

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky
Vyšší odborná škola informačních služeb v Praze

Lukáš Váša

Virtualizace v Linuxu

Bakalářská práce

2007/2008

Prohlašuji, že jsem bakalářskou práci na téma Virtualizace v Linuxu zpracoval samostatně a použil pouze zdrojů, které cituji a uvádím v seznamu použité literatury.

V Praze dne :

Podpis :

Obsah

1	Úvod	7
2	Co znamená pojem virtualizace	9
3	Použitá terminologie	10
4	Techniky virtualizace	11
5	Použití virtualizace	13
5.1	Konsolidace	13
5.2	Oddělené prostředí	13
5.3	Testování a vývoj	14
6	Red Hat Enterprise Linux	16
7	Xen	17
7.1	Popis Xenu	17
7.2	Typy instalace Domain 0	18
7.3	Instalace Domain 0	18
7.4	Domain U	20
7.5	Konfigurace Domain U	21
7.6	Instalace paravirtualizované Domain U	23
7.6.1	Instalace pomocí nástroje virt-install	24
7.6.2	Instalace pomocí virt-mamageru	25
7.6.3	Manuální instalace pomocí xm	27

7.7	Správa virtuálních domén.....	28
7.7.1	Správa pomocí příkazů.....	28
7.7.2	Grafický Virt-manager	30
7.8	Instalace plně virtualizované Domain U	33
8	VMware.....	36
8.1	Popis VMwaru.....	36
8.2	Typy instalace	37
8.3	Instalace VMwaru do hostitelského systému	37
8.4	Instalace virtuálního stroje	39
8.5	Správa virtuálních strojů	43
9	Migrace virtuálních strojů	45
10	Výkonnostní testy.....	46
11	Závěr.....	49
12	Použité zdroje.....	51

Práce obecně shrnuje výhody použití virtualizace a popisuje jednotlivé principy a technologie používané pro virtualizaci. Virtualizace je demonstrována na dvou nejpoužívanějších produktech pro virtualizaci v Linuxu, tedy Xen a VMware. Jednotlivé produkty jsou postupně popsány a jsou v nich provedeny ukázkové instalace. Práce je zakončena provedením výkonnostních testů.

1 Úvod

Tato práce pojednává o virtualizaci v prostředí operačního systému Linuxu. Cílem práce je objasnění jednotlivých principů a technologií používaných pro virtualizaci, vysvětlení výhod použití virtualizace, popis dvou vybraných produktů a provedení ukázkových instalací v těchto produktech, popis jejich možností a provedení následných testů. Virtualizace je v poslední době hodně diskutované téma a to zejména kvůli jeho mnoha nesporným výhodám oproti standardní instalaci operačního systému. Pro tuto práci mě zajímaly technologie použitelné v podnikovém prostředí, tedy dostatečně stabilní a bezpečné, se zázemmí vývojového týmu, který je schopen dostatečně rychle reagovat na případné vzniklé problémy, ať již bezpečnostní, nebo z hlediska funkčnosti. Produkty musí také nabízet dostatečnou funkcionalitu a komfort pracovního prostředí nutný pro práci v podnikovém prostředí. Dnes již existuje několik řešení, které jsou dostatečně vyspělé, aby mohli být použity v podnikovém prostředí.

V této práci budu srovnávat dvě v dnešní době nepoužívanější řešení, Xen a VMware. Virtualizaci budu provádět pouze s lokálním datovým úložištěm, bez použití sdílených úložišť za použití technologií jako je SAN či iSCSI. V práci však popíši výhody při použití těchto technologií.

Před samotným srovnáváním těchto řešení je potřeba trochu zasvětit do pojmu virtualizace. Nejdříve se zaměřím na popis termínu virtualizace, co je pod tímto pojmem běžně myšleno a v jakých spojitostech se používá. Abych mohl virtualizaci popisovat, je potřeba vysvětlit alespoň základní terminologii. Vysvětlím, co znamenají jednotlivé pojmy a stanovím jejich použití pro tuto práci. Popíši také jednotlivé používané druhy virtualizace. Dále je třeba objasnit výhody a nevýhody použití virtualizace, proč tuto technologii vůbec používat, co mi přinese její použití, proč investovat finanční prostředky pro přechod na tuto technologii. Popíši několik nejtypičtějších druhů použití virtualizace a budu se na nich snažit demonstrovat výhody virtualizace oproti standardnímu řešení, kdy na jednom hardware běží jeden operační systém.

Postupně předvedu instalaci virtuálních strojů v Xenu a následně ve VMware. V případě řešení Xen provedu instalaci jak paravirtualizovaného stroje, tak plně virtualizovaného. Jako operační systém bude vždy použit Red Hat Enterprise Linux update 1, tedy jako operační systém pro privilegovanou doménu i jako operační systém instalovaný do virtuálních strojů.

Budu tak činit z důvodů následné srovnatelnosti při výkonnostních testech, aby prostředí byla co nejvíce stejná a rozdíly ve výkonu byly způsobeny opravdu jen výkonem virtualizačního softwaru.

Virtualizační software Xen existuje v několika implementacích a verzích. Pro tuto práci použiji Xen implementovaný v Red Hat Enterprise Linuxu verze 5 update 1 (dále je RHEL 5). Jak jsem již uvedl, bude tak zároveň sloužit jako operační systém v privilegované doméně Dom0 i jako operační systém v jednotlivých virtuálních doménách DomU. Přesná verze použitého Xenu je 3.0.3.

Virtualizační software VMware použiji VMware server for Linux verze 1.0.4. Tato verze VMware na rozdíl od verze ESX nemá vlastní operační systém, proto jako základní platformu použiji opět Red Hat Enterprise Linux verze 5 update 1. Bude tak docíleno lepší srovnatelnosti obou produktů, neboť veškeré instalované operační systémy budou stejné.

Testy budou provedeny na jednom fyzickém serveru postupně, opět kvůli konzistenci prostředí a lepší srovnatelnosti. V testech bude změřeno zatížení samotného RHEL jako základní údaj o vytížení hardwaru samotným operačním systémem. Poté postupně změřím výkon v jednotlivých nainstalovaných virtuálních doménách.

2 Co znamená pojem virtualizace

Pojem virtualizace, ve spojitosti s osobními počítači, nejčastěji vyjadřuje jakousi abstrakci uživatelského prostředí od fyzického hardware. Tedy technologii, kdy jsme schopni zdroje jednoho fyzického systému poskytnout několika izolovaným a na sobě nezávislým a izolovaným prostředím. Jsme tedy schopni sdílet zdroje, mít kontrolu nad jejich využíváním a to nám umožní lepší správu i využití těchto zdrojů. Pojem virtualizace jako takový není žádnou novinkou, idea je již delší dobu známa a v dnešní době existuje již celkem dost implementací postavených na různých technikách, z nichž některé v této práci postupně popíši.

Velmi důležité je, že virtualizované prostředí zůstává konzistentní s nevirtualizovaným prostředím. Virtuální stroje jsou vidět jako klasické stroje v síti. Veškerá komunikace mezi virtualizovanými i nevirtualizovanými stroji mezi sebou je čistě záležitost standardní síťové komunikace. Toto je důležité pro provozované aplikace, nejsou tedy nutné žádné změny v rámci těchto aplikací.

Virtualizace je tedy technika, kdy umožníme na jednom počítači současný běh několika izolovaným prostředím, takzvaným doménám. V každé z těchto domén může běžet samostatný operační systém. Normálně umožňuje počítači běh pouze jedné instance operačního systému, pokud chceme spustit jinou instanci, musíme počítač do této instance přebootovat, jedná se takzvanou dual-boot nebo multi-boot konfiguraci, v závislosti na počtu instalovaných systémů. Pokud potřebujeme současný běh dvou instancí operačního systému, musíme mít dva fyzické stroje. Toto je klasický model používaný na i386 architekturách.

Naproti tomu, pokud použijeme virtualizaci, rozdělíme jeden fyzický počítač do několika virtuálních prostředí - strojů. Každý virtuální stroj používá svůj vlastní virtuální hardware. Každý obsahuje vlastní instanci operačního systému, tyto domény jsou od sebe navzájem izolované a mohou běžet současně. Díky izolovanosti není potřeba, aby byli tyto systémy stejné, je možné použít různé operační systémy, nebo rozdílné verze jednotlivých operačních systémů. Každý systém tedy používá své vlastní virtuální prostředky, jako jsou síťové karty, souborové systémy, procesory, paměť a další. Tyto virtuální systémové zdroje používají tedy stejné sdílené fyzické zdroje, avšak tyto skutečné zdroje jsou virtuálním strojům skryty.

3 Použitá terminologie

Pro virtualizaci tedy potřebujeme jakousi mezivrstvu, která nám bude obsluhovat skutečné fyzické zdroje systému a poskytovat je jednotlivým virtuálním strojům. Tato mezivrstva se nazývá hypervisor. Někdy se můžeme setkat s pojmem virtual machine monitor. Hypervisor je tedy software, který dovoluje současný běh virtuálních strojů, v každém tomto virtuálním stroji je nainstalován jeden operační systém. Hypervisor řídí a spravuje jednotlivé virtuální stroje, zprostředkovává jim přístup k virtuálním procesorům, paměti, diskům, síti a dalším. Současně jim brání v přístupu ke skutečnému fyzickému hardwaru a k ostatním virtuálním strojům. Hypervisorů existuje několik druhů. Pokud hypervisor běží přímo na fyzickém hardware, mluvíme o takzvaném hypervisoru typ 1. Tento typ hypervisoru používá například Linux KVM, Vmware ESX Server, nebo IBM z/VM. Hypervisor typ 2 je software instalovaný na hostujícím operačním systému, nad kterým pak běží jednotlivé virtuální stroje, tedy guesti. Tento typ hypervisoru používá například VMware Workstation, Microsoft Virtual Server, nebo Parallels desktop for Mac. Další typ hypervisoru, je takzvaný hybrid hypervisor, který používá například Xen implementovaný v Red Hat Enterprise Linuxu, jenž bude dále popisován v této práci. Tento typ hypervisoru běží také přímo na fyzickém hardware jako hypervisor typ 1, avšak je přímo závislý na systémové podpoře a driverech zařízení jednoho virtuálního stroje. Tento virtuální stroj se nazývá privilegovaná doména, Xen bez ní nedokáže fungovat. Jde tedy o určitou mutaci hypervisoru typ 1.

Privilegovaná doména je virtuální stroj, který má speciální práva pro řízení jak hypervisoru, tak ostatních domén. V terminologii Xenu je tato doména nazývána Doména 0, nebo také Dom0. Je to první doména, jenž musí být spuštěna po startu Xenu. Zatímco hypervisor zprostředkovává prostředky jednotlivým doménám a provádí scheduling, Doména 0 se používá pro řízení běhu jednotlivých domén. V Doméně 0 jsou tedy instalovány utility pro práci s ostatními doménami, pro jejich instalaci, řízení, uspání apod. Někdy se také můžeme setkat s pojmem management domain, nebo management console. V této doméně tedy běží operační systém, jenž se nazývá host. Privilegovaná doména je vždy jen jedna.

Doména U, nebo jenom doména, je další virtuální stroj běžící nad hypervisorem. Těchto domén může již nad hypervisorem běžet takové množství, které potřebujeme a které nám samozřejmě dovolí výkon hardware. V nich běží operační systémy, jenž budu nazývat geust. Normální domény nemají právo na řízení ostatních domén, ani přístup k fyzickému hardware přiděleného ostatním doménám. K tomuto účelu je nutno použít Domain 0.

4 Techniky virtualizace

Pro vlastní virtualizaci se používá několik technik, já zde popíši hlavně ty, které používají produkty, o kterých je tato práce, Xen a VMware. Jedná se o paravirtualizaci, emulaci a dynamický překlad a plnou virtualizaci.

Paravirtualizace potřebuje pro svůj běh modifikované jak jádro hosta, tedy Domain0, tak jádro ostatních virtuálních strojů, tedy gestů. Oba dva operační systémy musí rozumět paravirtualizaci, je potřeba spolupráce kernelu a hypervisoru pro zacházení s některými instrukcemi, jádra tedy musí být upravena. Z tohoto důvodu nelze například do Domain U nainstalovat neupravená MS Windows, lze instalovat pouze operační systémy, kde lze jádro patchovat, tedy například Linux. Výhodou však je, že stačí upravené pouze jádro, není potřeba upravovat ostatní software běžící ve virtuální instanci operačního systému. Další výhodou je, že paravirtualizace je velmi rychlá, výkon virtuálního stroje by se měl blížit výkonu operačního systému instalovaného přímo na fyzickém hardware. Paravirtualizace je znatelně rychlejší než například plná virtualizace nebo emulace. Pokud tedy provozujeme virtualizované guesty stejného typu a na stejné architektuře jako hosta, je vhodné použít právě paravirtualizaci. Další výhodou je nepotřebnost hardwarové podpory v procesoru. Tento typ virtualizace je tedy vhodný i pro starší modely procesorů, které nemají hardware podporu, jako jsou i386 nebo starší x86 architektury. Paravirtualizace je základní technikou virtualizace v Xenu.

