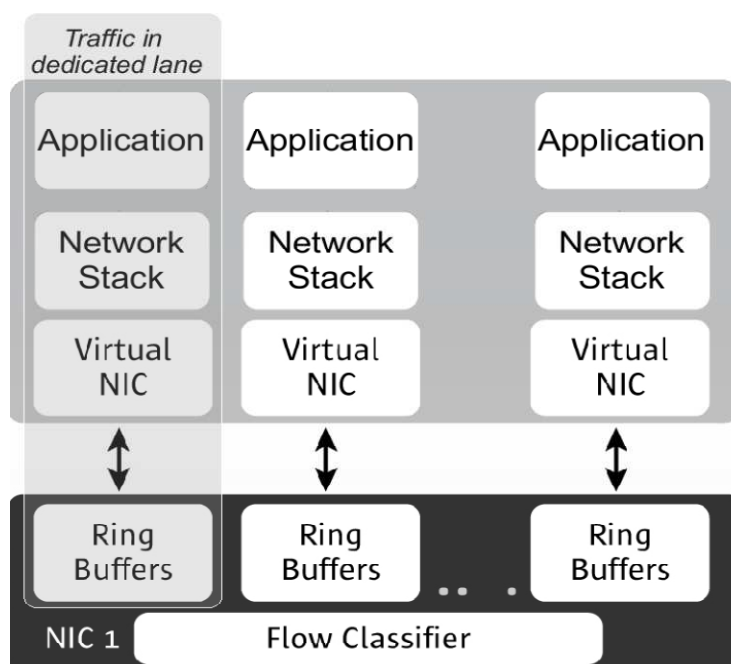
διασύνδεση δικτύου και για να έχει πρόσβαση και σε άλλα domains. Η συσχέτιση με τις ρυθμίσεις του δικτύου (bandwidth , πόροι CPU) σε LDOM guests δεν υποστηρίζεται στο Open Solaris.

Το Virtual Box είναι μια τεχνολογία virtualization η οποία επιτρέπει να δουλεύει μια ποικιλία από x86 λειτουργικά συστήματα σαν μια εφαρμογή από μια ποικιλία από hosts συμπεριλαμβανομένου και του Open Solaris. Το Virtual Box είναι μια ενδιαφέρουσα εφαρμογή του Virtualization ιδιαίτερα για ανάπτυξη διαφόρων πειραμάτων και γενικών δοκιμών. Τυπικά , ένα απλός server θα μπορούσε να στηθεί έχοντας πολλά λειτουργικά συστήματα για την χρήση διαφόρων σεναρίων. Για παράδειγμα μια 3-tier εφαρμογή Web/database αρχιτεκτονική. Ο σχεδιαστής θα μπορεί να αλληλεπιδρά με αυτά τα guest λειτουργικά μέσω πχ του VNC η του RPD με το να κάνει login στους guests. Αν το Virtual Box τρέχει πάνω σε Open Solaris τότε όλα τα πλεονεκτήματα που έχει το Crossbow στο virtualization δικτύου και διαχείριση των πόρων μπορούν να χρησιμοποιηθούν από τις εικονικές NICs που χρησιμοποιούνται πάνω σε Solaris ,Linux και Microsoft guests λειτουργικά συστήματα. Χρησιμοποιώντας το Crossbow είναι δυνατό να δημιουργήσουμε μια ολόκληρη δικτυακή τοπολογία μέσα σε ένα Server

Από τα παραπάνω , πρέπει να είναι πλέον κατανοητή η δράση της αρχιτεκτονικής του Crossbow που έχει σε μια εικονική προσομοίωση δικτύου και την διαχείριση των πόρων. Τώρα θα προχωρήσουμε σε λεπτομέρειες πάνω στην αρχιτεκτονική για να καταλάβουμε και αλλά χαρακτηριστικά όπως επίσης να καταλάβουμε γιατί το Crossbow είναι κάτι παραπάνω από virtualization. Το Crossbow είναι ένα project το οποίο έχει μεγάλη επίδραση στην στοίβα δικτύωσης και κυρίως το επίπεδο ζεύξης για την επανασχεδίαση μιας νέας γενιάς της στοίβας δικτύωσης.

4.3 Αρχιτεκτονική

Τα δομικά στοιχεία αυτής της νέας αρχιτεκτονικής είναι εικονικά NICS σε μια κατασκευή για την διαίρεση μιας φυσικής κάρτας δικτύου σε πολλές εικονικές. Μια εικονική κάρτα δικτύου μπορεί να διατρεχθεί από την μεριά μόνο των εφαρμογών όπως ακριβώς και στην φυσική κάρτα . Τα χαρακτηριστικά μιας Virtual NIC (bandwidth,πόροι , CPU) έχουν ανατεθεί στους χρηστές να τα διαχειρίζονται καθώς και άλλες προτεραιότητες μπορούν να ελέγχουν δυναμικά. Το Crossbow έχει σχεδιαστεί ως μια δομή συσχετισμένη παράλληλα με την στοίβα δικτύωσης. Μπορούμε να φανταστούμε την NIC σαν ένα δρόμο και το Crossbow επιτρέπει την διαίρεση του δρόμου σε πολλές λωρίδες .Κάθε λωρίδα αντιπροσωπεύει την ροή των πακέτων και οι ροές αυτές είναι ανεξάρτητες μεταξύ τους, καμιά κοινή ουρά και καμιά κοινή απειλή. Συγκεντρώνοντας αυτή την αρχιτεκτονική στην σύγχρονη NIC η οποία παρουσιάζεται στο διάγραμμα που ακολουθεί:



Εικόνα 18 Αρχιτεκτονική Crossbow

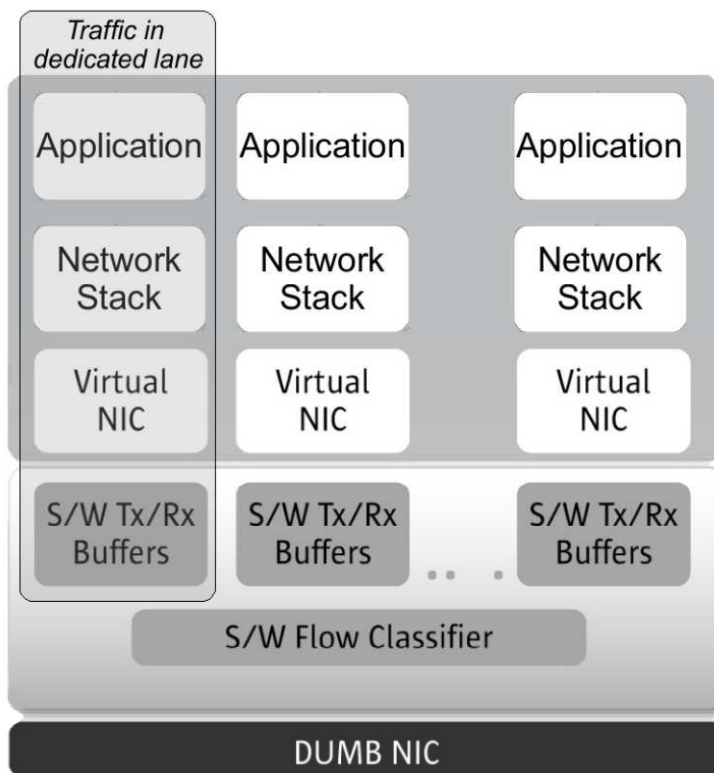
Ένα βασικό μέρος του σχεδιασμού είναι η ταξινόμηση της ροής που γίνεται στην NIC. Μόλις η εισερχόμενη ροή ταξινομηθεί τότε μπαίνει σε αποκλειστική ιδιωτική λωρίδα η οποία υποδεικνύεται στην σκιασμένη περιοχή. Η NIC έχει μια δομή μετάδοσης Tx και μια δομή λήψης Rx αποκλειστικά για κάθε λωρίδα. Τα δεδομένα είναι σχεδιασμένα σαν δακτυλίδι και αυτό γιατί τα δεδομένα λήψης που δεν έχουν μεταδοθεί στο λειτουργικό σύστημα θα αντικατασταθούν τελικά από νέα δεδομένα. Τα πακέτα που δεν πρέπει να χαθούν και θα δούμε στην συνέχεια ότι τα πακέτα θα πρέπει να χαθούν αν ο φόρτος είναι τόσο μεγάλος ,είναι πολύ καλύτερο να τα αφήνουμε μέσα στην NIC παρά να αφήνουμε στο OS να χρησιμοποιήσει πόρους για να αποφασίσει αν θα πρέπει να χαθούν

Ένας διαχειριστής υλικού δεν είναι σημαντική απαίτηση όπως θα δούμε παρακάτω, αλλά δίνει την δυνατότητα για μεγαλύτερη απόδοση. Ο διαχειριστής μπορεί να προγραμματιστεί στο να καθορίσει την εμβέλεια στο επίπεδο του OSI τα Layers 2,-3,-4 τα εξής ορίσματα :

- Διεύθυνση MAC
- Διεύθυνση και προορισμός στη διεύθυνση IP
- Πρωτόκολλο
- Πόρτα

Λειτουργικά αυτό σημαίνει ότι η HTTPS κίνηση μπορεί να διαχειριστεί από μια λωρίδα, η HTTP από μια άλλη και η FTP από μια τρίτη λωρίδα. Και άλλοι προσθετοί ή εναλλακτικοί έλεγχοι μπορούν να προστεθούν βασισμένοι στις διευθύνσεις πηγής και προορισμού. Σε ένα περιβάλλον όπου η ασφάλεια είναι σημαντική, η ταξινόμηση του Hardware θεωρείται μια πιο ασφαλής λύση. Θα μπορούσε να δημιουργηθεί ένα σύστημα Solaris Container ώστε να στεγάσει την εφαρμογή, να ορίσει ένα εικονικό VNIC και όλη την διαχείριση μέσω αποκλειστικού hardware σε ταξινομημένες λωρίδες . Από την πλευρά του όλου συστήματος όποτε το hardware ή το software εκτελούνται , η κατανομή στις λωρίδες διατηρείται με την ταξινόμηση μέσω του hardware να θεωρείται μια πιο χρηστική λύση.

Δεν έχει το κάθε NIC μια διαχείριση ροής ακόμη όμως και να είχε τότε ο αριθμός των λωρίδων που απαιτείται από το OS μπορεί να είναι παραπάνω από αυτό που έχει δοθεί στο NIC



Εικόνα 19 Λωρίδες κίνησης ταξινομημένες από το OS

Στην συγκεκριμένη περίπτωση , όλη η αρχιτεκτονική για την υποστήριξη εικονικών NICs είναι ένα μέρος του λογισμικού με αποκλειστικά δακτυλίδια λήψης και μετάδοσης για να δημιουργηθούν εξ ολοκλήρου οι λωρίδες κίνησης στο λογισμικό. Η ίδια προσέγγιση επιτρέπει σε εφαρμογές οι οποίες πχ χρειάζονται 20 VNICs σε ένα σύστημα να χρησιμοποιούν ένα διαχειριστή ο οποίος τα τροφοδοτεί με μόνο 16 δακτυλίδια κίνησης. Το λογισμικό επιτρέπει την δημιουργία πραγματικών λωρίδων κίνησης από hardware όσο και από εικονικό υλικό στο οποίο διαμοιράζεται μια πραγματική λωρίδα κίνησης . Το crossbow είναι ένα εργαλείο για την δημιουργία ενός εικονικού δικτύου αλλά και ένα σύστημα διαχείρισης πόρων. Η ικανότητα να καθορίζει τα χαρακτηριστικά αυτών των λωρίδων κίνησης είναι ένα βασικό στοιχείο της όλης του σχεδίασης. Τέλος το crossbow επιτρέπει την διαχείριση των πόρων μέσω των παρακάτω μηχανισμών :

- καθορισμός ορίων bandwidth
- καθορισμός προτεραιοτήτων κίνησης
- καθορισμός της διαχείρισης της κίνησης από CPUs

