



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Εικονικά συστήματα XEN



Του φοιτητή
Αϊβαλιώτη Αναστάσιου
Αρ. Μητρώου: 04/2504

Επιβλέπων Καθηγητής
κ. Ψαρράς Νικόλαος

Θεσσαλονίκη 2014

Περιεχόμενα

[Πρόλογος](#)

[Περίληψη](#)

[Abstract](#)

- 1 [Εικονικοποίηση \(Virtualization\)](#)
 - 1.1 [Τι είναι?](#)
 - 1.2 [Ιστορία](#)
- 2 [Γιατί είναι σημαντική](#)
 - 2.1 [Περιοχές που έχουν οφέλη από την εικονικοποίηση](#)
 - 2.2 [Κλιμάκωση - Υποψήφιοι για εικονικοποίηση](#)
 - 2.3 [Κοινές χρήσεις και οφέλη σήμερα](#)
- 3 [Τύποι εικονικοποίησης](#)
 - 3.1 [Πλήρης εικονικοποίηση](#)
 - 3.1.2 [Επίπεδα προστασίας επεξεργαστή](#)
 - 3.1.3 [Επόπτης \(Hypervisor\)](#)
 - 3.1.3.3 [Τύπος 1 Hypervisor](#)
 - 3.1.3.4 [Τύπος 2 Hypervisor](#)
 - 3.1.3.5 [Embedded Hypervisor](#)
 - 3.2 [Παραεικονικοποίηση](#)
 - 3.3 [Εικονικοποίηση επιπέδου Λειτουργικού συστήματος](#)
 - 3.4 [Εικονικοποίηση Επιπέδου Εφαρμογών](#)
- 4 [Xen Hypervisor](#)
 - 4.1 [Τι είναι το Xen](#)
 - 4.2 [Που χρησιμοποιείται](#)
 - 4.3 [Γιατί να μη διαλέξω... ?](#)
 - 4.4 [Ιστορία](#)
- 5 [Αρχιτεκτονική του Xen](#)
 - 5.1 [Domain 0 and DomU](#)
 - 5.2 [Πως λειτουργεί το Xen](#)
- 6 [Οικοδεσπότες και φιλοξενούμενοι - Host and Guests](#)
 - 6.1 [Host](#)
 - 6.2 [Φιλοξενούμενοι](#)
- 7 [Τι υλικό να διαλέξεις?](#)
- 8 [Πριν την εγκατάσταση](#)
- 9 [Εγχειρίδιο Χρήσης του Xen](#)
 - 9.1 [Εγκατάσταση Debian](#)
 - 9.2 [Εγκατάσταση του Xen](#)
 - 9.3 [Grub2 – Grand Unified Bootloader](#)
 - 9.4 [Παραμετροποίηση δικτύου](#)
 - 9.5 [Απομακρυσμένη πρόσβαση](#)

- 9.6 [Xen-tools](#)
 - 9.7 [Δημιουργία Παρά-εικονικού φιλοξενούμενου](#)
 - 9.8 [Χειροκίνητη δημιουργία Παραεικονικών Φιλοξενούμενων](#)
 - 9.9 [HVM guests](#)
 - 9.10 [Διαχειρίζοντας το Xen](#)
 - 9.10.2 [Αυτόματο ξεκίνημα των DomUs](#)
 - 9.10.3 [Backup!](#)
 - 9.10.4 [Επανεκκίνηση κωδικού](#)
 - 9.10.5 [Εκχώρηση περισσότερης υπολογιστικής δύναμης](#)
 - 9.10.6 [Σύνδεση πυρήνων επεξεργαστή σε συγκεκριμένο domain](#)
 - 9.10.7 [Διαχείριση μνήμης Ram στο Dom0](#)
 - 9.10.8 [Διαχείριση μνήμης Ram στα DomUs](#)
 - 9.11 [Πέρασμα συσκευών στα DomUs \(PCI Passthrough\)](#)
- 10 [Συμπεράσματα](#)

Πρόλογος

Η παρούσα εργασία παρουσιάζει τα οφέλη της εικονικοποίησης καθώς και δίνει στον αναγνώστη το ερέθισμα για να ασχοληθεί ενεργά με αυτήν την τεχνολογία. Μια τεχνολογία που υπάρχει γύρω μας για πολύ καιρό, αλλά μόλις τα τελευταία χρόνια έχει αρχίσει να χρησιμοποιείται ευρέως.

Διαβάζοντας αυτήν την εργασία ο αναγνώστης θα μάθει τι είναι η εικονικοποίηση, ποια είναι τα οφέλη της, θα είναι σε θέση να αποφασίσει αν τον συμφέρει η λύση της εικονικοποίησης, αν διαθέτει τα κατάλληλα μέσα για να πετύχει σωστά αποτελέσματα και τέλος θα έχει την κατάλληλη καθοδήγηση για να στήσει εικονικά μηχανήματα με το Xen Hypervisor.

Κατά τη διάρκεια της χρησιμοποίησης του Xen, ο χρήστης θα έρθει επίσης πιο κοντά στα λειτουργικά συστήματα Linux, στην περίπτωση μας το Debian.

Εικονικοποίηση, μια τεχνολογία που προσωπικά πιστεύω ότι είναι το μέλλον σε ότι αφορά την πληροφορική και είναι ο δρόμος που έχουν διαλέξει για να πορευτούν οι κολοσσοί της τεχνολογίας, όπως η Google, η Yahoo, η Amazon...

Περίληψη

Η πτυχιακή εργασία με θέμα «Εικονικά συστήματα Xen», έχει αρχικά σαν σκοπό τη γνωριμία με τον κόσμο της εικονικοποίησης. Ο αναγνώστης θα μάθει για την ιστορία της, για τις περιοχές που έχουν οφέλη από αυτήν, που χρησιμοποιείται κυρίως και ποια συστήματα είναι υποψήφια για εικονικοποίηση.

Στη συνέχεια θα αναλυθούν οι κυριότεροι τύποι εικονικοποίησης και θα αναφερθούν τα σημεία στα οποία υπερέχει αλλά και υστερεί η καθεμία. Οι κυριότεροι τύποι εικονικοποίησης, είναι η Πλήρης εικονικοποίηση, η οποία με τη σειρά της χωρίζεται σε Τύπου 1, Τύπου 2 και Ενσωματωμένη, η Παραεικονικοποίηση, η εικονικοποίηση επιπέδου Λειτουργικού Συστήματος και η εικονικοποίηση επιπέδου εφαρμογών.

Αφού ο αναγνώστης μάθει τι είναι η εικονικοποίηση και πως λειτουργεί γενικότερα, θα μάθει περισσότερα για τον επόπτη Xen. Το οποίο είναι ένα σύστημα εικονικοποίησης που υποστηρίζει και πλήρη και παρά-εικονικοποίηση. Θα γίνει αναφορά στην προέλευση του, ποιες εταιρίες βοήθησαν και βοηθούν στην ανάπτυξη του και στη συνέχεια θα αναλυθεί η αρχιτεκτονική του Xen και πως ακριβώς λειτουργεί.

Ακόμα υπάρχει ένας πλήρης οδηγός για το τι χρειάζεται κάποιος για να ξεκινήσει να χρησιμοποιεί το Xen, πώς να δημιουργήσει τα πρώτα παρά- αλλά και πλήρους εικονικοποίησης μηχανήματα, καθώς και ότι χρειάζεται κάποιος για τη σωστή διαχείριση των εικονικών μηχανημάτων - φιλοξενούμενοι μετά τη δημιουργία τους.

Abstract

This thesis with the name “Xen Virtual Systems”, has as its primary goal the acquaintance with the world of virtualization. The reader will learn the history, the areas that benefit most from this technology, where it is most used and which systems are candidates for virtualization.

Then the main types of virtualization will be analyzed and there is going to be pointed out where each of them is superior and where each lacks. The main types of virtualization are, hardware-assisted virtualization, which is divided in Type 1, Type 2 and Embedded, Paravirtualization, Operating system-level virtualization and Application virtualization.

After the reader understands what virtualization is and how it works in general, he will learn more about Xen Hypervisor. Xen is a virtualization system that supports both Hardware-assisted and Para- Virtualization. One will learn about its origin, which companies helped in its creation and development and subsequent is going to be analyzed how Xen functions and the architecture behind it.

Furthermore there is a full guide of what someone may need in order to begin using Xen, how to create the first virtual machines, both paravirtualized or hardware-assisted and whatever someone will need in order to manage the virtual machines after its creation

Εικονικοποίηση (Virtualization)

Τι είναι?

Ένας ευρύς ορισμός

Θα μπορούσαμε γενικώς να ορίσουμε την εικονικοποίηση ως:

«ένα πλαίσιο ή μια μεθοδολογία της διαίρεσης των πόρων ενός υπολογιστή σε πολλαπλά περιβάλλοντα εκτέλεσης, με την εφαρμογή μιας ή περισσοτέρων εννοιών ή τεχνολογιών, όπως είναι ο διαχωρισμός λογισμικού ή υλικού (*hardware and software partitioning*), ο διαμερισμός του χρόνου (*time-sharing*), μερική ή ολική προσομοίωση μηχανών (*machine simulation*), η εξομοίωση (*emulation*), ποιότητα των υπηρεσιών (*quality of service*) και πολλά άλλα.»

Εδώ πρέπει να σημειώσουμε ότι ο ορισμός είναι αρκετά γενικός και περιλαμβάνει έννοιες όπως ποιότητα υπηρεσιών που, παρόλο που είναι ένα ξεχωριστό πεδίο έρευνας αλλά συχνά χρησιμοποιείται παράλληλα με την εικονικοποίηση. Συχνά τέτοιες τεχνολογίες ενώνονται με περίπλοκους τρόπους ώστε να σχηματίσουν ενδιαφέροντα συστήματα, ένα από τα οποία είναι και η εικονικοποίηση.

Παρόλο που ορίσαμε την εικονικοποίηση με έναν ορισμό που υπονοεί το διαχωρισμό-διαμοιρασμό κάτι σε πολλαπλές έννοιες, δε σημαίνει πάντα αυτό. Για παράδειγμα μπορούμε να πούμε ότι, μπορείς να πάρεις ν δίσκους και να τους κάνεις να εμφανίζονται σαν ένας (λογικός - logical) δίσκος μέσω ενός στρώματος εικονικοποίησης.

Ιστορία

Ο Κρίστοφερ Στράσεϊ εξέδωσε μια εργασία με τίτλο Διαμοιρασμός Χρόνου σε Μεγάλους και Γρήγορους Υπολογιστές στο Παγκόσμιο συνέδριο Επεξεργασίας Πληροφοριών στην Ουνέσκο, Νέα Υόρκη, τον Ιούνιο του 1959. Αργότερα, το 1974, διευκρίνισε με ένα μήνυμα ηλεκτρονικού ταχυδρομείου στον Donald Knuthότι:

«...[η εργασία μου] ήταν κυρίως για τον πολύ-προγραμματισμό (για την αποφυγή αναμονής λόγω των περιφερειακών), παρόλο που αντιμετώπισα το ίδιο πρόβλημα σαν προγραμματιστής την ώρα που απασφαλμάτωνα το πρόγραμμα μου στην κονσόλα. Δεν οραματίστηκα ένα είδος κονσόλας που τώρα τόσο συγκεχυμένα ονομάζεται διαμοιρασμός χρόνου.»

"... [my paper] was mainly about multi-programming (to avoid waiting for peripherals) although it did envisage this going on at the same time as a programmer who was debugging his program at a console. I did not envisage the sort of console system which is now so confusingly called time sharing."

Η χρήση του πολύ-προγραμματισμού της ουράς μπορεί να αποδοθεί στον υπολογιστή Άτλας στις αρχές της δεκαετίας του 60. Το σχέδιο Άτλας ήταν μια κοινή προσπάθεια του πανεπιστημίου του Μάντσεστερ και της Φεράντι Ltd. Παράλληλα με τη διαχείριση ουράς, ο Άτλαντας πρωτοπόρησε στη ζήτηση σελιδοποίησης και στις κλήσεις του επόπτη (*demand paging and supervisor calls*).

Ένα εικονικό μηχάνημα χρησιμοποιήθηκε για τον επόπτη και ένα άλλο για να τρέχουν οι χρήστες προγράμματα.

Στα μέσα της δεκαετίας του 60, το Κέντρο Έρευνας του Watson της IBM ήταν το σπίτι του πρότζεκτ M44/44X. Ο στόχος ήταν να αξιολογήσει τα τότε αναδυόμενα συστήματα διαμοιρασμού χρόνου. Η αρχιτεκτονική του στηρίχτηκε σε εικονικά μηχανήματα. Το κύριο μηχάνημα ήταν ένα IBM 7044 (M44) και κάθε εικονικό μηχάνημα ήταν μία πειραματική εικόνα του κύριου μηχανήματος (44X). Ο χώρος διευθυνσιοδότησης ενός 44X κατοικούσε στην ιεραρχία της μνήμης του M44, εφαρμοσμένη μέσω εικονικής μνήμης και πολυπρογραμματισμού.

Η IBM παρείχε στον υπολογιστή IBM 704, μια σειρά από αναβαθμίσεις και επίσης έδωσε πρόσβαση, σε κάποια από τα συστήματα της, σε μηχανικούς του MIT. Ήταν σε υπολογιστές της IBM, όπου αναπτύχθηκε ένα Σύστημα Συμβατού Διαμοιρασμού Χρόνου (Compatible Time Sharing System - CTSS) από το MIT. Το επιβλέπον πρόγραμμα του CTSS χειριζόταν την κονσόλα εισόδου/εξόδου, όριζε το προσκήνιο και το παρασκήνιο των εργασιών, την προσωρινή αποθήκευση και επαναφορά των προγραμμάτων κατά την προγραμματισμένη εναλλαγή, τον έλεγχο της εισόδου και εξόδου του δίσκου, κ.α. Ο επόπτης είχε απευθείας έλεγχο σε όλες τις παγίδες διακοπών.

Περίπου την ίδια εποχή, η IBM άρχισε να φτιάχνει την οικογένεια υπολογιστών 360. Το πρότζεκτ MAC του MIT, που ιδρύθηκε το 1963, ήταν ένας μεγάλος και καλά χρηματοδοτούμενος οργανισμός που αργότερα μεταμορφώθηκε στο Εργαστήριο για την Επιστήμη των Υπολογιστών του MIT. Ο στόχος του πρότζεκτ MAC περιελάμβανε το σχεδιασμό και την εκτέλεση ενός συστήματος που θα κάνει καλύτερο διαμοιρασμό του χρόνου βασισμένο στις ιδέες του CTSS. Αυτή η έρευνα οδήγησε στο Multics, παρόλο που η IBM έχασε την προσφορά και χρησιμοποιήθηκε τελικά το GE 645 της General Electric.

Παρόλο την «ήττα», η IBM είναι ίσως η εταιρία με τη μεγαλύτερη συμβολή στον χώρο της εικονικοποίησης. Ανέπτυξε έναν αριθμό από συστήματα εικονικών μηχανών. Το CP-40 (αναπτύχτηκε για μια αλλαγμένη εκδοχή του IBM 360/40), το CP-67 (αναπτύχτηκε για το IBM 360/67), το περίφημο VM/370 και πολλά άλλα. Βασικά, τα εικονικά μηχανήματα ήταν ακριβείς αντιγραφές του υποκείμενου υλικού. Ένα μέρος του που ονομάστηκε «παρακολούθηση» εικονικών μηχανών (virtual machine monitor -VMM) έτρεχε μόνο απευθείας στο «πραγματικό» υλικό. Στη συνέχεια, πολλαπλά εικονικά μηχανήματα μπορούσαν να δημιουργηθούν μέσω του VMM.

Γιατί είναι σημαντική

Η εικονικοποίηση απέκτησε ένα σημαντικό μέρος δημοσιότητας τα τελευταία χρόνια. Δεδομένου αυτού του απότομου κύματος προσοχής θα ήταν εύκολο να πιστέψει κανείς ότι η έννοια της εικονικοποίησης είναι καινούργια. Όπως είδαμε παραπάνω, η εικονικοποίηση υπάρχει με τη μία ή την άλλη μορφή γύρω μας από το 1960.

Περιοχές που έχουν οφέλη από την εικονικοποίηση

Υπάρχουν 3 κύριοι τύποι εικονικοποίησης, όλοι από τους οποίους παίζουν σημαντικό ρόλο στις επιχειρήσεις:

Εικονικοποίηση αποθήκευσης

Αυτή η έννοια χρησιμοποιείται όταν οι πόροι από πολλές δικτυακές συσκευές αποθήκευσης συνδυάζονται ώστε να μοιάζουν σε να είναι μια μεγάλη αποθήκη. Η διαχείριση γίνεται μέσω ενός κεντρικού συστήματος, πράγμα που κάνει τη ζωή των διαχειριστών ευκολότερη, καθώς μπορούν να γνωρίζουν τον όγκο των δεδομένων που υπάρχουν στην εταιρία, το διαθέσιμο χώρο και κάνει τη ζωή ευκολότερη όσον αφορά τα αντίγραφα ασφαλείας.

Εικονικοποίηση δικτύου

Η εικονικοποίηση δικτύου είναι όταν όλοι οι ξεχωριστοί πόροι ενός δικτύου συνδυάζονται, επιτρέποντας το διαχειριστή να τους διανέμει μεταξύ των χρηστών του δικτύου. Αυτό γίνεται χωρίζοντας τους πόρους του εύρους ζώνης σε κανάλια και επιτρέποντας το διαχειριστή να τα διανείμει κατάλληλα όποτε χρειάζεται κάνοντας ευκολότερη τη διαχείριση του δικτύου. Με αυτόν τον τρόπο οι χρήστες μπορούν να έχουν πρόσβαση σε διάφορους δικτυακούς δίσκους, εκτυπωτές κ.α.

Εικονικοποίηση σέρβερ

Αυτή είναι η κύρια περιοχή της εικονικοποίησης, όπου ένας αριθμός από εικονικά μηχανήματα δημιουργούνται σε ένα σέρβερ και λειτουργούν παράλληλα. Αυτό σημαίνει ότι οποιοδήποτε σφάλμα συμβαίνει φαίνεται να δημιουργείται σε ξεχωριστό χώρο, πράγμα που σημαίνει ότι μπορεί να διαγνωσθεί και να λυθεί γρήγορα.

Κλιμάκωση - Υποψήφιοι για εικονικοποίηση

Υπάρχουν κάποιες διαφορές μεταξύ του κλιμακώνω «έξω» και κλιμακώνω «άνω» (scaling "out" and scaling "up"). Κάποιες εφαρμογές μπορούν να κλιμακωθούν πάνω, αυτό σημαίνει ότι μπορούν να λειτουργήσουν σε γρηγορότερο υλικό για να υποστηρίξουν περισσότερες διεργασίες.

Αυτό ονομάζεται επίσης «κάθετη κλιμάκωση». Ας πούμε ότι έχουμε ένα σέρβερ με βάση δεδομένων που μπορεί να χειριστεί 1 εκατομμύριο αιτήματα το δευτερόλεπτο, αλλά λόγω νέων απαιτήσεων χρειαζόμαστε η βάση να χειρίζεται τουλάχιστον 2 εκατομμύρια το δευτερόλεπτο. Οι βάσεις δεδομένων είναι κατάλληλες για άνω κλιμάκωση καθώς είναι συχνή η συμφόρηση που παρατηρείται στη μνήμη RAM και τη μονάδα κεντρικής επεξεργασίας.

Έξω κλιμάκωση (οριζόντια) σημαίνει να προσθέτεις περισσότερους σέρβερ και να μοιράζεις τον φόρτο ανάμεσα σε πολλαπλά μηχανήματα. Στις βάσεις δεδομένων αυτό θα ήταν δύσκολο, αφού όλοι οι σέρβερ θα έπρεπε να έχουν τα ίδια δεδομένα και θα έπρεπε να αναπαράγονται. Η οριζόντια κλιμάκωση παρόλα αυτά, είναι μια συνήθης πρακτική.

Πριν αποφασίσουμε όμως ποια κλιμάκωση θα ακολουθήσουμε πρέπει να καταλάβουμε ότι καθεμία έχει τα δικά της προβλήματα. Οι εφαρμογές ιστού πχ, χρειάζονται δεδομένα συνεδρίασης ώστε οι σέρβερ εξισορρόπησης φορτίου να είναι στην ίδια κατάσταση. Ένα παράδειγμα είναι η αυθεντικοποίηση. Αν ένας χρήστης αυθεντικοποιηθεί σε έναν σέρβερ και ένας άλλος σέρβερ εξισορρόπησης φορτίου αποφασίσει να εξυπηρετήσει την επόμενη αίτηση του, θα αποτύχει να αναγνωρίσει ότι ο χρήστης έχει αυθεντικοποιηθεί.

Κάποιες ερωτήσεις που πρέπει να απαντήσει κάποιος πριν προχωρήσει στην κατάλληλη κλιμάκωση είναι:

Η εφαρμογή θα δουλέψει σωστά σε ένα περιβάλλον εξισορρόπησης φόρτου?

Μπορεί η εφαρμογή να κλιμακωθεί άνω ώστε να εξυπηρετήσει αρκετούς χρήστες χωρίς εξισορρόπηση φόρτου?

Μπορώ να τρέξω πολλά στιγμιότυπα της ίδιας εφαρμογής σε έναν αυτοματοποιημένο και καλά διαχειριζόμενο τρόπο?

Αν βρίσκεσαι στη θέση που πρέπει να κλιμακώσεις άνω επειδή υπάρχουν περιορισμοί στην εφαρμογή, τότε δε θα έπρεπε να χρησιμοποιήσεις εικονικοποίηση. Μια εφαρμογή που χρειάζεται το δικό της σέρβερ δεν είναι υποψήφια για εικονικοποίηση. Η επιβάρυνση (overhead) της εικονικοποίησης παρόλο που είναι μικρή σήμερα, θα συμβάλει στο να περιοριστεί η απόδοση. Επιπρόσθετα δε θα αποκτήσεις κανένα από τα πλεονεκτήματα της

εικονικοποίησης όπως είναι η ενοποίηση και η μεταφορά ανάμεσα σε φυσικούς σέρβερ καθώς η εφαρμογή θα πρέπει να λειτουργεί σε δικό της σέρβερ.

Κοινές χρήσεις και οφέλη σήμερα

Ο επόπτης εικονικών μηχανημάτων (**hypervisor**) παρέχει έναν εύκολο τρόπο για να χρησιμοποιήσεις το φυσικό υλικό του υπολογιστή για διαφορετικά καθήκοντα. Τα λειτουργικά συστήματα το κάνουν αυτό για χρόνια, απλά με το να επιτρέπει τους χρήστες να χρησιμοποιούν διαφορετικές εφαρμογές παράλληλα. Παρόλα αυτά χωρίς την εικονικοποίηση η αποκλειστική χρήση ενός λειτουργικού συστήματος αποκλείει πολλές επιλογές. Τα τελευταία χρόνια υπάρχει μια τάση φυγής από τα λειτουργικά συστήματα windows και κερδίζουν έδαφος τα λειτουργικά MacOS της Apple και τα λειτουργικά συστήματα βασισμένα σε Linux. Έτσι μπορεί μια εταιρία να έχει Linux σέρβερ για να φιλοξενεί τα site της, windows σέρβερ για τα mail κοκ. Η εικονικοποίηση επιτρέπει διαφορετικά λειτουργικά συστήματα να λειτουργούν παράλληλα στο ίδιο μηχάνημα. Επίσης κάποιος χρήστης μπορεί να τρέχει παράλληλα στον προσωπικό του υπολογιστή διαφορετικά λειτουργικά. Έτσι θα μπορεί να έχει τη σταθερότητα που προσφέρει το Linux αλλά να μπορεί να χρησιμοποιεί και τα προγράμματα που είναι διαθέσιμα μόνο για windows. Για παράδειγμα αν χρησιμοποιείς Linux για να αναπτύξεις και να δοκιμάσεις ένα πρόγραμμα δε θα μπορείς να χρησιμοποιείς προγράμματα που είναι γραμμένα αποκλειστικά για windows. Επίσης αν ένας υπολογιστής έχει εγκατεστημένες όλες τις τελευταίες ενημερώσεις του λειτουργικού συστήματος, η αναπαραγωγή προβλημάτων για χρήστες με παλιότερες εκδόσεις θα ήταν αδύνατη.

Χωρίς την εικονικοποίηση θα αναγκαζόμασταν να έχουμε διαφορετικά φυσικά μηχανήματα παρόλο που θα μπορούσαν να τρέξουν όλα σε ένα, καθώς η υπολογιστική του δύναμη θα το επέτρεπε. Τα περισσότερα από τα πλεονεκτήματα της εικονικοποίησης προκύπτουν από την αφθονία της υπολογιστικής δύναμης διαθέσιμης σε ένα μηχάνημα. Όσο η δύναμη του μέσου μηχανήματος αυξάνεται, τόσο αυξάνεται και η υπολογιστική δύναμη που είναι ανεκμετάλλευτη, ειδικά με τα συστήματα με επεξεργαστές πολλών πυρήνων.

Εδώ έρχεται να αναλάβει δράση ο επόπτης εικονικών μηχανημάτων που ελέγχει το υλικό, επιτρέπει να χρησιμοποιείται από πολλούς χρήστες και δίνει την ψευδαίσθηση στον χρήστη ότι τρέχει στο δικό του φυσικό μηχάνημα. Ο επόπτης χωρίζει τους φυσικούς πόρους του ξενιστή υπολογιστή σε διακριτές εικονικές αντιστοιχίες που μπορούν να χρησιμοποιηθούν από ξεχωριστούς επισκέπτες. Οι εικονικοί επισκέπτες μεταχειρίζονται το εικονικό υλικό σα να ήταν πραγματικό. Επιπρόσθετα οι επόπτες πρέπει να εξασφαλίζουν κάποιο επίπεδο απομόνωσης μεταξύ των επισκεπτών.