Hardwarová emulace umožňuje souběžný běh virtuálních strojů bez modifikace operačních systémů jednotlivých guestů. Emulace je technika, při níž hypervisor softwarově plně simuluje hardware pro virtuální stroj. Je to umožněno díky analýze všech instrukcí a jejich následné příslušné konverzi před vstupem do procesoru. Emulace může být použita pro plnou virtualizaci s procesory bez hardwarové podpory. Znamená to tedy, že kvůli nutnosti překladu instrukcí je plná virtualizace s emulací pomalejší, než nativní plná virtualizace. Pro zrychlení se používá technika zvaná dynamický překlad, při níž jsou instrukce analyzovány ještě před jejich spuštěním a umožňuje bezpečně instrukce spustit nemodifikovaně a překládá citlivé instrukce ještě před jejich provedením. Přeložené instrukce mohou být cachovány pro případně zrychlení spuštění stejného kódu. Dynamický překlad navíc ještě optimalizuje často spouštěné instrukce za běhu. Optimalizace je však velmi výpočetně náročná a hypervisor proto spotřebovává více procesorového času. Plná virtualizace s dynamickým překladem je základní technika tradičně používaná ve VMware.

Nativní plná virtualizace neboli hardwarově asistovaná plná virtualizace umožňuje běh nemodifikovaných operačních systémů ve virtuálních strojích přesně jako kdyby byly spuštěny na fyzickém hardwaru. Prostředí virtuálního stroje s virtuálními rozhraními se tváří přesně jako fyzický hardware. Přístup k fyzickému hardware skrze virtuální zařízení je zprostředkováno hypervisorem. Tento typ plné virtualizace potřebuje ke svému fungování hardwarovou podporu přímo v procesoru. Procesory typu Intel používají technologii zvanou Intel VT-x, neboli Intel virtualization technology, procesory AMD používají technologii AMD-V, neboli Pacifica. Díky této podpoře jsou všechny instrukce spuštěny přímo v procesoru bez jakýchkoliv změn.

5 Použití virtualizace

5.1 Konsolidace

Mezi nejdůležitější argumenty pro použití virtualizace bezesporu patří konsolidace. Při klasickém použití, tedy jeden fyzický počítač na jeden operační systém, se kupuje hardware tak, aby výkonnostně pokryl zátěžové špičky. K těmto špičkám dochází jen občas, zbytek času je hardware nevyužit. Podle průzkumů je standardní vytížení hardware při tomto druhu použití kolem dvaceti procent. Při použití virtualizace můžeme několik instancí serverů provozovat virtuálně na jednom fyzickém hardwaru. Docílíme tak lepší využitelnosti hardware, neboť špičky budou rozloženy v čase a výkon hardware bude použit několika instancemi serverů současně. Dojde tak navíc ke značnému snížení nákladů, neboť není potřeba kupovat tolik fyzických serverů. Další výhodou je nižší spotřeba elektřiny a samozřejmě i z toho plynoucí menší nároky na chlazení. Dojde také ke snížení nároků na potřebné místo v serverovnách.

Při použití sdílené storage, jako například za použití technologií SAN nebo iSCSI, je dokonce možné servery za chodu přesouvat z jednoho fyzického hardware na jiný. Dochází pouze k přesunu malého množství nezbytných dat a obrazu paměti, neboť vlastní data serveru jsou umístěna na sdílené storage. Tímto způsobem můžeme s virtuálními servery dynamicky pracovat, přesouvat je například v případě potřeby na silnější hardware. Snížíme tím také náklady na správu, neboť v případě nutnosti zásahu na serveru a jeho vypnutí, není potřeba plánovat výpadek do ranních hodin a platit administrátorům příplatky, ale stačí pouze virtuální stroj pře migrovat na jiný fyzický hardware, server vypnout, provést údržbu a po zapnutí virtuální stroj pře migrovat zpět.

5.2 Oddělené prostředí

Stále více se také používají oddělená prostředí pro jednotlivé služby, hlavně z bezpečnostních důvodů. Služby velmi často sdílejí různé systémové prostředky či knihovny a navzájem se mohou ovlivňovat. Při použití virtualizace pak například při pádu jedné služby nejsou ovlivněny služby jiné a dojde tedy jen k částečnému odmítnutí služeb. Případně lze použít rozdílné knihovny, které by si jinak mohli překážet. Při použití virtualizace opět není nutné služby provozovat na odděleném hardware, ale je možné je pustit současně na jednom fyzickém hardware v oddělených virtuálních strojích. Díky absolutní separaci virtuálních strojů nám pád jedné virtuálního stroje neovlivní ty ostatní. Také při kompromitaci jedné

služby nám útočník nemůže ovlivnit jiné služby, pokud jsou puštěny v jiných virtuálních strojích.

Podobného principu lze také použít v případě, že máme odladěnou aplikaci, která nám běží na určité verzi operačního systému. Není tedy žádoucí měnit verzi operačního systému a provádět nové odladění aplikace. Může ale nastat případ, že z nějakého důvodu potřebujeme nasadit novější hardware, který ovšem není podporován starším operačním systémem. Díky virtualizaci můžeme toto elegantně vyřešit, neboť do privilegované domény nainstalujeme novější operační systém a skrze virtuální zařízení poskytneme toho zařízení starší verzi operačního systému, jenž pustíme ve virtuálním stroji.

V některých případech je také nutné poskytnout klientům plný přístup k operačnímu systému, například skrze ssh. Pokud na jednom serveru provozujeme službu pro více klientů, je toto téměř nemožné vyřešit, neboť root práva přidělená jednomu klientovi by mohla ohrozit bezpečnost dat pro ostatní klienty. Pokud použijeme virtualizaci, můžeme pro každého klienta provozovat vlastní virtuální stroj a tudíž mu můžeme ponechat plná práva k systému. Za prvé neuvidí data jiných klientů a za druhé si každý může provádět nastavení podle vlastní potřeby, aniž by to ovlivnilo ostatní.

5.3 Testování a vývoj

Velké výhody přináší virtualizace také v oblasti testování a vývoje. Tester nebo vývojář může mít na jednom systému množství testovacích virtuálních strojů. Pokud dojde k pádu virtuálního stroje, neovlivní to privilegovanou doménu ani ostatní testovací stroje. Je možné si udělat snapshot stavu před provedením testu a po pádu virtuálního stroje se opět vrátit k tomuto stavu. Navíc instalace virtuálního stroje z připraveného konfiguračního souboru je mnohem rychlejší, než klasická instalace. Z Domain 0 také můžeme lépe sledovat chod virtuálního stroje a následně lépe analyzovat proč k pádu vůbec došlo.

Virtualizace také přináší nové možnosti v oblasti práce s instalacemi a licencemi systémů. Ohledně ceny může ušetřit finanční náklady, jelikož některé operační systémy nelicencují další instalace virtuálních strojů na jednom fyzickém hardware jako nové instalace, tudíž není potřeba pro tyto virtuální stroje další softwarová licence. Například Red Hat Enterprise Linux, jež budu používat v této práci pro virtualizaci v Xenu jako základní platformu, dovoluje v jedné subskripci pro verzi AS nekonečné množství virtuálních instalací bez potřeby koupě

další subskripce. Toto bude později podrobněji rozepsáno v cenovém srovnání jednotlivých řešení, avšak finanční úspora je zde zřejmá.

Virtuální stroje je také možné použít pro určité předem definované instalace, například pro často používané služby. Budeme mít jasně definovanou konfiguraci virtuálního stroje pro určitou službu, instalace takového stroje je pak velmi rychlou záležitostí. Navíc budeme mít jednotný typ konfigurace pro virtuální stroje pro určitou službu. Nemusíme tedy udržovat seznamy nainstalovaných software balíčků a jednotlivých konfigurací, místo toho budeme mít kompletní předinstalovaný virtuální stroj s určitou konfigurací, verzemi software, odladěným prostředím.

6 Red Hat Enterprise Linux

V této práci je jako operační systém pro virtuální stroje, pro Xen privilegovanou doménu i hostující operační systém ve VMware použit Red Hat Enterprise Linux 5 update 1 standard edition. Red Hat Enterprise Linux je vyspělá open source distribuce vhodná jak pro nasazení v malých firmách až po kritické prostředí a velké datové sklady.

Red Hat Enterprise Linux je komerční produkt, jeho podpora se kupuje v rámci takzvaných subskripcí na jeden až tři roky. V této subskripci je obsažena podpora, jak telefonická tak e-mail, přístup do programových kanálů a přístup do správcovského prostředí Red Hat Network. Red Hat Network je modulární platforma pro správu systémů přístupná přes webové rozhraní. Od verze Red Hat Enterprise Linux 5 je podporována virtualizace, přímo v distribuci je implementován virtualizační software Xen. Red Hat Enterprise Linux je certifikován pro nasazení na hardware od mnoha výrobců hardware, zároveň mnoho aplikací je certifikováno pro provoz na Red Hat Enterprise Linuxu. Aktuální seznam pro všechny verze RHEL je dostupný na stránkách www.redhat.com. Red Hat Enterprise Linux je postavený na Red Hatem sponzorovaném projektu Fedora, kde je většina softwaru důkladně prověřena v reálném použití, než je zařazena do Red Hat Enterprise Linuxu.

Red Hat nabízí dvě verze pro serverové nasazení, RHEL standard edition a RHEL advanced edition. RHEL standard je limitován pro použití s dvěma fyzickými patičkami pro procesory. Počet podporovaných virtualizovaných strojů je omezen na čtyři, u advanced platform není počet virtuálních strojů omezen. Součástí RHEL 5 advanced platform jsou i produkty pro podporu sdílení dat, Red Hat Cluster suite a Red Hat Global File System.

7 Xen

7.1 Popis Xenu

Xen je open-source projekt, původně vyvinutý na Cambridge University, nadále vyvíjený open-source komunitou. Je šířen pod licencí GPL. Je široce akceptovaný mnoha výrobci softwaru a softwaru, jako je IBM, Red Hat, Intel, AMD, Novell a mnoho dalších. Xen je základní platforma používaná pro virtualizaci v Red Hat Enterprise Linuxu.

Podporuje 32bit a 64bit x86 platformu, v testování je Intel Itanium 2. Základní technika virtualizace v Xenu je paravirtualizace, pro 32bit x86 systémy je vyžadována PAE (physical adress extension) podpora v procesoru. Tuto podporu mají Intel kompatibilní procesory od řady Pentium Pro. Pro provoz nemodifikovaných operačních systémů používá plnou hardwarově asistovanou virtualizaci, potřebuje tudíž podporu virtualizace přímo v procesoru. Intel podporuje virtualizaci technologií jménem Intel Virtualization Technology, AMD má technologii nazvanou Virtualization extension. Jestli daný procesor potřebné technologie obsahuje ověříme v souboru `/proc/cpuinfo`. Pro paravirtualizaci musí obsahovat řádek `flags` obsahovat `pae`, pro plnou virtualizaci na Intelu `vmx` nebo `svm` na AMD.

Red Hat Enterprise Linux podporuje virtualizaci v domain 0 od verze RHEL 5. Jako paravirtualizovaného hosta můžeme použít RHEL 5 nebo RHEL 4 update 5 a novější.

Následující tabulka udává, jaké architektury lze instalovat ve virtuálních strojích s použitím jakých virtualizačních technologií a na jakých architekturách Dom0.

Architektura	DomU x86_32	DomU x86_64
Dom0 x86_32	Paravirtualizace / plná virtualizace	nelze
Dom0 x86_64	Plná virtualizace	Paravirtualizace / plná virtualizace

Jako hypervisor je použit hybrid hypervisor, je tedy spouštěn přímo na fyzickém hardware, avšak pro svůj běh potřebuje privilegovanou doménu. Tato privilegovaná doména, tedy domain 0 používá nativní Linuxové ovladače, ty jsou pak základem pro front-end virtuální zařízení pro jednotlivé gesty – domain U. Díky tomu, že využívá existující model ovladačů v

Linuxu, aplikace není nutné žádným způsobem modifikovat. V domain 0 jsou k dispozici veškeré nástroje pro správu a kontrolu gestů. Xen používá dynamické alokace zdrojů, jako je CPU, paměti, disky apod.

Při startu systémů nejdříve Grub načte hypervisoru, který vytvoří domain 0, tedy privilegovanou doménu a načte se modifikované jádro a initrd. Poté se již standardním způsobem spustí RHEL 5 v domain 0. V domain 0 je nutné spustit službu xend, která umožní utilitám instalaci a kontrolu ostatních domain U a také umožní správu hypervisoru. Domény U jsou kompletně separované od hypervisoru a mezi sebou navzájem.

7.2 Typy instalace Domain 0

Instalace Dom0 existují dva typy, tlustá a tenká Dom0. Toto rozdělení je z bezpečnostních důvodů, neboť z Dom0 lze ovlivňovat běh jak hypervisoru, tak ostatních DomU. Proto je nutné Dom0 co nejlépe zabezpečit. Z toho důvodu se na serverech používá takzvaná tenká instalace, při které instalujeme do Dom0 pouze minimální instalaci RHEL 5 plus nejpotřebnější nástroje pro správu hypervisoru a ostatních DomU.

Dom0 při tenké instalaci tedy neobsahuje žádné síťové služby a není tedy tak velké riziko její kompromitace skrze tyto služby. Slouží pouze jako základna pro provoz ostatních DomU, do kterých se tyto síťové služby instalují. Aby byly stabilní ostatní DomU, musí být stabilní i právě Dom0.

Při tlusté instalaci instalujeme i do Dom0 síťové služby a další software. Dom0 tedy slouží jako normální pracovní prostředí. Toto sice není vhodné pro serverové nasazení, ale často se používá například při testování nebo vývoji, kde Dom0 používáme k normální práci a DomU provádíme testy. Potřebujeme navíc znát výsledky komunikace mezi hypervisorem a DomU pro lepší odladění služeb.

Pro instalaci RHEL do DomU je doporučeno pro tuto doménu vyčlenit alespoň 512MB paměti a minimálně 1GB místa na disku. Virtuální disk pro DomU může být implementován jako soubor na disku v Dom0, nebo je možné použít vlastní diskový oddíl. Je možné použít i LVM svazek a využít tak všech výhod LVM, jako jsou například snapshoty.