Τα όρια του bandwidth μπορούν να χρησιμοποιηθούν είτε για να εξασφαλίσουν ένα ελάχιστο όριο πόρων είτε για να αποτρέψουν ένα εικονικό VNIC από το να χρησιμοποιήσει παραπάνω από όσο πρέπει. Μια σύνδεση 1Gb θα μπορούσε να σπάσει σε 3 εικονικές οι οποίες θα μπορούσαν να έχουν 100Mb, 100Mb και 800Mb/Sec. Με το να εκχωρήσουμε ένα πιο μεγάλο εύρος ζώνης στην τελευταία σύνδεση δείχνει την λειτουργική σημασία των εφαρμογών που χρησιμοποιούν αυτή την σύνδεση και γι αυτό μπορούν να χρησιμοποιηθούν πολύ πιο αποτελεσματικά ώστε να επιτύχουν μεγαλύτερους QoS στόχους. Αυτές οι δυνατότητες διαχείρισης των πόρων συγκροτούν την όλη αρχιτεκτονική του συστήματος. Παρακάτω παρατίθενται 2 χαρακτηριστικά παραδείγματα:

Η ώθηση της ροής έλεγχου κοντά στο NIC είναι σημαντική. Στην κλασική ρύθμιση κίνησης οι εφαρμογές απλά φέρνουν ένα πακέτο στην ουρά του OS από την οποία θα πιθανά να χρησιμοποιήσει ένα μεγάλο μέρος επεξεργασίας και διαχείρισης του πακέτου. Αν η ροη μετρηθεί στο NIC ένα χαμένο πακέτο θα έχει μικρότερη επίδραση στο σύστημα. Για NICs τα οποία ελέγχουν από μόνά τους τα Rx\Rx τα χαμένα πακέτα δεν επιβαρύνουν καθόλου την CPU. Άλλες υλοποιήσεις QoS έχουν συνήθως ένα στρώμα το οποίο βρίσκεται στην στοίβα δικτύου ένα σημείο κορεσμού το οποίο δεν υπολογίζει την ζήτηση πόρων από άλλα μέρη του συστήματος. Με το Open Solaris Resource management είναι δυνατή η παροχή μέσω επιβολής μιας εφαρμογής από μια ελάχιστη κατανομή πόρων μέσω όμως ο καθορισμός και μόνο του bandwidth δεν είναι αρκετός. Ο έλεγχος των πόρων δεν περιορίζεται μόνο σε VNICS. Μπορούν να χρησιμοποιηθεί επίσης για την διαχείριση πραγματικών NICs ή πιο συγκεκριμένα αφού ένα NIC μπορεί να έχει πολλές πόρτες για να διαχειριστεί κάθε μια από αυτές καθορίζοντας το bandwidth ,την CPU καθώς και άλλες προτεραιότητες.

Οι δικτυακοί πόροι έλεγχου δεν είναι περιορισμένοι στις VNICS. Μπορούν επίσης να ελέγχουν και πραγματικές NICS ή και αλλά πράγματα δεδομένου ότι η NIC μπορεί να έχει διάφορες φυσικές πόρτες και να διαχειρίζεται για κάθε μια το bandwidth,CPU και άλλες ρυθμίσεις. Ένα άλλο στοιχείο της όλης αρχιτεκτονικής το οποίο είναι και αξιοσημείωτο είναι ότι ο τρόπος με τον οποίο το Crossbow διαχειρίζεται τις διακοπές. Σε χαμηλές ταχύτητες ωφέλιμων φορτίων τα πακέτα διαχειρίζονται με τον κλασικό τρόπο διακοπής . Αυτή είναι μια κατάλληλη προσέγγιση για χαμηλού ρυθμού και μεγάλου όγκου διακίνησης δεδομένων. Σε υψηλές ταχύτητες με μεγάλο όγκο δεδομένων οι διακοπές μπορούν να έχουν μεγάλο αντίκτυπο στο συνολικό throughput του συστήματος.

Υπάρχουν διαφορές τεχνικές οι οποίες για τον περιορισμό αυτού το προβλήματος αλλά πιο ουσιαστικά μια διακοπή ανά κάθε πακέτο δεν είναι και πολύ αποτελεσματική σε αρκετά πυκνό δίκτυο. Το crossbow αλλάζει αυτόματα από την λειτουργία διακοπής σε λειτουργία ανάλυσης όταν ο ρυθμός αύξησης των πακέτων υπερβαίνει το απαιτούμενο όριο. Η ανάλυση έχει ένα πλεονέκτημα όσο αφορά τα πολύ φορτωμένα δίκτυα. Μια ανάλυση από τον χρήστη μπορεί να επιστρέψει μια σειρά από πολλά πακέτα σε μια εργασία πιο πολύ αποτελεσματική από μια διακοπή ανά κάθε πακέτο.

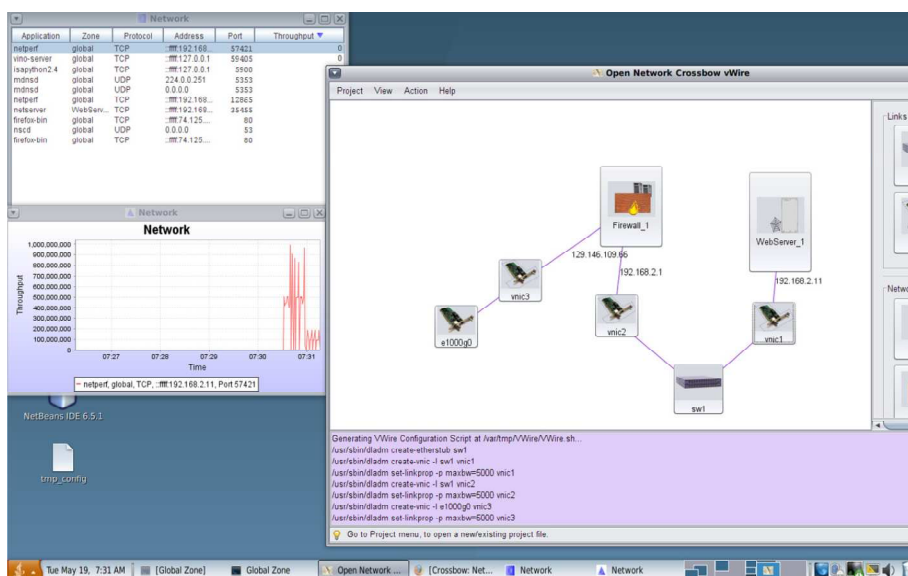
4.4 Εικονικά σενάρια δικτύωσης

Η αξία της εικονικής δικτύωσης όπως έχει περιγράψει παραπάνω είναι εμφανής. Ο συνδυασμός των εικονικών NICs και η δυνατότητα να διαχειριστούν οι πόροι των συστημάτων κάνει ένα σημαντικό βήμα για τεχνολογίες εικονικών servers. Για παράδειγμα η δυνατότητα να δημιουργηθεί μια φυσική διασύνδεση 10 Gb ή 1 Gb σε μικρότερες οι οποίες μπορούν να εκχωρηθούν σε ένα xVM ή σε ένα Solaris container είναι απαραίτητη καθώς ο διαχειριστής ή το λειτουργικό σύστημα είναι αυτό το οποίο προσδιορίζει το εύρος ζώνης, τις προτεραιότητες και τους πόρους CPU που θα διατεθούν στην εικονική εφαρμογή

4.4.1 VWireBuilder

Ένα πολύ καλό παράδειγμα του τι μπορεί να γίνει με το crossbow είναι μέσω ενός εργαλείου το οποίο ονομάζεται vWireBuilder. Το σχήμα που ακολουθεί παρουσιάζει ένα στιγμιότυπο της οθόνης του εργαλείου που χρησιμοποιείται. Σε αυτό το παράδειγμα έχουν τοποθετηθεί κάποια στοιχεία πάνω στον καμβά υλοποίησης. Κάνοντας κλικ πάνω τους μπορούν γίνουν οι απαραίτητες ρυθμίσεις που χρειάζονται και έπειτα πρέπει να γίνει η διαμόρφωση και η εκτέλεση της όλης τοπολογίας.

Το vWireBuilder έπειτα δημιουργεί όλη αυτή την τοπολογία του δικτύου μέσα στο σύστημα μέσω της δημιουργίας των Containers για κάθε μια από τις συσκευές που δημιουργήθηκαν (firewall, Web server) καθώς επίσης τα VNICS και τα switches για τις συνδέσεις δικτύου. Στην συνέχεια ξεκινά ο web server μέσα από τον web server container. Η IP QoS πραγματοποιείται κάνοντας κλικ πάνω στα VNICS για τον καθορισμό του εύρους ζώνης. Μετά προστίθεται στο σύστημα μια γεννήτρια δεδομένων και το vWireBuilder εμφανίζει την όλη κίνηση μέσω του μενού του. Αξίζει να σημειωθεί ότι δεν ρυθμιστικό το firewall μέσα από τον Container του firewall κάτι το οποίο να γίνει με χειρονακτικό τρόπο. Το σύνολο της δημιουργίας όλης της τοπολογίας και επίσης και των ρυθμίσεων δικτύου που χρειαστήκαν έγιναν μέσα σε πολύ μικρό χρονικό διάστημα .



Εικόνα 20 vWireBuilder

Κεφάλαιο 5 Επικαιροποίηση του Crossbow

5.1 Project (Network Virtualization and resource control)

Στο πλαίσιο αυτής της άσκησης θα εξετάζουμε :

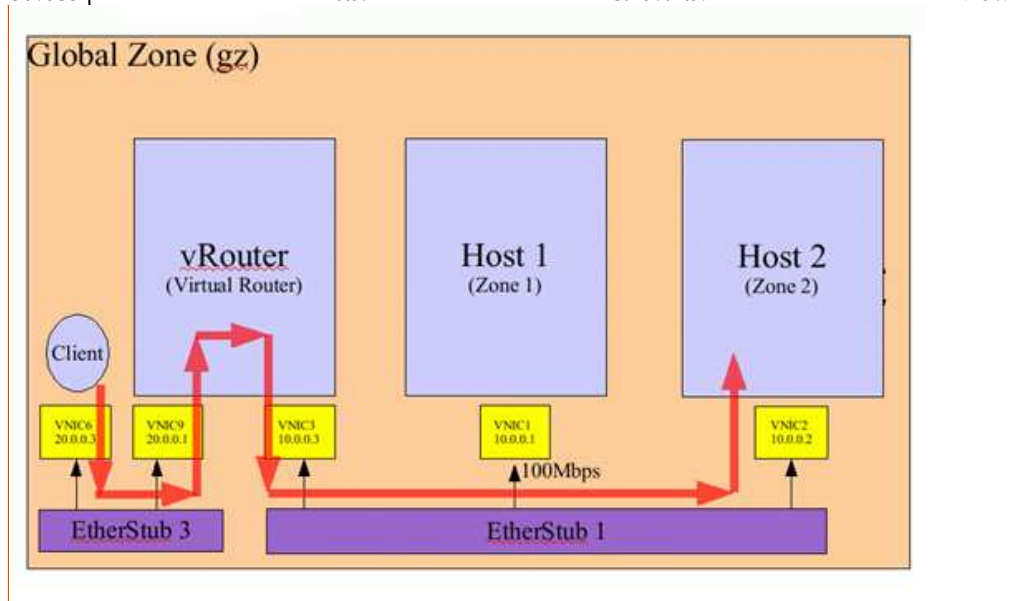
- Πώς να ρυθμίζουμε ένα εικονικό δίκτυο που έχει δύο υποδίκτυα και συνδέονται με ένα β εικονικό router χρησιμοποιώντας crossbow και zones.
- Πώς να ρυθμίσουμε τις ταχύτητες δικτύου για την προσομοίωση πολλαπλών ταχυτήτων στο δίκτυο

5.1.1 Σκοπός project

Σκοπός είναι η δημιουργία ενός πραγματικού δικτύου που να αποτελείται από Hosts,Switches και Routers σαν ένα εικονικό δίκτυο σε ένα φορητό υπολογιστή. Το Virtual Network δημιουργείται χρησιμοποιώντας το project Crossbow του Open Solaris και οι hosts κτλ δημιουργούνται χρησιμοποιώντας το Solaris zones. Όλα τα βήματα για τη δημιουργία της εικονικής τοπολογίας εξηγούνται παρακάτω.

