Wprowadzenie do wirtualizacji

J. Apelski G. Chimosz S. Kurek

Wydział Matematyki, Informatyki i Mechaniki

9 listopada 2007

Spis treści

Wirtualizacja
Co to jest?

- Warstwa pośrednia interfejsu producent-konsument
- „Abstrakcja zasobów”
  - Jeden jako wiele
  - Wiele jako jeden
- Wirtualizacja platform
  - Sprzętowych — emulacja
  - Systemowych
- Wirtualizacja zasobów systemowych
  - RAID,NAT,VPN,pamięć wirtualna...
  - ...o tym nie mówimy :)

Pojęcia

- Host,gospodarz
- Gość
- Hypervisor, VMM
  - Natywny
  - Gościnny

Kryteria oceny

Czego oczekujemy od wirtualizacji?

- Kryteria Popka-Goldberga
  - Równoważność
  - Kontrola zasobów
  - Wydajność

Zastosowania

- Inne OS, architektury
  - Dostęp do aplikacji na różne platformy
- Testowanie
  - Implementacja systemów operacyjnych
  - Potencjalnie niebezpieczne aplikacje
  - Dydaktyka
- Szybsze restarty
  - Zapisywanie obrazu serwera
- Konsolidacja serwerów
  - Wiele serwerów logicznych na jednej maszynie
  - Niski średni współczynnik wykorzystania
  - Bezpieczeństwo
- Izolacja użytkowników
  - Konta shellowe
- Honey-poty

Problemy z architekturą x86

- Pierwotna architektura x86 nie spełniała kryteriów Popka-Goldberga
- Bardzo trudno było stworzyć maszynę wirtualną
- Stronicowanie, mechanizm zabezpieczenia, segmentacja w założeniu miały być zarządzane tylko przez jeden system operacyjny
- Jedynym rozwiązaniem było programowe ominięcie różnych problemów powstających przy wirtualizacji

Problemy z architekturą x86

Instrukcje

- Istnieją instrukcje, które pozwalają na odczytanie rejestrów systemowych w trybie użytkownika
- Nie ma sprzętowych mechanizmów wykrywania wszystkich instrukcji, które nie działają poprawnie w wirtualnym środowisku
- Należy tak przerywać wykonanie kodu (tworzyć breakpointy), żeby nadzorca mógł odpowiednio reagować
- Modyfikacje jakich użyjemy nie mogą zostać wykryte przez gościa

Problemy z architekturą x86

Dalsze problemy

- Należy tak ustawić breakpointy, żeby nigdy nie dopuścić do wykonania kodu, który nie był jeszcze sprawdzony
- Trzeba uważać na kod, który zagląda do już przeskanowanej części programu
- Jeszcze większym problemem jest kod, który modyfikuje przeskanowaną część
- Istnieją sztuczki, dzięki którym tworzy się stronę z kodem wykonywalnym, którego nie można odczytać i zapisać

Rozszerzenia architektury x86

sprzętowe rozwiązanie problemu

Intel Virtualization Technology (IVT)

- zwana też Vanderpool
- Rozszerzenie architektury x86 o wirtualizację
- VT-x - dla 32-bitowych procesorów Intela
- VT-i - dla procesorów IA-64 (Itanium)

AMD virtualization (AMD-V)

- nazwa wewnętrzna Pacifica
- dla 64-bitowych procesorów x86
- stworzone równolegle z rozwiązaniem Intela

Ring $(-1)0 \div 3$

- Hierarchiczne domeny – Ringi są zorganizowane w hierarchię.
- Sprzętowe wsparcie dla SO.
- Tradycyjny podział:
  - 0 – kernel
  - 1, 2 – sterowniki urządzeń, wirtualizacja
  - 3 – procesy użytkownika
- Ring -1 jest jednym ze sposobów wirtualizacji SO o zamkniętych źródłach (AMD-V, Intel VT)