7.3 Instalace Domain 0

Jak již bylo řečeno, pro správnou funkci Xenu je nutná spolupráce privilegované domény, tedy Domain 0. Instalace Domain 0 probíhá tak, že se pouze nainstaluje upravené jádro s

podporou virtualizace. Toto upravené jádro při bootu samo zajistí, že se instalovaný Linux spustí v Domain 0 a ne standardním způsobem na čistém hardware. RHEL podporuje upravené jádro s podporou virtualizace až od verze RHEL 5. Po instalaci upraveného jádra je ještě nutné zajistit instalaci a spuštění xend démona, skrze kterého za použití příslušných utilit probíhá správa jak samotného hypervisoru, tak ostatních Domain U. Samozřejmě nakonec ještě doinstalujeme případné další utility pro správu, jako například virt-manager.

Virt-manager je virtual machine manager, slouží pro snadnější správu a instalaci virtuálních strojů. Je licencován pod GPL a používá knihovnu libvirt. Lze pomocí něj sledovat výpis a statistiky všech instalovaných virtuálních strojů, virtuální stroje lze spravovat, připojovat a odebírat virtuální zařízení pro jednotlivé stroje, zprostředkovává přístup do grafické i textové konzole pro virtuální stroje a nabízí automatizovaného průvodce pro instalaci nového virtuálního stroje.

Instalaci upraveného jádra tedy můžeme provést jak při instalaci samotného RHEL, nebo později z repositářů. Nebudu zde popisovat proces instalace samotného Linuxu, to není předmětem této práce, proto ukáži postup instalace na již nainstalovaném Linuxu. Předpokládám standardně nainstalovaný Linux, v případě serveru ideálně do minimální instalace, dále také předpokládám že je Linux subskribovaný, tedy že má přístup do repositářů skrze Red Hat Network. Nejdříve tedy ověříme podporu příslušných instrukcí v procesoru.

```
# grep -color pae /proc/cpuinfo
```

Dále je třeba nainstalovat příslušné softwarové balíky, tedy upravené jádro, démona xend s příslušnými utilitami pro správu a nakonec nainstalujeme grafické rozhraní virt-managera.

```
# yum -y install kernel-xen xen virt-manager
```

Po instalaci jádra by jsme měli zajistit spuštění tohoto jádra po startu systému. Zkontrolujeme tedy, že soubor `/etc/sysconfig/kernel` obsahuje direktivu `DEAULTKERNEL=kernel-xen` a že v souboru `/boot/grub/grub.conf` ukazuje direktiva `default` na upravené jádro. Pokud je tedy upravený kernel na prvním místě, bude direktiva `default=0`.

Dále zajistíme spuštění xend démona.

```
# chkconfig --level 35 xend on
```

Po restartu systému by měla již být podpora virtualizace funkční, což můžeme jednoduše zkontrolovat například pomocí příkazu `xm`, který by nám měl vypsat běžící `Dom0`.

xm list

7.4 Domain U

Jako paravirtualizovaný stroj v Domain U můžeme použít RHEL 4.5 a vyšší nebo RHEL 5.0 a vyšší. Jako plně virtualizovaný můžeme použít kteroukoliv verzi, neboť nepotřebuje žádné úpravy. Paravirtualizace je obecně rychlejší, avšak kvůli potřebě úpravy jádra nelze vždy použít. Starší verze RHEL než 4.5, jiné verze Linuxu než Red Hat Enterprise Linux, či jiné operační systémy než Linux jako třeba Windows je nutno instalovat jako plně virtualizované. Paravirtualizovaný systém se žádným způsobem neliší od standardního, jde pouze o upravené jádro a upravený spouštěč, `anaconda`, který sám pozná, že instalace se provádí do virtuálního stroje. Paravirtualizované a plně virtualizované Domain U mohou běžet zároveň na jednom systému.

Domain U se skládá ze dvou hlavních komponent. Jedna komponenta je konfigurační soubor se specifikacemi, jak se má příslušná doména vytvořit a druhý je blokové zařízení, které simuluje harddisk. Konfigurační soubory k jednotlivým doménám se umísťují standardně do adresáře `/etc/xen`. Každá doména by měla mít svůj konfigurační soubor, který je možné vytvořit buď ručně, nebo bude vytvořen automaticky například při použití `virt-manager`a. Je také možné spustit doménu bez použití konfiguračního souboru za použití utility `xm`.

Jako blokové zařízení je možné použít buď image soubor, v tom případě je standardní umístění v `/var/lib/xen/images`, nebo lze použít samostatný diskový oddíl, logical volume svazek, nebo jiné blokové zařízení. Virtuální stroj normálně nemá přímý přístup k blokovému zařízení, ale pouze přes hypervisor, to umožňuje například při použití sdílené storage přesouvat virtuální stroje za chodu z jednoho fyzického stroje na jiný, pokud je dodržena stejná architektura procesoru obou fyzických strojů.

Pokud je požadavek automatického spuštění DomU po startu Dom0, je potřeba vytvořit link konfiguračního souboru z adresáře `/etc/xen` do adresáře `/etc/xen/auto`. Linkem zajistíme, aby jsme neměli dvě kopie konfiguračního souboru a případně také dvě odlišné verze. `Xendomain` démon po startu prochází adresář `/etc/xen/auto` a podle v něm

umístěných konfiguračních souboru vytvoří příslušné domény. Při ukončení běhu Dom0 jsou všechny ostatní DomU domény automaticky hybernovány, nejsou ukončeny jako Dom0.

7.5 Konfigurace Domain U

Před vlastní instalací popíše jednotlivé direktivy konfiguračního souboru. Konfigurační soubory pro jednotlivé domény jsou umístěny v adresáři `/etc/xen`. Při instalaci skrze grafické rozhraní virt-managera je konfigurační soubor automaticky vygenerován.

Jako první je nutné specifikovat jméno domény, pod kterým bude zobrazena.

```
name = "domain1"
```

Při správě virtuálních domén je možné doménu volat právě pomocí tohoto jména, či je možné použít přidělené ID.

Každá doména musí mít přiřazen alespoň jeden virtuální procesor. Instrukce předávané virtuálnímu procesoru VCPU jsou hypervisorem zpracovány a předány fyzickému procesoru. Je možné nastavit více VCPU než je fyzických procesorů, avšak to by mělo negativní vliv na výkon systému. Je také možné přímo vyhradit pro doménu jeden či více fyzických procesorů. Existují tři direktivy vztahující se k procesoru.

```
vcpus = 2
```

```
cpu = 0
```

```
cpus = 1, 3-4
```

První direktiva `vcpus` udává, kolik virtuálních procesorů je přiřazeno doméně. Druhá direktiva `cpu` udává počet fyzických procesorů přiřazených doméně, přesně jádro procesoru. Pokud použijeme například dvou jádrový procesor, systém jej bude vidět jako dva samostatné procesory. Poslední direktiva `cpus` dovoluje spuštění domény na příslušných fyzických procesorech, tyto procesory však nejsou vyhrazeny jen pro tuto doménu. Doména bude mít pouze dovoleno používat příslušné procesory.

Dále je nutné definovat množství použité paměti. Každá doména potřebuje pro běh operačního systému v ní nainstalovaném dostatek operační paměti. Doporučené minimum pro instalaci RHEL je 512 MB paměti.

```
memory = "512"
```

Po startu systému Dom0 automaticky převezme správu nad veškerou pamětí. Po následném spuštění virtuálního stroje, DomU, je tato paměť odebrána z Dom0. Pokud je DomU zastavena, tato paměť není automaticky navrácena Dom0, zůstává alokována pro příslušnou DomU, aby mohla být případně následně rychle znova spuštěna. Tuto paměť je nicméně možné manuálně znovu přiřadit Dom0.

Pro simulaci pevného disku či jiného blokového zařízení slouží v DomU virtuální blokové zařízení. Virtuální blokové zařízení VBD je pro doménu DomU stejné jako standardní blokové zařízení. Disky mohou být děleny na oddíly a formátovány. První virtuální blokové zařízení je /dev/svda, druhé /dev/xvdb apod. V Dom0 je toto virtuální blokové zařízení buď diskový image soubor, diskový oddíl nebo jiné klasické blokové zařízení. Diskový image soubory jsou standardně uloženy ve /var/lib/xen/images. Tento soubor musí existovat ještě před startem příslušné domény, je tudíž nutné ho buď vytvořit ručně pomocí příkazu dd nebo ho za nás automaticky vytvoří instalátor. Příklad konfigurační direktivy pro namapování diskového image souboru

```
disk = [ 'tap:aio:/var/lib/xen/images/domain1.img, xvda,w', ]
```

a pro standardní sdb1 oddíl

```
disk = [ 'phy:/dev/sdb1,xvda,w', ]
```

Dále je potřeba nakonfigurovat síťové rozhraní pro DomU. Každá virtuální doména může mít definovány až tři síťová rozhraní. V konfiguraci je nutné zadat MAC adresu, která je při použití instalátoru vybrána náhodně automaticky začínající na 00:16:3E, například

```
vif = [ 'mac=00:16:3E:24:27:a0, bridge=xenbr0', ]
```

Pokud tuto MAC adresu volíme sami, můžeme samozřejmě definovat jakoukoliv chceme, měli by jsme ovšem dát pozor na jedinečnost této adresy v lokální síti. Za definicí MAC adresy je definice použitého virtuálního síťového bridže. Xenbr0 je nastaven automaticky při instalaci Xenu. Slouží jako HUB a propojuje všechny domény síťově mezi sebou. Informace o konfiguraci síťového bridže je možné vypsát pomocí příkazu brctl.

Fyzické síťové rozhraní eth0 je přejmenováno na peth0 a je propojeno s bridžem. Jednotlivá virtuální síťová rozhraní jsou pojmenována podle principu vifD.N kde D je domain ID a N je číslo v pořadí. Všechny síťové rozhraní jsou propojeny se síťovým bridžem. Oddělení eth0 a peth0 v Dom0 dovoluje nastavit rozdílná pravidla pro Dom0 a ostatní DomU.

Každá Domain U poskytuje pro práci v této doméně grafickou a textovou konzoli. Grafická konzole emuluje fyzický monitor a je asociována s virtuálním frame bufferem, klávesnicí a myší. Grafická virtuální konzole pracuje stejně jako klasická grafická konzole, ve virtuálním terminálu může být spuštěn X server a může být k němu skrz tento virtuální terminál přistupováno. Standardně se jako grafická konzole používá implementace kompatibilní s VNC. Můžeme k ní přistupovat buď přes grafickou konzoli virt-managera, nebo přes jakéhokoliv VNC klienta. Grafická konzole se zapíná direktivou

```
vfb = [ "type = vnc, vncunused = 1" ]
```

Virtuální textová konzole, neboli sériová konzole se jmenuje /dev/xvc0. Můžeme k ní opět přistupovat skrze virt-managera, nebo ji otevřít z příkazové řádky v Dom0 příkazem `xm console domain`. Sériová konzole může být použita jako systémová konzole místo grafické konzole. Sériová konzole může běžet souběžně s grafickou konzolí, avšak můžeme přesměrovat zprávy vypisované init démonem do této sériové konzole. V příslušné DomU přidáme v souboru /boot/grub/grub.conf příslušnému jádru parametry `console=tty0 console=xvc0`. Zprávy od init démona jsou automaticky posílány pouze na druhou definovanou konzoli v pořadí. Pro možnost přihlášení do sériové konzole je v příslušné DomU nutné přihlášení povolit v /etc/inittab řádkou

```
co:2345:respawn:/sbin/agetty xvc0 9600 vt100-nav
```

a pro root účet povolit přístup v souboru /etc/security přidáním řádku `xvc0`.

7.6 Instalace paravirtualizované Domain U

Při instalaci virtuálního stroje, tedy operačního systému do DomU máme na výběr ze tří variant, které zde postupně popíši. Můžeme použít příkazový řádek a nástroj virt-install, virtuální konzoli můžeme také inicializovat pomocí příkazu `xm`, nebo můžeme použít grafické rozhraní programu virt-manager.

7.6.1 Instalace pomocí nástroje virt-install

Nástroj virt-install je napsán v pythonu a je postaven na knihovně libvirt. Lze jej použít jak interaktivně, tedy že se nás na požadované parametry bude postupně ptát, tak neinteraktivně a vše definovat pomocí parametrů. Nástroj za nás podle zadaných parametrů vytvoří a nastaví doménu, bez nutnosti našeho ručního zásahu, vytvoří konfigurační soubor a případně i image diskový soubor.

Než začneme instalovat virtuální doménu, je nutné se rozhodnout, kam doménu uložíme, jestli na vlastní diskový oddíl, či do diskového image souboru.

Pokud použijeme diskový oddíl, je nutné vytvořit tento oddíl předem. Nyní ukážu postup pro vytvoření LVM svazku. V Dom0 tedy tedy například pomocí Fdisku vytvoříme dostatečně velký nový diskový oddíl, tento oddíl zaregistrujeme jako LVM pomocí příkazu pvcreate. Dále na tomto fyzickém lvm oddíle vytvoříme lvm volume group. V této volume group vytvoříme logický lvm svazek, který již bude použit jako virtuální disk pro virtuální doménu.

```
# pvcreate /dev/sda5  
  
# vgcreate vg0 /dev/sda5  
  
# lvcreate --size 5G --name lv_domain1 vg0
```

Měli by jsme mít nainstalovaný softwarový balík vnc, kvůli utilitě vncviewer. Nyní můžeme začít instalaci. Protože jsou ostatní domény, včetně Dom0 viděny pouze přes síťový most jako síťové rozhraní, je nutné zadat zdroj s instalačním stromem jako síťový, tedy pomocí nfs, ftp, či http. Po zavolání nástroje virt-install se nás systém bude postupně ptát na jméno domény, cestu k instalačním médiím, dále můžeme zadat cestu ke kickstart souboru z definicí instalace, diskový oddíl nebo umístění diskového image souboru, velikost paměti ram pro virtuální stroj, počet virtuálních procesorů a případně můžeme definovat vlastní MAC adresu, nebo ji nástroj vygeneruje za nás.