5.1.2 Υλοποίηση project

Χρησιμοποιούμε zones για τη δημιουργία των host1,host2 και virtual Router ενώ η παγκόσμια ζώνη(GZ) λειτουργεί σαν πελάτης(client).Οι Etherstubs λειτουργούν σαν εικονικοί διακόπτες για τη σύνδεση των εικονικών NICs.



Σχόλιο [x1]: Επιτέλους φτιάξτε αυτό σχήμα να μην φαίνεται το legend μέσα στο σχήμα.

Εικόνα 21 Crossbow virtual network

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Δημιουργία του virtual network:

Ξεκινάμε με τη δημιουργία των 2 εικονικών μεταγωγών με την εντολή dladm

```
gz# dladm create-etherstub vswitch1
gz# dladm create-etherstub vswitch3
gz # dladm show-etherstub
LINK
vswitch1
vswitch3
```

Κατασκευή των απαραίτητων Virtual NICs .Το VNIC1 έχει περιορισμένη ταχύτητα 100Mbps ενώ οι άλλοι δεν έχουν κανένα όριο.

```
gz#dladmcreate-vnic-lvswitch1vnic1
gz#dladm create-vnic-l vswitch1 vnic2
gz # dladm create-vnic-1 vswitch1 vnic3
gz # dladm create-vnic-1 vswitch3 vnic6
gz # dladm create-vnic-1 vswitch3 vnic9
gz # dladm show-vnic
LINK OVER SPEED MACADDRESS MACADDRTYPE
vnic1 vswitch1 - Mbps 2:8:20:8 d: de: b1 random
vnic2 vswitch1 - Mbps 2:8:20:4 a: B0: f1 random
vnic3 vswitch1 - Mbps 2:8:20:46:14:52 random
vnic6 vswitch3 - Mbps 2:08:20: bf: 13:2f random
vnic9 vswitch3 - Mbps 2:08:20: ed: 1:45 random
```

Δημιουργία hosts και εκχώρηση των VNICs που τους αντιστοιχούν .Επίσης, γίνεται και η κατασκευή του Virtual Router στον οποίο θα ανατεθούν τα VNIC3 και VNIC9 πάνω στα vswitch1 και vswitch3 αντίστοιχα. Τόσο το Virtual Router όσο και οι hosts δημιουργούνται με τη χρήση ζωνών σε αυτό το παράδειγμα.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Δημιουργία μιας βάσης ζώνης στην οποία μπορεί να γίνει η κλωνοποίηση .

```
gz# zfs create -o mountpoint=/vnm rpool/vnm
gz# chmod 700 /vnm gz# zonecfg -z vnmbase
vnmbase: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vnmbase> create
zonecfg:vnmbase> set zonepath=/vnm/vnmbase
zonecfg:vnmbase> set ip-type=exclusive
zonecfg:vnmbase> add inherit-pkg-dir
zonecfg:vnmbase:inherit-pkg-dir> set dir=/opt
zonecfg:vnmbase:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vnmbase:inherit-pkg-dir> end
zonecfg:vnmbase> verify
zonecfg:vnmbase> commit
zonecfg:vnmbase> exit
```

Αυτό το μέρος παίρνει 15-20 λεπτά

gz# zoneadm -z vnmbase install (Με αυτή την εντολή κάνουμε install τη βασική ζώνη από την οποία γίνεται η κλωνοποίηση των υπόλοιπων ζωνών)

Κατασκευή των 2 hosts και του virtual router ως έξης:

Δημιουργία του host1 και του αποδίδουμε το vnic1

Ο κώδικας για τη δημιουργία του host1 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας
Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 10.0.0.1 ως ip για το vnic1.Βάζουμε 10.0.0.3 για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ.

Δημιουργία του host2 και του ανάθεση του στο vnic2

Ο κώδικας για τη δημιουργία του host2 βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας
Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 10.0.0.2 ως ip για το vnic2.Βάζουμε 10.0.0.3 για default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ.

Δημιουργία του Virtual Router

Ο κώδικας για τη δημιουργία του vrouter βρίσκεται στο παραρτημα στο τέλος της πτυχιακής εργασίας
Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 10.0.0.3 ως ip για το vnic3 και 20.0.0.1 ως ip για το vnic9.Βάζουμε none για το default route.Μπορούμε να πούμε none για naming service,IPv6,Kerberos security κ.τ.λ.

Με τη παρακάτω εντολή γίνεται η ενεργοποίηση το ipv4 στο virtual router αφού έχουμε κάνει log in στη ζώνη :

```
vrouter# svcadm enable network/ipv4-forwarding:default
```

Ανοίγουμε το global zone για να ρυθμίσουμε το vnic6

```
gz# ifconfig vnic6 plumb 20.0.0.3/24 up
gz# route add 10.0.0.0 20.0.0.1
gz# ping 10.0.0.1
10.0.0.1 is alive
```

```
gz# ping 10.0.0.2
10.0.0.2 is alive
```

Στην ζώνη host1 κάνουμε την εντολή ping ότι όλα δουλεύουν σωστά

```
host1# ping 20.0.0.3
20.0.0.3 is alive
```


Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
host1# ping 10.0.0.2
10.0.0.2 is alive
```

Αφού έχουμε ρυθμίσει όλα τα παραπάνω μπορούμε να διαμορφώσουμε τη ταχύτητα σύνδεσης σε όλους τους συνδέσμους σ' αυτό το παράδειγμα θα ρυθμίσουμε τη ταχύτητα σύνδεσης του vnic1 σε 100Mbps

```
gz# dladm set-linkprop -p maxbw=100 vnic1
```

Έλεγχος απόδοσης δικτύου.

Στα εικονικά συστήματα που αντιπροσωπεύουν τους χρήστες γίνεται εκκίνηση του προγράμματος

```
host1#/opt/tools/netsserver&
```

```
host2#/opt/tools/netsserver&
```

Στη στη global zone με την εντολή netperf -H 10.0.0.2 και 10.0.0.1 θα βρούμε το Throughput

```
gz# /opt/tools/netperf -H 10.0.0.2
```

```
TCP STREAM TEST to 10.0.0.2 : histogram
```

```
Recv Send Send
```

```
Socket Socket Message Elapsed
```

```
Size Size Size Time Throughput
```

```
bytes bytes bytes secs. 10^6bits/sec
```

```
49152 49152 49152 10.00 2089.87
```

```
gz# /opt/tools/netperf -H 10.0.0.1
```

```
TCP STREAM TEST to 10.0.0.2 : histogram
```

```
Recv Send Send
```

```
Socket Socket Message Elapsed
```

```
Size Size Size Time Throughput
```

```
bytes bytes bytes secs. 10^6bits/sec
```

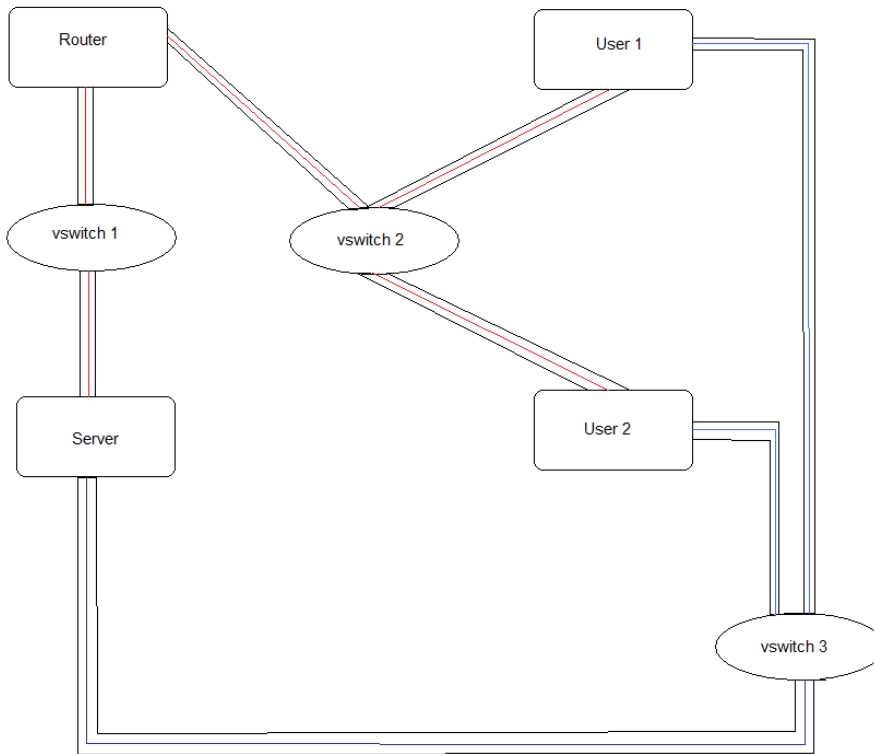
```
49152 49152 49152 10.00 98.78
```

```
Για ταχύτητα σύνδεσης vnic1=50Mbps:  
gz# dladm set-linkprop -p maxbw=50 vnic1  
gz# /opt/tools/netperf -H 10.0.0.2  
TCP STREAM TEST to 10.0.0.2 : histogram  
Recv Send Send  
Socket Socket Message Elapsed  
Size Size Size Time Throughput  
bytes bytes bytes secs. 10^6bits/sec  
49152 49152 49152 10.00 1429.86  
gz# /opt/tools/netperf -H 10.0.0.1  
TCP STREAM TEST to 10.0.0.2 : histogram  
Recv Send Send  
Socket Socket Message Elapsed  
Size Size Size Time Throughput  
bytes bytes bytes secs. 10^6bits/sec  
49152 49152 49152 10.00 45.72
```

```
Για ταχύτητα σύνδεσης vnic1=10Mbps :  
gz# dladm set-linkprop -p maxbw=10 vnic1  
gz# /opt/tools/netperf -H 10.0.0.1  
TCP STREAM TEST to 10.0.0.2 : histogram  
Recv Send Send  
Socket Socket Message Elapsed  
Size Size Size Time Throughput  
bytes bytes bytes secs. 10^6bits/sec  
49152 49152 49152 10.00 6.70
```

```
gz# /opt/tools/netperf -H 10.0.0.2  
TCP STREAM TEST to 10.0.0.2 : histogram  
Recv Send Send  
Socket Socket Message Elapsed  
Size Size Size Time Throughput  
bytes bytes bytes secs. 10^6bits/sec  
49152 49152 49152 10.00 14.09
```

5.2 Project (Network Virtualization and resource control with return link)



Εικόνα 22 Crossbow virtual network with return link

5.2.1 Σκοπός project

Σκοπός είναι η δημιουργία ενός πραγματικού δικτύου που να αποτελείται από 1 server, 3 Switches, 1 Router και 2 users σαν ένα εικονικό δίκτυο. Το Virtual Network δημιουργείται χρησιμοποιώντας το project Crossbow του Open Solaris και οι users, server, router δημιουργούνται χρησιμοποιώντας το Solaris zones. Όλα τα βήματα για τη δημιουργία της εικονικής τοπολογίας εξηγούνται παρακάτω.

5.2.2 Υλοποίηση project

Με την χρήση των zones γίνεται η δημιουργία των user1,user2,server και virtual Router.Οι Etherstubs λειτουργούν σαν εικονικοί διακόπτες για τη σύνδεση των εικονικών NICs.

Δημιουργούμε το virtual network :

Ξεκινάμε με τη δημιουργία των 3 εικονικών διακοπών με την εντολή dladm

```
# dladm create-etherstub vswitch1
# dladm create-etherstub vswitch2
# dladm create-etherstub vswitch3
~# dladm show-etherstub
LINK
vswitch1
vswitch2
vswitch3
```

Κατασκευή των απαραίτητων Virtual NICs.