Rodzaje wirtualizacji

- Możliwe jest kilka różnych podejść do wirtualizacji.
- Poszczególne rodzaje różnią się:
  - Możliwościami
  - Wydajnością
  - W praktyce: zastosowaniem

Rodzaje wirtualizacji

- Maszyny wirtualne
- Parawirtualizacja
- Wirtualizacja systemów operacyjnych
- Emulacja
- Emulacja API
- Specyficzny przypadek: UML

Maszyny wirtualne
Wprowadzenie

- Wirtualizacja pełna, właściwa...
  - Wirtualizacja natywna - przy wsparciu sprzętowym
- Symulacja wszystkich elementów sprzętu
- Ta sama architektura!
- Dowolne programy i (teoretycznie) systemy operacyjne
- Problem: przechwytywanie niebezpiecznych instrukcji
  - Pułapka i symulacja

Maszyny wirtualne
Przykłady

- Virtual PC
- VMWare Workstation
- QEMU (+ kqemu)
- Virtual Box

Microsoft Virtual PC

Przedstawienie

- Oryginalny producent - Connectix
  - Pierwsze wydanie - czerwiec 1997
  - Początkowo emulator x86 dla Mac-ów na PowerPC
  - Pierwsza wersja na MS Windows: czerwiec 2001
- Przejęcie przez Microsoft w lutym 2003
- Platforma docelowa: MS Windows
  - ...oraz Mac OS na PowerPC
- Darmowy od lipca 2006

Microsoft Virtual PC

Możliwości

- Wersja dla x86
  - Zasada działania - pełna wirtualizacja
    - Wersja 2007 - wykorzystuje AMD-V i IVT
  - Wsparcie dla sieci, napędów CD/DVD, dźwięku, drukarek...
  - Virtual Machine Additions - dla systemów Microsoftu
    - M.in. zwiększenie wydajności, wsparcie dla akceleracji 3D.
  - Zapisywanie obrazu maszyny
  - Wydajność - prawie natywna
- Wersja dla PowerPC
  - Emulator x86 dla Mac OS X
  - Bardzo niska wydajność
  - Praktycznie zażegnany

Microsoft Virtual PC

Systemy Gościa

- Obsługuje:
  - DOS
  - Windows 3.1 - Vista
  - OS/2
  - Linux - modyfikowany lub ograniczony
  - Mac OS X - zabrania licencja, dodatkowo mechanizmy anty-pirackie
- Zastosowania:
  - Hobbistyczne - prosty i darmowy
  - Dydaktyka
  - Development

VMWare Workstation

Przedstawienie

- Producent: VMWare
  - VMWare Server
  - VMWare Fusion
- Pierwsze wydanie - 1999
- Platformy docelowe: Windows i Linux
- 30-dniowy darmowy okres próbny
- VMWare Player - darmowy
  - Nie tworzy obrazów, jedynie uruchamia
  - Można je ściągnąć z Internetu
  - Trochę gorsza wydajność, brak SMP

VMWare Workstation

Możliwości

- Zasadniczo platforma wirtualizacyjna
  - Od wersji 6.0 - eksperymentalne wsparcie dla parawirtualizacji
  - Otwarty protokół VMI, włączony do jądra 2.6.21
- Rozbudowana funkcjonalność
  - Praktycznie wszystkie urządzenia peryferyjne
  - Zarządzanie snapshotami
  - Rozbudowane wsparcie dla sieci
  - Ułatwione administrowanie
  - Szyfrowanie
- Wydajność - prawie natywna

VMWare Workstation

Systemy Gościa

- Obsługuje:
  - DOS
  - Windows
  - Linux
  - FreeBSD
  - Netware
  - Solaris
  - „Broadest Host & Guest Operating System Support”
- Starsze wersje miewają problemy z nowymi jądrami

VMWare Workstation

Zastosowania

- Produkcja i testowanie oprogramowania
- Demonstracje produktów u klienta
- Również dydaktyka