Můžeme také vše zadat jako parametry a nepoužít interaktivní instalaci.

```
# virt-install --paravirt --name Domain1 --ram 512 --vnc \  
  
--file /dev/vg0/lv_domain1 \  
  
--location http://server.example.com/pub \  

```



```
--extra-args ks=http://server.example.com/ks/domain1.ks
```

Parametrů je možné zadat mnohem více, například typ architektury či typ instalovaného operačního systému, plný výpis je dostupný pomocí `man virt-install`. Místo instalace na zadaný lvm svazek či fyzickou oddíl, je možné instalovat do diskového image souboru.

```
-- file /var/lib/xen/images/Domain1.img --file-size 5
```

Při instalaci do souboru za nás `virt-install` tento soubor vytvoří, je pouze nutné specifikovat velikost souboru v GB.

Nyní se instalace spustí, otevře se grafická konzole kde můžeme sledovat průběh instalace, který se nijak neliší od klasické instalace. Po dokončení instalace a zvolení restartu v grafické konzoli se doména automaticky zastaví. Doménu restartujeme a spustíme příkazem

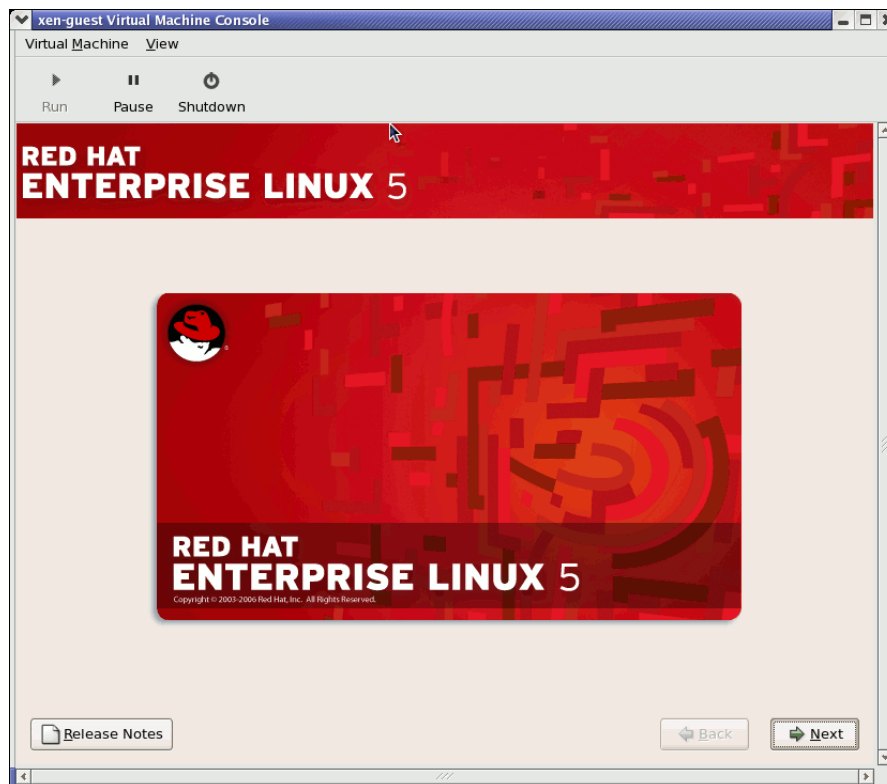
```
# xm create Domain1
```

7.6.2 Instalace pomocí virt-mamageru

Virt manager poskytuje grafické rozhraní pro instalaci virtuálních strojů. V prvním okně se nás instalátor zeptá, pokud to dovoluje náš hardware, jestli budeme instalovat plnou virtualizaci či paravirtualizaci. V dalším kroku zadáme jméno virtuálního stroje, zadáme zdroj s instalačními obrazy nebo instalačním stromem. Pokud chceme, můžeme použít kickstart soubor pro definici voleb instalace. Následně vybíráme místo pro uložení virtuálního stroje, buď můžeme specifikovat diskový oddíl, nebo image diskový soubor s velikostí.



Jako poslední specifikujeme velikost použité paměti a počet virtuálních procesorů pro doménu. Po rekapitulaci a následném potvrzení se spustí grafická konzole s instalací.



7.6.3 Manuální instalace pomocí xm

Instalaci lze provést i ručně bez pomoci instalačních nástrojů jako je virt-install či virt-manager. V tomto případě je ale nutné vytvořit ručně i diskový image soubor pomocí například příkazu dd. Pro 5GB velký image soubor zadáním

```
# dd if=/dev/zero of=/var/lib/xen/images/domain1.img bs=1M \
count=5000
```

Poté je potřeba zkopírovat jádro a initial RAM disk do adresáře /var/lib/xen/images. Můžeme použít například jádro pro PXE z prvního instalačního disku

```
# cp /media/cdrom/images/pxeboot/vmlinuz /var/lib/xen
```

```
# cp /media/cdrom/images/pxeboot/initrd.img /var/lib/xen
```

Dále je nutné vytvořit konfigurační soubor pro danou DomU doménu. Jednotlivé konfigurační direktivy jsem popsal výše, například

```
name = "domain1"
```

```
memory = "512"
```

```
disk = [ 'tap:aio:/var/lib/xen/images/domain1.img, xvda,w', ]
```

```
vif = [ 'mac=00:16:3E:24:27:a0, bridge=xenbr0', ]
```

```
uuid = "32f75ee1-2d0e-4564-5612-451b678e1212"
```

```
vcpus = 2
```

```
on_reboot = 'restart'
```

```
on_crash = 'restart'
```

```
vfb = [ "type = vnc, vncunused = 1" ]
```

Nakonec jádro, RAM disk a konfigurační soubory zadáme jako parametry příkazu xm create který je předá instalátoru

```
# xm create -c -f /etc/xen/domain1 on_reboot=destroy \
```

```
kernel=/var/lib/xen/vmlinuz \  
  
ramdisk=/var/lib/xen/initrd.img \  
  
extra="ks=http://server.example.com/ks/domain1.ks vnc  
askmethod"
```

Tímto se celá instalace spustí, pomocí vnc konzole můžeme sledovat její průběh.

7.7 Správa virtuálních domén

Správa virtuálních domén musí být prováděna z privilegované domény - Dom0, protože jednotlivé DomU se navzájem vidí pouze přes síťový bridge, jsou tedy jinak absolutně separované. Z privilegované domény se provádí také správa hypervizoru.

Pro správu se používá několik nástrojů, liší se způsobem přístupu k doménám a použitou technologií. Pro virtualizaci se v RHEL používá knihovna libvirt, pro kterou Red Hat vyvinul vlastní API, aby byla oddělena vlastní technologie virtualizace od programů virtualizaci používajících. To znamená, že programy pro správu využívající toto API jsou odolné vůči budoucím změnám v technologii virtualizace. API je kompatibilní s programovacími jazyky C a Python a má porty pro mnoho dalších, jako například Perl. API je také kompatibilní s dalšími virtualizačními programy jako KVM a QEMU, které používají pro virtualizaci knihovnu libvirt.

7.7.1 Správa pomocí příkazů

Virsh je utilita pro příkazovou řádku používající právě API pro knihovnu libvirt. Byla vyvinuta firmou Red Hat a umí pracovat nejen s Xenem, ale také například s KVM.

Nativní nástroj pro správu domén se jmenuje xm. Byl vyvinut v rámci projektu Xen. K doménám přistupuje na nižší vrstvě a nabízí detailnější informace než nástroje založené na knihovně libvirt, avšak umí pracovat pouze s programem Xen.

Xentop je utilita pro monitoring aktuálních informací a stavu domén založených na Xenu. Je podobná standardní utilitě top z Linuxu pro zobrazení informací o procesech. Zobrazuje detailní statistiky o síti, virtuálních cpu, virtuálních blokových zařízeních apod.

Pro identifikaci virtuálních domén potřebují nástroje pro správu tyto domény nějak způsobem pojmenovat. Existují tři identifikační znaky pro každou doménu, jméno domény, ID domény a UUID domény.

Jméno domény je čteno z konfiguračního souboru, když je doména vytvářena, s výjimkou Dom0. Jedná se většinou o krátký textový řetězec, který je dobře zapamatovatelný pro běžnou práci s doménami.

Číslo ID domény je dočasný identifikátor pro doménu, který je doméně přiřazen při spuštění. Pokud je doména zastavena a znovu spuštěna, dostává tento identifikátor nový. Privilegovaná doména má vždy ID 0, proto je nazývána Domain 0.

UUID je 128 bitový unikátní identifikátor, který je přiřazen doméně při jejím vytváření na stálo. Tento identifikátor zůstává stejný i při změně jména domény či při migraci na jiný fyzický stroj. Číslo je náhodně vygenerováno a je uloženo v konfiguračním souboru.

Běžně se u výpisu domén zobrazuje pouze ID a jméno, UUID se zobrazí v podrobnějších informacích o doméně. Nástroj pro správu virsh a xm dokáží konvertovat jednotlivé identifikátory mezi sebou.

Virsh domid name|UUID vypíše ID domény, virsh domname id|UUID vypíše jméno domény a virsh dominfo name|id vypíše informace o doméně včetně UUID. Stejně možnosti nabízí také příkaz xm.

Virsh byl vyvinut jako nástroj pro správu virtuálních domén založených na knihovně libvirt. Je navržen pro práci s rozdílnými virtualizačními metodami. Můžeme pomocí něj domény vytvářet, suspendovat, oživovat, ukládat, zastavovat. Můžeme také zobrazit stav všech domén. Může pracovat jak v interaktivním tak neinteraktivním módu.

Xm je nativní nástroj pro správu virtuálních domén. Funguje pouze s produktem Xen. Nabízí stejnou funkčnost jako virsh. Xm je utilita přímo kontrolující démona xend, který spravuje hypervizora.

```
# xm list
```

Name	ID	Mem(MiB)	VCPUs	State	Time(s)
Domain-0	0	5521	4	r-----	361.4

Start nové domény se provádí příkazem `xm create configfile`. `Xm` automaticky hledá konfigurační soubor v cestě `/etc/xen`. Parametr `-c` zajistí spuštění console.

Nástroje `xm` a `virsh` poskytují stejné možnosti pro zastavení virtuální domény. Doménu můžeme zastavit pomocí `virsh destroy domain`, který doménu nezničí kompletně, pouze ji jakoby odpojí od proudu. Pomocí příkazu `virsh shutdown domain` je doména nejdříve ukončena a pak je proveden `destroy`. Pomocí příkazu `virsh reboot domain` je proveden `shutdown` domény a následně je doména znovu spuštěna. Stejně možnosti nabízí nástroj `xm`.

Doménu je také možné suspendovat a znovu obnovit. Pokud provedeme příkaz `virsh suspend domain`, doména není zastavena pomocí `shutdown`, ale je pouze zastavena a virtuální doména je držena nadále v paměti. Následně můžeme doménu probudit příkazem `virsh resume domain`, kdy dojde k její rychlé obnově. `Xm` používá pro tuto funkci `xm pause domain` a `xm unpause domain`.

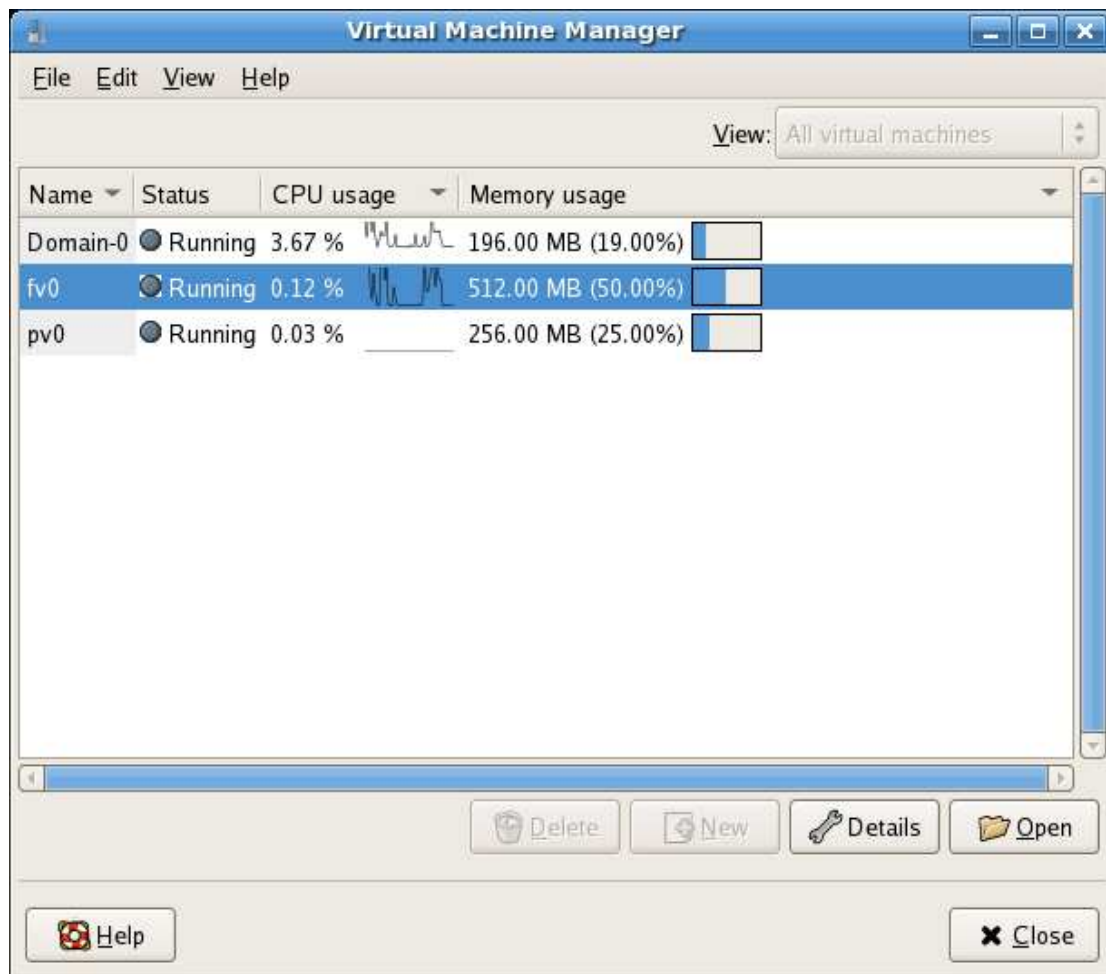
Pro virtuální domény můžeme také uložit jejich aktuální stav na disk. Je to podobné jako hibernace u notebooků. Standardní umístění souborů je `/var/lib/xen/save/domain-name`. Při uložení aktuálního stavu je doména suspendována, ale obraz paměti RAM je dumpován do souboru na disk. To znamená že počítač může být vypnut a později může být virtuální doména obnovena přesně do stavu před uložením, tedy i informace v paměti RAM které by byli jinak smazány.