```
# dladm create-vnic -l vswitch1 vnic1
#dladm create-vnic -l vswitch1 vnic3
#dladm create-vnic -l vswitch2 vnic4
# dladm create-vnic -l vswitch2 vnic5
# dladm create-vnic -l vswitch2 vnic7
# dladm create-vnic -l vswitch3 vnic2
# dladm create-vnic -l vswitch3 vnic6
# dladm create-vnic -l vswitch3 vnic8
# dladm show-vnic
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE
vnic1	vswitch1	0	2:8:20:56:c3:8	random
vnic3	vswitch1	0	2:8:20:e2:e3:44	random
vnic4	vswitch2	0	2:8:20:9:74:1a	random
vnic5	vswitch2	0	2:8:20:89:b2:b3	random
vnic7	vswitch2	0	2:8:20:3:2c:81	random
vnic2	vswitch3	0	2:8:20:23:15:63	random
vnic6	vswitch3	0	2:8:20:fd:a4:79	random
vnic8	vswitch3	0	2:8:20:4b:6b:de	random

Δημιουργούμε 2 users και 1 server και τους εκχωρούμε τα vnics που τους αντιστοιχούν . Επίσης, δημιουργούμε το Virtual Router και του αναθέτουμε τα VNIC3 και VNIC4 πάνω στα vswitch1 και vswitch2 αντίστοιχα. Τόσο το Virtual Router όσο και ο server,users δημιουργούνται με τη χρήση ζωνών σε αυτό το παράδειγμα.

Δημιουργία μιας βάσης ζώνης με την οποία μπορεί να γίνει η κλωνοποίηση .

```
# zonecfg -z vnmbase
vnmbase: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vnmbase> create
zonecfg:vnmbase> set zonepath=/vnm/vnmbase
zonecfg:vnmbase> set ip-type=exclusive
zonecfg:vnmbase> add inherit-pkg-dir
zonecfg:vnmbase:inherit-pkg-dir> set dir=/opt
zonecfg:vnmbase:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vnmbase: inherit-pkg-dir> end
zonecfg:vnmbase> verify
zonecfg:vnmbase> commit
zonecfg:vnmbase> exit
# zoneadm -z vnmbase install
A ZFS file system has been created for this zone.
Publisher: Using opensolaris.org (http://pkg.opensolaris.org/release/).
Image: Preparing at /vnm/vnmbase/root.
Sanity Check: Looking for 'entire' incorporation.
Installing: Core System (output follows)
DOWNLOAD          PKGS    FILES    XFER (MB)
Completed          20/20   3021/3021 42.55/42.55
PHASE              ACTIONS
Install Phase      5747/5747
Installing: Additional Packages (output follows)
DOWNLOAD          PKGS    FILES    XFER (MB)
Completed          37/37   5600/5600 32.53/32.53
PHASE              ACTIONS
Install Phase      7337/7337
Done: Installation completed in 259,081 seconds.
Next Steps: Boot the zone, then log into the zone console; (zlogin -C) to complete the
configuration process
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Δημιουργία των 2 users, to server και to virtual router
Δημιουργία του server και του αποδίδουμε το vnic1, vnic2

Ο κώδικας για τη δημιουργία του server βρίσκεται στο appendix στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 10.0.0.2 ως ip για το vnic1 και 192.168.0.1 ως ip για το vnic2. Βάζουμε 10.10.0.1 για default route και στα 2 vnic. Μπορούμε να πούμε none για naming service, IPv6, Kerberos security κ.τ.λ. Βάζουμε nfs4_domain=dynamic.

Κατασκευή του virtual Router και του αποδίδουμε το vnic3 και το vnic4

Ο κώδικας για τη δημιουργία του vRouter βρίσκεται στο appendix στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 10.0.0.1 ως ip για το vnic3 και 172.72.0.1 ως ip για το vnic4. Βάζουμε none για default route και στα 2 vnic. Μπορούμε να πούμε none για naming service, IPv6, Kerberos security κ.τ.λ. Βάζουμε nfs4_domain=dynamic.

Με τη παρακάτω εντολή ενεργοποιούμε το ipv4 στο virtual router αφού έχουμε κάνει log in στη ζώνη :

```
vRouter# svcadm enable network/ipv4-forwarding:default
```

Δημιουργία του user1 ρύθμιση των virtual διεπαφών vnic5 και vnic6.

Ο κώδικας για τη δημιουργία του user1 βρίσκεται στο appendix στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 172.72.0.2 ως ip για το vnic5 και 192.168.0.2 ως ip για το vnic6. Βάζουμε 172.72.0.1 για default route στο vnic5 και 192.168.0.1 για default route στο vnic6. Μπορούμε να πούμε none για naming service, IPv6, Kerberos security κ.τ.λ. Βάζουμε nfs4_domain=dynamic

Κατασκευή του user2 και του αποδίδουμε το vnic7 και το vnic8.

Ο κώδικας για τη δημιουργία του user2 βρίσκεται στο appendix στο τέλος της πτυχιακής εργασίας. Στις ρυθμίσεις που θα κάνουμε στη ζώνη βάζουμε 172.72.0.3 ως ip για το vnic7 και 192.168.0.3 ως ip για το vnic8. Βάζουμε 172.72.0.1 για default route στο vnic7 και 192.168.0.1 για default route στο vnic8. Μπορούμε να πούμε none για naming service, IPv6, Kerberos security κ.τ.λ. Βάζουμε nfs4_domain=dynamic.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Αφού έχουν γίνει οι ρυθμίσεις όλα τα παραπάνω μπορούμε να διαμορφώσουμε τη ταχύτητα σύνδεσης σε όλους τους συνδέσμους' αυτό το παράδειγμα θα ρυθμίσουμε τη ταχύτητα σύνδεσης του vnic2 σε 2Mbps και 100Mbps και του vnic4 σε 20Mbps, 15Mbps, 10Mbps

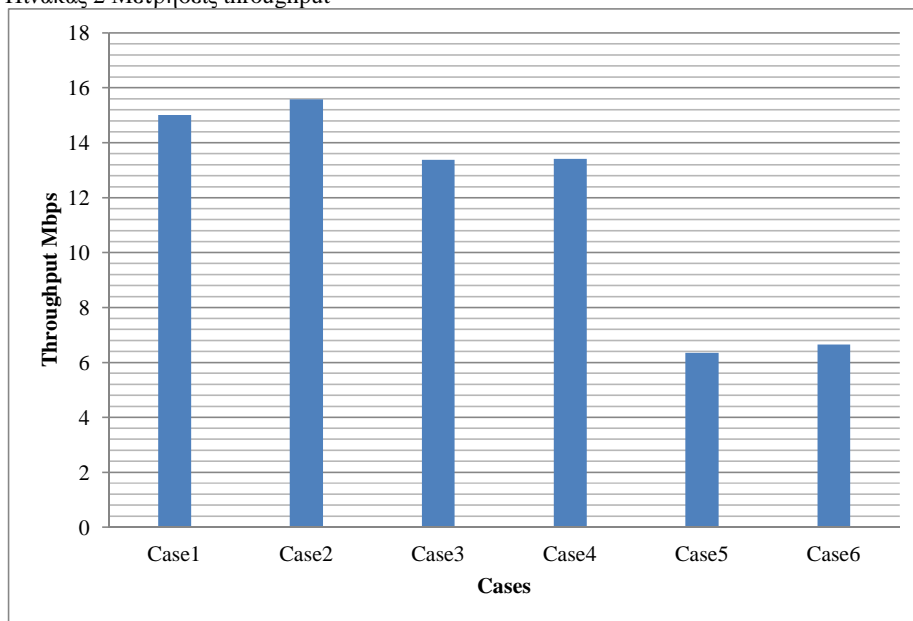
```
gz# dladm set-linkprop -p maxbw=20 vnic4
gz# dladm set-linkprop -p maxbw=2 vnic2
gz# dladm set-linkprop -p maxbw=100 vnic2
gz# dladm set-linkprop -p maxbw=15 vnic4
gz# dladm set-linkprop -p maxbw=2 vnic2
gz# dladm set-linkprop -p maxbw=100 vnic2
gz# dladm set-linkprop -p maxbw=10 vnic4
gz# dladm set-linkprop -p maxbw=2 vnic2
gz# dladm set-linkprop -p maxbw=100 vnic2
Γίνεται εγκατάσταση του netserver στο server και του netperf στο user1
user@opensolaris:/usr/bin$ pfexec cp netserver /vnm/server/root/var/tmp/.
user@opensolaris:/usr/bin$ pfexec zlogin server/var/tmp/netserver
el_GR.UTF-8: unknown locale
el_GR.UTF-8: unknown locale
Starting netserver at port 12865
Starting netserver at hostname 0.0.0.0 port 12865 and family AF_UNSPEC
user@opensolaris:/usr/bin$ pfexec cp netperf /vnm/user1/root/var/tmp/.
user@opensolaris:/usr/bin$ pfexec zlogin server/var/tmp/netperf -H 10.10.10.2
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Το vnic2 το χρησιμοποιούμε σαν uplink και το vnic4 σαν downlink.Απο τις μετρήσεις τις οποίες κάναμε βγάλαμε τις παρακάτω τιμές:

Ταχύτητες	Throughput στο user1
Case1: 20MBPS DOWN 2MBPS UP (adsl)	15.01 Mbits/sec
Case2: 20MBPS DOWN 100MBPS UP (Ethernet)	15.58 Mbits/sec
Case3: 15MBPS DOWN 2MBPS UP (adsl)	13.38 Mbits/sec
Case4: 15MBPS DOWN 100MBPS UP (Ethernet)	13,42 Mbits/sec
Case5: 10MBPS DOWN 2MBPS UP (adsl)	6.34 Mbits/sec
Case6: 10MBPS DOWN 100MBPS UP (Ethernet)	6.65 Mbits/sec

Πίνακας 2 Μετρήσεις throughput

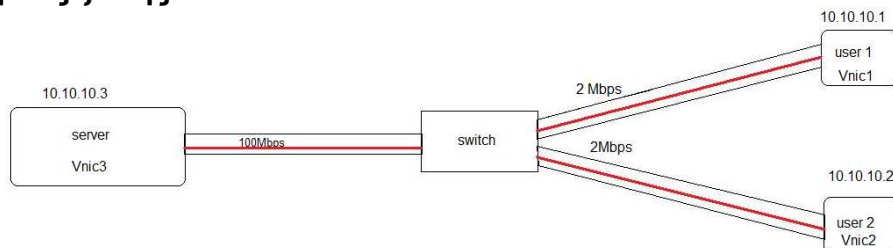


Εικόνα 23 Γραφική παράσταση throughput

ΚΕΦΑΛΑΙΟ 6

Σενάρια προσομοίωσης εικονικής πλατφόρμας

6.1 Σενάριο υλοποίησης συστήματος δυναμικής διαχείρισης εύρους ζώνης



Εικόνα 24 Σχηματική Αναπαράσταση του Σεναρίου

6.1.1 Σκοπός της υλοποίησης του σεναρίου :

Η δημιουργία ενός συστήματος μέσω του οποίου θα παρέχεται μια πολυμεσική υπηρεσία σε ένα αριθμό χρηστών από έναν Server και προοδευτικά θα γίνεται δυναμική διαχείριση του εύρους ζώνης στον ένα από τους χρηστές.

6.1.2 Υλοποίηση Σεναρίου

Χρησιμοποιούμε zones για τη δημιουργία των user1,user2 και server. Ο Etherstub λειτουργεί σαν εικονικός διακόπτης για τη σύνδεση των εικονικών NICs.

Δημιουργούμε δυο users και τους εκχωρούμε τα vnics που τους αντιστοιχούν (vnic1,vnic2) Πάνω στο vswitch1.Επίσης, δημιουργούμε τον server και του αναθέτουμε το vnic3 πάνω στο vswitch1. Τόσο ο server όσο και οι users δημιουργούνται με τη χρήση ζωνών σε αυτό το παράδειγμα.

```
~# dladm create-etherstub vswitch1
~# dladm show-etherstub
```

LINK

vswitch1

```
# dladm create-vnic -l vswitch1 vnic1
```

```
# dladm create-vnic -l vswitch1 vnic2
```

```
# dladm create-vnic -l vswitch1 vnic3
```

```
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS
vnic1	vswitch1	0 2:8:20:e0:84:e2	random
vnic2	vswitch1	0 2:8:20:7b:10:f5	random
vnic3	vswitch1	0 2:8:20:5b:c8:9e	random

Υπάρχουν 1 server,2 users και 1 switch.Το link του server είναι 100mbps, του user1 είναι 2mbps και του user2 είναι 2mbps.