QEMU

z kqemu w systemie Hosta

- QEMU bez kqemu - dynamiczna rekompilacja
- Moduł jądra KQEMU - akcelerator dla x86
- Kod użytkownika uruchamiany w procesorze, emulacja tylko kodu jądra
- Dla większości aplikacji: wzrost wydajności z 10%-20% do 80%-90% prędkości natywnej
- Nadal brak sterowników dla systemu Gościa - aplikacje multimedialne kuleją
- Istnieją porty na FreeBSD i Windows

VirtualBox

- Oryginalny producent - InnoTek
- Styczeń 2007 - VirtualBox Open Source Edition
- Wersja płatna - do darmowego użytku dla celów niekomercyjnych
- Zasada działania:
  - W miarę możliwości wirtualizuje kod gościa
  - Kod uprzywilejowany Gościa uruchamiany w ring 1
  - W razie potrzeb dokonuje dynamicznej rekompilacji - na bazie rekompilatora QEMU
  - Wsparcie dla IVT, eksperymentalne wsparcie dla AMD-V
- Elastyczny

Dostosowanie Gościa do wirtualizacji

- Zamiast emulować sprzęt, lepiej dostarczyć API – hypercall
- ...najlepiej mieć wsparcie w procesorze ;)

Parallels Desktop for Mac

- Wymaga MacOS X, wspiera w pełni Intel VT
- VirtualPC for Mac – protoplasta, obecnie porzucony
- Zaawansowane współdzielenie plików pomiędzy Hostem a Gościem
- Snapshoty
- DirectX z Wine
- Oficjalnie wspierane systemy Gościa:
  - Windows 3.1 - Vista
  - Linux (SUSE, Red Hat, Debian, Fedora Core, Mandriva, Xandros)
  - FreeBSD, OpenBSD 3.8
  - OS/2, eComStation
  - Solaris
  - MS-DOS

XEN

- Ian Pratt, University of Cambridge, 2003
- Wspiera IA-32, x86-64, IA-64 oraz PowerPC
- Host – Linux lub NetBSD (zmodyfikowane)
- Gość – różne systemy unixowe (modyfikowane); Windows (od Xen 3.0)
- Wolny (free), ale istnieje komercyjna implementacja: Citrix XenServer Enterprise Edition
- Novell SUSE 10, Red Hat's RHEL 5/Fedora 7, Sun Microsystems Solaris 10, Debian Etch, Ubuntu 6.10

XEN

wspierane systemy Gościa

- Linux 2.6.23+, patche dla wcześniejszych
- Minix
- Plan 9 from Bell Labs
- NetBSD 2.0+
- OpenBSD
- FreeBSD
- OpenSolaris
- NetWare
- GNU/Hurd/Mach (gnumach-1-branch-Xen-branch)
- OZONE (Xen v1.2)
- oraz Windows XP :)

XEN

Zalety:

- Szybkość
- Przemieszczanie domeny „w locie” (60-300 ms)

Wady:

- Konieczność modyfikowania jądra gościa lub posiadania odpowiedniego procesora
- Skomplikowany system
- Dość trudna instalacja (w porównaniu do UML czy VirtualPC)

Wirtualizacja OS

Wstęp

- Szybsze niż inne rodzaje wirtualizacji
- Wiele instancji tego samego systemu operacyjnego
- Wszystkie instancje pracują na tym samym jądrze
- Wygodne do testowania nowego oprogramowania
- Zwiększa bezpieczeństwo pracy dzięki odizolowaniu różnych części systemu

OpenVZ

- Tworzy wirtualne środowiska (VE)
- Obsługuje Linuksa
- VE jest tylko 1-3% wolniejsze
- Checkpoint - możliwość przeniesienia VE na inny serwer bez potrzeby restartu
- obsługuje do 64 procesorów i 64 GB pamięci RAM
- Pojedyncze VE można przeskalować tak, by wykorzystywało wszystkie zasoby komputera