Příkaz pro uložení domény je `virsh save domain state-file`, pro obnovení `virsh restore state-file`. `Xendomain init` skript může při bootu `Dom0` automaticky obnovit uložené domény.

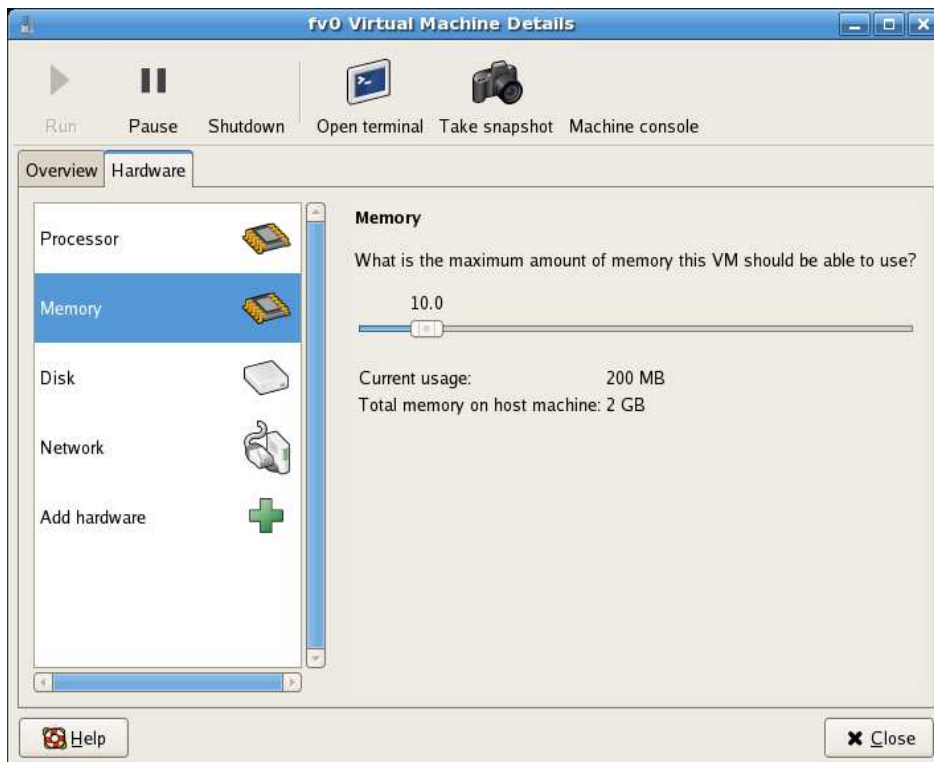
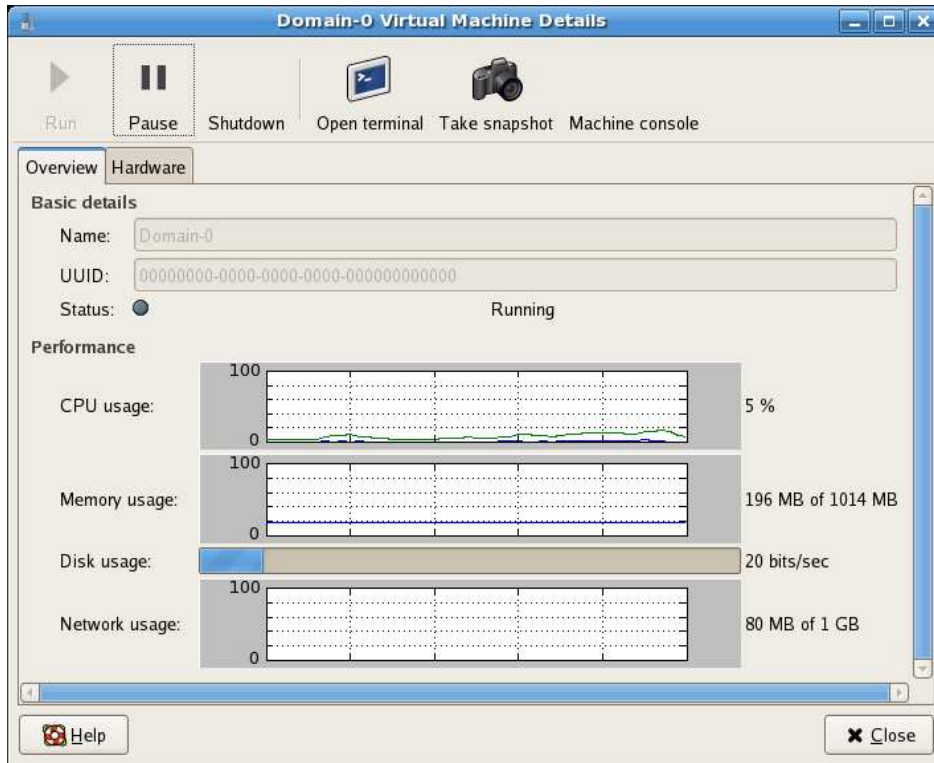
Plně virtualizované domény nemohou být uloženy, pouze paravirtualizované. `Domain 0` také nemůže být zastavena či uložena.

7.7.2 Grafický Virt-manager

Správu lze také provádět graficky pomocí `virt-manager`. `Virt-manager` je grafický nástroj pro správu domén a jejich monitoring také využívající knihovnu `libvirt`. V základním okně je vidět jednotlivé domény a jejich aktuální stav.



U každé domény lze zobrazit podrobné informace o doméně a jejím vytížení. Lze měnit stav domény, nebo také pracovat s virtuálními zařízeními.



7.8 Instalace plně virtualizované Domain U

Výhoda plné virtualizace oproti paravirtualizaci spočívá v nenutnosti modifikovat instalovaný operační systém. Je tak možné instalovat starší verze RHEL, jiné verze Linuxu, nebo můžeme instalovat proprietární operační systémy jako například Microsoft Windows, které nám modifikaci neumožňují. Xen používá hardwarově asistovanou plnou virtualizaci. Plná virtualizace však potřebuje podporu virtualizace přímo v procesoru a je pomalejší než paravirtualizace. Je z důvodů je, že při plné virtualizaci jsou fyzická zařízení emulována privilegovanou doménou a hypervisorem, aby jednotlivé DomU byli od sebe navzájem izolováni. Tato emulace je náročná na zdroje systému.

Jedna z uvažovaných metod pro urychlení plné virtualizace spočívá ve vytvoření speciálních driverů zařízení pro jednotlivé operační systémy. Díky těmto driverům by pak driver věděl, že je nainstalován ve virtuálním stroji a přístup k reálnému zařízení by tak mohl být přímější než skrze emulované virtuální zařízení.

Před instalací je tedy nutné ověřit podporu pro virtualizaci v procesoru

```
# cat /proc/cpuinfo | grep '(vmx|svm)'
```

Pro procesory Intel je to tedy Intel Virtualization Technology VT-x a pro procesory AMD je to technologie Pacifica, neboli AMD Virtualization AMD-V. Je také nutné zkontrolovat zapnutou podporu virtualizace v biosu.

I při použití plné virtualizace existují omezení týkající se architektury operačního systému instalovaného do virtuálního stroje. Pokud Dom0 je 32bit, je možné do DomU instalovat 32bit systém, jak plně virtualizovaný, tak paravirtualizovaný. Na 32bit architektuře ovšem zatím není možné instalovat do DomU 64bit systémy. Pokud je Dom0 64bit, je možné do DomU instalovat 32bit systém, avšak pouze jako paravirtualizovaný.

Verze Xenu v RHEL 5 update 1 dovoluje adresovat pro plně virtualizované domény maximálně 2 GB RAM. Pro plně virtualizované domény není možné ukládat aktuální stav na disk. To může způsobit problémy při ukončení běhu Dom0, která automaticky všechny paravirtualizované stroje uloží a po opětovném zapnutí Dom0 tyto domény obnoví. V souboru `/etc/sysconfig/xendomains` se vypnutím parametru `xendomains_save` nastaví, že se domény nebudou ukládat, ale normálně se ukončí. V tom případě se ukončí i plně virtualizované domény.

Plně virtualizované domény také nemohou být živě migrovány na jiný fyzický stroj.

Pro operační systémy v jednotlivých plně virtualizovaných doménách je nutné emulovat většinu zařízení jako virtuální, protože nemodifikované ovladače v těchto operačních systémech musí vidět jakoby pravé zařízení. Emulace je nutná, neboť pravé zařízení nemůže být sdíleno záraz několika operačními systémy, navíc by mohlo dojít k bezpečnostnímu problému kvůli separaci jednotlivých domén.

Hypervizor tedy nedovolí přímý přístup k zařízením jako je síťová karta, grafická karta, harddisk apod. Tyto zařízení jsou emulovány s použitím device modelu. Výjimkou je procesor, který není emulován, k procesoru je přistupováno napřímo. Jako device model je v Xenu použit QEMU emulátor. QEMU emulátor pro jednotlivé plně virtuální stroje běží jako samostatný proces `qemu-dm`. Například síťová karta je emulována jako Realtek-8139 a grafická karta jako Cirrus Logic GD 5446.

Vlastní instalace plně virtualizované domény je stejná jako paravirtualizované domény. Je pouze nutné specifikovat, že chceme plnou virtualizaci.

Pokud použijeme skript `virt-install` je nutné místo parametru `--paravirt` použít parametr `--hvm`. Je také nutné pomocí parametru `-c` zadat cestu k instalačnímu image souboru na disku, nebo cestu k cd (dvd) mechanice s instalačním médiem.

```
# virt-install --hvm --name hvmdomain --ram 512 \  
  
-c /dev/cdrom -f /dev/vg0/hvm_domain
```

Pokud instalujeme pomocí `virt-managera`, je pouze nutné vybrat Fully Virtualized místo Paravirtualized při definici instalace.

Konfigurační soubor pro plně virtualizovanou doménu obsahuje některé konfigurační direktivy navíc, některé jsou lehce změněny.

```
name = "fullvirt"  
builder = "hvm"  
memory = "512"  
disk = [ 'phy:/dev/vg0/lv_fullvirt, sda,w', ]  
vif = [ 'type=ioemu, mac=00:16:3E:24:27:a0, bridge=xenbr0', ]  
uuid = "32f75ee1-2d0e-4564-5612-451b678e1212"  
device_model= "/usr/lib/xen/bin/qemu-dm"  
kernel= "/usr/lib/xen/boot/hvmloader"
```

```
vnc = 1
vncunused = 1
vcpus = 2
pae = 1
apic = 1
acpi = 1
on_reboot = 'restart'
on_crash = 'restart'
```

Jsou přidány direktivy `builder`, `kernel` a `device_model`. U definici disku je možné použít direktivu `file` místo `tap:aio`. Sít'ové rozhraní obsahuje navíc direktivu `type=ioemu`, které říká že zařízení je emulováno skrz `device model`.

Do plně virtualizované domény je možné instalovat neupravené operační systémy, je tedy možné instalovat například i MS Windows a jiné. Instalace těchto operačních systémů však vyžaduje provést několik nestandardních kroků. V této práci se zabývám pouze virtualizací Linuxu, nebudu zde proto tyto výjimky uvádět.

8 VMware

8.1 Popis VMwaru

VMware je komerční proprietární produkt vyvíjený firmou VMware inc pro virtualizaci na platformách x86. Produkt pro virtualizaci VMware je nabízen v základních třech variantách, lišících se hlavně cílovou skupinou uživatelů. VMware nabízí dva serverové produkty určené pro podnikové řešení a jeden produkt pro osobní počítače nebo pracovní stanice. Serverové produkty lze potom dále rozšiřovat přídatnými moduly, například pro použití s více procesory, pro práci se storage apod. Pro pracovní stanice je určen produkt VMware Workstation, který ke své práci stejně jako VMware server potřebuje hostitelský operační systém.

