trap and emulate: http://www.cs.utexas.edu/~mwalfish/classes/s10-cs372h/lectures/l27.txt
segmentace! http://cs.wikipedia.org/wiki/Segmentace

high availability ~ minimum down-time

VM - virtual machine
VMM - virtual machine monitor - hypervizor, arbitr virtualizovanych PC

host - fyzicky pocitac, na kterem bezi guests
guest - virtualizovany PC

Technika virtualizace

Vývoj:

- mainframe → klientske terminaly pripojene k nejakemu superpocitaci
- jednoduche klient/server prostredi → spousta PC pripojenych pres sit k ruznym serverum
- distrib. prostredi → klienti jsou pripojeni pres web server, ten je pres nejakou branu pripojeny k siti, v siti jsou dalsi klienti, kteri jsou pres rozhrani pripojeni k serverovemu prostredi (nejaka skupina multicomputeru), kde jsou directory server, application server, email server, file server, DB server, management server, content server
    - clustering
        - vykon, stabilita
        - nakladnost
    - virtualizace
        - sdileni prostredku a vykonu, vyvazovani
- pozdeji se klienti pripojuji i pres wifi, VPN, na stanici bezi vice serveru, DB clustery
- cloud → virtualni HW platforma pro provozovani virt. serveru a sluzeb
    - skalovatelnost, elasticita
        - zmena vykonu podle potreb klienta
        - sluzby uctovany podle skutecneho vyuziti
    - vysoka dostupnost a spolehlivost
        - zalozni systemy na ruznych urovnich (servery, infrastruktura, DC)
        - SW, ktery umoznuje za behu rychlou vymenu vadneho clanku
    - automaticke aktualizace
    - sdileni systemu
        - thin provisioning - poskytuje dojem dostupnosti vice prostredku, nez fyzicky existuje - virtualni alokace prostoru na ulozisti dat
            - organizace provozuje jeden fyzický server a na něm např. čtyři virtuální servery. I když má fyzický server limitovanou velikost operační paměti (např. 8 GB), mohu nadefinovat na virtuálních

serverech v součtu více operační paměti, než kolik je fyzicky dostupné (např. každý virtuální server bude pracovat se 4 GB RAM). Pokročilými systémovými technikami (know-how výrobců virtualizačních technologií) je pak možné sdílet RAM fyzického serveru tak, že opravdu vzniká dojem nezávislých virtuálních serverů s definovanou virtuální velikostí RAM - tyto techniky např. umožňují sdílet podobné bloky dat, využívat nevyužitou paměť jednoho virtuální serveru jiným apod.
                - úspora při investicích do infrastruktury
            - multi-tenancy - poskytovani sluzby vice klientum
                - virtualni oddily dat a konfigurace - kazdy klient pracuje se svou customizovanou verzi
                    - virtualizovane zdroje jsou logicky oddeleny, nelze pracovat s cizimi daty
                - v kontrastu s: multi-instancove architektury, kde separatni instance SW pracuji s klienty
    - druhy cloudu
        - verejny - poskytovani IaaS, PaaS, SaaS treti stranou - nejcastejsi
            - zajistena vysoka skalovatelnost, uctovani dle vyuziti zdroju
        - soukromy
            - infrastruktura poskytujici sluzby pouze jedne organizaci
            - schopnost uctovani jednotlivym slozkam organizace
        - komunitni
            - sluzby vyuzivany komunitou - spolupracujici firmy, projekt
        - hybridni
            - cloud slozeny z vice ruznych zdroju
            - cloud interoperability - sky computing

Webhosting

- motivace pro virtualizaci
- postupny vyvoj
    - staticke stranky
        - 1 web = 1 adresar
        - 1 proces - apache - pro vsechny
    - dynamicke stranky
        - muze obsahovat potencialne nebezpecny kod
        - 1 web = 1 proces (Apache+PHP)
        - 1 počítač pro všechny
    - uzivatelske systemy
        - weby vyzaduji odlisne konfigurace
        - 1 web = 1 pocitac
        - bez virtualizace je to zbytecne drahe
- jak to zhruba vypada