Virtuozzo

- Stworzony przez SWsoft
- Oparty na OpenVZ
- Jedno wspólne jądro
- Obsługuje Linuksa od 2001, Windows od 2005
- Wikipedia twierdzi, że jest to jedyny program tego typu obsługujący Windows
- Pozwala uruchomić setki VE zachowując funkcjonalność serwera
- Można przekierować zasoby do różnych VE bez konieczności restartu
- Zawiera narzędzia do zarządzania wieloma VE naraz

FreeBSD Jail
na FreeSBD

- Mechanizm wirtualizacji na poziomie OS dla FreeBSD
- Pozwala administratorowi podzielić system na niezależne podsystemy - Jaile
- Każdy Jail to VE - posiada własne pliki, procesy, użytkowników i zasoby sieciowe
- Jaile są od siebie odseparowane, co zapewnia bezpieczeństwo
- Root Jaila nie ma dostępu do całego systemu

Emulacja
Wstęp

- Pełne emulowanie innej maszyny
- Zapewnia największą przenośność
- Duży spadek wydajności

Bochs

- Emulator PC klasy x86 i amd64
- Kevin Lawton
- Pierwotnie program komercyjny
- Popularny wśród twórców systemów operacyjnych
- Emuluje wszystkie elementy komputera
- Istnieją wersje pod Windows, Linuksa oraz Mac OS X

DOSBox

- Emulator PC klasy x86
- Open source
- Tworzy środowisko DOS-owe
- Przeznaczony głównie do uruchamiania starych gier
- Wersje pod Linuksa, OpenBSD, FreeBSD, Windows, Mac OS X, OS/2, Palm OS, RISC OS oraz BeOS

QEMU

bez kqemu w systemie Hosta

- Fabrice Bellard
- Bez kqemu może pracować w trybie pełnej emulacji
- Pozwala emulować komputer klasy x86, amd64, alpha, mips oraz sparc
- Emuluje procesor jak i inne podzespoły docelowej maszyny
- Może służyć do uruchomienia wielu wirtualnych komputerów różnego typu na jednym komputerze
- Dynamiczna translacja przyspiesza szybkość emulacji
- Wersje pod Linuksa, Windows, Mac OS X oraz OpenSolaris

Wine

- Wine Is Not an Emulator
- Bob Amstadt i Eric Youngdale
- Zamiast emulować, Wine zapewnia interfejs, którym można zastąpić jądro Windows
- Z małymi problemami uruchamia większość oprogramowania
- Posiada odpowiedniki większości bibliotek dll
- Moduł Direct3D jest ciągle rozwijany (np. dodawana jest obsługa Pixel Shaderów)
- Może używać oryginalnych bibliotek dll Microsoftu
- Napisany na Linuksa, ale są też wersja dla Mac OS X, FreeBSD oraz Solarisa

UML

- Jeff Dike, Paolo Giarrusso (aka BlaisorBlade), Bill Stearns
- user-mode-linux.sourceforge.net (w przebudowie)
- umlwin32.sourceforge.net
- Pierwotnie na x86, porty na IA-64, AMD64, PowerPC

UML
zastosowania

- Serwery wirtualne (linode.com, od $19.95 / miesiąc; linuxzoo.net)
- Wirtualne sieci (z wyjściem na świat) (www.netkit.org) (VNUML)
- X – poprzez sieć mogą wyświetlać się na hoście
- honey-pot
- debugowanie jądra

UML
dlaczego UML?