Ο επόπτης ακόμα δίνει μια έννοια διεπαφής με το υλικό. Αυτή η διεπαφή θωρακίζει τα συστήματα επισκεπτών από τις χαμηλού επιπέδου λεπτομέρειες του συστήματος, και έτσι προσδίδει στο σύστημα φορητότητα, που είναι ένα ακόμα θετικό της εικονικοποίησης.

Για την ακρίβεια, πολύ σύγχρονοι επόπτες επιτρέπουν τα εικονικά συστήματα να μετακινούνται από μια φυσική μηχανή σε άλλη χωρίς διακοπή. Έτσι ένα εικονικό σύστημα μπορεί να στηθεί σε έναν επόπτη και μετά να μεταφερθεί και σε άλλους μειώνοντας έτσι κατά πολύ το χρόνο που χρειάζεται για να στηθεί ένα σύστημα

και κάνοντας ευκολότερη τη διαχείριση τους ανεξάρτητα από το διαφορετικό υλικό και χαρακτηριστικά που υπάρχουν από «κάτω». Στο Xen μάλιστα η όλη διαδικασία της μεταφοράς μπορεί να γίνει ενώ το σύστημα χρησιμοποιείται, με μια διακοπή μόνο κάποιων μιλισεκόντ.

Ένα ακόμα πλεονέκτημα της εικονικοποίησης, είναι η ευκολία που δίνει στο χρήστη να κάνει διάφορες δοκιμές στο λειτουργικό σύστημα χωρίς να επηρεάζει τα υπόλοιπα. Για παράδειγμα ένα τέτοιο εικονικό σύστημα μπορεί να χρησιμοποιηθεί για διδακτικούς σκοπούς ώστε ο εκπαιδευόμενος να μπορέσει να εξοικειωθεί και να πειραματιστεί με ένα ακριβές αντίγραφο του κανονικού συστήματος χωρίς τον κίνδυνο να καταστρέψει κάτι στο αυθεντικό. Με το ίδιο σκεπτικό ερευνητές μπορούν να ερευνήσουν τα αποτελέσματα κάποιου ιού ή σκουληκιού χωρίς να επηρεάσουν το αρχικό σύστημα. Τέτοια συστήματα ονομάζονται “sandboxed” και μπορούν να χρησιμοποιηθούν και για να δοκιμάσει εκ του ασφαλούς ένας διαχειριστής το πώς θα επηρεάσουν νέες ενημερώσεις και προγράμματα το σύστημα και μετά να τα εφαρμόσει και στο αυθεντικό.

Ένα ακόμα πλεονέκτημα της εικονικοποίησης είναι η ικανότητα της γρήγορης ανάκτησης του συστήματος μετά από ένα σφάλμα προγράμματος ή κάποιον ιό, αρκεί να υπάρχει ένα αντίγραφο του συστήματος, μία εικόνα του, από όταν ήταν πλήρως λειτουργικό.

Η εικονικοποίηση μπορεί να προσφέρει μεγαλύτερη διαθεσιμότητα μεταφέροντας το εικονικό σύστημα (guest) όταν ο σέρβερ στον οποίο φιλοξενείται δε λειτουργεί σωστά.

Το μεγάλα περιβάλλοντα σέρβερ έχουν πολλά φυσικά μηχανήματα στα οποία φιλοξενούνται πολλά συστήματα που λειτουργούν παράλληλα. Τα συστήματα αυτά μπορούν να αντιγραφούν από μηχανήμα σε μηχανήμα εξυπηρετώντας έτσι περισσότερους χρήστες μέσω της εξισορρόπησης φόρτου.

Η εικονικοποίηση έκανε αυτές τις λειτουργίες των εξωτικών σέρβερ προσιτές σε μεγαλύτερο κοινό.

Τα περισσότερα πλεονεκτήματα της εικονικοποίησης εμφανίζονται σε περιβάλλον σέρβερ. Το πιο χαρακτηριστικό παράδειγμα είναι η ικανότητα της συγκέντρωσης πολλών υπηρεσιών σε ένα φυσικό μηχανήμα ενώ παραμένει η δυνατότητα της ξεχωριστής διαχείρισης τους. Σε ένα περιβάλλον πολλαπλής φιλοξενίας ένας φορέας παροχής υπηρεσιών μπορεί να φιλοξενήσει πολλά συστήματα από διαφορετικές εταιρίες στο ίδιο φυσικό μηχανήμα. Κάθε οντότητα έχει τα δικά της δικαιώματα διαχείρισης, κάνει ότι αλλαγές θέλει και μπορεί να τρέξει ότι προγράμματα θέλει χωρίς να χρειαστεί να συμβουλευτείται ή να συντονίζεται με τους ιδιοκτήτες των άλλων συστημάτων-επισκεπτών.

Ο επόπτης είναι πολύ χρήσιμος στους προγραμματιστές που θέλουν να δοκιμάζουν τις δημιουργίες στους σε διαφορετικά συστήματα καθώς μπορούν να εναλλάσσονται ανάμεσα σε διάφορα λειτουργικά χωρίς να χρειάζεται να κάνουν επανεκκίνηση τον υπολογιστή.

Η εικονικοποίηση μπορεί να οδηγήσει και στη μείωση του κόστους απόκτησης του εξοπλισμού γιατί από το υλικό χρησιμοποιούνται όλες οι δυνατότητες του. Για παράδειγμα, αν μια εταιρία σε κάθε σέρβερ φιλοξενούσε 2 από τα συστήματα της αντί για ένα θα μείωνε το κόστος απόκτησης στο μισό. Βέβαια δε μπορούν να εικονικοποιηθούν όλα τα συστήματα, αλλά υπάρχουν περιπτώσεις που ένα μηχανήμα είναι στην ουσία αδρανές. Το σύστημα το οποίο φιλοξενεί είναι κατάλληλος υποψήφιος για εικονικοποίηση.

Επίσης τα sandboxed συστήματα που αναφέραμε παραπάνω μπορούν να φιλοξενηθούν σε ένα μηχάνημα μειώνοντας έτσι εκτός από το ρίσκο και το κόστος εκπαίδευσης.

Επίσης, αργά ή γρήγορα οι σέρβερ που χρησιμοποιούνται θα παλαιώσουν και θα πρέπει να αντικατασταθούν με καινούργιους. Στην περίπτωση αυτή τα συστήματα που έτρεχαν στους παλιούς σέρβερ μπορούν να τρέξουν σαν εικονικά μηχανήματα στους καινούργιους συνεχίζοντας να προσφέρουν τις υπηρεσίες τους, επωφελούμενα κιόλας από το καινούργιο και πιο γρήγορο υλικό, μέχρι να στηθούν καινούργια μηχανήματα, εικονικά και αυτά, με τις καινούργιες εκδόσεις του λειτουργικού.

Το τελευταίο πλεονέκτημα είναι μείωση της κατανάλωσης ενέργειας. Η λεγόμενη πράσινη πληροφορική αναφέρεται στην πρόσφατη τάση για μείωση της κατανάλωσης ενέργειας στα υπολογιστικά συστήματα. Παρόλο που κάτι τέτοιο δεν είναι πρωταρχικό μέλημα για μεμονωμένους χρήστες ή μικρές επιχειρήσεις, εταιρίες με μεγάλο αριθμό σέρβερ στην κατοχή τους μπορούν να μειώσουν σημαντικά την κατανάλωση ρεύματος χρησιμοποιώντας λιγότερους σέρβερ μέσω της εικονικοποίησης. Επιπρόσθετα χρησιμοποιώντας λιγότερους σέρβερ, μειώνεται και η κατανάλωση ρεύματος από τα συστήματα ψύξης καθώς λιγότεροι σέρβερ προκαλούν και μικρότερη παράγωγή θερμότητας.

Τύποι εικονικοποίησης

Από όλους τους τύπους εικονικοποίησης, οι πιο σημαντικές μορφές της είναι η , πλήρης εικονικοποίηση (full virtualization) , η παρά-εικονικοποίηση (para-virtualization) και η εικονικοποίηση λειτουργικού συστήματος (operating system virtualization). Μια ακόμα μορφή εικονικοποίησης, η επιπέδου – εφαρμογών, τελευταία αποκτάει δημοτικότητα.

Πλήρης εικονικοποίηση

Παραδείγματα: XEN, VMware, z/VM

Αυτή είναι η ιδανική τεχνική εικονικοποίησης στην οποία ένα λειτουργικό σύστημα χρησιμοποιείται σαν οικοδεσπότης και σαν φιλοξενούμενο μπορούμε να χρησιμοποιήσουμε οποιαδήποτε **απαράλλακτη** έκδοση λειτουργικού.

Οι εξομοιωτές υλικού μπαίνουν στο παιχνίδι όταν θέλουμε να συνενώσουμε πολλούς σέρβερ. Για παράδειγμα μια τυπική επιχείρηση μπορεί να έχει έναν σέρβερ για SQL, έναν για Apache και έναν Exchange σέρβερ. Κάθε ένα από αυτά τα συστήματα μπορεί να χρησιμοποιεί μόνο το 5% από την υπολογιστική δυνατότητα.

Ένας εξομοιωτής υλικού παρουσιάζει μια διεπαφή προσομοίωσης υλικού στα φιλοξενούμενα λειτουργικά συστήματα. Στην πλήρη προσομοίωση, το λογισμικό εικονικοποίησης (hypervisor) δημιουργεί μια τεχνητή συσκευή υλικού με όλα όσα χρειάζεται για να δουλέψει ένα λειτουργικό σύστημα και παρουσιάζει ένα προσομοιωμένο περιβάλλον υλικού στο οποίο λειτουργούν τα φιλοξενούμενα λειτουργικά συστήματα. Αυτό το προσομοιωμένο περιβάλλον υλικού τυπικά αναφέρεται ως Επόμενη Εικονικών Μηχανών (EEM) γνωστό στα αγγλικά ως Virtual Machine Monitor (VMM).

Η προσομοίωση υλικού υποστηρίζει σαν φιλοξενούμενους πραγματικά λειτουργικά συστήματα και στο καθένα οι εφαρμογές τρέχουν σε ένα πλήρως απομονωμένο λειτουργικό σύστημα από τα υπόλοιπα. Ο EEM παρέχει στα φιλοξενούμενα λειτουργικά συστήματα μια πλήρη προσομοίωση του υπάρχοντος υλικού και με αυτήν την τεχνική υποστηρίζει αμετάβλητα λειτουργικά συστήματα. Αυτό σημαίνει ότι όλα τα λειτουργικά συστήματα και το λογισμικό που τρέχει σε έναν υπολογιστή, μπορεί να υποστηριχθεί και σαν εικονικό μηχάνημα. Γι' αυτούς τους λόγους η εικονικοποίηση αυτή ονομάζεται Πλήρης Εικονικοποίηση. Ο όρος αμετάβλητος αναφέρεται στον πυρήνα του λειτουργικού συστήματος που δεν έχει αλλαχθεί για να τρέξει και γι αυτό μπορεί ακόμα να εκτελέσει

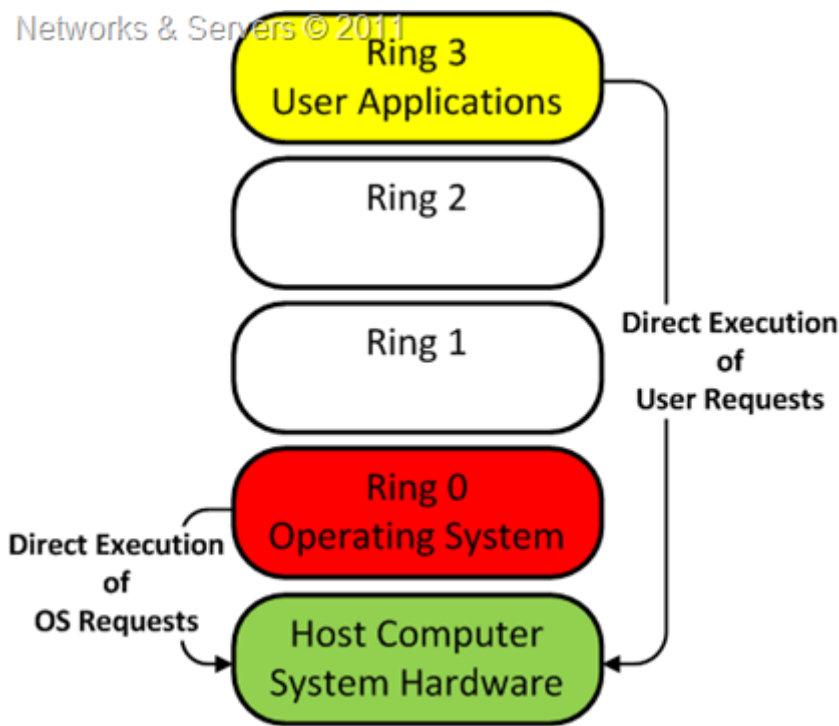
προνομιούχες (privileged) λειτουργίες στον δακτύλιο 0 του επεξεργαστή ([βλέπε παρακάτω](#)). Η πλήρης εικονικοποίηση χρησιμοποιεί το λογισμικό εικονικοποίησης (hypervisor) για να συντονίσει τον επεξεργαστή του σέρβερ και τους πόρους του μηχανήματος-οικοδεσπότη με σκοπό να διαχειριστεί τα λειτουργικά συστήματα χωρίς αλλαγή, προσφέροντας μια προσομοίωση του επεξεργαστή που χειρίζεται και μετατρέπει τις προνομιούχες και προστατευόμενες λειτουργίες του επεξεργαστή.

Το λειτουργικό σύστημα του φιλοξενούμενου δημιουργεί κλήσεις συστήματος στο προσομοιωμένο υλικό. Αυτές οι κλήσεις, που κανονικά θα αλληλεπιδρούσαν με το υπάρχον υλικό, μεταφράζονται από τον hypervisor ο οποίος τις κατευθύνει στο πραγματικό υπάρχον υλικό. Με αυτόν τον τρόπο ο hypervisor προσφέρει πλήρη και ανεξάρτητη ελευθερία και αυτονομία σε κάθε εικονικό σύστημα που υπάρχει σε έναν σέρβερ. Κάθε φιλοξενούμενο σύστημα έχει το δικό του λειτουργικό που μπορεί να είναι διαφορετικό από σύστημα σε σύστημα. Για παράδειγμα ένα φιλοξενούμενο σύστημα μπορεί να τρέχει Linux και ένα άλλο Windows.

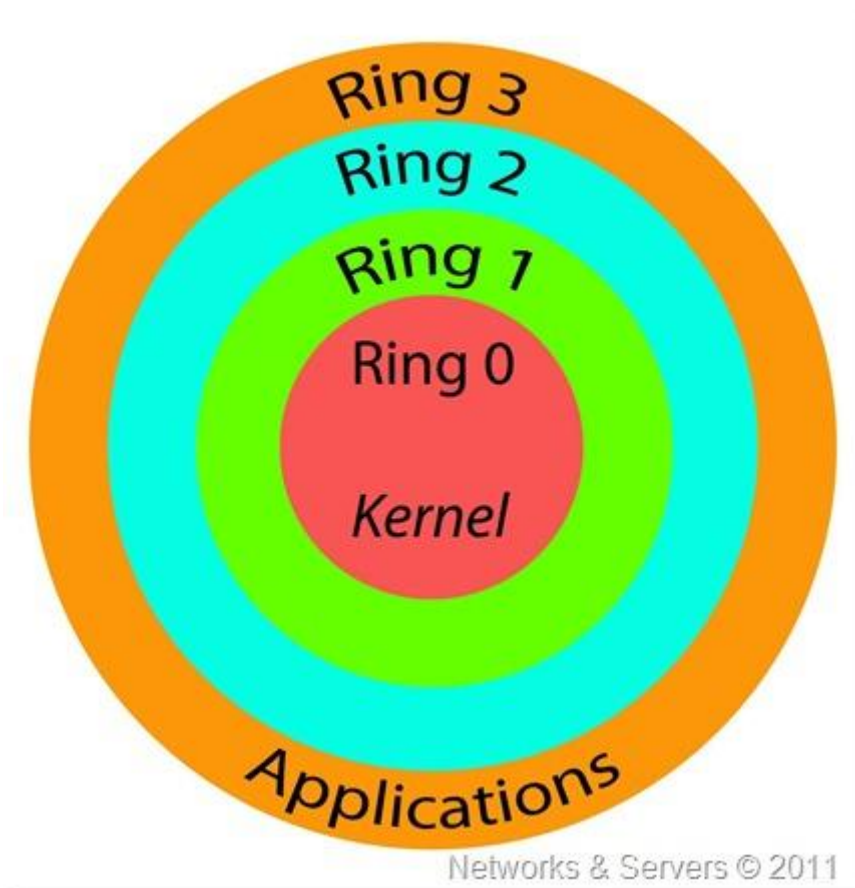
Ο hypervisor επίσης παρακολουθεί και ελέγχει τους φυσικούς πόρους του σέρβερ, κατανέμοντας όσο χρειάζεται σε κάθε λειτουργικό σύστημα και φροντίζει ότι τα λειτουργικά συστήματα των εικονικών μηχανημάτων δε θα διακόψουν το ένα το άλλο. Με την πλήρη εικονικοποίηση ο φιλοξενούμενος δε γνωρίζει ότι τρέχει σαν εικονικό μηχάνημα και γι αυτό δε χρειάζεται και κάποια αλλαγή.

Επίπεδα προστασίας επεξεργαστή

Η x86 αρχιτεκτονική προσφέρει ένα εύρος από επίπεδα προστασίας, γνωστά και σαν δακτύλιοι, στους οποίους μπορεί να εκτελεστεί κώδικας και αυτοί με τη σειρά τους χρησιμοποιούν το λειτουργικό σύστημα και τα προγράμματα για να αποκτήσουν πρόσβαση στο υλικό του υπολογιστή. Ο δακτύλιος 0 έχει το μεγαλύτερο επίπεδο προνομίου και σε αυτόν τον δακτύλιο εκτελείται ο πυρήνας του λειτουργικού συστήματος. Ο κώδικας που εκτελείται σε αυτόν τον δακτύλιο λέμε ότι τρέχει σε χώρο συστήματος, κατάσταση πυρήνα ή κατάσταση επόπτη. Όλος ο άλλος κώδικας όπως αυτός των εφαρμογών που τρέχουν στο λειτουργικό σύστημα λειτουργούν σε λιγότερο προνομιούχους δακτυλίους, συνήθως στο δακτύλιο 3.



Εδώ πρέπει να τονίσουμε ότι το επίπεδο προνομίου δεν έχει καμία σχέση με τα προνόμια που έχει ένας χρήστης σε ένα λειτουργικό σύστημα. Είτε είναι διαχειριστής (administrator), ρίζα(root) ή απλός χρήστης, δεν έχει σημασία ανεξάρτητα από το λειτουργικό που χρησιμοποιείς.



Ο όρος δακτύλιος φέρεται να αναφέρεται στο αυθεντικό εγχειρίδιο αναφοράς της 80386 αρχιτεκτονική που δείχνει τα τέσσερα επίπεδα προστασίας σαν ομόκεντρους κύκλους των εργασιών. Ο δακτύλιος 3 ήταν ο εξωτερικός δακτύλιος και προσέδιδε τη μεγαλύτερη προστασία, επιτρέποντας μόνο την εκτέλεση των οδηγιών που δε θα μπορούσαν να επηρεάσουν την συνολική κατάσταση του επεξεργαστή. Ο δακτύλιος 0 ήταν ο πιο εσωτερικός δακτύλιος και επέτρεπε τον πλήρη έλεγχο του επεξεργαστή.

Επόπτης (Hypervisor)

Οι Hypervisors έχουν σχεδιαστεί για να δημιουργούν όσο το δυνατόν την μεγαλύτερη απομόνωση μεταξύ των εικονικών μηχανημάτων. Όμως η πιο ισχυρή απομόνωση έχει επιπτώσεις στην απόδοση. Αυτό γίνεται λόγω της μεγαλύτερης επιβάρυνσης (overhead) που χρειάζεται για να εφαρμοστεί η μεγαλύτερη απομόνωση.

Αν η εικονικοποίηση ορίζεται ως η δυνατότητα να λειτουργούν παράλληλα πολλά λειτουργικά συστήματα σε ένα φυσικό μηχάνημα τότε ο hypervisor είναι απαραίτητο συστατικό. Ο hypervisor γνωστός και ως Επόπτης Εικονικών Μηχανών (EEM), δημιουργεί μια εικονική πλατφόρμα στο μηχάνημα-οικοδεσπότη, πάνω στον οποίο θα εκτελεστούν και θα παρακολουθούνται πολλαπλά λειτουργικά συστήματα. Με αυτόν τον τρόπο τα λειτουργικά συστήματα που μπορούν να είναι είτε πολλές φορές ένα λειτουργικό σύστημα, είτε διαφορετικά μοιράζονται τους διαθέσιμους πόρους του οικοδεσπότη.

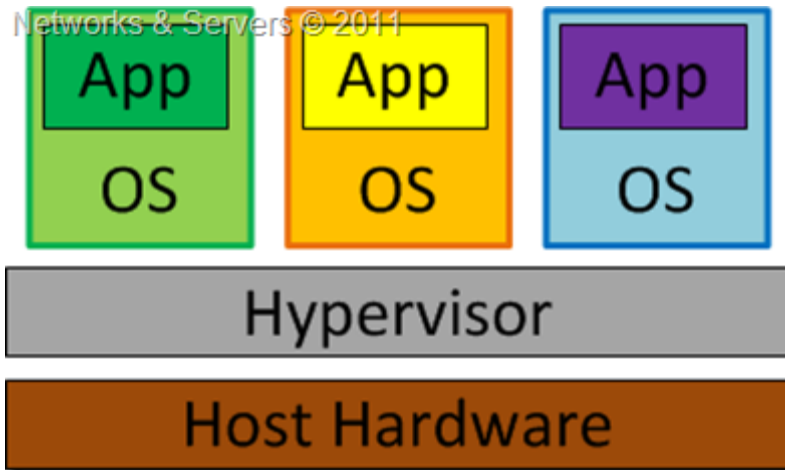
Επίσης είναι υπεύθυνος για να ξεκινήσει αρχικά το μηχάνημα μέχρι το σημείο που θα μπορεί να φορτωθεί το πρώτο μηχάνημα (Dom0).

Όπως οι εφαρμογές μπορούν να αλληλεπιδρούν με το λειτουργικό με κλήσεις syscall, τα εικονικά συστήματα αλληλεπιδρούν με τον hypervisor στέλνοντας κλήσεις hypercall. Ο hypervisor απαντά στέλνοντας ένα συμβάν, που κάνει την ίδια λειτουργία όπως μια γραμμή αιτήματος διακοπής IRQ στο φυσικό υλικό.

Τύπος 1 Hypervisor

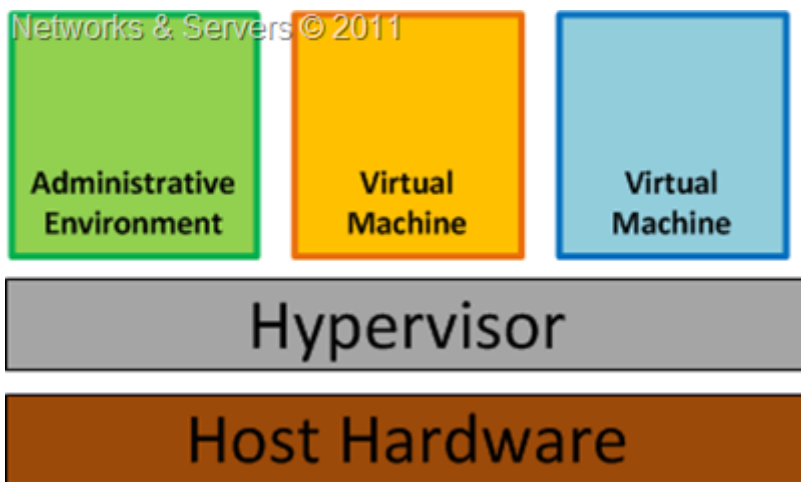
Παραδείγματα: Oracle VM, Microsoft Hyper-V, VMware ESX και Xen.

Γνωστός και ως Ντόπιος, Μητρικός ή Βασική Εικονικοποίηση, είναι η τεχνική όπου το στρώμα αφαίρεσης κάθεται απευθείας πάνω στο υλικό και όλα τα τμήματα εδρεύουν από πάνω του. Ο τύπος 1 τρέχει κατευθείαν πάνω στο υλικό του οικοδεσπότη στο δακτύλιο 0. Το έργο του είναι να χειρίζεται πόρους και να κατανέμει τη μνήμη στα εικονικά συστήματα και να προσφέρει διεπαφές για υψηλού επιπέδου διαχείριση και εργαλεία παρακολούθησης. Τα λειτουργικά συστήματα τρέχουν σε άλλο επίπεδο πάνω από τον hypervisor.