V této práci se zabývám virtualizací v podnikovém prostředí, proto se zaměřím na serverové produkty. Pro servery jsou určeny dva produkty, VMware server a VMware server ESX. VMware server ESX má vlastní operační systém, vlastní distribuci Linuxu založená na Red Hat Linuxu, která má přispět ke stabilitě a bezpečnosti celého řešení. Je to komerční placený produkt. Produktu VMware server je určen spíše pro použití na jednotlivých serverech, je šířen částečně jako open source a jeho použití je zdarma, má však při oproti ESX serveru omezenou funkčnost. VMware server ESX je určen spíše pro použití v kompletních virtualizačních infrastrukturách a náročnějších řešeních, například s použitím sdílených storage, jako je NAS či iSCSI. Proto je VMware server ESX dnes dodáván spíše jako součást produktu VMware Infrastructure, z důvodů stability má omezený seznam hardwaru, pro který je certifikován. Správa VMware Infrastructure se provádí přes webovou aplikaci nebo použitím nástrojů VMware ACE.

Pro tuto práci jsem použil produkt VMware server 1.0.4, který pro svůj běh potřebuje hostitelský operační systém, tím bude v mém případě RHEL 5, který použiji kvůli srovnatelnosti s řešením Xen. Je také možné použít MS Windows jako hostitelský operační systém. VMware server v maximální konfiguraci podporuje až 16 procesorů. Na jednom fyzickém procesoru je doporučeno provozovat maximálně 4 virtuální stroje. Hostitelský operační systém musí mít minimálně 512 MB paměti RAM, maximálně 64 GB.

Firma VMware nabízí ještě mnoho dalších nástrojů souvisejících s virtualizací. Například VMware Player je produkt, který sice neumožňuje tvorbu virtuálních strojů, dokáže ale

přehrát z datového souboru již vytvořené virtuální stroje. Je také částečně šířen jako open source.

Momentálně je také dostupný VMware server 2.0 beta. Verze VMware 2.0 bude obsahovat několik novinek, hlavně se jedná o nové rozhraní pro správu virtuálních strojů kompletně skrze webové rozhraní. Mělo by se tak sjednotit ovládání VMwaru pro Windows a Linux. Dále bude verze 2.0 podporovat až 8 GB paměti a až 2 procesory.

8.2 Typy instalace

I při virtualizaci s použitím VMware máme na výběr mezi tenkou a tlustou instalací. Toto rozdělení je opět z bezpečnostních důvodů, neboť z hostitelského operačního systému lze ovlivňovat běh jak hypervisoru, tak ostatních virtuálních domén. Proto je nutné hostitelský operační systém co nejlépe zabezpečit.

Hostitelský operační systém při tenké instalaci tedy neobsahuje žádné síťové služby a není tedy tak velké riziko její kompromitace skrze tyto služby. Slouží pouze jako základna pro provoz ostatních virtuálních strojů do kterých se tyto síťové služby instalují.

Při tlusté instalaci opět instalujeme do hostitelského operačního systému i síťové služby a další software, používáme jej tedy jako normální pracovní prostředí. Toto nasazení není vhodné pro nasazení na server, často se používá například při testování nebo vývoji, kde hostitelský operační

8.3 Instalace VMwaru do hostitelského systému

VMware server je šířen ve dvou variantách, buď jako balíček rpm, nebo jako zdrojový archiv tar.gz.

Jednodušší je samozřejmě instalace z rpm balíčku, pokud VMware server chceme instalovat na distribuci, která nepodporuje balíčkovací systém rpm, spustíme instalaci po rozbalení archivu skriptem vmware-install.pl. Instalaci z archivu tar.gz. také zvolíme v případě, že chceme VMware instalovat do jiné adresářové cesty než je pro VMware standardní.

```
# rpm -ivh VMware-server-1.0.4.i386.rpm
```

Je nutné také nainstalovat balíčky kernel-headers a kernel-devel.

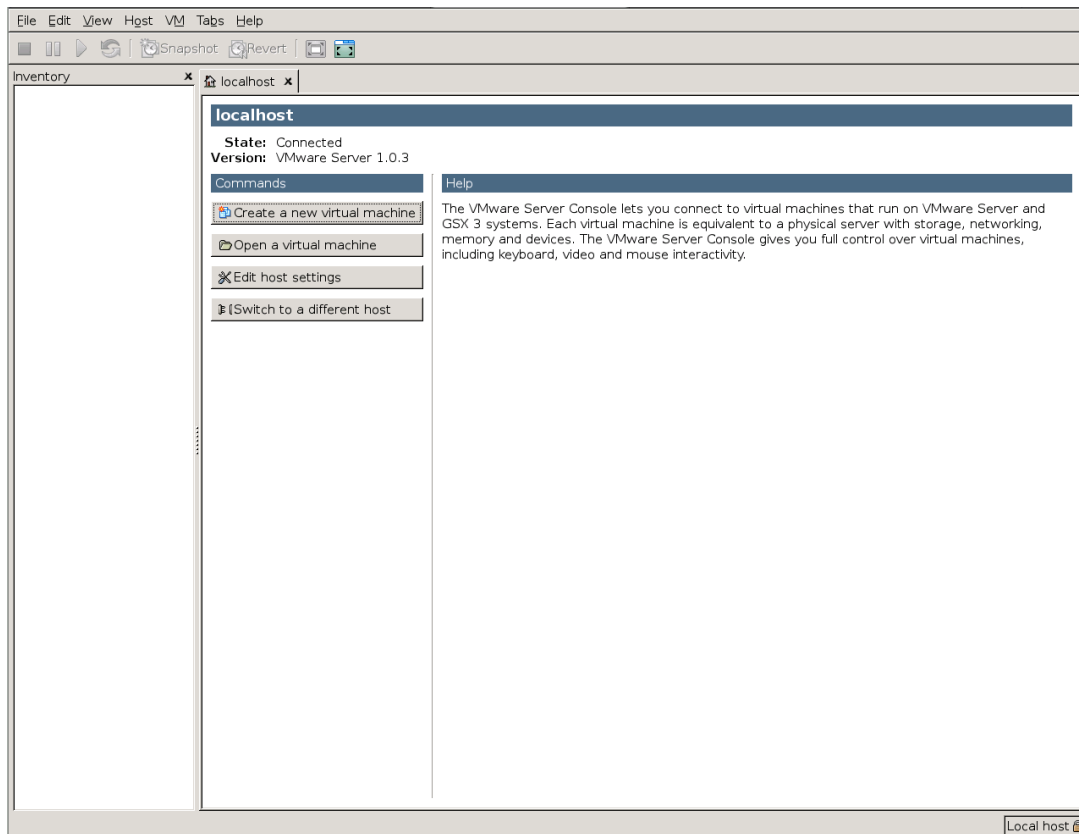
Dále je také nutné nainstalovat libXtst-devel a libXrender-devel kvůli potřebným knihovnám.

Po vlastní instalaci produktu, ať už z balíčku rpm, nebo z archivu tar.gz. je nutná konfigurace, která se spouští skriptem vmware-config.pl. Po odsouhlasení licenčních politik se nás konfigurátor ptá na umístění ikon a souboru menu. Jako další krok instalátor zjistí verzi našeho jádra a pokud pokusí se najít předkompilovaný jaderný modul. VMware pro svoji funkci potřebuje zavedený jaderný modul. Jaderný modul je velmi citlivý na strukturu jádra, proto se pro každou verzi liší. Pokud instalátor nenajde předkompilovaný modul, v dalším kroce jej zkompilujeme. Rekompilace modulu se musí provést při každé změně jádra. Pro kompilaci je potřeba mít nainstalovaný překladač gcc a nastavit správně cestu k hlavičkovým souborům jádra.

Následuje konfigurace síťových rozhraní. VMware používá síťový bridge, který propojuje virtuální síťové rozhraní s fyzickým síťovým rozhraním. Můžeme také použít NAT a virtuální síťové rozhraní před předáním na fyzickém rozhraní překládat. Je také možné zvolit síťové rozhraní pouze pro komunikaci s hostitelským operačním systémem bez přístupu do vnější sítě.

Před dokončením instalace se ještě nastaví port pro spojení prostřednictvím vzdálené konzole pro případnou komunikaci z jiných fyzických strojů. Předvolený port je tcp 902, pokud používáme firewall, například iptables, musíme povolit přístup na tento port. Nakonec musíme zadat sériové číslo které získáme registrací při stahování produktu z webových stránek.

Instalace je nyní dokončena, po spuštění příkazu vmware se nás VMware zeptá, ke kterému stroji se chceme připojit. Poté se objeví hlavní správcovské okno.



8.4 Instalace virtuálního stroje

VMware se na rozdíl od Xenu ovládá pouze skrze grafické rozhraní. Na serveru však může být nainstalován bez X serveru a pro správu můžeme používat grafickou konzoli z jiného systému.

Instalace virtuálního stroje se provádí volbou Create new virtual machine, který spustí průvodce instalací. První okno se dotazuje na typ instalace, předvolenou nebo vlastní. Následně se nás instalátor ptá na instalovaný operační systém, zde je výběr relativně velký, pokud však nenalezneme přesně který chceme, stačí vybrat nejpodobnější.



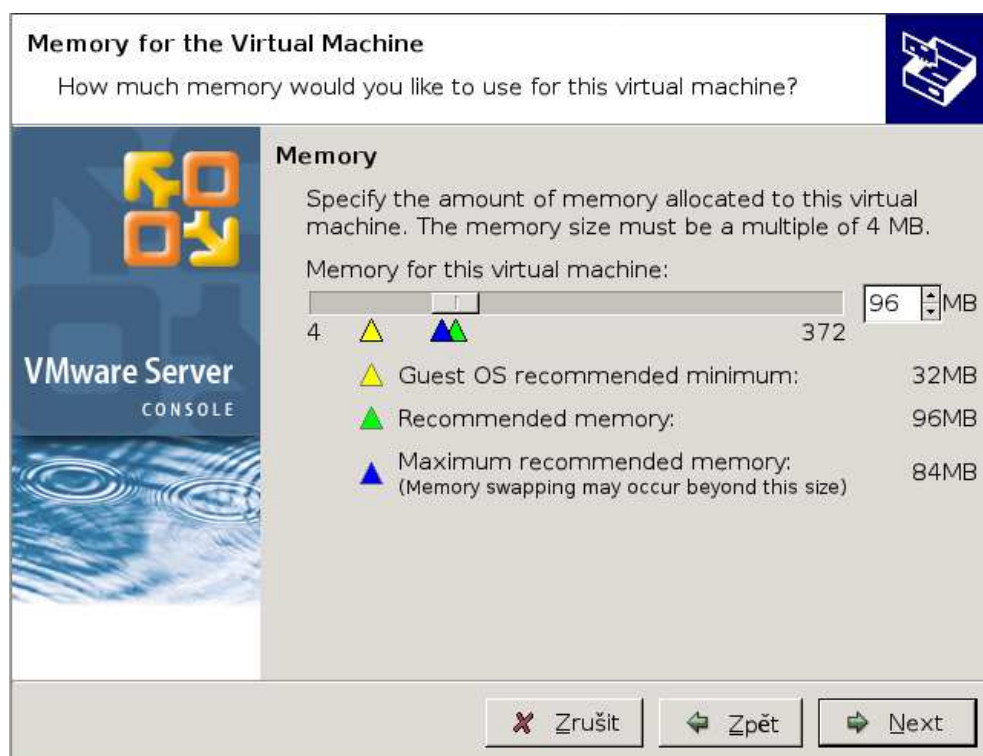
Následně je třeba zvolit jméno pro virtuální stroj a určit umístění zdrojových souborů pro instalaci. Pro instalaci je možné použít iso obrazy, fyzické mechaniky, nebo adresář na disku.

Dále je třeba zvolit kolik virtuálních procesorů bude použito pro virtuální stroj.

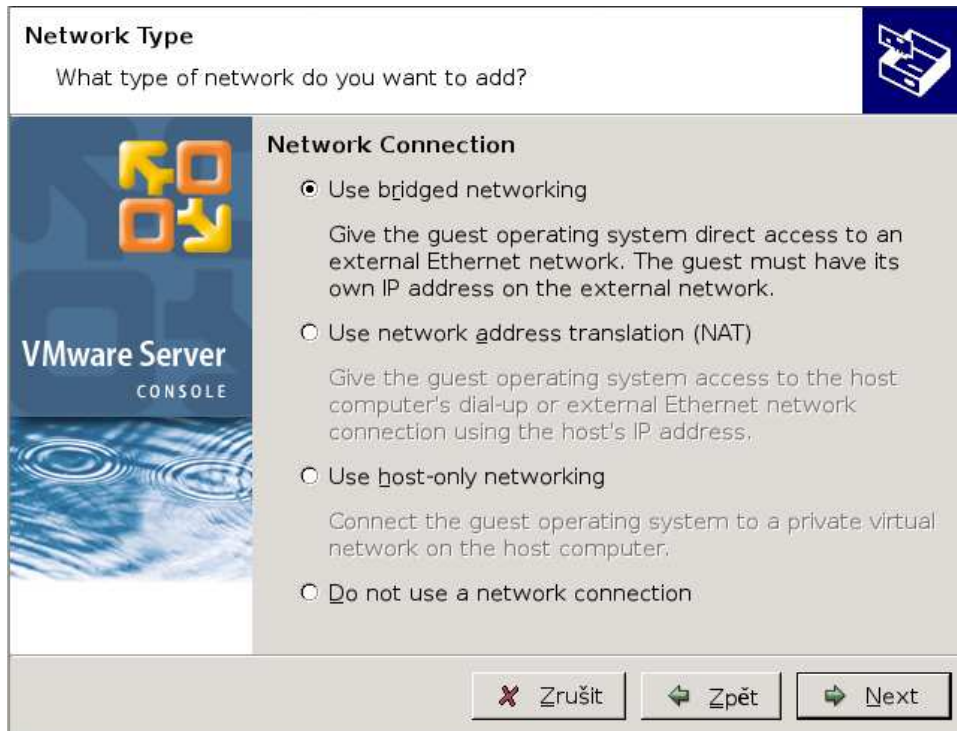
Následuje definice, jestli má být virtuální stroj privátní, tedy jestli na něj budeme moci přistupovat pouze z lokálního fyzického stroje, nebo jestli bude přístupný i ze vzdáleného systému.