Zalety:

- qemu – szybkość
- Xen, VMWare ESX – prostota (instalacja, koncepcja)
- *hostfs* – banalnie łatwy dostęp do plików hosta
- „krówki” (pliki Copy On Write)
- **MADV_REMOVE** – należy wkrótce oczekiwać wparcia hotplug pamięci w UML

Wady:

- szybkość

PTRACE_SYSEMU vs. PTRACE_SYSCALL

test	bez SYSEMU	z SYSEMU	zysk
getpid 10.000.000	real 6m16.461s user 0m5.077s sys 0m58.462s	real 3m55.052s user 0m5.962s sys 0m46.385s	60%
MySQL „run-all-tests”	real 18m38.329s user 2m37.370s sys 6m3.040s	real 13m57.353s user 2m17.370s sys 3m54.310s	33%
make bzImage modules	real 13m35.457s user 4m18.885s sys 1m40.481s	real 13m5.980s user 4m18.096s sys 1m40.115s	4%

Na podstawie: sysemu.sourceforge.net

SKAS
Separate Kernel Address Space

- W waniliowym Linuksie 2.6.0+ SKAS1
- SKAS3 v8.2 dla Linuksa 2.6.20-
- Szybciej i bezpieczniej niż z Traced Thread
- *[...]kernel build is almost twice as fast with skas mode as with tt mode, and is within 30% of the host's time. Bill Stearns saw a script's running time drop from 50 seconds to 14 seconds, almost a quadrupling in speed.*
- *It's SMP-safe, and it's running stably on SMP hosts.*

Porównanie wirtualizatorów

część 1

Rodzaj	Przykład	Zgodność	Wydajność
Emulacja	Bochs	Brak	2
Wirt. pełna	VMWare	Architektura	3 - 4
Parawirt.	Xen	Hypercall-e	4+
Wirt. OS	OpenVZ	Jądro	4 - 5
UML	UML :)	Jądro	3-
	UML+SKAS	Jądro	4+
Emulacja API	Wine	Architektura	4 - 5

Skala ocen od 1 (najgorzej) do 5 (najlepiej).

Porównanie wirtualizatorów

część 2

Rodzaj	Przykład	Równoważność	Bezpieczeństwo
Emulacja	Bochs	5	5
Wirt. pełna	VMWare	4+	5
Parawirt.	Xen	4+	5
Wirt. OS	OpenVZ	4+	4+
UML	UML :)	4+	4-
	UML+SKAS	4+	5
Emulacja API	Wine	Ile napiszą :)	3

Skala ocen od 1 (najgorzej) do 5 (najlepiej).

Transmeta
Crusoe i Efficeon

- Procesor Very Long Instruction Word
- Intel Itanium (core: Merced)
- Porażka biznesowa
- Morfing kodu (programowy ale zrealizowany w krzemie)
- Można zmienić mikrokod już po wyprodukowaniu układu
- Tłumaczenie instrukcji odbywa się tylko raz; pętle i odwołania do tego samego fragmentu kodu są pobierane w postaci rozkazów VLIW z pamięci cache
- Optymalizacja tłumaczonego kodu (usuwanie zbędnych instrukcji!)
- Torvalds i inne sławy miały swój wkład w projekt...

Blue Pill

- Rootkit zaprojektowany przez Joannę Rutkowską w 2006
- Teoretycznie umożliwia „całkowite” i „niewykrywalne” przejęcie kontroli nad systemem ofiary
- Bazuje na AMD-V
- Kiedy system „połknie niebieską tabletkę”:
  - Blue Pill przechwytuje system „w locie” i ładuje go do swojej VM
  - Użytkownik ofiary pracuje bez straty wydajności (...?) i bez możliwości wykrycia Blue Pill (...?)
- Utrata części zasobów systemowych na rzecz Blue Pill:
  - Czas procesora: można przestawiać zegar...
  - Pamięć: trzeba ukrywać rootkit i *jednocześnie* nie zmniejszyć rozmiaru dostępnej pamięci
- Prototypowa implementacja z opublikowanymi źródłami

lguest

- Ładowalny moduł *lg* - prosty mechanizm parawirtualizacyjny
- *„l" stands for „light"* - tylko 6000 linii
- Cel: prostota, łatwość użytkowania i modyfikacji
- Brak rozbudowanej funkcjonalności
- Niska wydajność - 25% do 50% natywnej
- Od wersji jądra 2.6.23 - włączony do kernela