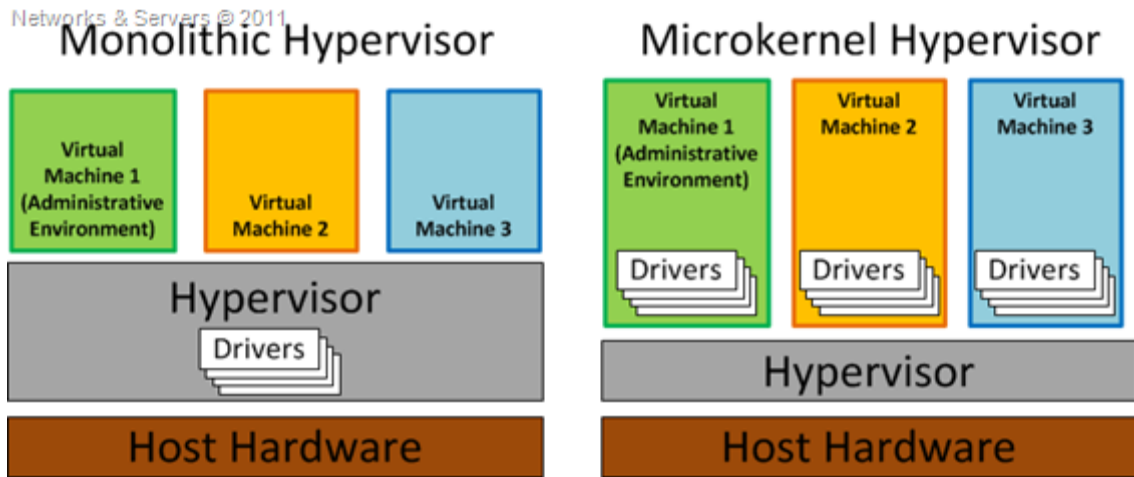


Με τον hypervisor να καταλαμβάνει το δακτύλιο 0 του επεξεργαστή, οι πυρήνες των φιλοξενούμενων λειτουργικών συστημάτων πρέπει να τρέξουν σε λιγότερο προνομιούχους δακτυλίους.

Δυστυχώς οι πυρήνες των περισσότερων λειτουργικών συστημάτων είναι γραμμένοι να τρέχουν αποκλειστικά στο δακτύλιο 0 για τον απλό λόγο ότι πρέπει να εκτελούν εργασίες που είναι διαθέσιμες μόνο σε αυτόν τον δακτύλιο, όπως η δυνατότητα να εκτελούν προνομιούχες εντολές στον επεξεργαστή και να διαχειρίζονται απευθείας τη μνήμη.



Επίσης ανάλογα με την αρχιτεκτονική του hypervisor, μπορεί να περιλαμβάνει τους οδηγούς για τους πόρους του υλικού (Μονολιθικός - Monolithic) ή οι οδηγοί μπορεί να βρίσκονται στο φιλοξενούμενο λειτουργικό (Μικροπυρηνικός - Microkernelized).



Καθώς έχει πρόσβαση χαμηλού επιπέδου στο υλικό ο Τύπος 1 είναι πιο αποτελεσματικός και καταφέρνει καλύτερη απόδοση επειδή χρησιμοποιεί λιγότερους πόρους.

Το μειονέκτημα αυτού του μοντέλου είναι ότι υπάρχει εξάρτηση στον hypervisor για τους οδηγούς (τουλάχιστον στο Μονολιθικό). Άλλωστε οι περισσότερες υλοποιήσεις απαιτούν συγκεκριμένη υποστήριξη στο επίπεδο υλικού. Το Xen έχει ένα ειδικό πεδίο το Dom0 που περιέχει τους οδηγούς για το υλικό, όπως και τα εργαλεία για να ελέγχει τα εικονικά μηχανήματα.

Τύπος 2 Hypervisor

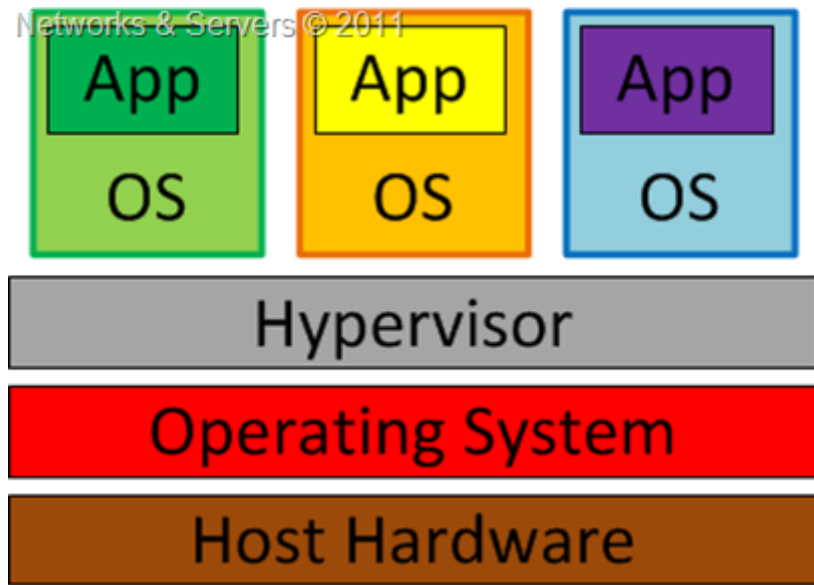
Παραδείγματα: Oracle's VirtualBox, VMware Server, VMware Workstation and Microsoft VirtualPC, QEMU και Parallels.

Guest OS Virtualization

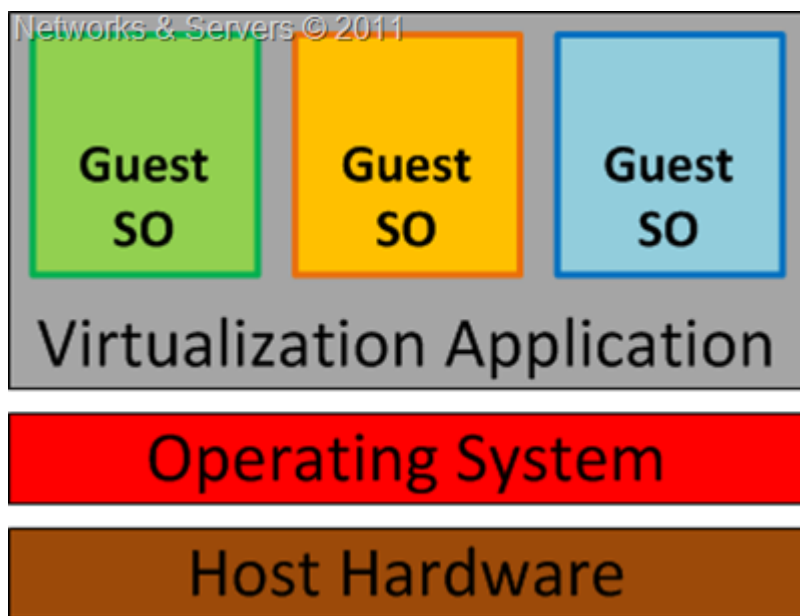
ΣΕ αυτό το σενάριο στον οικοδεσπότη τρέχει ένα αμετάβλητο λειτουργικό σύστημα όπως windows, Linux, Unix, MacOS και το στρώμα της εικονικοποίησης τρέχει πάνω σε αυτό το λειτουργικό στην ουσία σαν μία φιλοξενούμενη εφαρμογή. Σε αυτήν την αρχιτεκτονική, ο EEM παρέχει σε κάθε εικονικό μηχάνημα όλες τις υπηρεσίες του φυσικού συστήματος, περιλαμβάνοντας ένα εικονικό BIOS, εικονικές συσκευές και εικονική μνήμη. Αυτό δίνει την εντύπωση στο φιλοξενούμενο σύστημα όχι ότι είναι ένα εικονικό σύστημα μέσα σε μια εφαρμογή, αλλά ότι τρέχει απευθείας στο υλικό.

Το λειτουργικό παρέχει ένα στρώμα αφαίρεσης, γνωστό σαν Τύπος 2, που επιτρέπει άλλα λειτουργικά συστήματα να «κατοικούν» μέσα του. Γι' αυτό η αρχιτεκτονική αυτή μπορεί να ονομαστεί και σαν «Φιλοξενούμενη

Εικονικοποίηση», αφού ένα λειτουργικό σύστημα τη φιλοξενεί, όπως φαίνεται και στη φωτογραφία.



Το στρώμα εικονικοποίησης, της παραπάνω εικόνας περιέχει λογισμικό που χρειάζεται για να φιλοξενήσει και να διαχειριστεί τα εικονικά μηχανήματα, αλλά η ακριβής του λειτουργικότητα διαφέρει από κατασκευαστή σε κατασκευαστή. Σε αυτήν την προσέγγιση, το λειτουργικό-φιλοξενούμενος βασίζεται στο λειτουργικό-οικοδεσπότης για πρόσβαση στους πόρους του υλικού.



Τα πολλαπλά στρώματα αφαίρεσης μεταξύ του φιλοξενούμενου λειτουργικού και του οικοδεσπότη δεν είναι αγώγιμα στην απόδοση του εικονικού μηχανήματος.

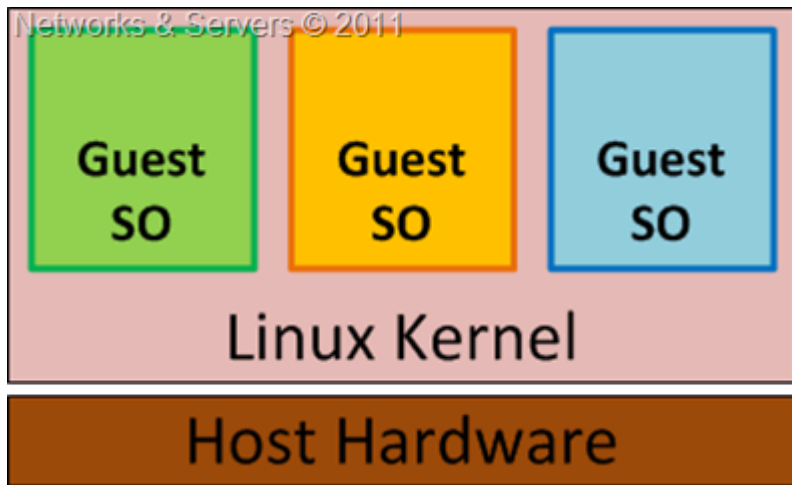
Αυτή η τεχνική έχει όμως το πλεονέκτημα ότι δε χρειάζονται αλλαγές στα λειτουργικά συστήματα, ούτε χρειάζεται κάποια υποστήριξη εικονικοποίησης από την πλευρά του Επεξεργαστή.

Embedded Hypervisor

Παραδείγματα: User Mode Linux (UML) and Kernel-based Virtual Machine (KVM).

Στην εικονικοποίηση επιπέδου Πυρήνα, το λειτουργικό σύστημα τρέχει σε έναν ειδικά διαμορφωμένο πυρήνα που περιέχει επεκτάσεις σχεδιασμένες για διαχείριση και έλεγχο πολλαπλών εικονικών μηχανημάτων, που το καθένα περιέχει ένα λειτουργικό σύστημα.

Το στρώμα εικονικοποίησης προστίθεται στον πυρήνα ενός λειτουργικού και κάθε φιλοξενούμενος τρέχει το δικό του πυρήνα. Υπάρχουν όμως περιορισμοί στους οποίους το φιλοξενούμενο λειτουργικό πρέπει να έχει μεταγλωττιστεί για το ίδιο υλικό όπως ο πυρήνας στον οποίο τρέχουν.



Το πραγματικό όφελος αυτής της προσέγγισης είναι ότι ο κώδικας του hypervisor είναι δραματικά μικρότερος. Με τον hypervisor εισαγμένο στον πυρήνα του Linux, τα φιλοξενούμενα λειτουργικά ωφελούνται από την πολύ καλή απόδοση του δίσκου και του δικτύου.

Πλεονεκτήματα Πλήρους Εικονικοποίησης

- Η προσέγγιση αυτή σημαίνει ότι η εικονικοποίηση τρέχει σε ένα πλήρως απομονωμένο λειτουργικό – οικοδεσπότη, με ένα ή περισσότερα λειτουργικά – φιλοξενούμενους, που τρέχουν εικονικοποιημένα και ταυτόχρονα στο υλικό. Επίσης υποστηρίζει διαφόρων ειδών λειτουργικά, ακόμα και διαφορετικά μεταξύ τους.
- Ο hypervisor μεταφράζει όλες τις οδηγίες του λειτουργικού συστήματος πολύ γρήγορα και αποθηκεύει τα αποτελέσματα για μετέπειτα χρήσεις, ενώ οι οδηγίες επιπέδου χρήστη τρέχουν αναλλοίωτες σε φυσική ταχύτητα.
- Ο ΕΕΜ παρέχει ένα στανταρισμένο περιβάλλον υλικού στο οποίο κατοικούν τα φιλοξενούμενα λειτουργικά και επικοινωνούν μαζί του. Έτσι τα φιλοξενούμενα λειτουργικά και ο ΕΕΜ δημιουργούν ένα συμπαγές πακέτο που μπορεί να μεταφερθεί από μηχάνημα σε μηχάνημα παρόλο που τα φυσικά μηχανήματα μπορεί μεταξύ τους να διαφέρουν.

Περιορισμοί Πλήρους Εικονικοποίησης

- Το λογισμικό της εικονικοποίησης έχει επιπτώσεις στην απόδοση, κάτι που σημαίνει ότι οι εφαρμογές τρέχουν κάπως πιο αργά στα εικονικά μηχανήματα. Ο hypervisor χρειάζεται επεξεργασία δεδομένων, που σημαίνει ότι μέρος της υπολογιστικής δύναμης και συσχετιζόμενοι πόροι είναι δεσμευμένοι από το λογισμικό εικονικοποίησης. Ο ΕΕΜ φαίνεται να λύνει κάπως αυτό το πρόβλημα, αλλά και αυτός με τη σειρά του προσθέτει κάποια παραπάνω επεξεργασία .
- Ο hypervisor μπορεί να χρειάζεται να περιέχει τους οδηγούς για τις διάφορες συσκευές του μηχανήματος. Επειδή η προσομοίωση υλικού χρησιμοποιεί λογισμικό για να ξεγελάσει τα φιλοξενούμενα λειτουργικά με το οποίο επικοινωνούν με προσομοιωμένο ή μη υπάρχον υλικό, υπάρχουν κάποιες φορές προβλήματα ασυμβατότητας. Το πρόβλημα δημιουργείται επειδή ο hypervisor περιέχει τους οδηγούς , μπορεί να είναι δύσκολο στα φιλοξενούμενα λειτουργικά να εγκατασταθούν νέοι οδηγοί.

Παραεικονικοποίηση

Παραδείγματα : XEN, User Mode Linux (UML)

Το πρόθεμα «πάρα» σημάνει «δίπλα σε» ή «παράλληλα». Η τεχνική αυτή απαιτεί αλλαγμένες εκδόσεις των λειτουργικών συστημάτων. . Για παράδειγμα δε μπορείς να χρησιμοποιήσεις την διαθέσιμη έκδοση των Debian, Fedora, CentOS κτλ, σε αυτούς τους εικονικούς σέρβερ, αλλά ειδικά σχεδιασμένες εκδόσεις αυτών των λειτουργικών συστημάτων.

Δηλαδή χρειάζονται αλλαγές στον πυρήνα τους για να αντικατασταθούν οι μη εικονικοποιημένες οδηγίες με υπερκλήσεις που επικοινωνούν απευθείας με το στρώμα εικονικοποίησης του hypervisor. Ο hypervisor επίσης παρέχει διεπαφές υπερκλήσεων για επείγουσες λειτουργίες, όπως η διαχείριση μνήμης, ο χειρισμός των διακοπών και η διαχείριση χρόνου.

Η προσέγγιση αυτή επιτρέπει πολλαπλά λειτουργικά συστήματα να τρέχουν την ίδια ώρα στο ίδιο υλικό κάνοντας πιο αποδοτική τη διαχείριση των πόρων, όπως της μνήμης και του επεξεργαστή μέσω αποτελεσματικού μοιράσματος των πόρων. Αντίθετα με την πλήρη εικονικοποίηση όπου όλο του υλικό προσομοιώνεται , η διαχείριση της παραεικονικοποίησης λειτουργεί με ένα λειτουργικό σύστημα που έχει αλλαχθεί ώστε να λειτουργεί σαν εικονικό μηχάνημα.

Η αφαίρεση που δημιουργείται στην παραεικονικοποίηση γενικά σημαίνει ότι το λειτουργικό σύστημα θα αποδώσει καλύτερα. Όμως αυτό έχει επιπτώσεις στην ασφάλεια και την ευελιξία. Αυτή χάνεται επειδή το λειτουργικό σύστημα πρέπει να αλλαχθεί ώστε να μπορεί να χρησιμοποιηθεί, πράγμα που δεν το κάνουν όλες οι διανομές. Για παράδειγμα, τα λειτουργικά Red Hat Linux, Enterprise Linux, Windows Server δεν είναι διαθέσιμα για παραεικονικοποίηση. Κάποια ασφάλεια χάνεται καθώς το λειτουργικό-φιλοξενούμενος έχει πιο κοντινό έλεγχο στο υπάρχον υλικό, αυξάνοντας έτσι τον κίνδυνο να επηρεάσει το επίπεδο του υλικού, κάτι που θα σήμαινε μειωμένη απόδοση για όλα τα φιλοξενούμενα συστήματα.

Μειονεκτήματα:

1. Πρέπει να χρησιμοποιήσεις αλλαγμένα λειτουργικά γιατί δε μπορεί να υποστηρίξει μη αλλαγμένα λειτουργικά (Windows), κάτι που σημαίνει ότι συμβατότητα και η δυνατότητα μεταφοράς δεν είναι καλή.
2. Τα 64-μπιτα λειτουργικά τρέχουν σε μη προνομιούχο δακτύλιο(δακτύλιος 3).
3. Για να προστατευτεί το λειτουργικό είναι απαραίτητη η εναλλαγή του πίνακα σελίδων (page table). Αυτό έχει σαν αποτέλεσμα να χρειάζονται 2 κλήσεις συστήματος και μια εκκαθάριση του TLB, που είναι πολύ δαπανηρό σε πόρους συστήματος.
4. Οι πρόσθετες εντολές μπορεί να είναι λιγότερες στην παραεικονικοποίηση, απ' ότι στην πλήρη εικονικοποίηση, αλλά αυτό εξαρτάται κατά πολύ από το φόρτο εργασιών.
5. Μπορεί να προκαλέσει σημαντικά ζητήματα υποστήριξης και συντήρησης καθώς χρειάζεται μεγάλες αλλαγές στον πυρήνα.

Εικονικοποίηση επιπέδου λειτουργικού συστήματος

Παραδείγματα : OpenVZ, Linux-VServer

Αυτή είναι η πιο διαδομένη τεχνική εικονικοποίησης. Σε αυτή τη μέθοδο η εικονικοποίηση βασίζεται σε ένα λειτουργικό σύστημα οικοδεσπότη. Αλλά μόνο το ίδιο στιγμιότυπο του λειτουργικού συστήματος του οικοδεσπότη μπορεί να χρησιμοποιηθεί από τα εικονικά μηχανήματα σε λειτουργικό τους σύστημα.

Η προσέγγιση αυτή εικονικοποίησης σέρβερ είναι που ο πυρήνας ενός λειτουργικού συστήματος, επιτρέπει πολλαπλά στιγμιότυπα αντί μόνο για ένα. Αυτές συχνά ονομάζονται και containers, VEs, VPSs ή φυλακές) και μπορεί να μοιάζουν και να φαίνονται σαν κανονικός σέρβερ από τη μεριά του χρήστη. Στα συστήματα Unix η τεχνολογία αυτή μπορεί να θεωρηθεί σαν μια προχωρημένη υλοποίηση του μηχανισμού chroot. Σε συνδυασμό με τους μηχανισμούς απομόνωσης ο πυρήνας συχνά προσφέρει χαρακτηριστικά διαχείρισης για να περιορίσει τις δραστηριότητες του container σε σχέση με τους υπολοίπους.

Η τεχνική αυτή εικονικοποίησης συνήθως προσφέρει πολύ λίγες ή και καθόλου πρόσθετες εντολές, επειδή τα προγράμματα τα σε ένα εικονικό partition χρησιμοποιούν το σύστημα κλήσεων του κανονικού λειτουργικού και δε χρειάζεται κάποια προσομοίωση ή κάποιο ενδιάμεσο λειτουργικό σύστημα. Επίσης δε χρειάζεται υποβοήθηση από το υλικό για να λειτουργήσει σωστά.

Η τεχνική αυτή όμως δεν είναι τόσο ευέλικτη όσο άλλες καθώς δε μπορεί να φιλοξενήσει διαφορετικά λειτουργικά συστήματα, απ ότι του οικοδεσπότη ή διαφορετικού πυρήνα.

Εικονικοποίηση Επιπέδου Εφαρμογών

Παραδείγματα: AppZero, BoxedApp, Cameyo, Ceedo, Evalaze, InstallFree, Citrix XenApp, Novell ZENworks Application Virtualization, Endeavors Technologies Application Jukebox, Microsoft Application Virtualization, Software Virtualization Solution, Spoon (former Xenocode), VMware ThinApp and P-apps.

Η εικονικοποίηση επιπέδου εφαρμογών τρέχει πάνω από το λειτουργικό οικοδεσπότης και τοποθετεί συγκεκριμένες εφαρμογές σε απομονωμένα περιβάλλοντα. Κάθε χρήστης που έχει πρόσβαση σε υπολογιστή φαίνεται να έχει μια δικιά του εγκατάσταση του προγράμματος. Μέσα σε αυτήν οι αλλαγές του συστήματος αρχείων, οι ρυθμίσεις της registry και άλλες λεπτομέρειες γίνονται σε ένα απομονωμένο (sandboxed) περιβάλλον.

Τα σύγχρονα λειτουργικά συστήματα μπορούν να περιλαμβάνουν περιορισμένη εικονικοποίηση εφαρμογών. Για παράδειγμα η χαρτογράφηση των αρχείων INI που εμφανίστηκαν στα Windows NT για να εικονικοποιηθούν, μέσα στη registry, τα legacy INI files για εφαρμογές που είχαν γραφεί για Windows 3.1. Παρομοίως, τα Windows Vista έχουν ένα shim που προσδίδει κάποια εικονικοποίηση αρχείων και registry έτσι ώστε οι εφαρμογές που προσπαθούν να σώσουν δεδομένα του χρήστη σε περιοχές συστήματος μόνο για ανάγνωση που ήταν εγγράψιμες σε προηγούμενες εκδόσεις Windows να μπορούν ακόμα να λειτουργήσουν.

Η προσέγγιση αυτή απαιτεί ένα στρώμα εικονικοποίησης που αντικαθιστά μέρος του περιβάλλοντος εκτέλεσης που φυσιολογικά θα υπήρχε στο λειτουργικό σύστημα. Το στρώμα αυτό παρεμβάλλεται σε όλες τις λειτουργίες αρχείων και registry των εικονικοποιημένων εφαρμογών και διαφανώς τις μεταφέρει σε μια εικονικοποιημένη τοποθεσία, συχνά ένα απλό αρχείο. Η εφαρμογή δε γνωρίζει ότι εισέρχεται σε εικονικό πόρο αντί για πραγματικό. Η εφαρμογή τώρα τρέχει με ένα αρχείο αντί για πολλά αρχεία και εγγραφές registry και αυτό κάνει την εφαρμογή να μπορεί να τρέχει εύκολα σε διαφορετικούς υπολογιστές και προηγούμενες εφαρμογές που δεν ήταν συμβατές να λειτουργούν μαζί.

Πλεονεκτήματα

- Επιτρέπει σε εφαρμογές να τρέξουν σε περιβάλλοντα, που κανονικά δε θα μπορούσαν (π.χ. το Wine επιτρέπει σε εφαρμογές που έχουν γραφεί για Windows να τρέξουν σε Linux).
- Προστατεύει τα λειτουργικά συστήματα από κακογραμμένο και με σφάλματα κώδικα.
- Χρησιμοποιεί λιγότερους πόρους από ένα εικονικό μηχάνημα.
- Διευκολύνει τη μεταφορά λειτουργικών συστημάτων
- Αυξημένη ασφάλεια, απομονώνοντας τις εφαρμογές από το λειτουργικό σύστημα
- Επιτρέπει σε εφαρμογές να αντιγραφούν και να μεταφερθούν σε άλλο μηχάνημα, χωρίς την ανάγκη να τα εγκαταστήσεις σε αυτά.

Περιορισμοί

- Δε μπορούν όλες οι εφαρμογές να εικονικοποιηθούν,
- Δημιουργούνται προβλήματα συμβατότητας σε επίπεδο αρχείων και registry.

Xen hypervisor

Τι είναι το Xen

Το Xen είναι ένα σύστημα εικονικοποίησης που υποστηρίζει και πλήρη και παρα-εικονικοποίηση.

Πήρε το όνομα του από τη «νέα γενιά» (neXtgENeration) εικονικοποίησης και είναι ανοικτού κώδικα (άδεια GPLv2). Το Xen είναι hypervisor Τύπου 1, παρόλο που πρώτα εγκαθιστάμε τη διανομή Unix που θέλουμε και μετά το Xen το οποίο στην ουσία εγκαθίσταται από κάτω.

Σαν τύπου 1, παρέχει ένα στρώμα αφαίρεσης από το υλικό, το οποίο του επιτρέπει να μοιράζεται το υλικό με πολλαπλά λειτουργικά συστήματα. Τα λειτουργικά τρέχουν σε domains. Το Xen τρέχει απευθείας στο υλικό και περνάει τους περισσότερους έλεγχους στο προνομιούχο domain-0 (Dom0 – 0 επειδή είναι το πρώτο που δημιουργείται). Τα εικονικά μηχανήματα εκτελούνται ασφαλή και με απόδοση κοντά στην native.

Με τις ρίζες του στον ανοικτό προγραμματισμό και με πολύ υποστήριξη από την κοινότητα ανοικτού κώδικα, το xen είναι ένας από τους πιο δημοφιλείς EEM. Κάποια λειτουργικά συστήματα έχουν αρχίσει να ενσωματώνουν το hypervisor του xen. Έτσι σύντομα δε θα είναι καν αναγκαίο να εγκαθιστάς το Xen.