    - pri pouziti fyzickych PC
        - na kazdem fyzickem PC je jedna domena
        - PC pripojeny do internetu
    - pri pouziti virtualnich PC
        - na jednom fyzickem PC je spousta virtualnich pocitacu, kazdy spravuje nejakou domenu
        - dale zde bezi hypervizor (virtual machine manager), ktery spravuje virt. PC
        - fyzicky PC pripojen do internetu
        - podrobneji:
            - u fyzickych PC je u kazdeho pocitace NIC (network interface controller)
            - u virtualnich PC je u kazdeho virt. PC virtualni NIC
            - hypervizor je pak napojen na fyzicky NIC, ktery je pripojeny k internetu a pozadavky na jednotliva virt. PC, ktera spravuje, organizuje on

Virtualizace

- virtuální
    - very close to being something without being it
- iluze fyzickeho zarizeni, ktere fyzicky neexistuje
- zname rozhrani fyzickeho zarizeni (stranka pameti, pocitac, disk, sitova karta) nebo software zarizeni (proces-kernel, HAL - HW abstraction layer) jsou implementovany jinak nez obvykle
    - softwarove/jinym HW (SCSI radic)/v kombinaci HW+SW (virtualni pamet, moderni virtualizace procesoru)
- pouzivat tento termin je otazka zvyku
    - SSH/RDP neni nazyvano virtualizace konzole
    - iSCSI neni virtualni disk
    - JVM neni virtualizace fyzickeho stroje
- proc?
    - muzeme mit vetsi pocet virtualnich objektu nez fyzickych
        - virtualni vs fyzicka pamet, virtualni pocitac
    - virtualni objekty mohou byt jineho druhu nez fyzicke
        - emulace PC jine architektury
    - virtualni objekty mohou byt vzdalene od fyzickych
        - vzdalene disky jsou prezentovany jako lokalni, vzdalena klavesnice+obrazovka
    - virtualni objekty mohou byt implementovane zcela jinak nez fyzicke
        - disky implementovane souborem
    - virtualni objekty mohou byt bez vazby na fyzicky svet
        - virtualni site

    - muzeme provadet zasahy do chovani, ktere by bez virtualizace nebyly mozne
        - ladeni, experimenty, mereni
        - šizení všeho druhu - thin provisioning, time sharing apod.
        - migrace, load balancing
- jaka zarizeni?
    - pocitac
        - virtualizovane rozhrani
            - fyzicke rozhrani SW-HW (nazyva se fyzicka virtualizace)
            - SW rozhrani uvnitr OS (paravirtualizace)
        - virtualizace zarizeni uvnitr
    - CPU
        - virtualizovane rozhrani
            - instrukcni sada
            - aplikacni+privilegovane instrukce (HW virtualizace)
            - aplikacni instrukce (paravirtualizace)
        - virtualizovat CPU samo o sobe nema smysl - chybi I/O
    - pamet
        - virtualizovane rozhrani
            - instrukce cteni a zapisu
    - I/O
        - virtualizace
            - na urovni I/O instrukci
            - na SW rozhrani uvnitr OS
- na jake urovni se virtualizuje ~ jak probiha sdileni fyzickeho PC virtualnimi
    - CPU
        - guest OS i hypervizor (ten bezi na host machine) pouzivaji preemptivni multitasking (umi prerusit prave vykonavajici proces bez spoluprace OS)
    - pamet
        - guest OS ma svoji virtualni pamet
        - hypervizor pridava druhou uroven pameti
    - disky
        - virtualni disk je namapovan do spolecneho diskoveho prostoru
        - iSCSI (Internet Small Computer System Interface, IP-based protokol)
            - protokol umoznuje klientum - iniciatorum posilat SCSI prikazy na SCSI disk. zarizeni, ktera jsou ale na vzdalenych serverech
            - organizace pomoci SAN protokolu
        - SAN protokol (storage area network)
            - dedikovaná (oddělená od LAN, WAN, atd) datová síť, která slouží pro připojení externích zařízení k serverům (disková pole, páskové knihovny a jiná zálohovací zařízení). SAN vznikla hlavně kvůli narůstajícím potřebám na zabezpečení a konsolidaci dat.
        - NAS (network attached storage)
            - označení pro datové úložiště připojené k místní síti LAN. Data toho

úložiště mohou být poskytována různým uživatelům. NAS nemusí mít pouze funkci souborového serveru, ale může mít i jiné specializované funkce. Například klient P2P sítě, webový server a další. Většinou obsahuje nějaký vestavěný počítač, který má za úkol sdílení dat a podporu různých protokolů.