```
# dladm set-linkprop -p maxbw=100 vnic3
```

```
# dladm set-linkprop -p maxbw=2 vnic2
```

```
# dladm set-linkprop -p maxbw=2 vnic1
```

Για t=0 ο server στέλνει στο user1 μια ροή udp 2mbps και στον user1 άλλη μια ροή 2mbps udp.

```
# flowadm add-flow -l vnic3 -a transport=UDP,local_port=623 asfflow
```

```
# flowadm add-flow -l vnic3 -a transport=UDP,local_port=692 hispflow
```

```
#flowadm set-flowprop -p maxbw=2 asfflow
```

```
# flowadm set-flowprop -p maxbw=2 hispflow
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Για $t=30\text{sec}$ ο server στέλνει στο user1 μια ροή udp 2mbps αλλά σε μια port2(s2) που ακούει ο user1

```
# flowadm add-flow -l vnic1 -a transport=UDP,local_port=712 prcflow
```

Για $t=60\text{sec}$ δίνουμε εντολή για αύξηση του link του user1 σε 5MBPS

```
# dladm set-linkprop -p maxbw=5 vnic1
```

Για $t=75\text{sec}$ δίνουμε εντολή για τερματισμό της ροής που πάει στο user1

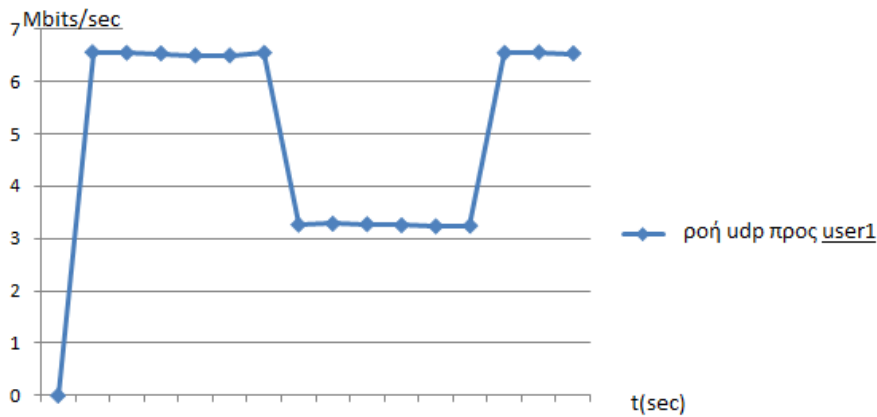
```
# flowadm remove-flow -t asfflow
```

Για $t=90\text{sec}$ δίνουμε εντολή για τερματισμό όλων

```
# flowadm remove-flow -t prcflow
```

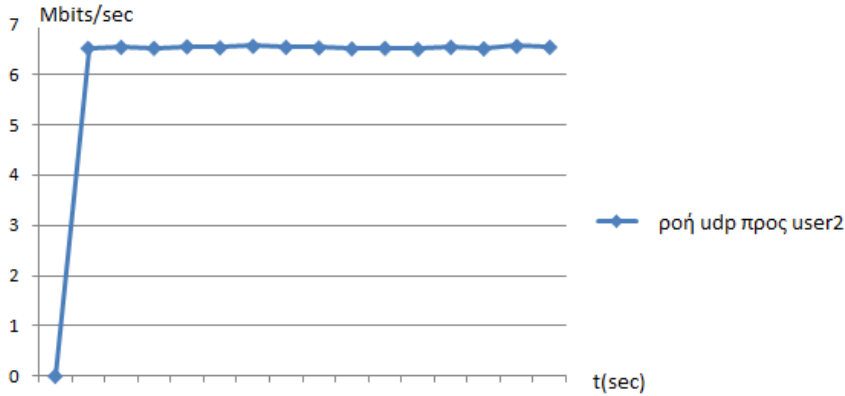
```
# flowadm remove-flow -t hispflow
```

Γίνεται χρήση του netperf για να βρούμε το bitrate των ροών που χρησιμοποιήθηκαν στο παραπάνω σενάριο . Οι πινάκες των μετρήσεων βρίσκονται στο παράρτημα. Παρακάτω ακολουθούν οι γραφικές παραστάσεις των μετρήσεων.



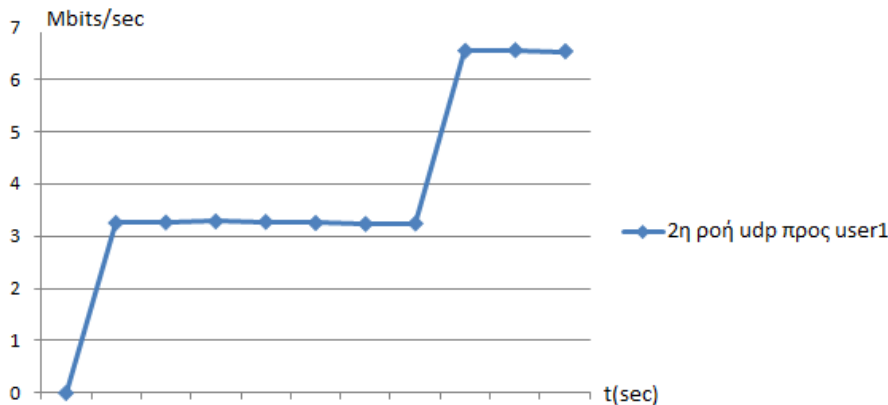
Εικόνα 25 Γραφική παράσταση πρώτης ροής UDP προς User1

Στην παραπάνω γραφική παρασταση βλέπουμε το bitrate της 2 Mbps UDP ροής που ξεκινάει για $t=0\text{sec}$ από τον Server και πάει στον User1. Για $t=30\text{sec}$ εισέρχεται άλλη μια ροή UDP 2 Mbps στο κανάλι οπότε το Bitrate της πρώτης ροής πέφτει στο μισό. Για $t=60\text{sec}$ το link του user1 ανεβαίνει στα 5 Mbps επομένως το bitrate της ροής αυξάνεται .



Εικόνα 26 Γραφική παράσταση των τιμών του Netperf User2(Mbps)

Στην παραπάνω γραφική παράσταση βλέπουμε το bit rate της 2Mbps UDP ροής που ξεκινάει για $t=0$ sec από τον Server και πάει στον User2. Παρατηρούμε ότι η ροή εκμεταλλεύεται και τα 2 Mbps του Link του User2 μέχρι το τέλος .



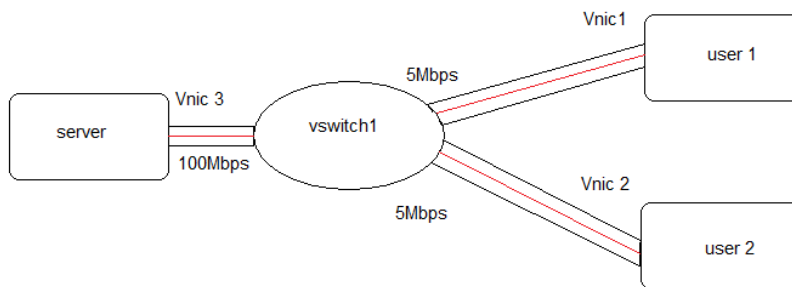
Εικόνα 27 Γραφική παράσταση δεύτερης ροής UDP προς User2

Στην παραπάνω γραφική παράσταση βλέπουμε το bitrate της δεύτερης 2 Mbps UDP ροής που για $t=30$ sec παει από τον server στον user2. Το bitrate της ροής αυξάνεται για $t=60$ sec επειδή το link του καναλιού του user2 αυξάνει σε 5 Mbps.

6.1.3 Παρατηρήσεις σεναρίου

Η όλη διαδικασία εκτέλεσης του παραπάνω σεναρίου μας οδηγεί στο συμπέρασμα ότι η δυναμική διαχείριση του εύρους ζώνης σε ένα εικονικό δίκτυο με την χρήση εικονικών συσκευών και ζεύξεων δικτύου είναι εφικτή καθώς από τις μετρήσεις που πήραμε όντως ανταποκρίνεται στις απαιτήσεις που καθορίσαμε στον σχεδιασμό του σεναρίου. Επίσης παρατηρούμε ότι δεν υπάρχουν απώλειες σε τέτοιο βαθμό ώστε το πρωτόκολλο (UDP) επικοινωνίας να μπορεί να προσφέρει όσο το δυνατόν καλύτερα την πολυμεσική υπηρεσία με βάση και το εύρος ζώνης που του παρέχεται στο κανάλι σε κάθε μια από τις περιπτώσεις.

6.2 Σενάριο υλοποίησης συστήματος προσομοίωσης επίγειας ψηφιακής τηλεόρασης



Εικόνα 28 σχηματική αναπαράσταση του σεναρίου

6.2.1 Σκοπός Σεναρίου

Η δημιουργία ενός συστήματος μέσω του οποίου θα έχουμε δυναμική διαχείριση εύρους ζώνης μαζί με την χρήση priorities των πρωτόκολλων UDP, TCP ώστε να επιτύχουμε μεγαλύτερη QoS στην μεταφορά πολυμεσικών δεδομένων .

6.2.2 Υλοποίηση Σεναρίου

Με την χρήση των zones γίνεται η δημιουργία των user1,user2 και server .Ο Etherstub λειτουργεί σαν εικονικός διακόπτης για τη σύνδεση των εικονικών NICs..Δημιουργούμε 2 users και τους εκχωρούμε τα vnics που τους αντιστοιχούν (vnic1,vnic2) Πάνω στο vswitch1.Επίσης, δημιουργούμε το server και του αναθέτουμε το vnic3 πάνω στο vswitch1. Τόσο ο server όσο και οι users δημιουργούνται με τη χρήση ζωνών σε αυτό το παράδειγμα.

```
# dladm create-etherstub vswitch1
# dladm show-etherstub
LINK
vswitch1
# dladm create-vnic -l vswitch1 vnic1
# dladm create-vnic -l vswitch1 vnic2
# dladm create-vnic -l vswitch1 vnic3
# dladm show-vnic
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE
vnic1	vswitch1	0	2:8:20:e0:84:e2	random
vnic2	vswitch1	0	2:8:20:7b:10:f5	random
vnic3	vswitch1	0	2:8:20:5b:c8:9e	random

Υπάρχουν 1 server, 2 users και 1 switch. Το link του server είναι 100mbps, του user1 είναι 5mbps και του user2 είναι 5mbps

```
# dladm set-linkprop -p maxbw=100 vnic3
# dladm set-linkprop -p maxbw=5 vnic2
# dladm set-linkprop -p maxbw=5 vnic1
```

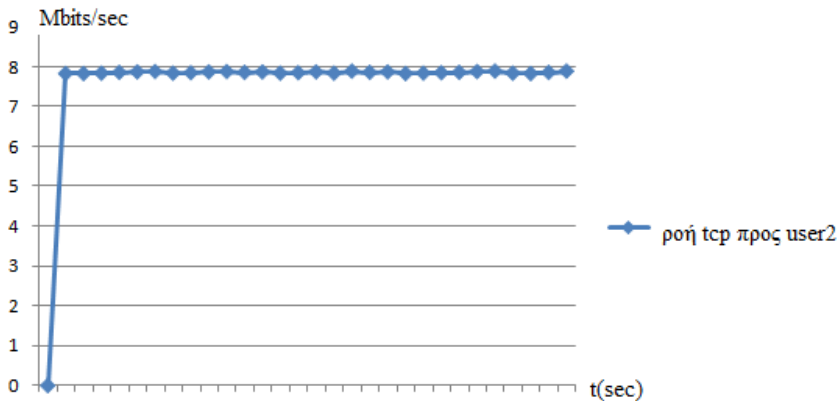
```
Για t=0 ο server στέλνει στο user2 μια ροή TCP low priority .
# flowadm add-flow -l vnic2 -a transport=TCP,local_port=750 rfflow
# flowadm set-flowprop -p priority=low rfflow
```

```
Για t=60 ο server στέλνει στο user1 μια ροή TCP high priority .
# flowadm add-flow -l vnic1 -a transport=TCP,local_port=753 rrhflow
# flowadm set-flowprop -p priority=high rrhflow
```

```
Για t=120 ο server στέλνει στο user1 μια ροή udp 2mbps .
# flowadm add-flow -l vnic1 -a transport=UDP,local_port=754 tsflow
# flowadm set-flowprop -p maxbw=2 tsflow
```