Η ανάπτυξη του xen στην κοινότητα καθοδηγείται τώρα από την XenSource, που ιδρύθηκε από τον Pratt. Παρόλο που υπάρχουν εμπορικές εκδόσεις για επιχειρήσεις βασισμένες στην τεχνολογία του Xen, η XenSource έχει δεσμευτεί στην ανάπτυξη της κοινότητας του Xen, προωθώντας και εμπνέοντας τους προγραμματιστές ώστε να οδηγήσουν το Xen ακόμα πιο μακριά.

Το Xenproject είναι αυτοδιοικούμενο. Το συμβούλιο, συμβουλεύει τον αρχηγό του Xenproject και είναι υπεύθυνο για το εμπορικό σήμα, το οποίο η Citrix έχει την αδεία να το χρησιμοποιεί ελεύθερα σε όλες τις διανομές του που είναι βασισμένες στο Xen.

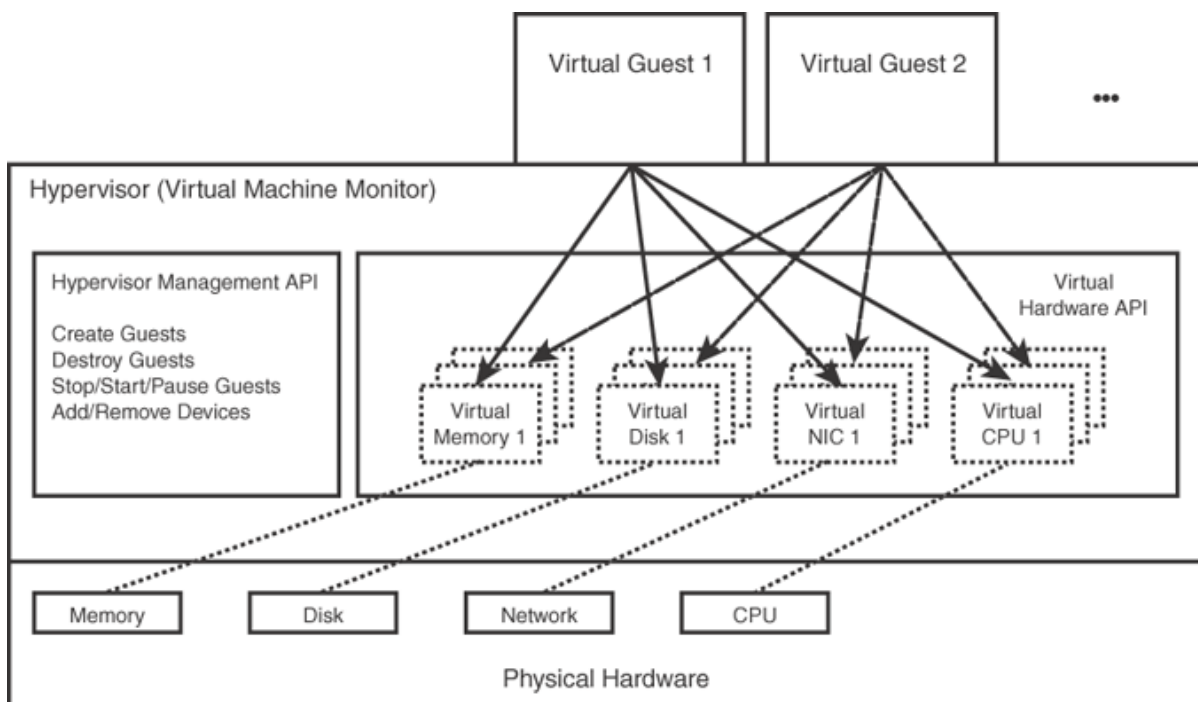
Ατυχώς, η Citrix έχει επίσης χρησιμοποιήσει το όνομα Xen και για προϊόντα που δε σχετίζονται με το Xen, όπως το XenApp και το XenDesktop.

Το Xen hypervisor βρίσκεται πάνω από το φυσικό υλικό και όπως είδαμε παραπάνω η δουλειά του είναι να παρουσιάζει στα φιλοξενούμενα domains ένα εικονικό υλικό. Με αυτόν τον τρόπο καθορίζει το εικονικό μηχάνημα που βλέπουν τα φιλοξενούμενα domains αντί για το φυσικό υλικό.

TIP

Χρησιμοποιώντας το Live CD του Xen είναι ένας πού καλός τρόπος να αρχίσεις να χρησιμοποιείς το Xen. Αν έχεις διαβάσει για το Xen αλλά δεν ξέρεις αν είναι η κατάλληλη λύση για σένα , μπορείς να το δοκιμάσεις χωρίς να αποθηκευτεί καμιά αλλαγή. Έτσι θα αποφασίσεις τελικά αν σου κάνει ή όχι πριν διαμορφώσεις και χωρίσεις το σκληρό σου δίσκο. Άπλα να έχεις στο νου σου ότι, ότι αλλαγές κάνεις δε θα αποθηκευτούν, γι αυτό μη ξοδέψεις πολύ χρόνο.

Το LiveCD πλέον δεν υποστηρίζεται για τις καινούργιες εκδόσεις του Xen. Παρόλα αυτά παραμένει ένα καλό σημείο για να αρχίσεις.



Από το 2009, οι περισσότερες διανομές του Linux έχουν συμπεριλάβει πακέτα για να επικοινωνούν με το Xen hypervisor και να ξεκινούν άλλα domains, αλλά επειδή το xen δεν έγινε δεκτό από τον κύριο πυρήνα του Linux (τελικά τα κατάφερε το Μάιο του 2013) και απαιτείται πρόσθετη εγκατάσταση πακέτων για τον πυρήνα, μερικές διανομές όπως η Red Hat Enterprise Linux 6 και η Ubuntu 8.10 απέσυραν

την υποστήριξη για το Dom0. Με την προσθήκη των πιο σημαντικών κομματιών του Xen στον πυρήνα 2.6.37 στις αρχές του 2011, πολλές διανομές έχουν ξανά υποστήριξη για το Dom0. Η έκδοση πυρήνα 3.0 υποστηρίζει και το Dom0 και DomU στον βασικό πυρήνα.

Χαρακτηριστικά του Xen:

- Μικρό αποτύπωμα και διεπαφή (περίπου 1MB σε μέγεθος). Επειδή το Xen χρησιμοποιεί μικροπυρηνική σχεδίαση, με μικρό αποτύπωμα και περιορισμένη διεπαφή στον χρήστη, είναι πιο συμπαγές και ασφαλές απ' ό,τι άλλοι hypervisor.
- Δεν επηρεάζεται από το λειτουργικό σύστημα. Τις περισσότερες φορές χρησιμοποιείται μια διανομή Linux για τον έλεγχο – Dom0, αλλά μπορεί να χρησιμοποιηθεί επίσης και το NetBSD και το OpenSolaris.
- Το Xen έχει τη δυνατότητα να επιτρέπει στους οδηγούς της κύριας συσκευής ενός συστήματος να χρησιμοποιηθούν σε ένα εικονικό μηχάνημα. Αν ο οδηγός καταρρεύσει ή χαλάσει, το εικονικό μηχάνημα που έχει αυτόν τον οδηγό μπορεί να επανεκκινηθεί χωρίς να επηρεάσει τα υπόλοιπα συστήματα.
- Υποστηρίζοντας την παραεικονικοποίηση μπορεί να υποστηρίξει εικονικά συστήματα χωρίς τη βοήθεια του υλικού.
- Υποστηρίζοντας και την πλήρη εικονικοποίηση μπορούν να χρησιμοποιηθούν αμετάβλητα λειτουργικά συστήματα που είναι ο μονός τρόπος ώστε να μπορείς να δημιουργήσεις domains με Microsoft Windows, απλά με λίγο μειωμένη απόδοση.

Που χρησιμοποιείται

Οι εταιρίες παροχής υπηρεσιών διαδικτύου χρησιμοποιούν εικονικούς σέρβερ. Οι Amazon EC2, Liquid Web, Fujitsu Global Cloud Platform, Linode, SparkNode, χρησιμοποιούν το Xen ως hypervisor.

Εκτός από τα προηγούμενα οφέλη της εικονικοποίησης που είπαμε παραπάνω το Xen προσφέρει κάποια ακόμη.

- Δυναμική ανοχή σε σφάλματα λογισμικού, μέσω γρήγορης επανεκκίνησης.
- Ξεχωρίζει την εκτέλεση του hypervisor από τη διαχείριση των λειτουργικών, της ουράς, των οδηγών των συσκευών και τους χρήστες.
- Ανοχή σε σφάλματα υλικού, μέσω της «ζωντανής Μεταφοράς». Η υπηρεσία αυτή βοηθά στην εξισορρόπηση του φόρτου και στην αποφυγή της μη διαθεσιμότητας του σέρβερ. Ο διαχειριστής μπορεί να μεταφέρει μέσω τοπικού δικτύου το εικονικό μηχάνημα χωρίς να σταματήσει αυτό να λειτουργεί. Η διαδικασία αυτή απαιτεί μια παύση περίπου 60-300ms για να πραγματοποιήσει τον τελικό συγχρονισμό πριν μπει σε λειτουργία στον καινούργιο προορισμό.
- Επιτρέπει το πάγωμα ενός εικονικού μηχανήματος και τη συνέχιση του από εκείνο το σημείο.

- Το Xen είναι λογισμικό ανοικτού κώδικα. Έτσι μπορείς να το επεκτείνεις ώστε να καλύπτει τις ανάγκες σου.
- Το Xen μπορεί να διαχειριστεί μέχρι 255 φυσικούς επεξεργαστές, 128 VCPUs ανά χρήστη και μέχρι 1 TB ram για κάθε χρήστη πλήρους εικονικοποίησης και 512 GB ram για χρήστες παραεικονικοποίησης.

Γιατί να μη διαλέξω... ?

- HyperV, Vmware ..., XenServer: Επειδή αυτά είναι εμπορικά προϊόντα και παρόλο που μπορεί να είναι χωρίς χρέωση, δεν είναι δωρεάν. Είναι άμεσα συνδεδεμένα με ιδιοκτησιακό λογισμικό . Επίσης τα προϊόντα αυτά είναι σχεδιασμένα ώστε να μη μπορείς να τα χρησιμοποιήσεις χωρίς ακριβά διαχειριστικά εργαλεία και συμβόλαια υποστήριξης.
- KVM (Kernel-based VirtualMachine): Ένας από τους κύριους αντιπάλους του Xen. Κάποιες μεγάλες εταιρίες που ήταν υποστηρικτές του Xen τώρα τάσσονται υπέρ του KVM. Όπως το Xen, το KVM υποστηρίζει πλήρη και παραεικονικοποίηση. Το μεγαλύτερο πλεονέκτημα του KVM είναι η ενσωμάτωση του στον πυρήνα Vanilla, ενώ οι πυρήνες του Xen πρέπει να είναι τροποποιημένοι. Σε σύγκριση με το Xen, το kvm προσφέρει καλύτερη απόδοση στη χρήση των συσκευών του υλικού ενώ το Xen καταφέρνει καλύτερη απόδοση του επεξεργαστή. Αντίθετα με το Xen, το kvm είναι hypervisor τύπου 2.
- Qemu: Είναι ένας εξομοιωτής , ικανός να προσομοιώνει έναν ολόκληρο υπολογιστή, ακόμα και διαφορετικές αρχιτεκτονικές αλλά με πολλές πρόσθετες εντολές. Συχνά χρησιμοποιείται μαζί με το KVM. Και το Xen κάνει χρήση του Qemu. Στους χρήστες πλήρους εικονικοποίησης, το XEN πρέπει να προμηθεύσει εικονικοποίηση συσκευών, πχ για τους σκληρούς δίσκους, κάρτες δικτύου, κάρτες γραφικών.. Αυτό επιτυγχάνεται μέσω το qemu-dm (dm-devicemanager-διαχειριστής συσκευών).

Ιστορία

Προέλευση του Xen

Η ανάπτυξη του ξεκίνησε από το εργαστηριακό τμήμα του πανεπιστημίου του Κέμπριτζ σαν μέρος του πρότζεκτ Xenoserver, το 2001. Το Xenoserver πρότζεκτ είχε σα στόχο την κατασκευή μιας δημόσιας υποδομής για μεγάλης περιοχής κατανεμημένα συστήματα πληροφορικής. Για να διασφαλιστεί ότι κάθε φυσικός κόμβος θα χρησιμοποιούταν στο έπακρο, χρειαζόταν ένας hypervisor για να φιλοξενεί πολλαπλά λειτουργικά συστήματα σε έναν σέρβερ. Γι αυτό το λόγο δημιουργήθηκε το Xen.

Για περισσότερες πληροφορίες για το πρότζεκτ Xenoserver, δες:

<http://www.cl.cam.ac.uk/research/srg/netos/xeno/>

και

(Evangelos Kotsovinos, ηγείτο το Xenoservers project)

<http://www.xenoservers.net/>

Το Xen παρουσιάστηκε για πρώτη φορά στο κοινό σε μια ακαδημαϊκή δημοσίευση το 2003 στη συνεδρίαση του the Association for Computing Machinery (ACM) Symposium on Operating Systems Principles (SOSP). Οι ισχυρισμοί για γρήγορη εικονικοποίηση συγκέντρωσε μεγάλο ενδιαφέρον από την ακαδημαϊκή κοινότητα. Σύντομα ένας μεγάλος αριθμός ανθρώπων ενδιαφέρθηκε για αυτήν την καινούργια προσέγγιση της εικονικοποίησης.

Είναι αξιοσημείωτο ότι κατά την ανάπτυξη του Xen 1.x το κέντρο ερευνών της Microsoft σε συνεργασία με το πανεπιστήμιο του Κέμπριτζ, ανέπτυξε μια έκδοση των Windows XP για το Xen. Δυστυχώς όμως λόγω των όρων άδειας του, δεν εκδόθηκε ποτέ παρόλο που αναφέρεται στη αρχική δημοσίευση στο SOSP.

Μια ξεχωριστή εταιρία, η XenSource ιδρύθηκε το 2004 για να προωθήσει την υιοθέτηση ενός hypervisor ανοικτού κώδικα στις εταιρίες. Η XenSource επικεντρώθηκε στην ανάπτυξη του ανοικτού λογισμικού πυρήνα του Xen ενώ ταυτόχρονα πουλούσε εταιρικά πακέτα και διαχειριστικό λογισμικό στους πελάτες της. Ενώ η XenSource οδηγεί και συντονίζει την ανάπτυξη, συνεισφορές έχουν γίνει και από έναν μεγάλο αριθμό εταιριών και οργανισμών όπως των IBM, Sun, HP, RedHat, Intel, AMD, SGI, ovell, the NSA, the USNavy, Samsung, Fujitsu, Qlogic και πολλών άλλων, συμπεριλαμβανομένων πολλών ακαδημαϊκών ερευνητών από πολλά πανεπιστήμια. Μαζί δημιουργούν κάποια στάνταρ που όλα τα συστήματα στο οικοσύστημα του Xen μπορούν να στηριχτούν, μειώνοντας το ρίσκο και επιταχύνοντας την ανάπτυξη για όλους τους συμμετέχοντες.

Στα τέλη του 2004, η έκδοση 2.0 του Xen ήρθε στο προσκήνιο. Η νέα έκδοση πέτυχε μεγαλύτερη ευελιξία στη ρύθμιση των συσκευών υλικού στα φιλοξενούμενα λειτουργικά συστήματα. Στην έκδοση αυτή οι χρήστες μπορούσαν να ρυθμίζουν, αυθαίρετους κανόνες στο τοίχος ασφαλείας, δρομολόγηση και γεφύρωμα των φιλοξενούμενων διεπαφών του δικτύου. Επίσης συμπεριλήφθηκε η υποστήριξη συμμετρικής πολυεπεξεργασίας, παρόλο που οι εικόνες των φιλοξενούμενων παρέμεναν σε ένα μόνο επεξεργαστή. Η πιο εντυπωσιακή βελτίωση ήταν η πρόσθεση της ζωντανής μεταφοράς.

Κατά τη διάρκεια του 2005, υπήρξε ένα αυξανόμενο ενδιαφέρον γύρω από το Xen. Αυτή η τάση υπήρξε και στη βιομηχανία και στους κλάδους των ακαδημαϊκών και στις αρχές του 2006 το Xen πέτυχε σημαντική αναγνωσιμότητα στο χώρο της εικονικοποίησης.

Το 2006, το Xen 3.0 παρουσίασε ένα αφαιρετικό στρώμα για τις τεχνολογίες εικονικοποίησης υλικού, που παρέχονταν από τις εφαρμογές Intel Vanderpool και AMD Pacifica. Αυτό έδωσε τη δυνατότητα στο να χρησιμοποιηθούν αμετάβλητα λειτουργικά παράλληλα με τους φιλοξενούμενους της παραεικονικοποίησης. Άλλα αξιοσημείωτα χαρακτηριστικά ήταν η βελτίωση της απόδοσης του δικτύου από την κατάτμηση των πακέτων φόρτου εργασίας, ενισχυμένη υποστήριξη IA64 και την αρχική θύρα στην αρχιτεκτονική Power processor.

Το 2007 η Citrix απέκτησε τη XenSource έναντι του ποσού των 500 εκατομμυρίων δολαρίων και η ομάδα της έγινε η Citrix XenServer ProductGroup. Επίσης το 2007 κυκλοφόρησε το xen 3.1. Παρείχε υποστήριξη για XenAPI, μια προγραμματιστική διεπαφή στις εντολές του Xen που επέτρεπε την ενσωμάτωση εργαλείων διαχείρισης και από άλλες εταιρίες. Επίσης έδωσε τη δυνατότητα για την αποθήκευση/αποκατάσταση/μεταφορά και το δυναμικό έλεγχο μνήμης των φιλοξενούμενων πλήρους εικονικοποίησης.

Η έκδοση xen 4.0.0 κυκλοφόρησε στις 7 Απριλίου του 2010.

Μερικά από τα χαρακτηριστικά που εισήγαγε αυτή η νέα έκδοση, είναι:

- Καλύτερη απόδοση και επεκτασιμότητα: 128 vcpus για κάθε χρήστη, 1 TBRAM για κάθε χρήστη, μέχρι 1 TBRAM για κάθε χρήστη πλήρους εικονικοποίησης ή 512 GBRAM για κάθε χρήστη παραεικονικοποίησης και 128 φυσικούς επεξεργαστές για κάθε χρήστη.
- Βελτιωμένο πέρασμα του δίαυλου PCI χρησιμοποιώντας τεχνικές επιταχυνόμενης εικονικοποίησης υλικού (Intel VT-d and AMD IOMMU).
- Πέρασμα της κύριας κάρτας γραφικών σε ένα χρήστη για υψηλής ταχύτητας γραφικών 3D και βίντεο επιταχυνόμενο από το υλικό.
- Το TMem που επιτρέπει τη βελτιωμένη χρησιμοποίηση της μη χρησιμοποιούμενης μνήμης για τους φιλοξενούμενους παραεικονικοποίησης. Για περισσότερες πληροφορίες: <http://oss.oracle.com/projects/tmem/>

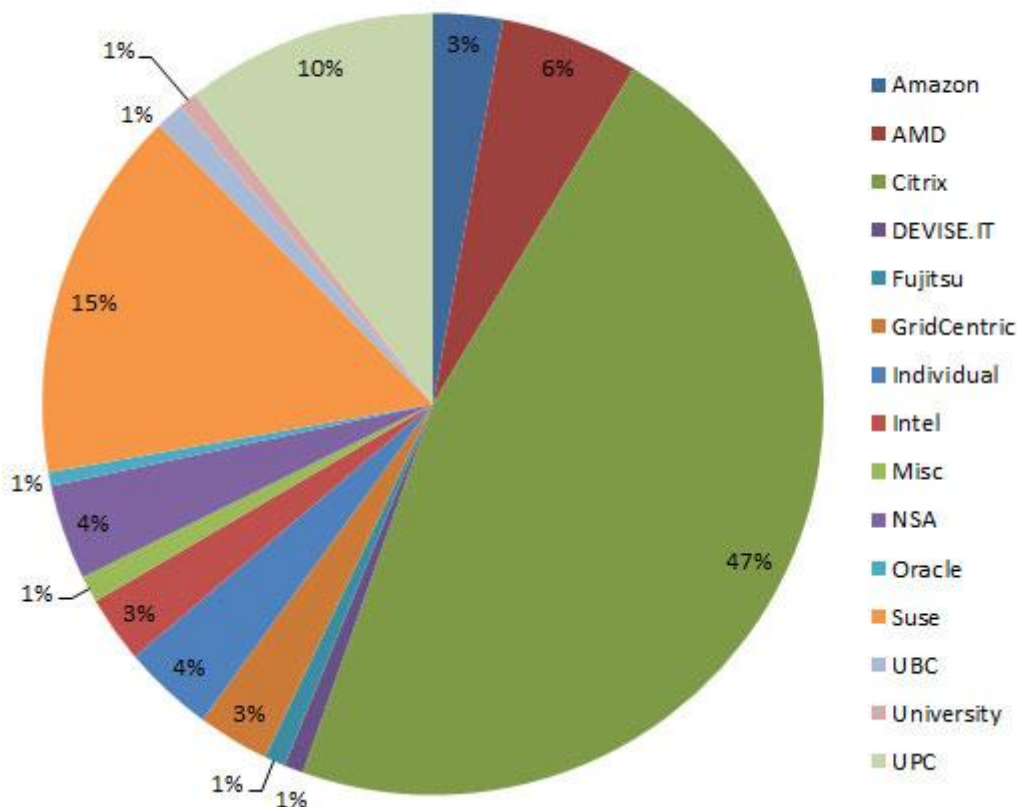
- Χρήση του πυρήνα Dom0 2.6.32 σαν προεπιλογή. Μπορείς να χρησιμοποιήσεις ακόμα τον πυρήνα 2.6.18 αν επιθυμείς.
- Αλλαγή μεγέθους στο χώρο δίσκου των φιλοξενούμενων χωρίς κλείσιμο/επανεκκίνηση
- Άμεση σύνδεση φυσικών επεξεργαστών/μνήμης.
- Παραεικονικό πέρασμα USB για φιλοξενούμενους πλήρους και παρά - εικονικοποίησης.
- Υποστήριξη για χρήστες παραεικονικοποίησης χρησιμοποιώντας GRUB2.
- Υποστήριξη για χρήστες /boot με σύστημα αρχείων ext4

Το 2013 το Xen έγινε μέρος του Οργανισμού Λίνουξ σαν ένα πρότζεκτ συνεργασίας του Οργανισμού ([τι σημαίνει αυτό](#)). Η ιστοσελίδα της κοινότητας μεταφέρθηκε σε ένα πιο μοντέρνο χώρο [XenProject.org](#) και το νέο εμπορικό σήμα "XenProject" εκδόθηκε στον Οργανισμό Λίνουξ.

Συνεισφορές στο Xen Hypervisor 4.x

Το παρακάτω διάγραμμα δείχνει τους οργανισμούς που συνεισέφεραν παραπάνω από 1% σε γραμμές κώδικα στην έκδοση 4.x. Το individual δείχνει τη συνεισφορά από ιδιώτες. Το Misc καλύπτει την προσφορά από εμπορικούς οργανισμούς που η συνεισφορά τους δεν ήταν πάνω από 1% και το University καλύπτει την προσφορά των Πανεπιστημίων.

Contributions to Xen in Lines of Code



TIP

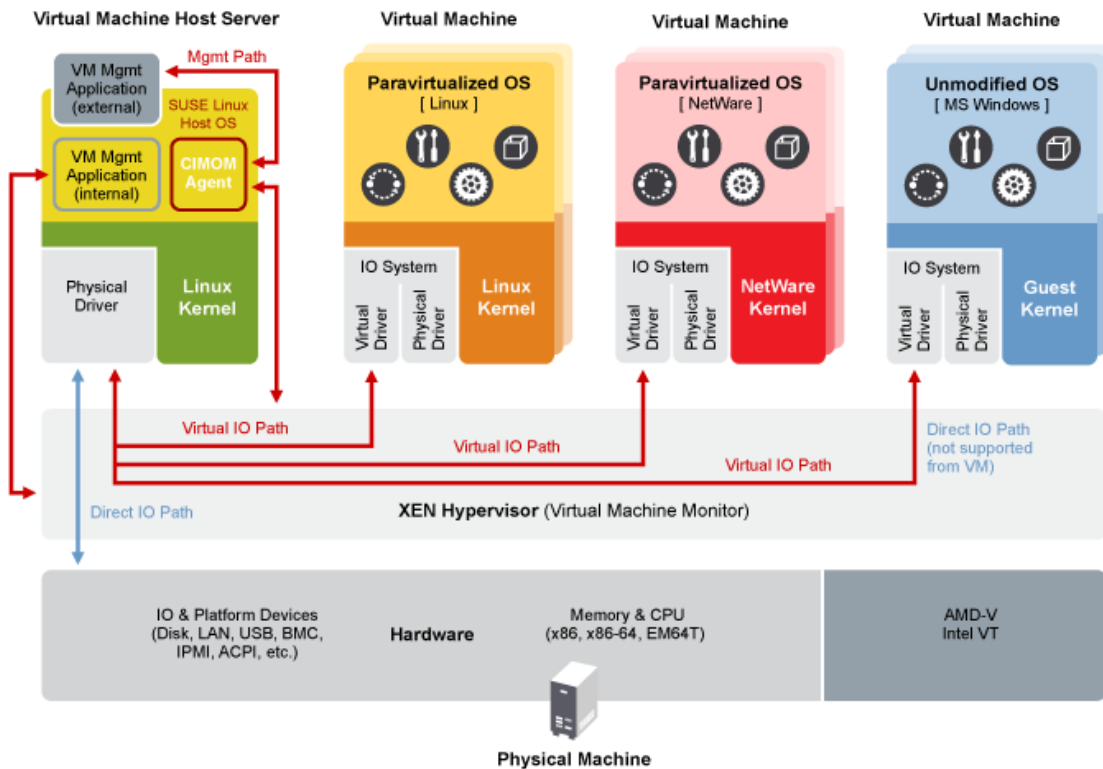
Αν ενδιαφέρεσαι να δεις τον κώδικα του Xen ή να αποκτήσεις τον κώδικα της υπό κατασκευή έκδοσης του Xen επισκέψου το:

<http://xenbits.xensource.com>.