    - site
        - trunk mode
            - metoda, ktera systemu poskytuje sitovy pristup pro spousty klientu sdilenim mnoziny linek a frekvenci misto toho aby je kazdemu klientovi poskytovala individualne
            - analogicke strukture stromu, kde je jedna hlavni pater a spoustu vetvich k ni pripojenych
        - NAT (network address translation)
            - síťová maškaráda
            - způsob úpravy síťového provozu přes router přepisem výchozí a/nebo cílové IP adresy, často i změnu čísla TCP/UDP portu u průchozích IP paketů
        - virtuální sítě
            - VPN, VLAN (logical LANs based on physical LANs)
    - dalsi zarizeni
        - resi se exkluzivni pristup/sdileny pristup, vzdalene porty (USB) apod.
- varianty virtualizace
    - aplikační virtualizace
        - technologie, ktera zapouzdruje (oddeluje) aplikacni SW od podkladoveho OS, na kterem bezi
        - plne virtualizovana aplikace neni nainstalovana v tradicnim smyslu, nicmene spousti se, jakoby byla
        - pri behu se aplikace chova jakoby primo komunikovala s rozhranim originalniho OS a vsemi prostredky, ktere OS spravuje, ale muze byt ve skutecnosti izolovana (napr. sandboxing - bezpečnostní mechanismus v rámci počítačové bezpečnosti, který slouží pro oddělování procesů běžících se stejným oprávněním - jail, chroot, nebo prave i VM apod.)
        - v tomto kontextu termin virtualizace reflektuje zapouzdreni, narozdil od HW virtualizace, kde termin odrazi abstrakci
        - chroot, WoW (windows on windows), UAC (user account control)
        - skupinam procesu je prezentovano jine prostredi
        - implementuje se operacnim systemem
    - paravirtualizace
        - Xen, MS Hyper-V
        - na fyzickem stroji bezi nekolik upravenych OS
        - misto nebezpecnych operaci vola VMM
        - hypervizor resi alokaci zdroju a serializaci pristupu k zarizenim
        - prezentuje SW rozhrani virtualnim strojum, ktere je podobne, ale neni

identicke jako rozhrani pro HW, nad kterym lezi
        - zamer je redukovat cas, ktery stravi guest OS pri provadeni narocnych operaci ve virtualnim prostredi (ktere bezi jednoduseji na nevirtualizovanem prostredi) - poskytuji se specialne definovane hooks, ktere umoznuji guestovi/hostovi pozadat/potvrdit o nativni beh techto operaci, ktere by jinak byly spoustene ve virt. domene, coz by melo za nasledek horsi vykon
        - vyzaduje aby guest OS umoznoval explicitni podporu para-API (nekdy se pouzivaji alespon nektere komponenty, ktere API podporuji - napr. poskytuji balik ovladacu zarizeni, ktere podporuji para-API, ktere se instaluji do guest OSu)
    - virtualizace
        - VMWare, MS Hyper-V
        - na fyzickem stroji bezi nekolik neupravenych OS
        - hypervizor vytvari kazdemu z nich iluzi fyzickeho HW
        - http://en.wikipedia.org/wiki/Virtualization
        - HW virtualizace
            - IBM 370
            - VMWare pro 64bit stroje
            - fyzicky proc. vykonava puvodni instrukce
            - nebezpecne instrukce vyvolavaji VMM a jsou jim emulovany - trap and emulate
        - SW virtualizace s prekladem
            - fyzicky procesor vykonava upravene (prelozene) instrukce
            - instrukce preklada VMM z binarniho kodu (BT - binary translation)
            - nebezpecne instrukce jsou pri prekladu nahrazeny volanim VMM
            - preklad se obvykle tyka pouze jadra OS hosta
                - kod urceny pro privilegovany rezim procesoru
                - ve virt. stroji bezi v neprivileg. rezimu
            - aplikacni procesy bez prekladu
        - casto se kombinuje s prvky paravirtualizace
            - ve virtualnim stroji bezi SW (ovladace zarizeni) komunikujici s VMM
            - omezuje ztraty vykonu pri emulaci privilegovanych instrukci a IO
            - zvysuje uzivatelsky komfort
- motivace
    - lepsi vyuziti CPU
        - vetsina PC se totiz vetsinu doby flaka
    - lepsi vyuziti pameti, diskoveho prostoru
        - OS neumi bezbolestne pridat novy prostor - velikosti byvaji predimenzovane
        - virtualizace dokaze prezentovat vetsi nez skutecny prostor
    - moznost migrace virtualnich PC