```
Για t=180 stop .
# flowadm remove-flow -t tsflow
# flowadm remove-flow -t rrhflow
# flowadm remove-flow -t rfflow
```

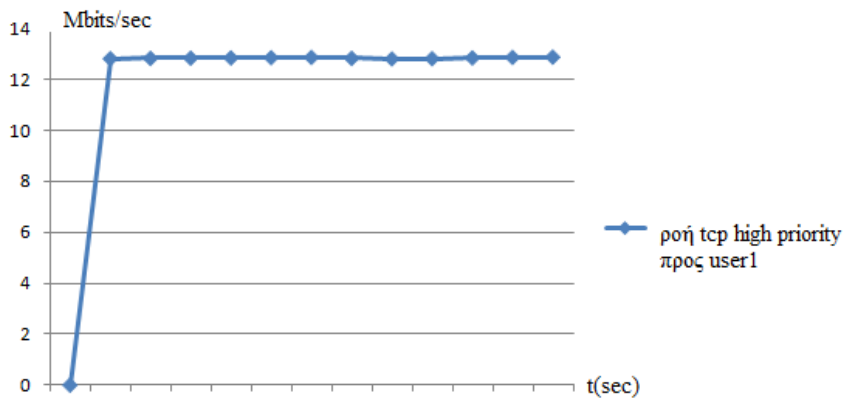
Χρησιμοποιήσαμε το netperf για να βρούμε το bitrate των ροών που χρησιμοποιηθήκαν στο παραπάνω σενάριο . Οι πίνακες των μετρήσεων βρίσκονται στο Appendix. Παρακάτω ακολουθούν οι γραφικές παραστάσεις των μετρήσεων.



Εικόνα 29 Γραφική παράσταση των τιμών του Netperf User2 με την low priority TCP ροή

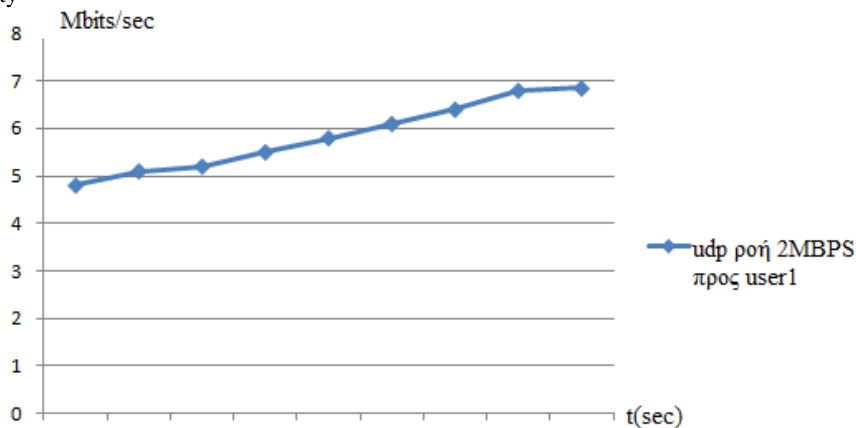
Στην παραπάνω γραφική παράσταση βλέπουμε το Bitrate της low priority tcp ροής που για t=0 sec ξεκινάει από τον Server και πάει στον user2. Το bit rate της ροής μένει αμετάβλητο μέχρι τέλους.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris



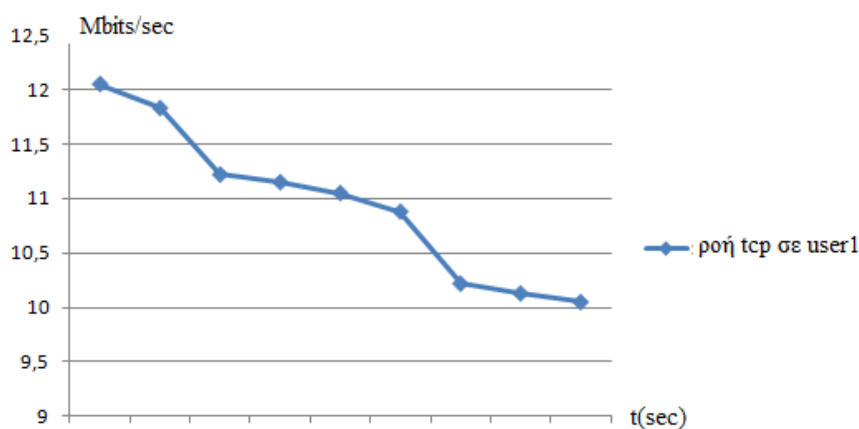
Εικόνα 30 Γραφική παράσταση των τιμών του Netperf User1 με την high priority tcp ροη

Στην παραπάνω γραφική παράσταση βλέπουμε το Bit rate της high priority tcp ροής που για $t=60$ sec ξεκίνα από τον Server και πάει στον user1. Το bit rate της ροής είναι μεγαλύτερο από την low priority που πάει στον User2 και αυτό γιατί έχουμε δηλώσει την συγκεκριμένη ροη σαν high priority



Εικόνα 31Γραφική παράσταση των τιμών του Netperf User1 με την 2Mbps UDP ροη

Στην παραπάνω γραφική παράσταση βλέπουμε το Bitrate της 2Mbps UDP ροής που για $t=120$ sec ξεκίνα από τον Server και παιει στον user1. Το bitrate της ροής αυξάνεται καθώς παράλληλα το bit rate της udp ροής που υπάρχει στο link του user1 μειώνεται.



Εικόνα 32 Γραφική παράσταση των τιμών του Netperf TCP ροής προς User1 καθώς εισέρχεται μια UDP 2 Mrbs ροη στο ίδιο κανάλι

Στην παραπάνω γραφική παράσταση βλέπουμε το Bitrate της TCP ροής που υπάρχει ήδη στο κανάλι του User 1 όταν εισέρχεται η UDP ροη

6.2.3 Παρατηρήσεις σεναρίου

Από την παραπάνω υλοποίηση διαπιστώσαμε ότι μπορεί να καθοριστεί ο τρόπος μετάδοσης των ροών στο δίκτυο ανάλογα με την σπουδαιότητα που έχει η κάθε ροη ως προς το να φτάσει όσο το δυνατόν πιο ακέραιη (πχ Web με χρήση TCP ή video streaming, με χρήση UDP) στους users ακόμα και σε ένα εικονικό δίκτυο. Αυτό δίνει την δυνατότητα σε μια εικονική δικτυακή τοπολογία να μπορούν να καθοριστούν όλες τις ρυθμίσεις που χρειάζονται ώστε να υπάρχει μεγαλύτερη ποιότητα παροχής της πολυμεσικής υπηρεσίας προς τους χρηστές πράγμα το οποίο μπορεί να συγκριθεί εξίσου σε ένα υπαρκτό δίκτυο με τις ίδιες ρυθμίσεις.

6.3 Συμπεράσματα

Η πτυχιακή εργασία επικεντρώθηκε στην κάλυψη των περισσότερων, κατά το δυνατόν, θεμάτων που αφορούν την υλοποίηση ιδεατών δικτυακών υποδομών με την χρήση του OpenSolaris για την παροχή οπτικοακουστικού υλικού με ποιότητα υπηρεσίας. Τα προτεινόμενα μοντέλα δικτύου βασίστηκαν στη φιλοσοφία των τεχνολογιών ευρείας εκπομπής και προσωπικών επικοινωνιών με ποιότητα υπηρεσίας. Με αυτή τη λογική υλοποιήθηκαν κάποιες ιδεατές δικτυακές υποδομές με χρήση του OpenSolaris, όπου δοκιμάστηκαν διάφορα ιδεατά δίκτυα πριν την ολοκλήρωση των δύο τελικών σεναρίων όπου εκεί επιτύχαμε να δημιουργήσουμε δύο ιδεατές δικτυακές αρχιτεκτονικές για παροχή πολυμεσικών υπηρεσιών με ποιότητα υπηρεσίας, μέσω της χρήσης του network virtualization. Στο πρώτο σενάριο υλοποιήσαμε ένα ιδεατό δίκτυο με δυναμική διαχείριση εύρους ζώνης και στο δεύτερο σενάριο υλοποιήσαμε ένα ιδεατό δίκτυο με προσομοίωση επίγειας ψηφιακής τηλεόρασης. Η πτυχιακή εργασία επικεντρώθηκε στις δυνατότητες που μας δίνει το network virtualization με το να μπορεί ένα χρήστης να ιδεατοποιήσει ολόκληρη δικτυακή τοπολογία χωρίς να χρειάζεται να επενδύσει περαιτέρω σε δικτυακό λογισμικό.

Όσον αφορά τα αποτελέσματα των μετρήσεων, κρίνονται ικανοποιητικά με βάση την αρχιτεκτονική και τη λειτουργία της ιδεατής δικτυακής υποδομής με τη συμμετοχή ενός server, δύο users και ενός etherstub. Κάνοντας σωστή διαχείριση των δικτυακών πόρων καταφέραμε να επέμβουμε στα όρια χρήσης εύρους ζώνης στο κάθε link και να δώσουμε προτεραιότητες ανά κίνηση στη κάθε ροή που χρησιμοποιήσαμε. Έτσι καταφέραμε να δείξουμε την βελτίωση των επιδόσεων των υπηρεσιών που παρέχονται.

ΠΑΡΑΡΤΗΜΑ

Κώδικας για παράδειγμα 2 ενότητα 3.3

```
~# zonecfg -z zweb1
zweb1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zweb1> create
zonecfg:zweb1> set zonepath=/vnm/zweb1
zonecfg:zweb1> set ip-type=exclusive
zonecfg:zweb1> add inherit-pkg-dir
zonecfg:zweb1:inherit-pkg-dir> set dir=/opt
zonecfg:zweb1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:zweb1:inherit-pkg-dir> end
zonecfg:zweb1> add net
zonecfg:zweb1:net> set physical=vphys1
zonecfg:zweb1:net> end
zonecfg:zweb1> add net
zonecfg:zweb1:net> set physical=vweb1
zonecfg:zweb1:net> end
zonecfg:zweb1> verify
zonecfg:zweb1> commit
zonecfg:zweb1> exit
giannis@opensolaris:~# zoneadm -z zweb1 clone vnmbase
sys-unconfig started 21 Φεβρουαρίου 2011 10:29:50
rm: /vnm/zweb1/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/zweb1/root/etc/dumpadm.conf
sys-unconfig completed Mon Feb 21 10:29:51 2011
giannis@opensolaris:~# zoneadm -z zweb1 boot
~# zlogin -C zweb1
System identification is completed.

zweb1 console login: root
Password:
Feb 21 00:36:11 zweb1 login: ROOT LOGIN /dev/console
Sun Microsystems Inc. SunOS 5.11 snv_111b November 2008
root@zweb1:~# route -p add net 192.168.7.0/24 192.168.0.254
add net 192.168.7.0/24: gateway 192.168.0.254
add persistent net 192.168.7.0/24: gateway 192.168.0.254
root@zweb1:~# ping 192.168.7.254
192.168.7.254 is alive
root@zweb1:~# ping 192.168.7.1
192.168.7.1 is alive

~$ su
Password:
# zonecfg -z zweb2
zweb2: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zweb2> create
zonecfg:zweb2> set zonepath=/vnm/zweb2
zonecfg:zweb2> set ip-type=exclusive
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:zweb2> add inherit-pkg-dir
zonecfg:zweb2:inherit-pkg-dir> set dir=/opt
zonecfg:zweb2:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:zweb2:inherit-pkg-dir> end
zonecfg:zweb2> add net
zonecfg:zweb2:net> set physical=vweb2
zonecfg:zweb2:net> end
zonecfg:zweb2> add net
zonecfg:zweb2:net> set physical=vphys2
zonecfg:zweb2:net> end
zonecfg:zweb2> verify
zonecfg:zweb2> commit
zonecfg:zweb2> exit
giannis@opensolaris:~# zoneadm -z zweb2 clone vnmbase
sys-unconfig started 21 Φεβρουαρίου 2011 10:38:42
rm: /vnm/zweb2/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/zweb2/root/etc/dumpadm.conf
sys-unconfig completed Mon Feb 21 10:38:42 2011
giannis@opensolaris:~# zoneadm -z zweb2 boot
giannis@opensolaris:~# zlogin -C zweb2
[Connected to zone 'zweb2' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

What type of terminal are you using?