Παρόλο την πρωτοπορία και το πολύ καλό προϊόν που προσέφερε, το Xenproject εφτασε πολύ κοντά στον αφανισμό. Μάλιστα υπήρξε ο κίνδυνος το 2011 να εγκαταλειφθεί από την κοινότητα FOSS κάτι που θα δήλωνε ουσιαστικά και το τέλος του. Μεγάλο ρόλο σε αυτό έπαιξε και η πώληση της τεχνολογίας στη Citrix. Σε συναντήσεις της FOSS αυτό που ακουγόταν για το Xen ήταν: «Το Xen πέθανε, σωστά?», «Το Xen δεν είναι δωρεάν.», «Υπάρχουν ακόμη άνθρωποι που χρησιμοποιούν το Xen?». Μάλιστα σε [αυτό](#) αναφέρθηκε και ο Evangelist Russell Pavnicek (αν και υπάλληλος της Citrix) σε μια παρουσίαση στο Texas Linux Fest 2013 στο Όστιν του Τέξας.

Αρχιτεκτονική του Xen

Η παρακάτω φωτογραφία απεικονίζει ένα μηχάνημα με 4 εικονικά μηχανήματα. Το 1^ο αριστερά είναι το Dom0 και φαίνεται να τρέχει απευθείας στο φυσικό υλικό. Το domain έλεγχου είναι επίσης απλά ένα εικονικό μηχάνημα παρόλο που έχει διάφορα πρόσθετα καθήκοντα διαχείρισης σε σύγκριση με τα υπόλοιπα.



Τα δυο εικονικά μηχανήματα στη μέση χρησιμοποιούν παραεικονικοποίηση ενώ το τέρμα δεξιά πλήρης εικονικοποίηση.

Domain 0 and DomU

Ο hypervisor δεν είναι μόνος του στο έργο της διαχείρισης των φιλοξενούμενων domain στο σύστημα. Ένα ειδικά προνομιούχο domain, το ονομαζόμενο Domain0 – Dom0, λειτουργεί ως μια διαχειριστική διεπαφή με το Xen. Σε ένα σύστημα Xen το Dom0 προσθέτει κάποια λειτουργικότητα που μπορεί να περιγραφεί σαν μια λογική συνάρτηση του hypervisor. Αυτό επιτρέπει το Xen να είναι ένα λεπτό στρώμα.

Το Dom0 είναι το πρώτο domain που ξεκινάει όταν το σύστημα εκκινεί και χρησιμοποιείται για την κατασκευή και τη ρύθμιση των άλλων φιλοξενούμενων domain τα οποία ονομάζονται DomU ή μη προνομιούχα domain.

Το Dom0 έχει απευθείας πρόσβαση σε όλο το υλικό και το εξάγει στα υπόλοιπα domains σαν συσκευές απλοποιημένης γενικής κλάσης. Αντί να εξομοιώνει τις πραγματικές φυσικές συσκευές όπως είναι, τις παρουσιάζει σαν εξιδανικευμένες συσκευές. Για παράδειγμα, ένα DomU βλέπει την κάρτα δικτύου σαν μια γενικής κλάσης συσκευή δικτύου ή το δίσκο σα μια γενική συσκευή από μπλοκς. Το Dom0 έχει έναν οδηγό για τη συσκευή συγκεκριμένο για κάθε φυσική συσκευή και μετά επικοινωνεί με τα άλλα domains μέσω μιας μεταφοράς ασύγχρονης κοινόχρηστης μνήμης.

Το Dom0 μπορεί επίσης να αναθέσει την ευθύνη για μια συγκεκριμένη συσκευή σε ένα άλλο domain. Ένα domain οδηγός είναι ένα DomU που τρέχει τον μικρότερο δυνατό πυρήνα και το σύστημα υποστήριξης για τη συγκεκριμένη συσκευή. Αυτό έχει το πλεονέκτημα ότι μεταφέρεται η πολυπλοκότητα και ο κίνδυνος της διαχείρισης συσκευών έξω από το Dom0. Μετακινώντας τη διαχείριση συσκευών έξω από το Dom0 κάνει το σύστημα πιο σταθερό επειδή οι οδηγοί του υλικού είναι μακρά οι πιο επιρρεπείς για λάθος σε ένα λειτουργικό σύστημα. Ένα domain οδηγός μπορείς να το σταματήσεις και να το επανεκκινήσεις χωρίς να διαταραχθεί το υπόλοιπο σύστημα.

Ο οδηγός της συσκευής που έχει ο Dom0 ή ένα domain οδηγός ονομάζεται ως σύστημα υποστήριξης και κάθε domain με πρόσβαση στη συσκευή τρέχει έναν οδηγό γενικής συσκευής. Το σύστημα υποστήριξης δίνει την ψευδαίσθηση σε κάθε domain ότι έχει μια γενική συσκευή αφιερωμένη σε αυτό.

Γενικά ένα μικρότερο και απλούστερο Dom0 είναι καλύτερο για την ασφάλεια και τη σταθερότητα του συστήματος. συνιστάται ιδιαίτερα να περιορίσεις των κώδικα επιπέδου χρήστη που εγκαθίσταται και τρέχει στο Dom0 σε εφαρμογές που είναι απολύτως απαραίτητες για τη διαχείριση των εικονικών μηχανημάτων του συστήματος. Όλες οι άλλες εφαρμογές μπορούν να εγκατασταθούν στα φιλοξενούμενα domains. Επίσης είναι καλό να κλείσεις όλες τις μη χρησιμοποιούμενες θύρες δικτύου. Γενικά ένα ελαφρύ Dom0 είναι λιγότερο ευάλωτες στις επιθέσεις ή σε σφάλματα λογισμικού.

Πως λειτουργεί το Xen

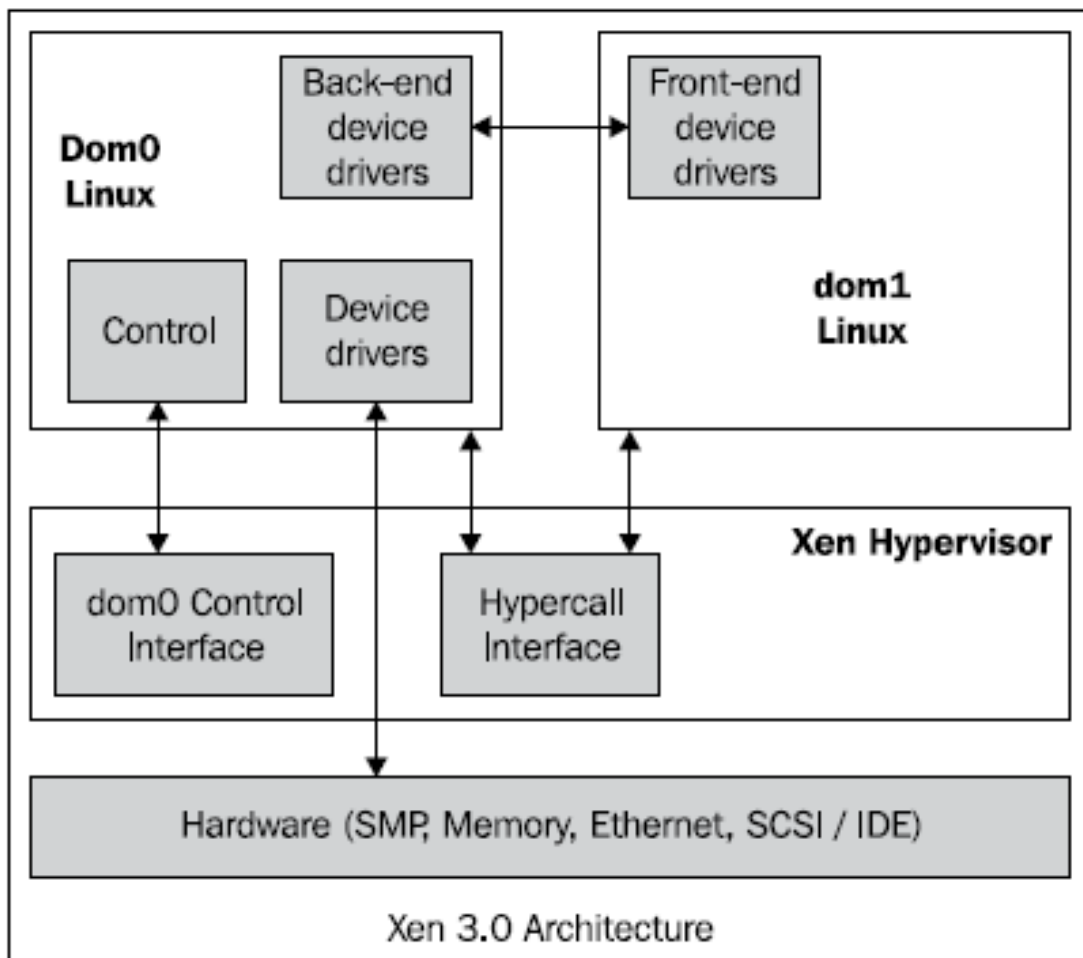
Το Xen αναφέρεται σε κάθε εικονικό μηχανήμα που τρέχει σε ένα σύστημα ως ένα domain. Όταν εκκινήσει ο Xen σέρβερ, πρώτα ξεκινάει τον hypervisor, που είναι υπεύθυνος για να εκκινήσει το domain που ονομάζεται Domain0 (Dom0) και στο οποίο τρέχει το λειτουργικό σύστημα. Το Dom0 είναι ένα προνομιούχο domain που έχει πρόσβαση στους υλικούς πόρους του σέρβερ και επίσης περιέχει

εργαλεία για τη διαχείριση όλων των άλλων domain. Ο hypervisor επίσης ελέγχει τους πίνακες σελίδων (pagetables) και είναι υπεύθυνος για την εύρεση πόρων για τα καινούργια domains. Κάθε domain αλληλεπιδρά με τον hypervisor στέλνοντας του μια υπερκλήση. Ο hypervisor απαντά σε αυτήν την υπερκλήση στέλνοντας ένα γεγονός στο domain. Νέα στιγμιότυπα μη προνομιούχων domain δημιουργούνται χρησιμοποιώντας εργαλεία που είναι διαθέσιμα στο Dom0. Αυτά τα εργαλεία με τη σειρά τους κάνουν κλήσεις στη διεπαφή ελέγχου API στον hypervisor για να εκτελέσουν τις ζητηθείσες λειτουργίες. Αυτά τα μη προνομιούχα domains αναφέρονται ως DomU.

Για παράδειγμα αν έχεις ένα Xen σέρβερ με τρεις επισκέπτες, θα έχεις σαν domains το Dom0, dom1, dom2. Το Dom0 είναι πανομοιότυπο με τα άλλα domains, αλλά θα έχει πρόσβαση στο φυσικό υλικό του σέρβερ.

Όλα τα αιτήματα από τα DomU για πρόσβαση στο υλικό γίνονται με βοηθητικούς οδηγούς συσκευών στο Dom0 και αυτό με τη σειρά του θα τα περάσει στο πραγματικό υλικό. Αυτό είναι δυνατόν επειδή τα λειτουργικά συστήματα γνωρίζουν ότι τρέχουν στο Xen hypervisor.

Η αρχιτεκτονική ενός συστήματος Xen με δύο domains είναι η ακόλουθη:



Μια προνομιούχος θέση

Η δουλειά του hypervisor είναι να διανέμει τους πόρους στα DomUs, να προστατεύει τα φιλοξενούμενα συστήματα το ένα από το άλλο, να παρέχει καθарές και φορητές διεπιφάνειες για τις συσκευές. Για να τα καταφέρει όλα αυτά πρέπει να καταλαμβάνει μια προνομιούχα θέση στο σύστημα.

Σε ένα παραδοσιακό, μη εικονικοποιημένο σύστημα, το λειτουργικό πρέπει να καταλαμβάνει προνομιούχα θέση σε σχέση με τις εφαρμογές σε επίπεδο χρήστη. Για να το καταφέρει αυτό, οι περισσότερες αρχιτεκτονικές επεξεργαστών έχουν τουλάχιστον 2 επίπεδα. Το λειτουργικό τρέχει σε επίπεδο υψηλότερης προτεραιότητας απ ότι ο κώδικας σε επίπεδο χρήστη, επιτρέποντας το έτσι να αναγκάζει τις εφαρμογές επιπέδου χρήστη να ακολουθούν τους κανόνες του.

Σε ένα εικονικοποιημένο σύστημα, έχουμε τον hypervisor, τα φιλοξενούμενα λειτουργικά και τις εφαρμογές επιπέδου χρήστη και όπως το λειτουργικό σύστημα «δαιτητεύει» ανάμεσα στις διάφορες εφαρμογές επιπέδου χρήστη, ο hypervisor δαιτητεύει μεταξύ των λειτουργικών συστημάτων. Γι αυτό το λόγο ο Xen hypervisor πρέπει να τρέχει σε ένα υψηλότερο, πιο προνομιούχο επίπεδο απ ότι τα υπόλοιπα λειτουργικά. Ταυτόχρονα όμως θέλουμε μέσα στα φιλοξενούμενα συστήματα τα λειτουργικά να τρέχουν σε υψηλότερο, πιο προνομιούχο επίπεδο σε σχέση με τις εφαρμογές επιπέδου χρήστη.

Ευτυχώς, όπως είδαμε και παραπάνω, η αρχιτεκτονική x86 παρέχει περισσότερα από δυο επίπεδα. Για την ακρίβεια έχει 4 επίπεδα τα οποία ονομάζονται δακτύλιοι προστασίας. Ο δακτύλιος 0 είναι ο πιο προνομιούχος και ο δακτύλιος 3 ο λιγότερο προνομιούχος. Σε ένα παραδοσιακό σύστημα το λειτουργικό εκτελείται στο δακτύλιο 0 και οι εφαρμογές επιπέδου χρήστη στο δακτύλιο 3 με τους δακτυλίους 1,2 να μένουν ανεκμετάλλευτοι. Σε ένα σύστημα Xen, ο hypervisor εκτελείται στο δακτύλιο 0, τα φιλοξενούμενα λειτουργικά στο δακτύλιο 1 και οι εφαρμογές επιπέδου χρήστη παραμένουν στο δακτύλιο 3.

Αυτοί οι δακτύλιοι προστασίας, δίνουν στον hypervisor την δύναμη που χρειάζεται για να εφαρμόσει το διαμοιρασμό και την απομόνωση ανάμεσα στα φιλοξενούμενα domains. Στο δακτύλιο 0 ο hypervisor έχει πλήρη έλεγχο του φυσικού υλικού. Τα φιλοξενούμενα domains έχουν τόση πρόσβαση στο φυσικό υλικό όση τους επιτρέπει ο hypervisor. Όσους πόρους χρειάζεται ένα φιλοξενούμενο domain, του χορηγούνται από τον hypervisor.

Τα φιλοξενούμενα domains στέλνουν αιτήματα στον hypervisor μέσω ενός σετ υπερκλήσεων, όπως κάνουν οι εφαρμογές επιπέδου χρήστη στο λειτουργικό σύστημα με ένα σετ κλήσεων. Στην αρχιτεκτονική x86, οι κλήσεις τυπικά γίνονται με μία οδηγία διακοπής του λογισμικού, int 0x80. Με τον ίδιο τρόπο οι υπερκλήσεις γίνονται με την οδηγία int 0x82. Ο hypervisor άπαντα στις αιτήσεις αυτές στέλνοντας στο domain ένα ασύγχρονο γεγονός όπως είναι ένα σήμα UNIX ή μια διακοπή στο φυσικό υλικό.

Με την υποστήριξη υλικού για εικονικοποίηση, όπως έχει η Intel τις επεκτάσεις VT-x και η AMD το AMD-V, αυτοί οι πρόσθετοι δακτύλιοι, γίνονται λιγότερο κρίσιμοι. Οι επεκτάσεις αυτές, παρέχουν λειτουργίες ριζας και οχι-ριζας (root and nonroot) που η καθεμιά έχει δακτυλίους από το 0 μέχρι το 3. Ο xen hypervisor μπορεί να

τρέχει σε λειτουργία ρίζας, ενώ τα φιλοξενούμενα λειτουργικά σε λειτουργία οχι-ρίζας στο δακτύλιο που ήταν σχεδιασμένα αρχικά να τρέχουν.

Τύποι εικονικοποίησης στο Xen

Το Xen υποστηρίζει δυο διαφορετικές προσεγγίσεις στο να τρέχει τα φιλοξενούμενα λειτουργικά συστήματα.

Παρά-εικονικοποίηση

Το Xen υποστηρίζει την παραεικονικοποίηση, στην οποία οι χρήστες τρέχουν ένα ειδικά διαμορφωμένο λειτουργικό σύστημα. Οι χρήστες είναι διαμορφωμένοι ώστε να τρέχουν μια ειδική υπερκλήση [ABI](#) αντί για κάποιες άλλες συγκεκριμένες λειτουργίες.

Μέσω της παραεικονικοποίησης, το Xen μπορεί να επιτύχει υψηλότερη απόδοση, ακόμα και στην αρχιτεκτονική x86, που φημίζεται για την όχι καλή συνεργασία με παραδοσιακές τεχνικές εικονικοποίησης.

Το Xen μπορεί να τρέξει παραεικονικοποιημένους φιλοξενούμενους ακόμη και σε επεξεργαστές που δε διαθέτουν υποστήριξη εικονικοποίησης .

Επίσης με την παραεικονικοποίηση δεν υπάρχει η ανάγκη για πλήρη εξομοίωση των υπηρεσιών και οδηγών υλικού κάτι που κάνει το σύστημα απλούστερο στη διαχείριση και μειώνει τον κίνδυνο να εκτεθεί το πραγματικό υλικό σε κακόβουλες επιθέσεις.

Ο πυρήνας του Xen τρέχει στο δακτύλιο 0, ενώ τα φιλοξενούμενα domains τρέχουν στο δακτύλιο 1 (πυρήνας) και στον πυρήνα 3 (εφαρμογές).

- Χρησιμοποιεί έναν αλλαγμένο πυρήνα Linux
- Οι φιλοξενούμενοι φορτώνουν το pygrub του Dom0 ή τον πυρήνα του Dom0
- Η διαχείριση των οδηγών συσκευών γίνεται από το Dom0
- Δε γίνεται να εγκατασταθούν με αυτόν τον τρόπο λειτουργικά των Windows
- Ο φιλοξενούμενοι γνωρίζουν ότι είναι εικονικά μηχανήματα και μιλάνε απευθείας με το hypervisor

Πληρης εικονικοποίηση – Υποβοηθούμενη από το υλικό:

Η υποβοηθούμενη από το υλικό εικονικοποίηση (hardware-assisted virtualization – HVM), επιτρέπει στην υποβοηθούμενη από το υλικό εικονικοποίηση, τη δημιουργία μη τροποποιημένων φιλοξενούμενων, συμπεριλαμβανομένων και κλειστών λειτουργικών συστημάτων όπως τα Windows.

Οι επεκτάσεις HVM παρέχουν επιπρόσθετες λειτουργίες εκτέλεσης, με μια σημαντική διαφοροποίηση μεταξύ των πιο προνομιούχων λειτουργιών που χρησιμοποιούνται από τον hypervisor για την πρόσβαση στο κανονικό υλικό (αυτό που καλείται «λειτουργία ρίζας» στην αρχιτεκτονική x86) και στις λιγότερο προνομιούχες λειτουργίες που χρησιμοποιούνται από τους πυρήνες των φιλοξενούμενων και τις εφαρμογές που έχουν πρόσβαση στο υλικό κάτω από τον απόλυτο έλεγχο του hypervisor (στην x86 αρχιτεκτονική γνωστό και ως λειτουργία μη ρίζας - non-root mode).

Η Intel και η AMD έχουν συνεισφέρει με αλλαγές στο XEN ώστε να υποστηρίζει τις επεκτάσεις αρχιτεκτονικής τους, IntelVT-x και AMD-V. Η υποστήριξη της αρχιτεκτονικής ARMv7A είναι υπό κατασκευή (κατά της διάρκειας της συγγραφής ανακοινώθηκε η έκδοση μιας δοκιμαστικής έκδοσης, με ενθαρρυντικά αποτελέσματα).

Οι επεκτάσεις HVM προσφέρουν επίσης νέες οδηγίες για να υποστηρίξουν απευθείας κλήσεις από ένα παραεικονικό φιλοξενούμενο/οδηγό στον hypervisor. Αυτό επιτρέπει στους φιλοξενούμενους πλήρους εικονικοποίησης με μικρές αλλά κατάλληλες αλλαγές να αποκτήσουν πολλά από τα πλεονεκτήματα της απόδοσης της παραεικονικοποίησης.

Μέχρι την έκδοση 4.2 του Xen μόνο οι πλήρως εικονικοποιημένοι φιλοξενούμενοι μπορούν να κάνουν χρήση της υποστήριξης υλικού για πολλαπλά ανεξάρτητα επίπεδα της προστασίας μνήμης και σελιδοποίησης. Σαν αποτέλεσμα, κάποιες φορές, σε σχέση με το φόρτο εργασίας, οι χρήστες HVM με οδηγούς PV (γνωστό και ως PV-on-HVM) έχουν καλύτερη απόδοση από τους απλούς χρήστες PV.

Η HVM στο Xen χρησιμοποιεί προσομοίωση συσκευών που βασίζεται στο πρότζεκτ QEMU για να παρέχει εικονικοποίηση συσκευών στα εικονικά μηχανήματα. Το σύστημα προσομοιώνει το υλικό μέσω ενός QEMU διαχειριστή υλικού που τρέχει σαν σύστημα υποστήριξης στο Dom0. Αυτό σημαίνει ότι τα εικονικά μηχανήματα βλέπουν μια προσομοιωμένη έκδοση ενός βασικού υπολογιστή. Σε ένα περιβάλλον όπου η απόδοση είναι το ζητούμενο, οι οδηγοί PV-on-HVM δισκου και δικτύου χρησιμοποιούνται κανονικά κατά τη λειτουργία και το προσομοιωμένο υλικό χρησιμοποιείται κυρίως κατά την εκκίνηση.

Υπάρχουν βέβαια συζητήσεις στο κατά πόσο η παραεικονικοποίηση είναι ξεπερασμένη ή όχι. Η HVM παράγει μια επιβάρυνση, ειδικά χωρίς τους ειδικά τροποποιημένους οδηγούς στο φιλοξενούμενο λειτουργικό.

Η επιβάρυνση στον επεξεργαστή ποικίλει, αλλά ανέρχεται περίπου στο 10-15% που υστερεί σε σύγκριση με το 2-3% που έχει η παραεικονικοποίηση.

Υποστήριξη Επεξεργαστών

Αρχικά πρέπει να εξακριβωθεί αν ο επεξεργαστής υποστηρίζει την πλήρη εικονικοποίηση.

Από την έκδοση 3.0 το Xen υποστηρίζει και τις τεχνολογίες IntelVT-x και AMD-V. Ένα από τα πλεονεκτήματα τους είναι η χρησιμοποίηση ενός buffer μετάφρασης ετικετών (tagged translation lookaside buffer - TLB). Χρησιμοποιώντας αυτές τις ετικέτες TLB οι χρήστες χαρτογραφούνται σε ένα χώρο διεύθυνσης που μπορεί να είναι διαφορετική από όπου βρίσκεται το εικονικό μηχάνημα. Η ετικέτα περιλαμβάνει επιπρόσθετες πληροφορίες γνώστες και ως αναγνωριστικά χώρου διεύθυνσης (address space identifiers - ASIDs). Οι ASIDs εξασφαλίζουν ότι δε χρειάζεται καθάρισμα της TLB σε κάθε πλαίσιο αλλαγής.

Για τα συστήματα AMD, πρέπει να υπάρχει υποστήριξη 'svm'. Για να το ελέγξεις αυτό χρησιμοποίησε την εντολή: `# grep svm /proc/cpuinfo`

Για τα συστήματα INTEL, πρέπει να υπάρχει υποστήριξη 'vmx'. Για να το ελέγξεις αυτό χρησιμοποίησε την παρακάτω εντολή: `# grep vmx /proc/cpuinfo`

Επίσης η Intel έχει μια λίστα διαθέσιμη που μπορείς να δεις αν ένας επεξεργαστής υποστηρίζεται: [supported cpus](#)

Αν αυτές οι εντολές δεν επιστρέψουν τίποτα, τότε έλεγξε αν έχει ενεργοποιηθεί η συγκεκριμένη ρύθμιση εικονικοποίησης στο BIOS (από προεπιλογή είναι στο off). Αν γνωρίζεις ότι ο επεξεργαστής υποστηρίζει την πλήρη εικονικοποίηση ίσως χρειάζεται αναβάθμιση τοBIOS.

Δίσκοι και ελεγκτές – Discs and Controllers

Το Dom0 του Xen μοιάζει με τα άλλα λειτουργικά συστήματα, στο ότι χρειάζεται κάποιο συγκεκριμένο-σταθερό χώρο για να συγκρατεί τα δεδομένα του. Συνήθως σε ένα προσωπικό υπολογιστή ή έναν μικρό server ο σταθερός χώρος αποθήκευσης προέρχεται από έναν τοπικό δίσκο στον υπολογιστή. Το Xen, όπως και τα υπόλοιπα λειτουργικά λειτουργεί καλύτερα με έναν τοπικό δίσκο. Κάποιες λειτουργίες του Xen, όπως η ζωντανή μεταφορά (live migration), απαιτούν αποθηκευτικό χώρο βασισμένο σε δίκτυο. Η λειτουργία αυτή όμως είναι για τα DomUs, διότι ένα Dom0 δεν μπορεί να μεταφερθεί, καθώς πρέπει να παραμείνει στο υλικό στο οποίο στήθηκε. Παρόλο που μπορείς να στήσεις ένα Dom0 που να χρησιμοποιεί αποθηκευτικό χώρο βασισμένο σε δίκτυο, προτείνεται τουλάχιστον το Dom0 να βρίσκεται σε τοπικό δίκτυο.