        - load-balancing, fault-tolerance
    - vzdalena sprava
        - CD pro instalaci OS lze do virtualni mechaniky vlozit kliknutim mysi
    - checkpointy
        - nepovedene zmeny v konfiguraci lze vratit
    - vyuka uzivatelu a spravcu
        - viz. administrace windows, unixu
    - testovani a ladeni OS, siti, aplikaci
        - napr. zkoumani malware
- problemy
    - ztraty vykonu
        - silne zavisi na charakteru aplikaci i technologii virtualizace
        - nekdy jednotky, nekdy desitky procent
    - zmena charakteristik pri migraci
        - ruzna CPU, apod.
    - nespolehlive mereni/ladeni vykonu
    - nepripravenost fyzicke sitove infrastruktury
        - migrace virt. sitovych karet mezi fyzickymi
    - nepripravenost dodavatelu SW
        - nevyhodne licencni podminky
        - problemy s individualnimi checkpointy v komunikujicich systemech

Historie virtualizace

- prvni era
    - 1972 - IBM VM na S/370 mainframes
        - koexistence ruznych OS
        - time-sharing a virtualni pamet nad OS, ten tyto pojmy nezna
        - ladeni OS
    - 1980 - postupny zanik
        - levnejsi architektury - miniPC, PC
        - novy HW nepodporuje virtualizaci
        - nastup Unixu (VM zbytecne komplikuji komunikaci mezi procesy)
- druha era
    - 1999 - VMWare workstation
        - SW virtualizace
        - VMM (hypervizor) jako aplikace Windows NT
    - 2002 - VMWare ESX Server
        - VMM nahrazuje OS hosta
    - 2003 - Xen
        - paravirtualizace - modifikace OS guesta
- architektura x86 je nevhodna
    - dedictvi procesoru 80286
    - pokusy o napravu - rozsireni procesoru: 2005 Intel VT-x, 2006 AMD-V

    - naprava neni dodnes dokonala
        - velky podil SW virtualizace (v 1.ere byla HW)
        - nezanedbatelna ztrata vykonu - volani jadra, preruseni, virtualizace virt. pameti
    - situace se ale rok od roku zlepsuje