Je potřeba také definovat velikost paměti pro virtuální stroj. Zde nám instalátor vypisuje minimální paměť, doporučenou a maximální. Po překročení maximální paměti instalátor začne swapovat na pevný disk.

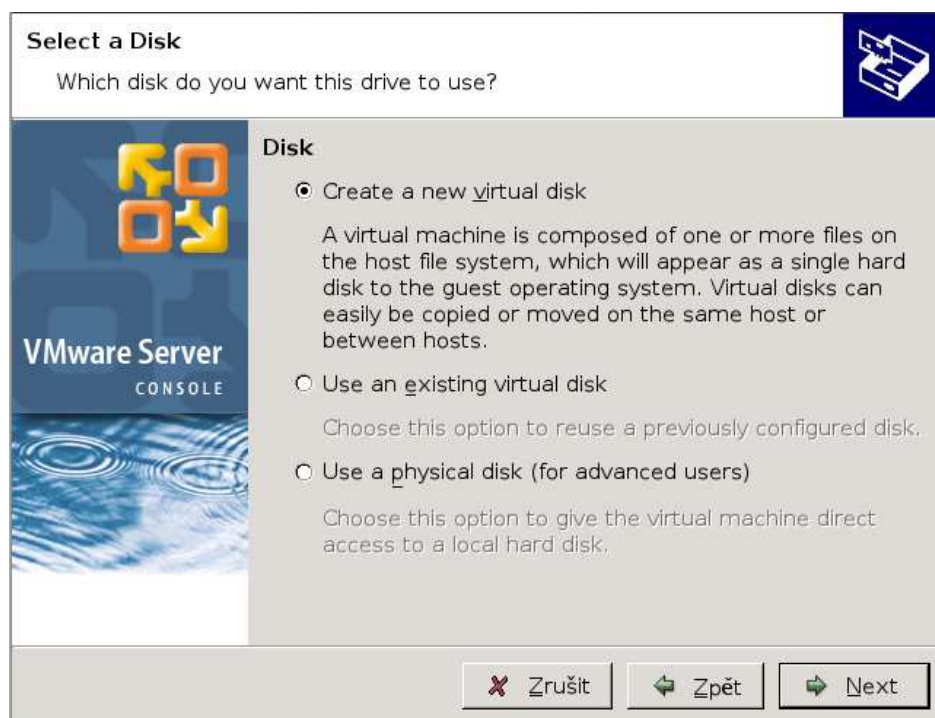


U nastavení sítě je možné vybrat ze čtyř variant, můžeme použít sdílené rozhraní a bridge, dáváme tedy virtuálnímu stroji přímý přístup do sítě. Pokud použijeme překlad ip adres, neboli nat, bude odchozí provoz maskován ip adresou hostitelského systému. Je možné také nastavit, že se k virtuálnímu stroji lze připojit pouze z lokálního systému, nebo nedefinovat síťové rozhraní vůbec.



U definice místa pro uložení virtuálního stroje nejdříve definujeme typ scsi adaptéru pro emulaci virtuálních disků. Následně definujeme disk, kam budeme data virtuálního stroje ukládat. Můžeme zvolit vytvoření virtuálního disku, v tomto případě je virtuální stroj uložen do image diskového souboru. Místo můžeme předem alokovat, nebo nechat postupně narůstat velikost při potřebě. Alokace místa předem vede ke zvýšení rychlosti. Při volbě virtuálního disku je nutné také zadat jméno souboru, do kterého bude virtuální disk uložen. Standardní cesta pro uložení image souborů je `/var/lib/vmware/Virtual Machines/`.

Je možné také zvolit fyzický oddíl, který si musíme předem vytvořit, například pomocí Fdisku. Poslední možnost je použití již existujícího virtuálního disku, který bude v tomto případě přepsán.



Následně proběhne alokace místa na disku ve správcovské konzoli se zobrazí nový virtuální stroj.

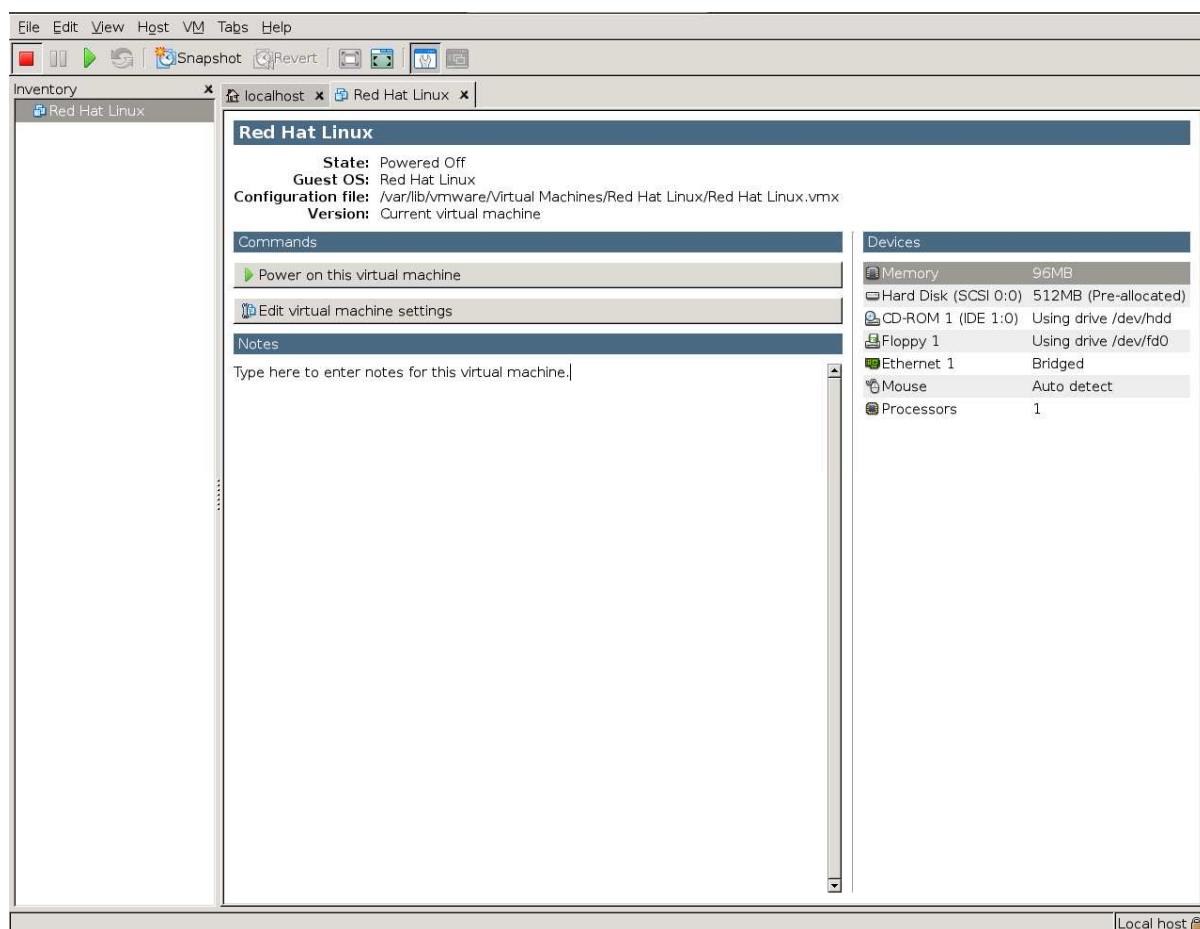
8.5 Správa virtuálních strojů

Pro přístup k virtuálním strojům se používá vmware-server-console.

V hlavním správcovském okně se v levém svislém panelu inventory objeví nový virtuální stroj. V pravém okně vidíme podrobnější informace o virtuálním stroji. Můžeme zde přes tlačítko edit virtual machines editovat jednotlivé nastavení virtuálního stroje. Okno s editací má dvě záložky, hardware a options. V nich můžeme přesně editovat jednotlivé nastavení stroje a jednotlivá virtuální zařízení.

V horní levé části správcovského okna jsou umístěny tlačítka pro ovládání stavu virtuálního stroje. Pokud zvolíme volbu spustit, dojde k emulaci hardware, dokonce se emuluje i BIOS a následně se spustí virtuální stroj v konzoli a začne se instalace operačního systému ve virtuálním stroji.

Jednotlivé stroje můžeme stejně jako u Xenu z správcovského okna zastavovat, spouštět, restartovat, uspávat a probouzet. Lze také provést snapshot aktuálního stavu. Bohužel verze server umí pouze jeden snapshot z každého virtuálního stroje. K provedenému snapshotu se můžete kdykoliv vrátit.



VMware používá jako identifikátor domény pro uživatele pouze jméno domény, které vidíme v panelu Inventory.

9 Migrace virtuálních strojů

Domény mohou být migrovány z jednoho fyzického stroje na druhý. Migrace virtuálních strojů je dvojího typu, živá a mrtvá.

Mrtvou migrací nazýváme způsob přesunu virtuálního stroje, kdy virtuální stroj nejdříve suspendujeme a pak diskový soubor a obraz paměti překopírujeme na jiný fyzický stroj a tam následně stroj znovu oživíme. Při mrtvé migraci dojde k zastavení virtuálního stroje a tedy i služeb jím poskytovaných.

Při živé migraci nedojde k zastavení virtuálního stroje a stroj je migrován za chodu. Dojde pouze zhruba na sekundu ke zpomalení chodu, ale pro uživatele služeb nedojde k odmítnutí. Uživatel vůbec neví, že došlo k migraci virtuálního stroje. Pro živou migraci je zapotřebí splnit několik požadavků.

Za prvé je nutné migrovat pouze obraz paměti, tedy vlastní data virtuálního stroje musí být uložena na sdílené storage, například pomocí technologií iSCSI nebo SAN. Oba dva fyzické stroje musí mít přístup na tuto sdílenou storage.

Protože při migraci dochází i k migraci MAC adresy a IP adresy virtuálního síťového rozhraní, je nutné aby oba dva fyzické stroje byly v jedné síti, jednom subnetu, aby bylo možné dobře přesměrovat provoz z jednoho virtuálního rozhraní na druhé.

Architektura procesorů musí být stejná. Nelze migrovat virtuální stroj z 32 bitu na 64 bit, ani mezi jinými druhy procesorů, jako třeba x86 a Intel Itanium. Dokonce není ani možné migrovat z Intelu 64bit na AMD 64bit. Ač toto není nikde popsáno, při testech migrace mezi Intelu a AMD nefungovala, ikdyž architektury byly stejné.

Pro použití sdílené storage je nutné v případě VMware použít verzi ESX server. Verze testovaná v této práci, tedy VMware server 1.0.4 tuto funkcionalitu nenabízí. Xen nabízí možnost použití sdílené storage i ve verzi Xen server 3.0.3.

10 Výkonnostní testy

Pro testování bylo potřeba vytvořit pro oba produkty co nejpodobnější a nejstálejší prostředí, aby naměřená výkonnost byla co nejméně ovlivněna dalšími vnějšími vlivy. Testy byly proto provedeny na jednom hardware postupně. Jako operační systém pro privilegovanou doménu a hostitelský operační systém v případě VMware byl pro oba produkty Red Hat Enterprise Linux 5 update 1. Do virtuálních strojů byl také ve všech případech instalován Red Hat Enterprise Linux 5 update 1, mělo byt tak být docíleno co největší srovnatelnosti při výkonnostních testech.

Jako hardware pro testovací prostředí byl použit server IBM x3400. Tento model je svými vlastnostmi určen pro malé a středně velké firmy. Je široce konfigurovatelný, lze ho tak dobře přizpůsobit aktuálním potřebám. Obsahuje dvě patice pro umístění procesorů, dále osm paměťových slotů pro postupné navyšování paměti, volitelně je možné server osadit redundantními komponenty, lze tak do určité míry předcházet výpadkům z důvodu výpadku hardware.

Pro disky je k dispozici až osm hot-swap pozic nebo čtyři simple-swat pozice, podle modelu. Při použití SATA disků je možné použít maximální kapacitu disků až 4 TB, při použití SAS disků 2,4 TB.

Jako zdroj napájení je možné použít jeden až dva hot-swap zdroje 670W nebo 835W. Integrovaný RAID dovoluje použití RAID 0,1 nebo 10.

Přesná konfigurace serveru:

Processor: 1x Intel Xeon QuadCore Processor E5335 s EM64T 2,00GHz

Paměť: 6 GB PC2-5300 667MHz CL5 ECC DDR2

Cache: 4MB L2

Řadič: Adaptec AIC-9580W 8 port SAS(300MB/sec) nebo SATA(150MB/sec) hot-swap

Sloty: 3x PCI-Express, 2x PCI-X 64-bit/133MHz , 1x PCI 32-bit/33MHz

Grafika: Grafika: X VGA ATI RN50(ES1000) 16MB

Disky: 4x IBM 250GB SATA II Hot-swap 3GB/s
Síť: 1x 10/100/1000 gigabit Broadcom BCM 5721
Zdroj: 635W Hot-swap
Provedení : Tower

Virtualizační software pro testování byl použit Xen verze 3.0.3 a VMware Server verze 1.0.4.