Το κυριότερο πράγμα που πρέπει να σκεφτούμε όταν διαλέγουμε έναν τοπικό δίσκο, είναι ότι πολλοί χρήστες θα προσπαθούν να αποκτήσουν πρόσβαση στον δίσκο την ίδια στιγμή. Το καλύτερο θα ήταν να μπορείς να αφιερώσεις έναν δίσκο

ή συστοιχία δίσκων και έναν ελεγκτή σε κάθε εικονικό μηχάνημα. Κάτι τέτοιο όμως είναι μια πολυτέλεια που οι περισσότερες εφαρμογές του Xen δεν θα έχουν. Έτσι γίνεται ξεκάθαρο ότι πρέπει να χρησιμοποιήσεις δίσκους με χαμηλή καθυστέρηση και υψηλές ταχύτητες, καθώς προσφέρουν καλύτερη απόδοση.

Το καλύτερο θα ήταν να χρησιμοποιούνται SATA δίσκοι σε Raid ή ακόμη καλύτερα υβριδικοί. Επίσης πρέπει να ρίξουμε ένα βλέμμα και σε δίσκους SSD που προσφέρουν πολύ μεγαλύτερες ταχύτητες αλλά έχουν πολύ υψηλό κόστος ακόμη σε αναλογία με τα gigabytes που προσφέρουν.

Το Xen υποστηρίζει την Αποθήκευση Βασισμένη σε Δίκτυο (Network Attached Storage - NAS). Για την ακρίβεια μπορεί να υποστηρίξει οποιοδήποτε εξωτερικό αφιερωμένο “κουτί” δικτύου που μπορείς να “κουμπώσεις” στο σύστημα σου, κάτω από την ομπρέλα του NAS. Προτείνεται όμως γενικότερα να διαλέξεις πρωτόκολλα βασισμένα σε blocks.

Συσκευές Δικτύου –Networking Devices

Γενικά η υποστήριξη δικτύου στο Xen, βασίζεται στους οδηγούς που βρίσκονται στο Dom0. Στην πραγματικότητα, το μόνο για το οποίο πρέπει να ανησυχείς είναι ότι το Dom0 έχει τους σωστούς drivers. Αν ένα DomU αναγνωρίσει μια συσκευή δικτύου, δεν θα έχεις πρόβλημα να την χρησιμοποιήσεις. Επιπρόσθετα όμως θα πρέπει να επιβεβαιώσεις ότι ο πυρήνας του Dom0 περιλαμβάνει υποστήριξη για γεφύρωση δικτύου και loopback, αν επιθυμείς τα DomU να έχουν συσκευές dev/ethX devices. Αν ο πυρήνας του οικοδεσπότη δεν υποστηρίζει γεφύρωση ή κάτι τέτοιο δεν είναι στα μέτρα σου, τότε μπορείς να χρησιμοποιήσεις δρομολόγηση IP στο Dom0. Η τεχνική αυτή μπορεί να είναι χρήσιμη αν θέλεις να απομονώσεις DomUs από κάποιο εξωτερικό δίκτυο.

Συσκευές Γραφικών

Παραδοσιακά τα συστήματα εικονικοποίησης δεν ασχολούνται με την εικονικοποίηση των συσκευών γραφικών. Η πρόσβαση στους φιλοξενούμενους συνήθως γίνεται μέσω ενός δικτυακού πρωτοκόλλου γραφικών ή ενός πρωτοκόλλου απομακρυσμένης σύνδεσης, από έναν απομακρυσμένο υπολογιστή που ενεργεί σαν viewer. Παλιότερες εκδόσεις του Xen απλά προσομοίωναν τη VGA μέσω ενός πρωτοκόλλου Εικονικού Δικτύου Computing (Virtual Network Computing - VNC) backend. Αυτό έδινε στους χρήστες πρόσβαση σε γραφικά περιβάλλοντα, αλλά αυτό δεν είναι απαραίτητα ούτε ο καλύτερος ούτε ο γρηγορότερος τρόπος για πρόσβαση σε ένα σύστημα.

Οι developers του Xen συνειδητοποίησαν ότι δεν μπορούσαν να βασίζονται τέτοια πρωτόκολλα και από την έκδοση 3.0.4, οι χρήστες μπορούν να χρησιμοποιούν

κάτι που ονομάζεται virtual frame buffer, χρησιμοποιώντας αυτό, το Dom0 μπορεί να αιχμαλωτίσει δεδομένα βίντεο γραμμένα σε συγκεκριμένο μέρος μνήμης και μετά να το στείλει στην οθόνη. Επίσης υπάρχει και η τεχνική VGA passthrough όπου η διαχείριση μιας κάρτας γραφικών μπορεί να περάσει ολόκληρη σε ένα DomU.

Διαχείριση ενέργειας

Μέχρι τώρα, δεν υπάρχει καμία ανεπτυγμένη μορφή διαχείρισης ενέργειας στον hypervisor ή στο Dom0. Το να εκτελέσεις το Xen σε ένα laptop είναι σίγουρα κάτι το πιθανό, αλλά δεν προτείνεται εξαιτίας της μείωσης της ζωής της μπαταρίας.

Σε μερικές περιπτώσεις έχει επίσης παρατηρηθεί αύξηση της θερμοκρασίας του επεξεργαστή. Έτσι αν τρέχεις το Xen σε ένα περιορισμένο χώρο ή έχεις ήδη ένα σύστημα που ζεσταίνεται αρκετά, το Xen δεν είναι μάλλον η κατάλληλη επιλογή αυτή τη στιγμή. Πρέπει όμως να σημειωθεί ότι σε συστήματα server με σωστή ψύξη όπως τα συστήματα blade, το Xen προσφέρει μεγάλο κέρδος από την μείωση ενέργειας λόγω της ενοποίησης που προσφέρει.

Απαιτήσεις Μνήμης

Οι πραγματικές απαιτήσεις μνήμης που χρειάζεται μια υλοποίηση διαφέρουν ανάλογα με το για τί πρόκειται να χρησιμοποιηθεί. Για παράδειγμα αν θέλεις να στήσεις ένα σύστημα με λειτουργικά Linux οι απαιτήσεις θα είναι πολύ μικρότερες απ' ότι αν χρησιμοποιούσες λειτουργικά windows. Μια καλή τεχνική είναι, αν γνωρίζεις τι συστήματα θα χρησιμοποιήσεις, να προσθέσεις τις ελάχιστες απαιτήσεις μνήμης που δίνει ο κατασκευαστής και να το πολλαπλασιάσεις επί 1.5 ή 2. Επίσης δεν πρέπει να ξεχάσεις να προσθέσεις τις απαιτήσεις μνήμης που έχει το Dom0 και ο πυρήνας. Η τεχνική αυτή συνήθως αποδίδει γιατί μπορεί κάποια συστήματα να χρησιμοποιούν πολύ μνήμη για τις διεργασίες τους ενώ άλλα λιγότερη και από την προτεινόμενη. Επίσης εξασφαλίζει και κάποια περίσσεια μνήμης για κάποιες εξειδικευμένες λειτουργίες του Xen, όπως η ζωντανή μεταφορά.

Το Dom0, έχει αποδειχθεί ότι μπορεί να τρέξει με minimum 64 mb RAM, κάτι όμως που δεν προτείνεται. Αυτό μπορεί να γίνει αν απενεργοποιήσεις πρακτικά οτιδήποτε στο Dom0. Η κατανάλωση μνήμης στο Dom0 είναι μια ισορροπία ανάμεσα στην σμίκρυνση του ρόλου του Dom0, κάτι που αυξάνει την ασφάλεια, και στον ακρωτηριασμό των δυνατοτήτων του συστήματος για συντήρηση.

Επιπρόσθετα ο τρόπος με τον οποίο το Dom0 θα μοιράζει το δίκτυο επηρεάζει την κατανάλωση μνήμης. Αν επιλέξεις να χρησιμοποιήσεις την μέθοδο δρομολόγησης μέσω του Dom0, θα έχεις αυξημένο overhead από την διαδικασία δρομολόγησης

αλλά και από τους κανόνες του τείχους προστασίας. Αυτοί οι κανόνες του τείχους προστασίας και της μετάφρασης διευθύνσεων δικτύου (NAT) προκαλούν μείωση της απόδοσης του συστήματος λόγω της μεγαλύτερης κατανάλωσης μνήμης RAM, αλλά μπορεί να είναι απαραίτητα για το σύστημα.

Η επιλογή του λειτουργικού συστήματος μπορεί επίσης να επηρεάσει τις απαιτήσεις μνήμης. Μια έκδοση stripped-down Linux για μικρούς server έχει μικρότερες απαιτήσεις από μια απλή έκδοση με γραφικό περιβάλλον. Γενικά προτείνεται 1gb RAM για το Dom0. Όμως αν χρησιμοποιείς κάποια εργαλεία που σε βοηθάνε στη συντήρηση πρέπει σκεφτείς ότι και αυτά θα δεσμεύσουν χώρο στη μνήμη.

Καλό είναι να κανονίσεις το budget σου ώστε να αγοράσεις όσο περισσότερη Ram μπορείς (ανάλογα πάντα με τις ανάγκες σου) αν και μπορείς να κλέψεις λίγο και για μερικούς φιλοξενούμενους να χρησιμοποιήσεις ακόμα και λιγότερο από τις ελάχιστες απαιτήσεις που έχει κλείνοντας-απενεργοποιώντας υπηρεσίες που γνωρίζεις ότι δεν θα σου χρειαστούν. Με αυτό τον τρόπο θα μπορέσεις να “χωρέσεις” περισσότερα εικονικά μηχανήματα στην ίδια ποσότητα μνήμης RAM. Αν όμως σκοπεύεις να τρέξεις έναν server εφαρμογών ή έναν server με βάσεις δεδομένων καλό είναι να χρησιμοποιήσεις όση μνήμη RAM θα χρησιμοποιούσες αν το έτρεχες σε μη εικονικό μηχανήμα.

Μια άλλη σκέψη που πρέπει να κάνεις, είναι ποιο χώρο swap θα χρησιμοποιήσεις. Τα Linux και άλλα λειτουργικά χρησιμοποιούν τον χώρο swap όταν η φυσική μνήμη είναι γεμάτη ή κοντά στο να γεμίσει. Παρόλο που ο χώρος swap είναι χρήσιμος για την ανακούφιση ενός συστήματος με πολλαπλούς χρήστες και με μικρή ποσότητα μνήμης RAM, ο χώρος αυτός δεν μπορεί να θεωρηθεί σαν αντικαταστατή της μνήμης RAM, γιατί βασίζεται στο σκληρό δίσκο που είναι πολύ πιο αργός από τη RAM και τελικά θα επηρεάσει αρνητικά τις επιδόσεις.

Αν σκοπεύεις να χρησιμοποιήσεις λειτουργικό Windows XP προτείνεται να χρησιμοποιήσεις τουλάχιστον 512 MB RAM. Η Microsoft προτείνει 128 mb, αλλά στην πράξη το νούμερο αυτό αποδεικνύεται μικρό.

Εν κατακλείδι, έχοντας υπολογίσει πόση μνήμη RAM θα χρειαστούμε για τα εικονικά συστήματα που σκοπεύουμε να χρησιμοποιήσουμε, πρέπει να προσθέσουμε και κάποιο έξτρα χώρο για να δουλεύει το σύστημα πιο άνετα.

Οικοδεσπότες και φιλοξενούμενοι - Host and Guests

Host

Στις περισσότερες διανομές Linux και πολλές διανομές Unix έχεις τη δυνατότητα να κατεβάσεις και να εγκαταστήσεις το Xen μέσω ενός εγγενή συστήματος διαχείρισης πακέτων. Αν η διανομή σου περιλαμβάνει το Xen, προτείνεται να χρησιμοποιήσεις το ίδιο υπάρχον καθώς θα επωφεληθείς από την καλή συνεργασία με τον πυρήνα, την άμεση υποστήριξη καθώς και από τις αναβαθμίσεις.

Μερικές από τις διανομές που είναι δυνατή η εγκατάσταση του Xen είναι στον παρακάτω πίνακα:

Distro	Κύριος ιστότοπος	Περιγραφή
Arch Linux	archlinux.org	Το Arch Linux είναι μια διανομή ελαφριά και ευέλικτη που προσπαθεί να «κρατηθεί απλή».
Alpine Linux 2.4.x	alpinelinux.org	Μια ελαφριά έκδοση που έχει προσανατολισμό προς την ασφάλεια και βασίζεται στο uClibc και στο Busybox.
CentOS 5.x, 6.4 και νεότερα	centos.org	Το CentOS είναι μια διανομή που προορίζεται για επιχειρήσεις. Το CentOS είναι δωρεάν. Καθώς το CentOS βασίζεται στο RedHat 6 και αυτό σταμάτησε την υποστήριξη του Xen στην έκδοση, πρέπει να βρούμε ένα πυρήνα συμβατό από Τρίτη πηγή.
Distro	Κύριος ιστότοπος	Περιγραφή

Fedora 15,16,17	fedoraproject.org	Το Fedora είναι μια διανομή βασισμένη στο Red Hat Package Manager με κύκλο ανανέωσης τους 6 μήνες.
FreeBSD	freebsd.org	Το FreeBSD® είναι ένα προχωρημένο λειτουργικό σύστημα για μοντέρνους σέρβερ και ενσωματωμένες πλατφόρμες υπολογιστών.
Finnix	finnix.org	Το Finnix είναι ένα LiveCD με λειτουργία διαχειριστή υπολογιστών και περιλαμβάνει υποστήριξη Xen απο μόνο του.
NetBSD	netbsd.org	Το NetBSD είναι ένα ελεύθερο, γρήγορο, ασφαλές και απόλυτα φορητό λειτουργικό σύστημα βασισμένο σε Unix.
openSuSE 10.x, 11.x, 12.x	opensuse.org	Το openSuSE είναι ένα δωρεάν βασισμένο σε linux λειτουργικό σύστημα για PC, Laptop ή server.
RedHatEnterpriseLinux (RHEL) 5.x	redhat.com	Από την έκδοση του RHEL 5.x περιλαμβάνεται το Xen από την αρχή.
SUSE Linux Enterprise Server (SLES) 10.x, 11.x		Βασισμένο στον ανοικτό πυρήνα Linux και είναι η πιο παλιά εμπορική διανομή.
Distro	Κύριος ιστοτόπος	Περιγραφή

Ubuntu 11.10, 12.04 LTS	ubuntu.com	Γρήγορο, ασφαλές και απλό το Ubuntu χρησιμοποιείται από 20 εκατομμύρια ανθρώπους καθημερινά.
--------------------------------	--	--

Φιλοξενούμενοι

Τα φιλοξενούμενα συστήματα μπορούν να εικονικοποιηθούν πλήρως ή να παραεικονικοποιηθούν, με τους περιορισμούς κάθε φορά που αναφέραμε παραπάνω.

Τα περισσότερα λειτουργικά συστήματα που μπορούν να τρέξουν σε ένα pc μπορούν να τρέξουν σαν φιλοξενούμενοι πλήρους εικονικοποίησης.

Επιπρόσθετα τα παρακάτω συστήματα έχουν υποστεί αλλαγές για να μπορούν να λειτουργήσουν σαν παραεικονικά συστήματα στο Xen:

Διανομές Linux

Διανομή	Δυνατότητα παραεικονικοποίησης
Alpine Linux 2.3.x	Ναι
Alpine Linux 2.4.x	Ναι
Debian Lenny	Ναι
Debian Squeeze	Ναι
Ubuntu 10.04	Ναι [1]
Ubuntu 11.04	Ναι [1]
Ubuntu 12.04	Ναι [1]
Fedora 14	Ναι
Fedora 16	Ναι
Διανομή	Δυνατότητα παραεικονικοποίησης

Oracle Linux 5	Ναι [2]
Oracle Linux 6	Ναι [2]
RHEL 5	Ναι [3]
SLES 10	Ναι
SLES 11	Ναι
OpenSUSE 11.4	Ναι

- [1] OEC2 πυρήνας μόνο
[2] Υποστηρίζεται χρησιμοποιώντας τον συμβατό πυρήνα Red-Hat
[3] Υπάρχουν γνωστά προβλήματα κατά το σώσιμο/επαναφορά

BSDs και άλλα συστήματα Unix

Διανομή	Δυνατότητα παραεικονικοποίησης
FreeBSD 8.2	Ναι
NetBSD 5.1	Ναι
Solaris 11	Ναι

Τι υλικό να διαλέξεις?

Η επιλογή του υλικού είναι κρίσιμη, ειδικά στο θέμα επεξεργαστή και μητρικής που πρέπει να διαλεχτούν προσεκτικά. Πιθανόν να θέλεις έναν επεξεργαστή και μια μητρική ικανά να υποστηρίξουν τεχνολογίες όπως, Intel VT-x ή AMD-V και IOMMU (Passthrough) ή VT-d.

Ένα άλλο σημαντικό στοιχείο είναι η μνήμη RAM. Βασικά από αυτήν εξαρτάται πόσοι χρήστες μπορούν να τρέξουν παράλληλα. Αν έχεις μικρά εικονικά συστήματα όπως DHCP ή DNS σέρβερ που δε χρειάζονται παραπάνω από 128 ή 256MB RAM, μπορείς να χωρέσεις σε 2 GB ram 10 με 15 μηχανήματα. Μια καλή συμβουλή είναι να πάρεις περισσότερη από την ελάχιστη μνήμη που υπολογίζει να χρειάζονται τα συστήματα που θα εγκαταστήσεις όπως είδαμε παραπάνω.

Ο χώρος αποθήκευσης είναι ένας ακόμα παράγοντας. Με τους δίσκους τεχνολογίας HDD ο χώρος αποθήκευσης δεν είναι πλέον θέμα. Αλλά η διαχείριση του είναι! Έχε στο μυαλό σου ότι όλοι οι χρήστες συναγωνίζονται για πρόσβαση στον σκληρό καθώς είναι ένας κοινός πόρος. Να περιμένεις υψηλή πρόσβαση διαβάσματος κατά τη διάρκεια της εκκίνησης, ειδικά αν όλα τα φιλοξενούμενα συστήματα ξεκινάνε μαζί. Γι αυτό το λόγο πρέπει να προστατέψεις την επένδυση και τη δημιουργία σου με πλεόνασμα. Μια καλή λύση είναι η χρησιμοποίηση ενός συστήματος RAID, το οποίο είναι πιο γρήγορο και θα υπάρχει καθόλου ή πολύ λίγος χρόνος που ο σέρβερ δε θα είναι διαθέσιμος. Μια καλή πρόταση είναι το RAID 1 ή ακόμα καλύτερα το πιο γρήγορο αλλά και πιο ακριβό RAID 10. Αν τα χρήματα δεν είναι θέμα μπορείς να αγοράσεις έναν ελεγκτή RAID ή αλλιώς να θυσιάσεις μια μικρή μείωση της απόδοσης του επεξεργαστή για να φτιάξεις ένα RAID λογισμικού.

Σημείωση: Απ' όσο γνωρίζω δε μπορείς να εκκινήσεις απευθείας από Raid λογισμικού 5 ή 6. Αλλά είναι δυνατόν να έχεις ξεχωριστό partition για την εκκίνηση /boot (ως RAID 1) και τα υπόλοιπα όπως επιθυμείς.

Επίσης είναι καλό να υπάρχουν δύο αντάπτορες δικτύου για να μοιράζεται η κίνηση. Έτσι μπορείς να αγοράσεις μια μητρική με δυο NICs ή μια επιπλέον κάρτα δικτύου.

Πριν την εγκατάσταση

- Αναβάθμισε – Αγόρασε υλικό ανάλογα με τις ανάγκες σου
- Αναβάθμισε το BIOS
- Φόρτωσε τις προεπιλεγμένες ρυθμίσεις και κάνε τις έξης αλλαγές:
- Ενεργοποίησε τις επεκτάσεις του επεξεργαστή (χρειάζεται για το HVM)
 - Intel: vmx
 - AMD: svm (Secure Virtual Machine)
- Ενεργοποίησε το IOMMU αν υποστηρίζεται από τη μητρική (αλλιώς θα έχεις μόνο περιορισμένη διέλευση PCI(e)).
 - INTEL: Virtualization Technology for Directed I/O“ (short VT-d)
 - AMD: IOMMU
- Ενεργοποίησε την εκκίνηση μετά από διακοπή ρεύματος, αν νομίζεις ότι κάτι τέτοιο θα βοηθήσει.
- Θέσε την κατάλληλη σειρά εκκίνησης των μεσων.

Είμαστε έτοιμοι! Ας ξεκινήσουμε την εγκατάσταση...

Εγχειρίδιο Χρήσης του Xen

Εγκατάσταση Debian

Τοποθετούμε το CD εγκατάστασης του Debian στο cd-rom και αλλάζουμε τις ρυθμίσεις στο BIOS ώστε η πρώτη πηγή εκκίνησης να είναι το CDROM (αν δεν το έχουμε ήδη κάνει).

Όταν εμφανιστεί το μενού, επιλέγουμε την επιλογή "Install" για να ξεκινήσει η διαδικασία εγκατάστασης. Η διαδικασία είναι πολύ απλή. Ακολουθούμε τις οδηγίες μέχρι να φτάσει στο σημείο που γίνεται η κατάτμηση του δίσκου.

Επιλέγουμε την προχωρημένη διαμόρφωση για να δημιουργήσουμε 4 διαμερίσματα. Ένα για το /boot, ένα για το /, ένα για το swap και ένα τελευταίο για το LVM (αν επιλέξεις αυτόν τον τρόπο για να αποθηκεύσεις τα εικονικά μηχανήματα).

Δημιουργούμε πρώτα το διαμέρισμα /boot με μέγεθος 200Mb, φορμάτ ext2 και επιλέγουμε το /boot σαν mountpoint.

Επαναλαμβάνουμε τη διαδικασία για το " / " αλλά αλλάζουμε το mountpoint σε / και με μέγεθος 15Gb ή μεγαλύτερο. Το φορμάτ να είναι ext3.

Έπειτα δημιουργούμε άλλο ένα διαμέρισμα περίπου 1,5 φορά του μεγέθους της RAM που θα έχει το μηχάνημα και επέλεξε το για τόμο swap.

Τέλος δημιουργούμε ένα διαμέρισμα που πιάνει το υπόλοιπο μέγεθος του δίσκου αλλά δεν το διαμορφώνουμε ή θέτουμε κάποιο σημείο προσάρτησης και δηλώνουμε ότι θα χρησιμοποιηθεί για LVM.

Εν τέλει πρέπει να έχουμε ένα αποτέλεσμα:

/dev/sda :

sda1 - /boot 200MB

sda2 - / 15GB

sda3 – swap

sda4 - reserved for LVM

Όταν φτάσει στην επιλογή πακέτων επέλεγουμε μόνο το βασικό σύστημα και όχι κάποιο GUI όπως επίσης και το ssh server και το Standard system utilities.

Όταν ολοκληρωθεί η εγκατάσταση θα ξεκινήσουμε τη διαδικασία εγκατάστασης του xen. Αλλά πριν καλό θα ήταν να κάνουμε κάποιους ελέγχους ότι όλα πήγαν καλά.

Επιβεβαίωση πυρήνα:

Αν τώρα το μηχάνημα στο οποίο σκοπεύεις να εγκαταστήσεις το Xen είναι παλιά εγκατάσταση και δε θυμάσαι ποια έκδοση του Debian τρέχει, χρησιμοποίησε τις παρακάτω εντολές:

```
uname -a
```

```
lsb_release -a
```

Επιβεβαίωση Συστήματος αρχείων:

```
df -h
```

Επιβεβαίωση Swap:

```
swapon -s
```

Επιβεβαίωση δυνατότητας εικονικοποίησης υλικού:

Intel:

```
grep "vmx" /proc/cpuinfo
```

ή AMD:

```
grep "svm" /proc/cpuinfo
```

Ενημέρωση πηγών:

```
apt-get update
```

Με την εντολή αυτή κατεβαίνουν και ανανεώνονται οι λίστες των πακέτων για τις καινούργιες εκδόσεις και οι εξαρτήσεις τους.

Επίσης μπορούμε να αλλάξουμε τις ρυθμίσεις του `/etc/apt/sources.list`:

```
nano /etc/apt/sources.list
```

και να βάλουμε σε σχόλια τις γραμμές που αφορούν το cd-rom, ώστε να χρησιμοποιούνται τα πακέτα από το internet και όχι από το cd κατά την εγκατάσταση.

Έπειτα μπορούμε να εγκαταστήσουμε το vim (vim-nox καθώς δε θα έχουμε γραφικό περιβάλλον). Αλλιώς μπορούμε να χρησιμοποιήσουμε το ενσωματωμένο nano.

```
aptitude install vim-nox
```

Εφόσον επιλέξαμε το LVM για τρόπο αποθήκευσης, αν δεν είναι ήδη εγκατεστημένο, πρέπει να το εγκαταστήσουμε.