- 1) ANSI Standard CRT
- 2) DEC VT100
- 3) PC Console
- 4) Sun Command Tool
- 5) Sun Workstation
- 6) X Terminal Emulator (xterms)
- 7) Other

Type the number of your choice and press Return: 3

Creating new rsa public/private host key pair

Creating new dsa public/private host key pair

Configuring network interface addresses: vphys2

System identification is completed.

zweb2 console login: root

Password:

Feb 21 00:43:42 zweb2 login: ROOT LOGIN /dev/console

Sun Microsystems Inc. SunOS 5.11 snv_111b November 2008

root@zweb2:~# route -p add net 192.168.7.0/24 192.168.0.254

add net 192.168.7.0/24: gateway 192.168.0.254

add persistent net 192.168.7.0/24: gateway 192.168.0

```
giannis@opensolaris:~# zonecfg -z vdb1
```

```
vdb1: No such zone configured
```

Use 'create' to begin configuring a new zone.

```
zonecfg:vdb1> create
```

```
zonecfg:vdb1> set zonepath=/vnm/vdb1
```

```
zonecfg:vdb1> set ip-type=exclusive
```

```
zonecfg:vdb1> add inherit-pkg-dir
```

```
zonecfg:vdb1:inherit-pkg-dir> set dir=/opt
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:vdb1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vdb1:inherit-pkg-dir> end
zonecfg:vdb1> add net
zonecfg:vdb1:net> delete
Are you sure you want to delete zone vdb1 (y/[n])? y
vdb1: No such zone configured
zonecfg:vdb1:net> exit
Resource incomplete; really quit (y/[n])? y
giannis@opensolaris:~# zonecfg -z zdb1
zdb1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zdb1> create
zonecfg:zdb1> set zonepath=/vnm/zdb1
zonecfg:zdb1> set ip-type=exclusive
zonecfg:zdb1> add inherit-pkg-dir
zonecfg:zdb1:inherit-pkg-dir> set dir=/opt
zonecfg:zdb1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:zdb1:inherit-pkg-dir> end
zonecfg:zdb1> add net
zonecfg:zdb1:net> set physical=vdb1
zonecfg:zdb1:net> end
zonecfg:zdb1> verify
zonecfg:zdb1> commit
zonecfg:zdb1> exit
giannis@opensolaris:~# zoneadm -z zdb1 clone vnmbase
sys-unconfig started 21 Φεβρουαρίου 2011 11:00:12
rm: /vnm/zdb1/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/zdb1/root/etc/dumpadm.conf
sys-unconfig completed Mon Feb 21 11:00:12 2011
giannis@opensolaris:~# zoneadm -z zdb1 boot
giannis@opensolaris:~# zlogin -C zdb1
[Connected to zone 'zdb1' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
What type of terminal are you using?
1) ANSI Standard CRT
2) DEC VT100
3) PC Console
4) Sun Command Tool
5) Sun Workstation
6) X Terminal Emulator (xterms)
7) Other
Type the number of your choice and press Return: 3
Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
Configuring network interface addresses: vdb1
System identification is completed.

opensolaris:~$ su
Password:
giannis@opensolaris:~# zonecfg -z zdb2
zdb2: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zdb2> create
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:zdb2> set zonepath=/vnm/zdb2
zonecfg:zdb2> set ip-type=exclusive
zonecfg:zdb2> add inherit-pkg-dir
zonecfg:zdb2:inherit-pkg-dir> set dir=/opt
zonecfg:zdb2:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:zdb2:inherit-pkg-dir> end
zonecfg:zdb2> add net
zonecfg:zdb2:net> set physical=vdb2
zonecfg:zdb2:net> end
zonecfg:zdb2> verify
zonecfg:zdb2> commit
zonecfg:zdb2> exit
giannis@opensolaris:~# zoneadm -z zdb2 clone vnmbase
sys-unconfig started 21 Φεβρουαρίου 2011 11:04:33
rm: /vnm/zdb2/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/zdb2/root/etc/dumpadm.conf
sys-unconfig completed Mon Feb 21 11:04:33 2011
giannis@opensolaris:~# zoneadm -z zdb2 boot
giannis@opensolaris:~# zlogin -C zdb2
[Connected to zone 'zdb2' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

```
giannis@opensolaris:~# zonecfg -z zdb3
zdb3: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zdb3> create
zonecfg:zdb3> set zonepath=/vnm/zdb3
zonecfg:zdb3> set ip-type=exclusive
zonecfg:zdb3> add inherit-pkg-dir
zonecfg:zdb3:inherit-pkg-dir> set dir=/opt
zonecfg:zdb3:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:zdb3:inherit-pkg-dir> end
zonecfg:zdb3> add net
zonecfg:zdb3:net> set physical=vdb3
zonecfg:zdb3:net> end
zonecfg:zdb3> verify
zonecfg:zdb3> commit
zonecfg:zdb3> exit
giannis@opensolaris:~# zoneadm -z zdb3 clone vnmbase
sys-unconfig started 21 Φεβρουαρίου 2011 11:09:36
rm: /vnm/zdb3/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/zdb3/root/etc/dumpadm.conf
sys-unconfig completed Mon Feb 21 11:09:37 2011
giannis@opensolaris:~# zoneadm -z zdb3 boot
giannis@opensolaris:~# zlogin -C zdb3
[Connected to zone 'zdb3' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

System identification is completed.

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
opensolaris:~$ su
Password:
giannis@opensolaris:~# zonecfg -z zrouter
zweb2: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zrouter> create
zonecfg:zrouter> set zonepath=/vnm/zrouter
zonecfg:zrouter> set ip-type=exclusive
zonecfg:zrouter> add inherit-pkg-dir
zonecfg:zrouter:inherit-pkg-dir> set dir=/opt
zonecfg:zrouter:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:zrouter:inherit-pkg-dir> end
zonecfg:zrouter> add net
zonecfg:zrouter:net> set physical=vweb0
zonecfg:zrouter:net> end
zonecfg:zrouter> add net
zonecfg:zrouter:net> set physical=vdb0
zonecfg:zrouter:net> end
zonecfg:zrouter> verify
zonecfg:zrouter> commit
zonecfg:zrouter> exit
giannis@opensolaris:~# zoneadm -z zrouter clone vnmbase
sys-unconfig started 21 Φεβρουαρίου 2011 10:38:42
rm: /vnm/zweb2/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/zweb2/root/etc/dumpadm.conf
sys-unconfig completed Mon Feb 21 10:38:42 2011
giannis@opensolaris:~# zoneadm -z zrouter boot
giannis@opensolaris:~# zlogin -C zrouter
[Connected to zone 'zweb2' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

Κώδικας για παράδειγμα project ενότητα 5.1

```
gz# zonecfg -z host1
host2: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vnmbase> create
zonecfg:vnmbase> set zonepath=/vnm/host1
zonecfg:vnmbase> set ip-type=exclusive
zonecfg:vnmbase> add inherit-pkg-dir
zonecfg:vnmbase:inherit-pkg-dir> set dir=/opt
zonecfg:vnmbase:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vnmbase:inherit-pkg-dir> end
zonecfg:vnmbase> add net
zonecfg:vnmbase:net> set physical=vnic1
zonecfg:vnmbase:net> end
zonecfg:vnmbase> verify
zonecfg:vnmbase> commit
zonecfg:vnmbase> exit
```

```
gz# zoneadm -z host1 clone vnmbase
gz# zoneadm -z host1 boot
gz# zlogin -C host1
  gz# zonecfg -z host2
host2: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vnmbase> create
zonecfg:vnmbase> set zonepath=/vnm/host2
zonecfg:vnmbase> set ip-type=exclusive
zonecfg:vnmbase> add inherit-pkg-dir
zonecfg:vnmbase:inherit-pkg-dir> set dir=/opt
zonecfg:vnmbase:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vnmbase:inherit-pkg-dir> end
zonecfg:vnmbase> add net
zonecfg:vnmbase:net> set physical=vnic2
zonecfg:vnmbase:net> end
zonecfg:vnmbase> verify
zonecfg:vnmbase> commit
zonecfg:vnmbase> exit
  gz# zoneadm -z host2 clone vnmbase
gz# zoneadm -z host2 boot
gz# zlogin -C host2
```

```
gz# zonecfg -z vRouter
vRouter: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vnmbase> create
zonecfg:vnmbase> set zonepath=/vnm/vRouter
zonecfg:vnmbase> set ip-type=exclusive
zonecfg:vnmbase> add inherit-pkg-dir
zonecfg:vnmbase:inherit-pkg-dir> set dir=/opt
zonecfg:vnmbase:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vnmbase:inherit-pkg-dir> end
zonecfg:vnmbase> add net
zonecfg:vnmbase:net> set physical=vnic3
zonecfg:vnmbase:net> end
zonecfg:vnmbase> add net
zonecfg:vnmbase:net> set physical=vnic9
zonecfg:vnmbase:net> end
zonecfg:vnmbase> verify
zonecfg:vnmbase> commit
zonecfg:vnmbase> exit
  gz# zoneadm -z vRouter clone vnmbase
gz# zoneadm -z vRouter boot
gz# zlogin -C vRouter
```

Κώδικας για παράδειγμα project ενότητα 5.2

```
~# zonecfg -z server

server: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:server> create
zonecfg:server> set zonepath=/vnm/server
zonecfg:server> set ip-type=exclusive
zonecfg:server> add inherit-pkg-dir
zonecfg:server:inherit-pkg-dir> set dir=/opt
zonecfg:server:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:server:inherit-pkg-dir> end
zonecfg:server> add net
zonecfg:server:net> set physical=vnic1
zonecfg:server:net> end
zonecfg:server> add net
zonecfg:server:net> set physical=vnic2
zonecfg:server:net> end
zonecfg:server> verify
zonecfg:server> commit
zonecfg:server> exit
giannis@opensolaris:~# zoneadm -z server clone vnmbase
sys-unconfig started 07 Μαρτίου 2011 12:06:59 MM EE
rm: /vnm/server/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/server/root/etc/dumpadm.conf
sys-nconfig completed Mon Mar 07 12:07:00 2011
~# zoneadm -z server boot
# zlogin -C server
[Connected to zone 'server' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
What type of terminal are you using?
 1) ANSI Standard CRT
 2) DEC VT100
 3) PC Console
 4) Sun Command Tool
 5) Sun Workstation
 6) X Terminal Emulator (xterms)
 7) Other
Type the number of your choice and press Return: 3
Configuring network interface addresses: vnic1
vnic2
giannis@opensolaris:~# zonecfg -z vRouter
vRouter: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vRouter> create
zonecfg:vRouter> set zonepath=/vnm/vRouter
zonecfg:vRouter> set ip-type=exclusive
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:vRouter> add inherit-pkg-dir
zonecfg:vRouter:inherit-pkg-dir> set dir=/opt
zonecfg:vRouter:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:vRouter:inherit-pkg-dir> end
zonecfg:vRouter> add net
zonecfg:vRouter:net> set physical=vnic3
zonecfg:vRouter:net> end
zonecfg:vRouter> add net
zonecfg:vRouter:net> set physical=vnic4
zonecfg:vRouter:net> end
zonecfg:vRouter> verify
zonecfg:vRouter> commit
zonecfg:vRouter> exit
giannis@opensolaris:~# zoneadm -z vRouter clone vnmbase
sys-unconfig started 07 Μαρτίου 2011 12:13:14 MM EE
rm: /vnm/vRouter/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/vRouter/root/etc/dumpadm.conf
sys-unconfig completed Mon Mar 07 12:13:14 2011
giannis@opensolaris:~# zoneadm -z vRouter boot
giannis@opensolaris:~# zlogin -C vRouter
[Connected to zone 'vRouter' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
What type of terminal are you using?
1) ANSI Standard CRT
2) DEC VT100
3) PC Console
4) Sun Command Tool
5) Sun Workstation
6) X Terminal Emulator (xterms)
7) Other
Type the number of your choice and press Return: 3
Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
Configuring network interface addresses: vnic3
vnic4
```

```
giannis@opensolaris:~# zonecfg -z user1
user1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:user1> create
zonecfg:user1> set zonepath=/vnm/user1
zonecfg:user1> set ip-type=exclusive
zonecfg:user1> add inherit-pkg-dir
zonecfg:user1:inherit-pkg-dir> set dir=/opt
zonecfg:user1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:user1:inherit-pkg-dir> end
```


Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:user1> add net
zonecfg:user1:net> set physical=vnic5
zonecfg:user1:net> end
zonecfg:user1> add net
zonecfg:user1:net> set physical=vnic6
zonecfg:user1:net> end
zonecfg:user1> verify
zonecfg:user1> commit
zonecfg:user1> exit
giannis@opensolaris:~# zoneadm -z vRouter clone vnmbase
zoneadm: zone 'vRouter': clone operation is invalid for running zones.
giannis@opensolaris:~# zoneadm -z user1 clone vnmbase
sys-unconfig started 07 Μαρτίου 2011 12:20:25 MM EE
rm: /vnm/user1/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/user1/root/etc/dumpadm.conf
sys-unconfig completed Mon Mar 07 12:20:26 2011
giannis@opensolaris:~# zoneadm -z user1 boot
giannis@opensolaris:~# zlogin -C user1
[Connected to zone 'user1' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

What type of terminal are you using?