Operační systém byl Red Hat Enterprise Linux 5 update 1 x86_64, použitá verze kernelu bez úpravy pro Xen byla 2.6.18-53.1.4.el5, s Xen úpravou 2.6.18-53.el5xen, Gcc 4.1.2-14.el5

Při testování výkonnosti je třeba změřit jak vytížení procesoru a paměťovou náročnost, ale je také nutné změřit rychlost vstupně/výstupních operací.

Pro měření výkonnosti jsem použil software Lmbench. Lmbench je software pro vytváření srovnávacích testů, benchmarků. Lmbench je standardní součástí systému RHEL 5, výstup z programu je textový soubor s benchmarky pro jednotlivé předem zvolené součásti systémů.

Testy byly provedeny na jednom fyzickém serveru postupně, kvůli konzistenci prostředí a lepší srovnatelnosti. Nejdříve bylo změřeno zatížení samotného RHEL, výsledná hodnota benchmark testu posloužila jako základní údaj o vytížení hardwaru samotným operačním systémem. Následně jsem změřil výkon v jednotlivých nainstalovaných virtuálních doménách a to v Xenu s paravirtualizací, v Xenu s plnou virtualizací a ve VMware s plnou virtualizací.

Z výsledků testů jsem vybral devět hodnot. První čtyři hodnoty, null I/O, stat, open/close a shell proc ukazují latence pro aktivitu procesoru a procesů. Hodnoty jsou průměrem z deseti nezávislých testů a jsou v mikrosekundách. Menší hodnota je tedy lepší.

Mem read, mem write jsou výsledky pro šířku pásma pro lokální komunikaci s pamětí, větší hodnota je lepší. AF Unix je šířka pásma pro lokální komunikaci v mikrosekundách, větší hodnota je lepší. File create a file write jsou hodnoty pro zápis a čtení ze souborového systému, v testu se zapisují se soubory o velikosti 10K, menší hodnota je lepší.

Latence je čas procesoru nebo systémových komponent, který musí čekat na dostupnost zdrojů po jejich vyžádání. Šířka pásma je speciální typ rychlosti, která měří průtok informací za jednotku času, které jsou doručeny mezi částmi systému.

Tabulka výstupů z programu Lmbench:

	Null I/O	Stat	Open/close	Sh proc	Mem read	Mem write	Unix AF	File create	File delete
RHEL 5 bez virtualizace	1,02	2,98	4,09	4986	3509	1706	2172	46,9	24,1
RHEL5 v Xenu paravirtualizace	2,63	5,98	8,26	3820	3464	1684	1800	56,1	29,1
RHEL5 v Xenu plná virtualizace	4,03	7,80	10,13	3113	3436	1676	1756	58,1	31,5
RHEL5 ve VMware plná virtualizace	3,84	7,61	9,72	3329	3442	1680	1777	58,0	31,2

Z výsledků testů je vidět, že pokud je možné použít paravirtualizaci, je tato varianta nejvýkonnějším řešením. Při použití plné virtualizace byl o něco výkonnější VMware než Xen. Je vidět, že rychlost přístupu na souborový systém a přístup do paměti je u paravirtualizace i plné virtualizace skoro stejná.

Při cenovém srovnání oba dva produkty vycházejí stejně. Xen je součástí Red Hat Enterprise Linuxu, platíme tedy pouze za operační systém. VMware server je ke stažení sice zdarma, pro svůj chod však potřebuje operační systém, doporučovaný je Red Hat Enterprise Linux. Pokud doporučení dodržíme, obě varianty nás budou stát stejně. Rozdíl je ovšem v nabízené funkcionalitě. Pokud chceme použít sdílenou storage nebo více fyzických procesorů, musíme v případě VMware zvolit VMware server ESX. Ten již má svůj vlastní operační systém odvozený z Red Hat Enterprise Linuxu. Za operační systém tedy ušetříme, ESX server je však výrazně dražší.

11 Závěr

Z této práce vyplývá, že použití virtualizace může být v podnikovém prostředí, ale i mimo něj, velmi přínosné. I ve stabilním prostředí, kde relativně není potřeba nic nového nasazovat, nám virtualizace může přinést mnoho výhod při správě prostředí a dokonce i úsporu finančních nákladů.

Při použití virtualizace můžeme ušetřit finanční náklady na hardware, konsolidací je možné daleko lépe využít výkonnostního potenciálu hardwaru, navíc ušetříme finanční investice za místo v serverovně, chlazení a za elektrickou energii. Další úspory mohou být v ceně softwaru, neboť například Red Hat Enterprise Linux nabízí v ceně jedné subskripce v případě verze standard server čtyři a v případě verze advanced platform dokonce nekonečný počet instalací do virtuálních strojů zdarma. Můžeme tedy provozovat na jednom hardware několik virtuálních serverů s instalovaným Red Hat Enterprise Linuxem za cenu jedné subskripce.

Ale i v případě správy systémů nám virtualizace nabízí mnoho výhod a úspor nákladů. Za použití migrace virtuálních strojů můžeme naplánovat údržbu hardware v jakýkoliv čas, aniž by byla přerušena dostupnost služby, není potřeba platit navíc čas administrátorů v ranních hodinách apod. Navíc značně přínosná je možnost vytvářet a ukládat aktuální obrazy virtuálních strojů, tedy snapshoty, nebo možnost ozkoušení funkčnosti služby ve virtuálním stroji před ostrým nasazením.

Toto vše jasně podporuje použití virtualizace, k tomu však je potřeba dostatečně kvalitního, bezpečného a stabilního softwaru. Vyzkoušel jsem a otestoval dva nejpoužívanější produkty, Xen a VMware server. Oba dva produkty nabízí pro firemní použití dostatečně vyspělé prostředí a funkcionalitu. Xen nabízí dva typy virtualizace, paravirtualizaci a hardwarově asistovanou plnou virtualizaci, VMware server nabízí plnou virtualizaci hardware emulací. Který produkt a který druh virtualizace použít se nedá jednoznačně říci, záleží na prostředí kde chceme virtualizaci nasadit. Hardwarově asistovaná plná virtualizace potřebuje pro svůj běh podporu virtualizace přímo v procesoru, paravirtualizace zase potřebuje úpravu jádra operačního systému ve virtuálním stroji.

Pokud tedy chceme instalovat do virtuálních strojů operační systémy, které lze upravit pro podporu paravirtualizace, v případě Red Hat Enterprise Linuxu jsou to verze 4.5 a vyšší a 5.0 a vyšší, je nasazení paravirtualizace nejvýhodnějším řešením. To jasně potvrzují provedené výkonnostní testy.

Pokud však nelze jádro operačního systému upravit pro použití s paravirtualizací, musíme použít plnou virtualizaci. Zde je pak nutné přihlídnout k použitému hardware. Jestliže nemáme hardware podporu virtualizace v procesoru, není možné plnou virtualizaci na Xenu použít, musíme tedy zvolit VMware. Pokud vlastníme hardware s podporou virtualizace, máme na výběr mezi Xenem a VMwarem. Z výkonnostních testů obou produktů za použití plné virtualizace vyšel o něco lépe VMware, výkonnostní rozdíl je však v našem případě téměř zanedbatelný.

VMware server a Xen nabízejí pro virtualizaci s použitím lokálního úložiště podobné možnosti. Xen je trochu lépe konfigurovatelný, nabízí například více způsobů instalace virtuální domény, více druhů přístupů k informacím o doménách. Na druhou stranu je zase při další konfiguraci nutno více používat příkazovou řádku, což v případě VMware není potřeba. V případě VMware se veškerá práce provádí skrz grafické rozhraní, je pro uživatele po této stránce více přívětivý. Podobný grafický nástroj pro Xen, virt-manager, nabízí o něco horší funkcionalitu. Pokud někomu vyhovuje čistě grafické rozhraní, bude pro něj lepší volbou VMware. V případě nutnosti podrobnější kontroly, případně konfigurovatelnosti, bude vhodnější Xen. V případě použití sdílené storage Xen tuto funkcionalitu nabízí, u VMware by bylo nutné použít robustnější VMware ESX server, který však není zdarma.

12 Použité zdroje

Xen

- (1) Red Hat, Inc. - Virtualization guide [Průvodce virtualizace] – [dokument ve formátu PDF] – 14.3.2007 [cit. 20.10.2007]. Dostupné z: <https://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/en-US/Virtualization.pdf>
Obsahuje informace o instalaci, konfiguraci a administraci virtualizace Xen na OS RHEL 5.
- (2) University of Cambridge Computer Laboratory – Xen architecture [architektura Xenu]-[web site]- poslední aktualizace 3.5.2007 – [cit. 20.10.2007]. Dostupné z: <http://www.cl.cam.ac.uk/research/srg/netos/xen/architecture.html>
Dokumenty a prezentace popisující architekturu Xenu.
- (3) University of Cambridge Computer Laboratory – Xen 3.0 User manual [uživatelská příručka Xen verze 3.0] - [dokument ve formátu PDF] – [cit. 20.10.2007]
Dostupné z: <http://www.cl.cam.ac.uk/research/srg/netos/xen/readmes/user.pdf>
Kompletní uživatelská příručka
- (4) Libvirt organization – [web site] – [cit. 20.10.2007] – Dostupné z: <http://libvirt.org/index.html>
Domovská stránka organizace udržující knihovnu libvirt potřebnou pro paravirtualizaci Xen.
- (5) Xen Source, Inc. - The Xen Wiki [Wiki stránky projektu Xen Source] – [web site] – [cit. 20.10.2007]. Dostupné z: <http://wiki.xensource.com/xenwiki/>
Informace a praktické rady a zkušenosti uživatelů Xenu a lidí z komunity Xen Source.
- (6) Rami Rosen - Introduction to the Xen Virtual Machine [úvod do problematiky Xen virtuální stroj]- [www dokument]- Linux Journal – 1.10.2005 [cit. 20.10.2007].
Dostupné z: <http://www.linuxjournal.com/article/8540>
Článek popisující architekturu fungování Xen.

VMware

- (7) VMware Inc. - Introduction to VMware Infrastructure [Úvod do VMware Infrastructure] - [dokument ve formátu PDF] – 10.7.2007 [cit. 20.10.2007]- Dostupné z: http://www.vmware.com/pdf/vi3_301_201_intro_vi.pdf
Popis principů fungování a jednotlivých komponent VMware Infrastructure.
- (8) VMware Inc. - Basic System Administration [Základní administrace] - [dokument ve formátu PDF] – 13.7.2007 [cit. 20.10.2007]- Dostupné z: http://www.vmware.com/pdf/vi3_301_201_admin_guide.pdf
- (9) VMware Inc. - Installation and Upgrade Guide [Instalační a upgrade příručka] -

[dokument ve formátu PDF] – 11.7.2007 [cit. 20.10.2007]- Dostupné z:
http://www.vmware.com/pdf/vi3_301_201_installation_guide.pdf

- (10) VMware Inc. - Server Configuration Guide [Příručka konfigurace serveru] - [dokument ve formátu PDF] – 13.7.2007 [cit. 20.10.2007]- Dostupné z:
http://www.vmware.com/pdf/vi3_301_201_server_config.pdf
Kompletní popis konfigurace serveru
- (11) VMware Inc. - SAN Configuration Guide [Příručka konfigurace SAN] - [dokument ve formátu PDF] – 21.6.2007 [cit. 20.10.2007]- Dostupné z:
http://www.vmware.com/pdf/vi3_301_201_san_cfg.pdf
Popisuje způsoby konfigurace a instalace SAN úložiště.
- (12) VMware Inc. - Virtual Machine Backup Guide [Příručka pro zálohování virtuálních strojů] - [dokument ve formátu PDF] – 11.7.2007 [cit. 20.10.2007]- Dostupné z: http://www.vmware.com/pdf/vi3_301_201_vm_backup.pdf
Zálohování virtuálních strojů

iSCSI

- (13) Ben Rockwood - A Quick Guide to iSCSI on Linux – [Úvod do problematiky iSCSI na Linuxu] - [www dokument]- LiSCSI enterprise target project – 6.8.2004 [cit. 20.10.2007]. Dostupné z: <http://www.cuddletech.com/articles/iscsi/index.html>
Rychlý pohled na iSCSI, základní idey a termíny, nastavení target a initiator.
- (14) iSCSI target HowTo on Enterprise Linux [Instalační manuál iSCSI target na Enterprise Linuxu] – [www dokument] – 12.10.2006 [cit. 20.10.2007]. Dostupné z: <http://mail.digicola.com/wiki/index.php?title=User:Martin:iSCSI>
Příručka popisující krok za krokem instalaci iSCSI target.
- (15) IETF organization - Rfc 3720: iSCSI – [proposed standard] – duben 2004.
Dostupné z: <http://tools.ietf.org/html/rfc3720>
RFC dokument popisující iSCSI

Ostatní

- (16) LinWorx s.r.o. - Stanislav Polášek. - sp@linworx.cz – certifikovaný školitel v oblasti Red Hat linuxu, specialista pro nasazení velkých podnikových cluster řešení a storage.
- (17) CZLUG: elektronická konference okolo linuxu
List owner: Jan Kasprzak <linux-owner@linux.cz>
[cit. 20.10.2007]
[adresa konference] <linux@linux.cz>
[www archiv příspěvků] dostupný z:<http://www.linux.cz/lists/archive/linux/>