Ελέγχουμε αν επιστρέψει αποτελέσματα εντολή:

```
dpkg --get-selections | grep ^i
```

Αν ναι τότε είναι ήδη εγκατεστημένο και δε χρειάζεται κάτι άλλο. Αν όχι εκτελούμε την εντολή:

```
aptitude install lvm2
```

και στη συνέχεια επιβεβαιώνουμε ότι έγινε σωστά:

```
dpkg --get-selections | grep ^i
```

Τώρα που εγκαταστήσαμε το LVM θα το διαμορφώσουμε ώστε να χρησιμοποιήσουμε το /dev/sda7, στη δική μας περίπτωση ως το φυσικό τόμο.

Αρχικά ελέγχουμε αν είναι όντως αυτό:

```
fdisk -l
```

και στη συνέχεια:

```
pvccreate /dev/sda7
```

Με αυτόν τον τρόπο το LVM έχει κάπου να αποθηκεύσει τα blocks του. Στη συνέχεια θα δημιουργήσουμε ένα γκρουπ τόμου με ότι ονομασία θέλουμε. Η προεπιλογή στο Xen είναι το όνομα vg0. Εμείς θα το ονομάσουμε vgDomU.

```
vgcreate vgDomU /dev/sda7
```

Ελέγχουμε αν έγινε σωστά:

```
/sbin/vgdisplay vgDomU
```

Το LVM έχει αρχικοποιηθεί και διαμορφωθεί ώστε στη συνέχεια να μπορέσουμε να δημιουργήσουμε λογικούς τόμους για τα εικονικά μας μηχανήματα.

Εγκατάσταση του Xen

Αν χρησιμοποιείς το Debian testing (Wheezy) χρησιμοποίησε την εντολή:

```
aptitude install xen-linux-system-amd64
```

αυτή η εντολή είναι αρκετή, καθώς το aptitude θα επιλέξει τα κατάλληλα πακέτα.

Αν χρησιμοποιείς Debian stable (Squeeze) χρησιμοποίησε:

```
aptitude install xen-linux-system xen-qemu-dm-4.0
```

Σημείωση: στο Squeeze το έξτρα πακέτο xen-qemu-dm-4.0 είναι απαραίτητο για την υποστήριξη του HVM.

Μόλις τελειώσει η εγκατάσταση μπορούμε να κάνουμε επαλήθευση των πακέτων που εγκαταστάθηκαν:

```
dpkg --get-selections | grep '^i'
```

Grub2 – Grand Unified Bootloader

Στη συνέχεια πρέπει να αλλάξουμε τις ρυθμίσεις στο Grub2, τον επόπτη εκκίνησης δηλαδή, ώστε να ξεκινάει πρώτα το Xen αντί του πυρήνα του Linux.

```
mv -i /etc/grub.d/10_linux /etc/grub.d/21_linux
```

Το Grub προσπαθεί να ανιχνεύσει άλλα λειτουργικά και επίσης να βρει όποιο λειτουργικό εγκατασταθεί στο μέλλον. Γι' αυτό πρέπει να απενεργοποιήσουμε τη λειτουργία αυτή στο αρχείο ρυθμίσεων.

```
nano /etc/default/grub
```

και προσθέτουμε τις παρακάτω σειρές:

```
# Disable OS prober to prevent virtual machines on logical volumes from  
appearing in the boot menu.  
GRUB_DISABLE_OS_PROBER=true
```

Επίσης καλό είναι να προσθέσουμε τις παρακάτω σειρές

```
# Xen boot parameters for all Xen boots  
GRUB_CMDLINE_XEN="dom0_mem=512M"  
  
# Xen boot parameters for non-recovery Xen boots (in addition to  
GRUB_CMDLINE_XEN)  
GRUB_CMDLINE_XEN_DEFAULT=""
```

Με το «dom0_mem=512M» διατηρεί και περιορίζει 512Mb RAM για χρήση του dom-0. Προσωπικά επιλέγω 512Mb αν υπάρχει πολύ RAM. Με αυτό το μέγεθος το dom-0 μπορεί να εκτελέσει εντατικές εργασίες, χωρίς όμως να σπαταλείται άδικα παραπάνω μνήμη.

Τέλος πρέπει να ενημερώσουμε το Grub για τις αλλαγές:


```
update-grub
```

Τώρα που αλλάξαμε τις ρυθμίσεις του Grub, πρέπει να σκεφτούμε τη συμπεριφορά των φιλοξενούμενων όταν το dom-0 σβήνει. Από προεπιλογή τα domains παγώνουν όταν τερματίζεται το dom-0 και συνεχίζουν όταν εκκινεί ξανά. Πάγωμα σημαίνει ότι τα domains σώζονται, στην κατάσταση που είναι εκείνη τη στιγμή, και αποθηκεύονται στο /var/lib/xen/save. Αυτό μπορεί να είναι προβληματικό για διάφορους λόγους, όπως να μην υπάρχει αρκετός χώρος στο σκληρό, διάφορα σφάλματα λογισμικού και συνήθως δημιουργούνται προβλήματα και στις συνδέσεις δικτύου. Για να το αλλάξεις αυτό άλλαξε κατάλληλα το αρχείο:

```
nano /etc/default/xendomains
```

```
XENDOMAINS_SAVE=""  
...  
XENDOMAINS_RESTORE=false
```

Σώζουμε και έπειτα επανεκκινούμε το σύστημα:

```
Reboot
```

Όταν επανεκκινήσει τρέχουμε την εντολή:

```
xm dmesg
```

αν επιστρέψει το παρακάτω:

```
WARNING! Can't find hypervisor information in sysfs!  
Error: Unable to connect to xend: No such file or directory. Is xend  
running?
```

...σημαίνει ότι τρέχει σε πυρήνα έτοιμο για Xen, αλλά όχι στον ίδιο τον πυρήνα Xen. Αυτό σημαίνει ότι έχει γίνει κάποιο λάθος στον επόπτη εκκίνησης και δεν ξεκινάει πρώτα το Xen αντί του πυρήνα του Linux.

Παραμετροποίηση δικτύου

Η παραμετροποίηση δικτύου στο Debian βρίσκεται στο:

`/etc/network/interfaces`

Αρχικά κρατάμε αντίγραφο της τωρινής ρύθμισης του αρχείου

```
cp /etc/network/interfaces /etc/network/interfaces.orig
```

Έπειτα ανοίγουμε το αρχείο για επεξεργασία:

```
nano /etc/network/interfaces
```

Και το αλλάζουμε ώστε να μοιάζει με το παρακάτω:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet manual
    pre-up ifconfig eth0 up
    post-down ifconfig eth0 down

# Network Bridge, we will call ours xenbr0
```

```
auto xenbr0

iface xenbr0 inet static

    bridge_ports eth0

    address 192.168.0.100

    netmask 255.255.255.0

    broadcast 192.168.0.255

    gateway 192.168.0.1

#if you want the dhcp server to assign an address automatically -
#uncomment and comment the above

# iface xenbr0 inet dhcp

#     bridge_ports eth0
```

Για να ανέβει και ενεργοποιηθεί η γέφυρα:

```
ifup xenbr0
```

Σημείωση: Επειδή θα αλλάξει η διεύθυνση του, καλό θα είναι αν χειριζόμαστε το μηχάνημα απομακρυσμένα να είμαστε σίγουροι ότι οι αλλαγές που κάνουμε είναι σωστές, αλλιώς υπάρχει κίνδυνος να κλειδωθούμε απ' έξω και να χρειαστεί η φυσική παρουσία στο μηχάνημα για να διορθωθεί.

Χρησιμοποιούμε την εντολή `brctl show` για να εμφανιστούν όλες οι Ethernet γέφυρες στο σύστημα:

```
brctl show
```

bridge name	bridge id	STP enabled	interfaces
xenbr0	8000.001d7d06845d	no	eth0

Εδώ πρέπει να υπάρχει μια λίστα με όλες τις γέφυρες που φτιάξαμε και όλες τις διαθέσιμες θύρες.

```
ifconfig
```

Αποτελέσματα:

```
eth0  Link encap:Ethernet HWaddr 00:1d:7d:06:84:5d
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:122 errors:0 dropped:0 overruns:0 frame:0
      TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:17293 (16.8 KiB) TX bytes:9710 (9.4 KiB)

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:65536 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

xenbr0  Link encap:Ethernet HWaddr 00:1d:7d:06:84:5d
      inet addr:192.168.0.100 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::21d:7dff:fe06:845d/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:103 errors:0 dropped:0 overruns:0 frame:0  
TX packets:63 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:10151 (9.9 KiB) TX bytes:9710 (9.4 KiB)
```

Για να ελέγξουμε αν είναι σωστή η σύνδεση:

```
ping 192.168.0.1
```

ή όποια άλλη είναι η διεύθυνση του δρομολογητή
για ακύρωση:

```
Ctrl+C
```

Αν επιστρέψει σωστά αποτελέσματα το τοπικό δίκτυο δουλεύει σωστά.

```
ping 8.8.8.8
```

αν επιστραφούν σωστά αποτελέσματα, τότε είναι συνδεδεμένο στο internet.
και τέλος

```
ping google.com
```

αν αυτό αποτύχει, τότε ελέγχουμε αν έχει δηλωθεί σωστά ο nameserver:

```
nano /etc/resolv.conf
```

Αν επιστραφούν σωστά αποτελέσματα, τότε είμαστε έτοιμοι να προχωρήσουμε.

Απομακρυσμένη πρόσβαση

Το SSH είναι ένα πρωτόκολλο δικτύου για ασφαλή επικοινωνία δεδομένων που σχεδιάστηκε για να αντικαταστήσει το telnet. Υπήρχε η δυνατότητα να γίνει η εγκατάσταση του κατά την εγκατάσταση του λειτουργικού συστήματος, αλλά αν δεν επιλεχθηκε, μπορεί να γίνει και τώρα με τις εξής εντολές:

```
tasksel
```

μετά επέλεξε το `ssh server`

ή

```
aptitude install openssh-server
```

Έπειτα μπορούμε να συνδεθούμε στο Dom0 από οποιονδήποτε υπολογιστή. Ένα καλό πρόγραμμα για windows είναι το [putty](#) και το [MobaXterm](#).

Xen-tools

Υπάρχουν πολλά εργαλεία για να διαχειριστείς τα καινούργια domains στο Xen. Τα καλύτερο με βάση τις λειτουργίες και την απόδοση είναι το Xen-tools που μας παρέχει ότι χρειαζόμαστε, είναι απλό και δε χρειάζεται γραφικό περιβάλλον.

Η εγκατάσταση γίνεται μέσω της εντολής:

```
aptitude install xen-tools
```

Έλεγχος καλής λειτουργίας:

Η εντολή `xm` είναι αυτή που χρησιμοποιείς για να αλληλεπιδράς με το Xen. Η εντολή `xm list` τυπώνει μια λίστα με όλα τα domains που τρέχουν

```
xm list
```

Εδώ μπορείς να ελέγξεις και αν η μνημη είναι αυτή που είχες κρατήσει για το `dom-0`.

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	511	6	r—	14.9

Αν θέλεις περισσότερες πληροφορίες:

```
xm info Domain-0
```

Για να ελέγξεις ποια έκδοση του Xen χρησιμοποιείς, αρκεί να τρέξεις την εντολή:

```
dmesg | grep Xen\ version
```

```
[ 0.000000] Xen version: 4.0.1 (preserve-AD) (dom0)
```

Η εντολή `xm info` μεταξύ άλλων έχει τις εξής γραμμές:


```
xen_major: 4  
xen_minor: 0  
xen_extra : .1
```

που σημαίνει ότι χρησιμοποιείς την έκδοση 4.0.1.

Η παραμετροποίηση για τα εικονικά μηχανήματα που θα φτιάξουμε στη συνέχεια βασίζεται στις ρυθμίσεις αυτού του αρχείου: /etc/xen-tools/xen-tools.conf

```
nano /etc/xen-tools/xen-tools.conf
```

αν αποφασίσουμε να χρησιμοποιήσουμε το LVM για αποθήκευση πρέπει να να επιλέξουμε τον γκρουπ τόμου που δημιουργήσαμε παραπάνω.

```
lvm = vgDomU
```

επίσης μπορούμε να ρυθμίσουμε τις παρακάτω προκαθορισμένες ρυθμίσεις:

```
size = 4Gb    # Μέγεθος δίσκου  
memory = 128Mb # Μέγεθος μνήμης RAM  
swap  = 128Mb # Μέγεθος swap  
  
# μπορούμε διαδραστικά να δηλώσουμε τον κωδικό της ριζας κατά τη  
δημιουργία  
passwd = 1
```

Επίσης μπορούμε να δηλώσουμε τη γέφυρα του δικτύου που θα χρησιμοποιήσουν τα εικονικά μας μηχανήματα.

```
bridge = xenbr0
```

Οι ρυθμίσεις αυτές είναι που θα χρησιμοποιήσουν τα καινούργια domains κατά τη δημιουργία τους, αλλά μπορούμε να τις αντικαταστήσουμε κατά τη δημιουργία ενός καινούργιου.

Δημιουργία Παρά-εικονικού φιλοξενούμενου

Η εντολή `xen-create-image` δημιουργεί ένα αρχείο ρυθμίσεων που έχει όλες τις ρυθμίσεις στο `domain` και βρίσκεται στο αρχείο `/etc/xen/hostname.cfg`

Μερικές από τις παραμέτρους που μπορείς να ρυθμίσεις:

`--hostname= name` το όνομα του νέου μηχανήματος,

`--dhcp` χρησιμοποίηση του πρωτοκόλλου `dhcp` για να λάβει δικτυακή διεύθυνση.

`--size=5Gb --swap=256Mb` δημιουργούνται δυο λογικοί τόμοι στο `/dev/vgDomU` ο ένας ονομάζεται `hostname-disk` (5Gb) και εκεί βρίσκεται το καινούργιο εικονικό μας μηχάνημα και ο άλλος είναι ο `hostname-swap` και χρησιμοποιείται σαν διαμέρισμα `swap` (256 Mb).

`--memory=512Mb` δηλώνουμε ότι το μηχάνημα μας θα έχει 512Mb ram.

`--dist=dist` δηλώνεις τη διανομή που θέλεις να εγκαταστήσεις

`--noswap` δε δημιουργείται διαμέρισμα `swap`.

`--password=passphrase` δηλώνεις τον κωδικό για το `root` στο νέο μηχάνημα

`--passwd` ο κωδικός θα ζητηθεί διαδραστικά από τον χρήστη κατά τη δημιουργία

`--pygrub` το DomU θα εκκινηθεί χρησιμοποιώντας το `pygrub`

`--vcpus=num` δηλώνεις το αριθμό των πυρήνων του επεξεργαστή που θα χρησιμοποιήσεις

`--ip=x.x.x.x` δηλώνεις την `ip` σε περίπτωση που δε χρησιμοποιείς `dhcp`

`--gateway= x.x.x.x` δηλώνεις το `gateway` σε περίπτωση που δε χρησιμοποιείς `dhcp`

`--netmask= x.x.x.x` δηλώνεις την `netmask` σε περίπτωση που δε χρησιμοποιείς `dhcp`

`--nameserver= "123.456.789.ABC 123.456.789.DEF"` δηλώνεις τους `nameservers` σε περίπτωση που δε χρησιμοποιείς `dhcp`. Μπορείς να δηλώσεις πολλαπλούς αρκεί να αφήσεις ένα κενό μεταξύ τους

Πρόσθεσε την εντολή `--force` σαν παράμετρο αν θέλεις να γράψεις πάνω από ένα υπάρχον σύστημα.

Για να δεις επιπρόσθετες παραμέτρους μπορείς να τρέξεις την εντολή:

```
man xen-create-image
```

Στη συνέχεια θα δημιουργήσουμε το αρχείο ρύθμισης για ένα debian pv με όνομα testdebian, που θα χρησιμοποιεί dhcp, μέγεθος δίσκου 5gb, μέγεθος swap 256mb, μνήμη ram 512mb και διανομή squeeze.

```
xen-create-image --hostname=testdebian --dhcp --size=5Gb --  
swap=256Mb --memory=512Mb --dist=squeeze
```

Στη συνέχεια από εδώ μπορείς να αλλάξεις τη συμπεριφορά του DomU:

```
nano /etc/xen/testdebian.cfg
```

Με αυτόν τον τρόπο δημιουργείς έναν παραεικονικό φιλοξενούμενο στο Xen και τώρα το μόνο που μένει είναι να εκκινήσει:

```
xm create testdebian.cfg
```

στη συνέχεια αν δεν υπάρξει κάποιο λάθος πρέπει να βρίσκεται στη λίστα των domains:

```
xm list
```

για να συνδεθείς μέσω κονσόλας στο φιλοξενούμενο:

```
xm console testdebian
```

Επίσης αντί για το όνομα μπορείς να χρησιμοποιήσεις το id του DomU όπως φαίνεται στα αποτελέσματα τις εντολής xm list.

Μπορείς ακόμα κατά τη δημιουργία του DomU να συνδεθείς αυτόματα στο εικονικό μηχάνημα μέσω της εντολής:

```
xm create -c testdebian.cfg
```

Σημείωση: Για να επιστρέψεις πίσω στο dom-0, αρκεί να πατήσεις ctrl+]

Μπορείς να τερματίσεις ή να επανεκκινήσεις ένα domain από το dom-0 χρησιμοποιώντας τις εντολές:

```
xm shutdown testdebian
```

```
xm reboot testdebian
```

Αν θέλεις να τερματίσεις αμέσως ένα domain μπορείς να χρησιμοποιήσεις την εντολή:

```
xm destroy test
```

(Είναι σα να τραβάς το καλώδιο από ένα κανονικό μηχάνημα)

Για να διαγράψεις ένα domain χρησιμοποίησε την εντολή:

```
xen-delete-image testdebian
```

Μαζί θα διαγραφούν και τα διαμερίσματα που είχαν δημιουργηθεί για να το φιλοξενήσουν.

Αν θέλεις να διαχειριστείς τα domains ή θέλεις να δεις πως μπορείς να χρησιμοποιήσεις το xm, χρησιμοποίησε την εντολή:

```
xm help
```

Χειροκίνητη δημιουργία Παραεικονικών Φιλοξενούμενων

Μπορούμε να δημιουργήσουμε παραεικονικούς φιλοξενούμενους κάνοντας την εγκατάσταση του λειτουργικού στην περίπτωση μας το Ubuntu, αλλά το ίδιο μπορεί να γίνει και με άλλες διανομές όπως Fedora και CentOS.

Αρχικά δημιουργούμε το χώρο στο δίσκο που θα χρησιμοποιήσουμε για το DomU:

```
lvcreate -n ubuntu -L 5G vgDomU
```

Στη συνέχεια θα διαλέξουμε έναν καθρέφτη για να κατεβάσουμε τις εικόνες των αρχείων που μας χρειάζονται για την εγκατάσταση. Μπορείς να διαλέξεις έναν καθρέφτη από εδώ: <https://launchpad.net/ubuntu/+archivemirrors>

```
wget http://ubuntu.otenet.gr/ubuntu/dists/precise/main/installer-amd64/current/images/netboot/xen/initrd.gz
```

```
wget http://ubuntu.otenet.gr/ubuntu/dists/precise/main/installer-amd64/current/images/netboot/xen/vmlinuz
```

Τώρα ήρθε η ώρα να φτιάξουμε το αρχείο ρυθμίσεων για το DomU:

```
nano /etc/xen/ubuntu.cfg
```

```
name = "ubuntu"

memory = 256

disk = ['phy:/dev/vgDomU/ubuntu,xvda,w']

vif = [ 'mac=00:16:3E:4A:E0:E7,bridge=xenbr0' ]

#bootloader = "pygrub"

kernel = "/var/lib/xen/images/ubuntu-netboot/vmlinuz"
```

```
ramdisk = "/var/lib/xen/images/ubuntu-netboot/initrd.gz"  
extra = "debian-installer/exit/always_halt=true -- console=hvc0"
```

Έπειτα ήρθε η ώρα να το ξεκινήσουμε:

```
xm create -c /etc/xen/ubuntu.cfg
```

Και κάνουμε κανονικά την εγκατάσταση του λειτουργικού διαλέγοντας τις επιλογές που μας ενδιαφέρουν. Μόλις τελειώσει η εγκατάσταση, τερματίζουμε τη λειτουργία του, θέτουμε το pygrub υπεύθυνο για την εκκίνηση του:

```
ln -s /usr/lib/xen-4.1/bin/pygrub /usr/bin/pygrub
```

βάζουμε σε σχόλια τις τιμές για το kernel, ramdisk, extra και προσθετούμε τον bootloader:

```
name = "ubuntu"  
  
memory = 256  
disk = ['phy:/dev/vgDomU/ubuntu,xvda,w']  
vif = [ 'mac=00:16:3E:4A:E0:E7,bridge=xenbr0' ]  
  
bootloader = "pygrub"  
  
#kernel = "/var/lib/xen/images/ubuntu-netboot/vmlinuz"  
#ramdisk = "/var/lib/xen/images/ubuntu-netboot/initrd.gz"  
#extra = "debian-installer/exit/always_halt=true -- console=hvc0"
```

Έτσι μπορούμε να εκκινήσουμε και να χρησιμοποιήσουμε κανονικά το DomU μας :

```
xm create -c /etc/xen/ubuntu.cfg
```

HVM guests

Ενώ το Xen έχει έναν αυτοματοποιημένο τρόπο για να δημιουργείς παραεικονικούς φιλοξενούμενους, δεν ισχύει το ίδιο και για τους φιλοξενούμενους πλήρους εικονικοποίησης που χρειάζονται λίγο περισσότερη δουλειά. Εδώ θα δημιουργήσουμε ένα DomU που θα τρέχει windows server 2008.

Αρχικά θα πρέπει να έχουμε ή να δημιουργήσουμε ένα αρχείο iso που θα περιέχει τα αρχεία εγκατάστασης για το λειτουργικό μας. Αυτό στη συνέχεια μπορούμε είτε να το τοποθετήσουμε σε ένα cd/dvd, σε ένα εξωτερικό δίσκο είτε ακόμα και στον φυσικό δίσκο.

Το επόμενο βήμα είναι να δημιουργήσουμε έναν ή περισσότερους λογικούς τόμους για το σύστημα μας

```
lvcreate -n windows2008 -L 20G vgDomU
```

Στη συνέχεια αν επιλέξεις να γίνει η εγκατάσταση από εξωτερικό δίσκο ή από flash-drive πρέπει πρώτα να τον φορτώσεις.

Αρχικά βλέπουμε τους διαθέσιμους δίσκους με φορμάτ NTFS

```
fdisk -l | grep NTFS
```

δημιουργούμε έναν φάκελο για τον φορτώσουμε εκεί:

```
mkdir -p /media/demo-flash-drive
```

και τέλος ολοκληρώνουμε την φόρτωση:

```
mount -t ntfs /dev/sdc1 /media/demo-flash-drive
```

Κάνουμε και τον κατάλληλο έλεγχο για να δούμε αν έγινε σωστά


```
ls -latr /media/demo-flash-drive
```

Τώρα αν θέλεις η εγκατάσταση να γίνει από οπτικό δίσκο:

```
mkdir -p /dev/cdrom  
mount -t iso9660 /dev/cdrom /cdrom
```

Τώρα ήρθε η ώρα να δημιουργήσουμε το αρχείο ρύθμισης παραμέτρων

```
nano /etc/xen/windows2008.cfg
```

που θα πρέπει να μοιάζει όπως παρακάτω:

```
import os, re  
  
arch = os.uname()[4]  
if re.search('64', arch):  
    arch_libdir = 'lib64'  
else:  
    arch_libdir = 'lib'  
  
kernel = "/usr/lib/xen-4.0/boot/hvmloader"  
builder='hvm'  
memory = 1024  
shadow_memory = 8  
name = "winserver2008"  
vif = [ 'type=ioemu, bridge=xenbr0' ]  
acpi = 1  
apic = 1
```

```
disk = ['phy:/dev/vgDomU/windows2008,hda,w', 'file:/media/demo-flash-  
drive/server2008.iso,hdc:cdrom,r']  
  
#disk = ['phy:/dev/vgDomU/windows2008,hda,w']  
  
device_model = '/usr/' + arch_libdir + '/xen/bin/qemu-dm'  
  
#-----  
  
# boot on floppy (a), hard disk (c) or CD-ROM (d)  
  
boot="dc"  
  
sdl=0  
  
vnc=1  
  
vncconsole=1  
  
vncpasswd=""  
  
serial='pty'  
  
usbdevice='tablet'
```

Όπως μπορείς να παρατηρήσεις, ο hvmloader έχει αντικαταστήσει τον πυρήνα του Linux για τους παραεικονικούς φιλοξενούμενους. Ο hvmloader «εξαπατά» το φιλοξενούμενο και πιστεύει ότι τρέχει απευθείας σε αληθινό υλικό και επίσης παρέχει κάποιο είδος bios. Το μοντέλο συσκευών , qemu-dm παρέχει στους χρήστες με εικονικές συσκευές όπως κάρτες δικτύου και σκληρούς δίσκους.

Τώρα είμαστε σχεδόν έτοιμοι να δημιουργήσουμε το εικονικό μας σύστημα πλήρους εικονικοποίησης

```
xm create windows2008.cfg
```

Το DomU έχει τώρα δημιουργηθεί, αλλά όπως μπορείς να φανταστείς δε μπορείς να το χειριστείς όπως τα υπόλοιπα παραεικονικοποίησης καθώς χρειάζεσαι κάποια διεπαφή για να αλληλεπιδράσει με το χρήστη.

Αν ανατρέξεις παραπάνω στο αρχείο ρύθμισης παραμέτρων θα δεις τις εξής ρυθμίσεις:

```
vnc=1  
vncconsole=1  
vncpasswd=""
```

Το vnc είναι ένα γραφικό σύστημα διαμοιρασμού της επιφάνειας εργασίας ενός συστήματος που χρησιμοποιεί το πρωτόκολλο RFB (Remote Frame Buffer) για να μπορείς να χειριστείς απομακρυσμένα ένα σύστημα από έναν άλλο υπολογιστή. Επίσης μεταφέρει τα γεγονότα του πληκτρολογίου και του ποντικιού από τον έναν υπολογιστή στον άλλο. Όλα αυτά τα καταφέρνει μέσω του δικτύου.

Πριν μπορέσουμε να συνδεθούμε πρέπει πρώτα να αλλάξουμε κάποιες ρυθμίσεις στο Xend:

```
nano /etc/xen/xend-config.sxp
```

Θέλουμε ο server να ακούει σε όλες τις διεπαφές και επίσης αν θέλουμε να θέσουμε κάποιον κωδικό. Αν όχι το αφήνουμε κενό.

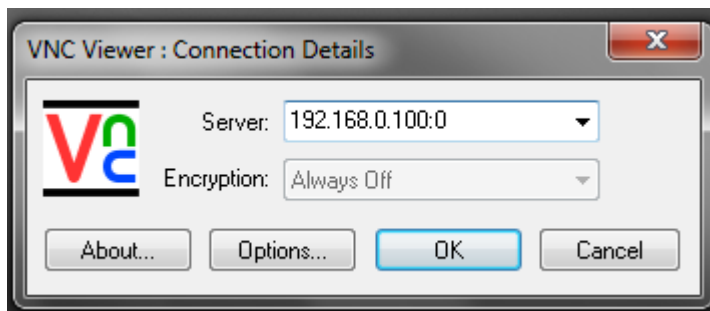
```
(vnc-listen '0.0.0.0')  
(vncpasswd 'password')
```

Το σώζουμε και κάνουμε επανεκκίνηση το Xend:

```
/usr/lib/xen-4.0/bin/xend restart
```

Έπειτα μπορούμε να συνδεθούμε από έναν άλλο υπολογιστή, καθώς στο δικό μας δεν έχουμε κάνει εγκατάσταση κάποιου γραφικού περιβάλλοντος και να συνεχίσουμε κανονικά την εγκατάσταση.

Εγώ προσωπικά βρήκα το [real vnc viewer](#) ένα πολύ καλό εργαλείο που ανταπεξήλθε στις απαιτήσεις μου.



Απλά γράφουμε την ip του Dom0 ακολουθούμενο από άνω κάτω τελεία και το 0 (π.χ. 192.168.0.100:0) αν πρόκειται για το πρώτο εικονικό μηχάνημα που χρησιμοποιεί το vnc, για το επόμενο :1 κοκ.

Μετά το τέλος της εγκατάστασης, πρέπει να κλείσουμε το DomU και να κάνουμε κάποιες αλλαγές στο αρχείο ρυθμίσεων του, ώστε να μην προσπαθεί να ξεκινήσει κάθε φορά από το cd-rom και το αρχείο iso.