- 1) ANSI Standard CRT
- 2) DEC VT100
- 3) PC Console
- 4) Sun Command Tool
- 5) Sun Workstation
- 6) X Terminal Emulator (xterms)
- 7) Other

Type the number of your choice and press Return: 3

```
Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
Configuring network interface addresses: vnic5
vnic6
```

```
giannis@opensolaris:~# zonecfg -z user2
user1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:user1> create
zonecfg:user1> set zonepath=/vnm/user2
zonecfg:user1> set ip-type=exclusive
zonecfg:user1> add inherit-pkg-dir
zonecfg:user1:inherit-pkg-dir> set dir=/opt
zonecfg:user1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:user1:inherit-pkg-dir> end
zonecfg:user1> add net
zonecfg:user1:net> set physical=vnic7
zonecfg:user1:net> end
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:user1> add net
zonecfg:user1:net> set physical=vnic8
zonecfg:user1:net> end
zonecfg:user1> verify
zonecfg:user1> commit
zonecfg:user1> exit
giannis@opensolaris:~# zoneadm -z user2 clone vnmbase
zoneadm: zone 'user2': clone operation is invalid for running zones.
giannis@opensolaris:~# zoneadm -z user2 clone vnmbase
sys-unconfig started 07 Μαρτίου 2011 12:20:25 MM EE
rm: /vnm/user2/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/user2/root/etc/dumpadm.conf
sys-unconfig completed Mon Mar 07 12:20:26 2011
giannis@opensolaris:~# zoneadm -z user2 boot
giannis@opensolaris:~# zlogin -C user2
[Connected to zone 'user1' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
What type of terminal are you using?
 1) ANSI Standard CRT
 2) DEC VT100
 3) PC Console
 4) Sun Command Tool
 5) Sun Workstation
 6) X Terminal Emulator (xterms)
 7) Other
Type the number of your choice and press Return: 3
Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
Configuring network interface addresses: vnic7 vnic8
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Κωδικας για 1^ο σενάριο 6.1

```
giannis@opensolaris:~# dladm create-etherstub vswitch1
giannis@opensolaris:~# dladm show-etherstub
LINK
vswitch1
giannis@opensolaris:~# dladm create-vnic -l vswitch1 vnic1
giannis@opensolaris:~# dladm create-vnic -l vswitch1 vnic2
giannis@opensolaris:~# dladm create-vnic -l vswitch1 vnic3
giannis@opensolaris:~# dladm show-vnic
LINK      OVER      SPEED  MACADDRESS      MACADDRTYPE      VID
vnic1     vswitch1  0      2:8:20:e0:84:e2  random           0
vnic2     vswitch1  0      2:8:20:7b:10:f5  random           0
vnic3     vswitch1  0      2:8:20:5b:c8:9e  random           0
giannis@opensolaris:~# zonecfg -z server
server: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:server> create
zonecfg:server> set zonepath=/vnm/server
zonecfg:server> set ip-type=exclusive
zonecfg:server> add inherit-pkg-dir
zonecfg:server:inherit-pkg-dir> set dir=/opt
zonecfg:server:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:server:inherit-pkg-dir> end
zonecfg:server> add net
zonecfg:server:net> set physical=vnic3
zonecfg:server:net> end
zonecfg:server> verify
zonecfg:server> commit
zonecfg:server> exit
giannis@opensolaris:~# zoneadm -z server clone vnmbase
sys-unconfig started 30 Μαρτίου 2011  5:42:26 MM EE
rm: /vnm/server/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/server/root/etc/dumpadm.conf
sys-unconfig completed Wed Mar 30 17:42:26 2011
giannis@opensolaris:~# zoneadm -z server boot
giannis@opensolaris:~# zlogin -C server
[Connected to zone 'server' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

```
giannis@opensolaris:~# zonecfg -z user1
user1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:user1> create
zonecfg:user1> set zonepath=/vnm/user1
zonecfg:user1> set ip-type=exclusive
zonecfg:user1> add inherit-pkg-dir
zonecfg:user1:inherit-pkg-dir> set dir=/opt
zonecfg:user1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:user1:inherit-pkg-dir> end
zonecfg:user1> add net
zonecfg:user1:net> set physical=vnic1
zonecfg:user1:net> end
zonecfg:user1> verify
```

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

```
zonecfg:user1> commit
zonecfg:user1> exit
giannis@opensolaris:~# zoneadm -z user1 clone vnmbase
sys-unconfig started 30 Μαρτίου 2011 5:46:49 MM EE
rm: /vnm/user1/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/user1/root/etc/dumpadm.conf
sys-unconfig completed Wed Mar 30 17:46:49 2011
giannis@opensolaris:~# zoneadm -z user1 boot
giannis@opensolaris:~# zlogin -C user1
[Connected to zone 'user1' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
giannis@opensolaris:~$ su
Password:
giannis@opensolaris:~# zonecfg -z user2
user1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:user1> create
zonecfg:user1> set zonepath=/vnm/user2
zonecfg:user1> set ip-type=exclusive
zonecfg:user1> add inherit-pkg-dir
zonecfg:user1:inherit-pkg-dir> set dir=/opt
zonecfg:user1:inherit-pkg-dir> set dir=/etc/crypto
zonecfg:user1:inherit-pkg-dir> end
zonecfg:user1> add net
zonecfg:user1:net> set physical=vnic2
zonecfg:user1:net> end
zonecfg:user1> verify
zonecfg:user1> commit
zonecfg:user1> exit
giannis@opensolaris:~# zoneadm -z user2 clone vnmbase
sys-unconfig started 30 Μαρτίου 2011 5:46:49 MM EE
rm: /vnm/user1/root/etc/vfstab.sys-u: No such file or directory
grep: can't open /vnm/user1/root/etc/dumpadm.conf
sys-unconfig completed Wed Mar 30 17:46:49 2011
giannis@opensolaris:~# zoneadm -z user2 boot
giannis@opensolaris:~# zlogin -C user2
[Connected to zone 'user2' console]
69/69
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
```

7.1 Πινάκες ροών για τα σενάρια της ενότητας 6

ΠΙΝΑΚΕΣ ΡΟΩΝ ΓΙΑ ΤΑ ΤΟ ΠΡΩΤΟ ΣΕΝΑΡΙΟ

Ροή προς user1	udp	
Mbits/sec		T(sec)
0		0
6,55		5
6,54		10
6,53		15
6,49		20
6,49		25
6,54		30
3,26		35
3,28		40
3,27		45
3,25		50
3,23		55
3,24		60
6,54		65
6,55		70
6,53		75

Mbits/sec	t(sec)
3,26	30
3,28	35
3,27	40
3,25	45
3,23	50
3,24	55
6,54	60
6,55	65
6,53	70

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Ροή προς user2 udp	
Mbits/sec	T(sec)
0	0
6,55	5
6,54	10
6,53	15
6,49	20
6,49	25
6,54	30
6,56	35
5,58	40
6,57	45
6,55	50
6,53	55
6,56	60
6,54	65
6,55	70
6,53	75

ΠΙΝΑΚΕΣ ΡΟΩΝ ΓΙΑ ΤΑ ΤΟ ΔΕΥΤΕΡΟ ΣΕΝΑΡΙΟ

Ροή tcp low priority προς user2	
Mbits/sec	t(sec)
0	0
7,82	5
7,82	10
7,83	15
7,85	20
7,86	25
7,87	30
7,83	35
7,84	40
7,86	45
7,87	50
7,85	55
7,86	60
7,83	65
7,84	70
7,86	75
7,83	80
7,88	85
7,85	90
7,86	95
7,82	100
7,83	105
7,84	110
7,85	115
7,87	120
7,88	125
7,83	130
7,82	135
7,85	140
7,88	145
7,86	150
7,87	155
7,85	160

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Ροή tcp high priority προς user1 Mbits/sec	t(sec)
12,83	60
12,84	65
12,85	70
12,86	75
12,87	80
12,88	85
12,84	90
12,82	95
12,83	100
12,86	105
12,88	110
12,89	115
12,05	120
11,83	125
11,22	130
11,15	135
11,05	140
10,88	145
10,22	150
10,13	155
10,05	160

Σχεδιασμός και υλοποίηση ιδεατής δικτυακής αρχιτεκτονικής με χρήση OpenSolaris

Ροή udp 2MBPS προς user 1	
Mbits/sec	t(sec)
4,8	120
5,1	125
5,2	130
5,5	135
5,8	140
6,1	145
6,4	150
6,8	155
6,85	160

7.2 Αναφορές

- [1]http://en.wikipedia.org/wiki/Application_virtualization
- [2]<http://eucalyptus.cs.ucsb.edu/>
- [3]<http://kvm.qumranet.com/>
- [4]<http://openvz.org/>
- [5]<http://www.openxvm.org/>
- [6]<http://x2go.berlios.de/index-en.html>
- [7]<http://www.xen.org/>
- [8]<http://thanos.terentes.com/el/oracle/2010/12/oracle-vm-virtualbox-4-0-now-available/>
- [9]http://www.opensolaris.com/use/crossbow_network_containers.pdf.
- [10]<http://wikis.sun.com/display/OpenSolaris/vWireExample>
- [11]www.opensolaris.com/networking
- [12]www.opensolaris.org/os/project/crossbow
- [13] <http://www.cl.cam.ac.uk/research/srg/netos/papers/2003-xensosp.Pdf>
- [14]Wiley.OpenSolaris.Bible.Feb.2009
- [15]Peter Baer Galvin **Pete** 's All**ThingsSun: Crossbow**
- [16] Xen and the art of virtualization, Paul Barham et al., 2003
- [17] Arthur Callado, Carlos Kamienski Member, IEEE, Géza Szabó, Balázs Péter Gerő, Judith Kelner,Stênio Fernandes Member, IEEE, and Djamel Sadok, Senior Member, IEEE, „Title“, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 11, NO. 3, THIRD QUARTER 2009
- [18]Jeff McMeekin, OpenSolaris Networking Product Manager Version 1.0 | Last updated: 09/29/09
- [19]Sunay Tripathi, Distinguished Engineer Solaris Core Technology group Nov 6th, 2009 Crossbow: Network Virtualization and Resource Control
- [20]Nicolas Droux Solaris Core Technologies June 2007 Crossbow: Network Virtualization and Bandwidth Partitioning Nicolas Droux
- [21]IP Instances – Network solution meets Zones Erik Nordmark Distinguished Engineer Solaris Networking October 2006
- [22]VLAN as VNICs: LAN Virtualization meets NIC Virtualization Shrikrishna Khare summer 2007 Intern, Crossbow Mentor: Nicolas Droux Manager: Markus Flierl
- [23]IP Instances Interface Document PSARC 2006/366 2006-12-28 Erik Nordmark Yukun Zhang Dong-Hai Han
- [24] Greg Shields, MVP, vExpert , Jeff McMeekin,OpenSolaris Networking Product Manager CTT Virtualization Keynote, Head Geek, Concentrated Technology. On-line: <http://www.ConcentratedTech.com>