```
xm shutdown winserver2008  
nano /etc/xen/windows2008.cfg
```

Και βγάζουμε από το σχόλιο:

```
disk = ['phy:/dev/vgDomU/windows2008,hda,w']
```

Και βάζουμε σε σχόλιο τη σειρά και αλλάζουμε τη σειρά εκκίνησης των μέσων, ώστε να μοιάζει τελικά ως εξής:

```
#disk = ['phy:/dev/vgDomU/windows2008,hda,w', 'file:/media/demo-flash-  
drive/server2008.iso,hdc:cdrom,r']  
disk = ['phy:/dev/vgDomU/windows2008,hda,w']  
boot="c"
```

Διαχειρίζοντας το Xen

Αυτόματο ξεκίνημα των DomUs

Παραπάνω είχαμε επέμβει στη συμπεριφορά των DomUs όταν κλείνουν ή επανεκκινούν και είχαμε απενεργοποιήσει το αυτόματο σώσιμο και εκκίνηση από εκείνο το σημείο που είχαν σωθεί. Αυτό το κάναμε όχι μόνο για λόγους χώρου στο δίσκο αλλά και για λόγους καλύτερης και ομαλότερης λειτουργίας του συστήματος. Όμως έτσι κάθε φορά που για κάποιο λόγο έκλεινε-επανεκκινούνταν το Dom0 θα έπρεπε να εκκινούμε χειροκίνητα τα DomUs μας.

Ευτυχώς υπάρχει ένας εύκολος τρόπος για να γίνεται αυτό αυτόματα. Αρχικά δημιουργούμε τον εξής φάκελο (που είναι αυτός που κοιτάζει το xen όταν εκκινεί το Dom0 για το αν πρέπει να εκκινήσει και κάποιο άλλο σύστημα)

```
mkdir /etc/xen/auto
```

και δημιουργούμε μια συμβολική ένωση του αρχείου ρύθμισης παραμέτρων από τον έναν φάκελο στον άλλο

```
ln -s /etc/xen/testdebian.cfg /etc/xen/auto/
```

Η ένωση αυτή σημαίνει ότι όποια αλλαγή γίνεται στο ένα αρχείο θα γίνεται και στο άλλο.

Backup!

Από τη στιγμή που δημιουργήσουμε τα εικονικά μα μηχανήματα και βεβαιωθούμε ότι δουλεύουν σωστά, είναι ώρα να κρατήσουμε κάποιο αντίγραφο πριν προβούμε σε ενέργειες που μπορεί να προκαλέσουν κάποια ζημιά. Αυτό μπορούμε να το κάνουμε κρατώντας ένα αντίγραφο-ομοίωμα (image) του σκληρού δίσκου. Ένα καλό εργαλείο για αυτό είναι το dd.

Πριν δημιουργήσεις το αντίγραφο βεβαιώσου ότι το εικονικό μηχάνημα δε λειτουργεί γιατί αλλιώς μπορεί να έχουμε ασυνεπή αντίγραφο.

Το dd έπειτα μπορεί να χρησιμοποιηθεί ως εξής:

```
dd if=/dev/vgDomU/testdebian-disk of=/backup/testdebian-disk.img  
bs=20M
```

lf: είναι το αρχείο εισόδου, δηλαδή ο τόμος που θέλουμε να κρατήσουμε αντίγραφο

of: είναι το αρχείο εξόδου, δηλαδή το αντίγραφο μας

bs: είναι το μέγεθος των blocks, το μέγεθος των 20mb επιτρέπει γρήγορη ανάγνωση και εγγραφή

Έπειτα για συμπίεση μπορούμε να χρησιμοποιήσουμε το bzip2.

```
aptitude install bzip2 pbzip2
```

Μπορούμε ακόμα με τη βοήθεια των linux pipes να το κάνουμε αυτόματα:

```
dd if=/dev/vgDomU/testdebian-disk bs=20M | pbzip2 | dd  
of=/backup/testdebian-disk.img.bzip2 bs=20M
```

Στη συνέχεια αρκεί να μεταφέρεις το αντίγραφο σε έναν άλλο δίσκο είτε κάποια άλλη δικτυακή θέση για περισσότερη ασφάλεια.

Αντιθέτως για την αποσυμπίεση:

```
dd if=/backup/testdebian-disk.img.bzip2 bs=20M | pbzip2 -d | dd  
of=/dev/vgDomU/testdebian-disk bs=20M
```

Αν δε μπορείς να έχεις για πολύ ώρα το DomU κλειστό, τότε μπορείς να το τερματίσεις, να κρατήσεις ένα lvm snapshot, να επανεκκινήσεις το DomU και να κάνεις τα παραπάνω με βάση το snapshot. Επίσης αν απλά θέλεις να δοκιμάσεις κάτι μπορείς να κρατήσεις το snapshot να δοκιμάσεις ότι θέλεις στο εικονικό σου μηχάνημα και στη συνέχεια αν πάει κάτι στραβά να επιστρέψεις στην προηγούμενη κατάσταση.

Επανεκκίνηση κωδικού

Αν για οποιοδήποτε λόγο ξεχάσεις τον κωδικό από οποιοδήποτε εικονικό σου μηχάνημα, μπορείς να τον αλλάξεις πολύ εύκολα με την προϋπόθεση βεβαία ότι ο δίσκος σου δεν είναι κρυπτογραφημένος. Αυτό μπορεί να γίνει είτε μέσω του Dom0 είτε ενός άλλου συστήματος linux από live-cd.

Αρχικά φορτώνουμε το ξένο σύστημα αρχείων. Εδώ το φορτώνουμε στο /mnt.

```
mount /dev/vgDomU/testdebian-disk /mnt
```

και έπειτα με την εντολή,

```
chroot /mnt
```

το σύστημα θα αντιμετωπίζει το /mnt σαν /

και έτσι αλλάζουμε τον κωδικό απλά χρησιμοποιώντας την εντολή

```
Passwd
```

για να γυρίσουμε στην αρχική κατάσταση τώρα:

```
Exit  
umount /mnt
```

Αν πιστεύεις ότι ο υπολογιστής σου πρέπει να είναι περισσότερο ασφαλής, πρέπει να χρησιμοποιήσεις κρυπτογράφηση σκληρού δίσκου.

Σημείωση: Αν έχεις έναν φιλοξενούμενο χωρίς καθόλου πρόσβαση στο διαδίκτυο και θέλεις να εγκαταστήσεις κάτι καινούργιο σε αυτό μπορείς να κάνεις την ίδια διαδικασία και να χρησιμοποιήσεις την εντολή aptitude αντι για τη passwd

Εκχώρηση περισσότερης υπολογιστικής δύναμης

Το Xen χρησιμοποιεί έναν προγραμματιστή για τη διανομή της υπολογιστικής δύναμης του επεξεργαστή σε κάθε domain. Σε κάθε ένα από αυτά έχει ανατεθεί ένα ειδικό βάρος (weight) και ένα μέγιστο μέγεθος (cap). Το πρώτο είναι υπεύθυνο για έναν αναλογικό διαμοιρασμό της υπολογιστικής δύναμης και το cap είναι μια προαιρετική ανώτερη τιμή στο χρόνο που καταναλώνει κάθε DomU στη χρήση του επεξεργαστή. Για παράδειγμα, ένα domain με βάρος 512 θα λαμβάνει διπλάσια υπολογιστική δύναμη από ένα domain με βάρος 256. Το εύρος του βάρους που μπορείς να χρησιμοποιήσεις κυμαίνεται από το 1 μέχρι το 65535. Το cap από την άλλη μπορεί να πάρει τιμές από 0-(αριθμο των πυρήνων του επεξεργαστή που χρησιμοποιεί το domain)*100. Έτσι για παράδειγμα σε ένα domain που χρησιμοποιεί 4 πυρήνες, η τιμή 100 σημαίνει ότι θα χρησιμοποιεί μόνο ένα φυσικό πυρήνα, 50 μισό πυρήνα, 400 και τους 4 κοκ. Η τιμή 0 σημαίνει ότι δεν υπάρχει ανώτερο όριο.

```
xm sched-credit -d Domain-0
```

Name	ID	Weight	Cap
Domain-0	0	256	0

Έπειτα κάνουμε την αλλαγή:

```
xm sched-credit -d Domain-0 -w 512
```

Επιβεβαίωση:

```
xm sched-credit -d Domain-0
```

Name	ID	Weight	Cap
Domain-0	0	512	0

Σύνδεση πυρήνων επεξεργαστή σε συγκεκριμένο domain

Συχνά ένα εικονικό μηχάνημα χρειάζεται συγκεκριμένους πόρους για να δουλέψει σωστά. Σε ένα σύστημα με πολυπύρηνο επεξεργαστή ένας ή περισσότεροι πυρήνες μπορούν να ανατεθούν σε ένα domain. Από προεπιλογή ένα domain χρησιμοποιεί όλους τους διαθέσιμους πυρήνες. Παρόλα αυτά μπορούμε να περιορίσουμε ένα μηχάνημα να χρησιμοποιεί μόνο τους δυο και μάλιστα δυο συγκεκριμένους, κάτι το οποίο όμως δεν είναι απαραίτητο. Αυτό μπορεί να γίνει στο αρχείο ρύθμισης παραμέτρων του domain προσθέτοντας:

```
cpus = "2,3"  
vcpus = 2
```

Έπειτα έλεγξε το αποτέλεσμα:

```
xm vcpu-list
```

Name	ID	VCPUs	CPU	State	Time(s)	CPU Affinity
node1	4	0	2	-b-	1.9	2-3
node1	4	1	3	-b-	2.8	2-3

Διαχείριση μνήμης Ram στο Dom0

Στο αρχείο ρύθμισης του Xend /etc/xen/xend-config.sxp μπορούμε να ρυθμίσουμε τη συμπεριφορά του Dom0 κατά τη λειτουργία, όταν παρουσιαστεί ανάγκη στο σύστημα για περισσότερη μνήμη Ram. Υπάρχει η ρύθμιση:

```
(enable-Dom0-ballooning yes)
```

Αλλάζοντας την κατάσταση αυτή σε «ναι», σημαίνει ότι όταν το σύστημα χρειάζεται περισσότερη Ram, το το Dom0 θα αποδεσμεύει από το Dom0 για να τη διαθέσει στο σύστημα. Αν νομίζεις ότι κάτι τέτοιο θα ωφελήσει το σύστημα σου μπορείς να το δοκιμάσεις.

Αυτό βεβαίως μπορεί να δημιουργήσει πρόβλημα στη σωστή λειτουργία του Dom0 γι αυτό είναι καλό να συνοδεύεται με μια ακόμα ρύθμιση που θα δηλώνει την ελάχιστη μνήμη που θα κρατήσει το Dom0 για τη λειτουργία του.

```
(Dom0-min-mem 196)
```

Διαχείριση μνήμης Ram στα DomUs

Μπορούμε να αλλάξουμε τη διαθέσιμη μνήμη Ram σε ένα DomU, ακόμα και όταν αυτό λειτουργεί. Τρέχοντας την εντολή:

```
xm list
```

μπορούμε να δούμε τη διαθέσιμη μνήμη σε κάθε DomU

Αρχικά πρέπει να δηλώσουμε τη μέγιστη μνήμη που θα μπορεί να λάβει το domain μας:

```
xm mem-max testdebian 1024
```

και έπειτα να δηλώσουμε πως θέλουμε να αλλάξουμε τη ποσότητα της μνήμης:

```
xm mem-set testdebian 512
```

Με αυτό τον τρόπο μπορούμε να αυξήσουμε ή να μειώσουμε τη διαθέσιμη μνήμη στο DomU.

Πρέπει να σημειωθεί ότι την επόμενη φορά που θα εκκινήσει το εικονικό μηχάνημα η διαθέσιμη μνήμη του θα επανέλθει πάλι στα προηγούμενα επίπεδα. Εκεί δηλαδή που ορίζει το αρχείο ρυθμίσεων του.

Σώσιμο και επαναφορά

Η κατάσταση των εικονικών μηχανημάτων μπορεί να σωθεί και να αποκατασταθεί. Αυτό μπορεί να γίνει με την εντολή

```
xm save testdebian /var/tmp/testdebian.save
```

Εννοείται ότι μετά την εντολή αυτή το DomU δεν είναι διαθέσιμο. Μπορείς να το επαναφέρεις με την εξής εντολή:

```
xm restore /var/tmp/testdebian.save
```

Εντολές διαχείρισης των domain

Για να δεις τους πόρους που καταναλώνει κάθε domain, τρέξε την εντολή:

```
xm top
```

ή

```
xentop
```

Για να δεις πόση ώρα λειτουργούν τα domains:

```
xm uptime
```

Πέρασμα συσκευών στα DomUs (PCI Passthrough)

Όταν εμβαθύνεις και δημιουργήσεις κάποια εικονικά μηχανήματα, μπορεί να αποφασίσεις ότι χρειάζεσαι να χρησιμοποιείς περισσότερα τις συσκευές υλικού του υπολογιστή. Είδαμε ότι αυτό δεν είναι πρόβλημα σε ότι αφορά συσκευές που μπορούν να δουλέψουν αφαιρετικά, όπως η κάρτα δικτύου, αλλά αν θέλεις πραγματικά να αποκτήσεις τον έλεγχο μιας συσκευής από το DomU, τότε τα πράγματα γίνονται πιο σύνθετα. Ο επόπτης Xen έχει τον απόλυτο έλεγχο πάνω στο υλικό, αλλά για τις περισσότερες συσκευές ο επόπτης δεν ξέρει και δε θα έπρεπε να ξέρει για λόγους προστασίας, πώς να χρησιμοποιήσει μια συσκευή. Ο έλεγχος των συσκευών περνάει στο Dom0 και από εκεί θα πρέπει να τον πάρεις.

Στο εγχειρίδιο του Xen λέει ότι υπάρχει υποστήριξη USB 1.1 για κάποιες (!) συσκευές στα domains πλήρους εικονικοποίησης. Επίσης υπήρχε σε παλιότερες εκδόσεις υποστήριξη για πέρασμα usb συσκευών που απαιτούσε ειδικούς οδηγούς και στο Dom0 και στα υπόλοιπα domains, οδηγοί που όμως δε γράφτηκαν για τον καινούργιο πυρήνα. Το xen έχει μια λογική ότι οι περισσότερες εφαρμογές θα γίνουν για επαγγελματικά περιβάλλοντα και το xen είναι πιο πολύ για υπολογιστές προσωπικής χρήσης.

Παρόλο όμως που δε μπορούμε να περάσουμε μια θύρα ή μια συσκευή USB, μπορούμε να περάσουμε όλη την PCI(e) κάρτα στο DomU. Αυτό όμως απαιτεί υποστήριξη υλικού για VT-d ή IOMMU. Αν δεν υπάρχει τέτοια υποστήριξη από τη μητρική τότε πρέπει να αποδεχτείς το γεγονός ότι δε θα μπορεί να χρησιμοποιείς το υλικό σου από τα DomUs ή πρέπει απλά να αγοράσεις καινούργιο υλικό που θα υποστηρίζει VT-d ή IOMMU.

Επίσης αν σκοπεύεις να περάσεις στο DomU μια κάρτα γραφικών, πρέπει να προσφέρει και αυτή υποστήριξη για κάτι τέτοιο.

Όπως μπορείς να φανταστείς, το πέρασμα συσκευών σε ένα domain και η ζωντανή μεταφορά δε μπορούν να συνυπάρξουν για προφανής λόγους.

Αν θέλεις να περάσεις μια συσκευή σε ένα DomU πρέπει πρώτα να το «ελευθερώσεις» από το Dom0.

Αρχικά πρέπει να δούμε τις συσκευές που έχουμε στο Dom0:

```
Lspci
```

00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge (rev 01)

00:06.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Secondary PCI Express Bridge (rev 01)

00:1a.0 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #4 (rev 02)

00:1a.1 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #5 (rev 02)

00:1a.2 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #6 (rev 02)

00:1a.7 USB controller: Intel Corporation 82801I (ICH9 Family) USB2 EHCI Controller #2 (rev 02)

00:1b.0 Audio device: Intel Corporation 82801I (ICH9 Family) HD Audio Controller (rev 02)

00:1c.0 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 1 (rev 02)

00:1c.3 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 4 (rev 02)

00:1c.4 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 5 (rev 02)

00:1c.5 PCI bridge: Intel Corporation 82801I (ICH9 Family) PCI Express Port 6 (rev 02)

00:1d.0 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #1 (rev 02)

00:1d.1 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #2 (rev 02)

00:1d.2 USB controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #3 (rev 02)

00:1d.7 USB controller: Intel Corporation 82801I (ICH9 Family) USB2 EHCI Controller #1 (rev 02)

00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev 92)

00:1f.0 ISA bridge: Intel Corporation 82801IR (ICH9R) LPC Interface Controller (rev 02)

```
00:1f.2 IDE interface: Intel Corporation 82801IR/IO/IH (ICH9R/DO/DH) 4
port SATA Controller [IDE mode] (rev 02)

00:1f.3 SMBus: Intel Corporation 82801I (ICH9 Family) SMBus Controller
(rev 02)

00:1f.5 IDE interface: Intel Corporation 82801I (ICH9 Family) 2 port SATA
Controller [IDE mode] (rev 02)

01:00.0 VGA compatible controller: Advanced Micro Devices, Inc.
[AMD/ATI] RV770 [Radeon HD 4850]

01:00.1 Audio device: Advanced Micro Devices, Inc. [AMD/ATI] RV770
HDMI Audio [Radeon HD 4850/4870]

04:00.0 SATA controller: JMicron Technology Corp. JMB363 SATA/IDE
Controller (rev 02)

04:00.1 IDE interface: JMicron Technology Corp. JMB363 SATA/IDE
Controller (rev 02)

05:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 02)

06:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 02)

07:06.0 FireWire (IEEE 1394): Texas Instruments TSB43AB23 IEEE-
1394a-2000 Controller (PHY/Link)
```

Όταν αποφασίσουμε ποια συσκευή θέλουμε να περάσουμε πρέπει να γνωρίζουμε τους ακριβείς αριθμούς του. Έστω ότι θέλουμε να περάσουμε αυτή τη συσκευή:

```
01:00.0 VGA compatible controller: Advanced Micro Devices, Inc.
[AMD/ATI] RV770 [Radeon HD 4850]
```

Μαθαίνουμε τους αριθμούς του, τρέχοντας την εντολή:

```
lspci -n | grep 01:00.0
```

Και μας δίνει σαν αποτέλεσμα:

```
01:00.0 0300: 1002:9442
```

Έχοντας αυτές τις πληροφορίες, μπορούμε να ξεκινήσουμε, αφαιρώντας στην αρχή τη συσκευή από το Dom0:

```
modprobe pci_stub
```

```
echo "1002 9442" > /sys/bus/pci/drivers/pci-stub/new_id  
echo "0000:01:00.0" > /sys/bus/pci/devices/0000\:01:00.0/driver/unbind  
echo "0000:01:00.0" > /sys/bus/pci/drivers/pci-stub/bind
```

Έπειτα τρέχοντας την εντολή:

```
xm pci-list-assignable-devices
```

μπορείς να δεις τις διαθέσιμες συσκευές για πέρασμα στα DomUs. Αν όλα πήγαν καλά θα πρέπει να δούμε εδώ τη συσκευή που ελευθερώσαμε από το Dom0.

Στη συνέχεια για να περάσουμε τη συσκευή σε ένα DomU πρέπει να τη προσθέσουμε στο αρχείο ρυθμίσεων:

```
nano /etc/xen/windows2008.cfg
```

και να προσθέσουμε στο τέλος:

```
pci = [ '01:00.0' ]
```

στη συνέχεια αν αποφασίσουμε να περάσουμε και άλλες συσκευές αφού τις ελευθερώσουμε από το Dom0, απλά τις προσθέτουμε με κόμμα:

```
pci = [ '01:00.0', '01:00.1' ]
```

Σημείωση: Καλό θα ήταν να έχουμε κάνει από πριν εγκατάσταση τους οδηγούς της συσκευής στο DomU

Εγκατάσταση οδηγών GPLPV

Οι οδηγοί αυτοί αναπτύχθηκαν από τον James Harper και επιτρέπει να την αλλαγή τον προσομοιωμένων συσκευών με παραεικονικές συσκευές. Με αυτούς τους νέους οδηγούς βελτιώνεται η ταχύτητα των νέων συσκευών καθώς θα χρησιμοποιεί το δίκτυο και θα σταματάει τη χρήση των οδηγών που βρίσκονται στο Dom0.

Οι ελεγμένοι οδηγοί GPLPV είναι διαθέσιμοι [εδώ](#).

Συμπεράσματα

Ολοκληρώνοντας την ανάγνωση τόσο του θεωρητικού όσο και του πρακτικού κομματιού της εργασίας, ο αναγνώστης θα να είναι σε θέση να γνωρίζει τι είναι η εικονικοποίηση, γιατί κάποιος να την προτιμήσει καθώς και αν κάτι τέτοιο θα τον συνέφερε.

Το μεγαλύτερο όμως όφελος θα είναι η γνωριμία με το Xen Hypervisor. Έναν επόπτη εικονικών μηχανημάτων που θα του επιτρέπει εύκολα και κυρίως δωρεάν να φτιάξει και να διαχειριστεί εικονικά μηχανήματα.

Το Xen έφτασε πολύ κοντά στην απαξίωση και νομίζω ότι το πάθημα έγινε μάθημα και από εδώ και στο εξής θα βελτιώνεται συνεχώς. Η μεταφορά στο καινούργιο site συντελεί σε αυτό το δρόμο καθώς και οι πολλές παρουσίες που έχει το Xen σε διάφορα συνέδρια ανά τον κόσμο.

Ακόμα όμως και αν κάποια στιγμή το xen σταματήσει να ανταποκρίνεται στις ανάγκες του τεχνολογικού κόσμου, τα ερεθίσματα που θα έχει ο αναγνώστης από αυτήν την εργασία θα τον βοηθήσει να μεταβεί πιο εύκολα σε έναν άλλο επόπτη, όπως ο KVM.

Βιβλιογραφία

Chisnall David - The definitive Guide to the Xen Hypervisor

Matthews Jeanna N.; Eli M. Dow; Todd Deshane; Wenjin Hu; Jeremy Bongio; Patrick F. Wilbur; Brendan Johnson - Running Xen: A Hands-On Guide to the Art of Virtualization

Prabhakar Chaganti (2008) Xen Virtualization – A practical Handbook

Takemura Chris and Luke S. Crawford - The Book of Xen: A Practical Guide for the System Administrator by

Williams David E., Juan Garcia - Virtualization with Xen

William von Hagen - Professional Xen Virtualization

<http://www.howtoforge.com>

<http://www.colestock.com>

<http://www.techinferno.com>

<http://www.debian.org/doc/user-manuals.en.html>

<https://wiki.debian.org/Xen>

http://wiki.xen.org/wiki/Main_Page

<http://www.virtuatopia.com>

<https://help.ubuntu.com/>

<http://wiki.xenproject.org>

<http://lists.xen.org>

<http://serverfault.com>

<http://www.virtualmin.com>

<http://www.xen-tools.org>

<http://www.debian-administration.org>