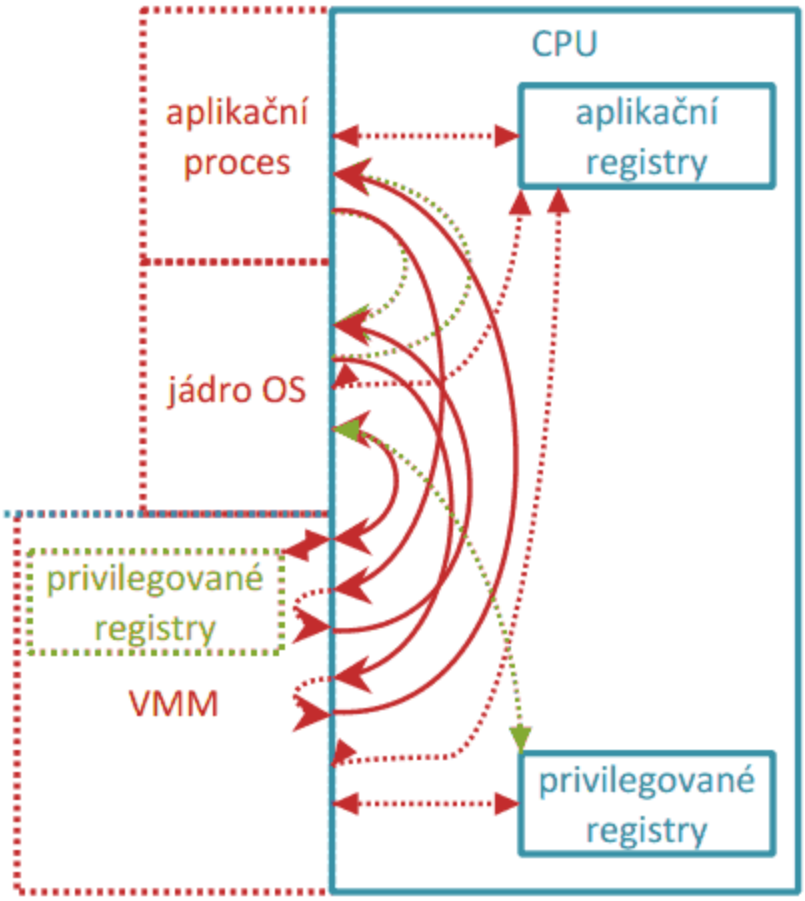
Požadavky na virtualizaci

- equivalence / fidelity (vernost)
    - program bezici pod VMM se musi chovat v zasade stejne jakoby bezel na ekvivalentnim stroji primo
- resource control / safety
    - VMM musi mit uplnou kontrolu nad virt. zdroji
- efficiency / performance
    - statisticky prevladajici cast strojovych instrukci musi byt provadena bez zasahu VMM

Implementace VMM

Virtualizace CPU

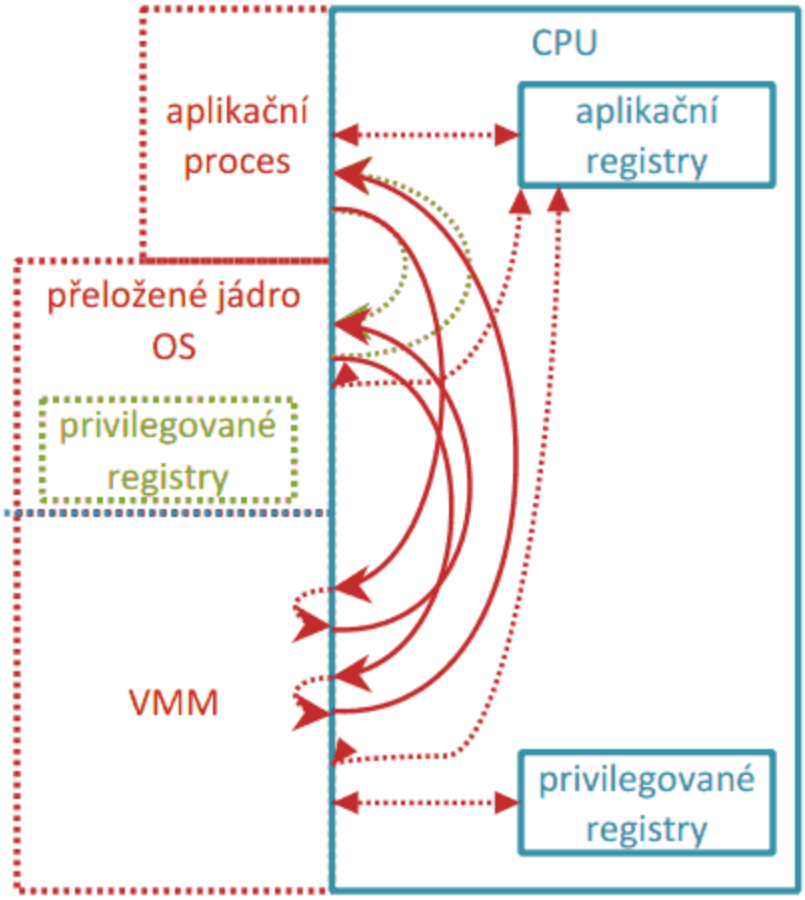
- situace na klasickem fyzickem CPU
    - rezim CPU je bud aplikacni nebo privilegovany
        - je to odliseno priznakem v privilegovanem registru
    - pro vstup do privilegovaneho rezimu
        - dojde nejprve k preruseni
        - instrukce pro volani jadra
        - vyvola se chyba
    - navrat z privilegovaneho rezimu
        - pouziti instrukce navratu
    - aplikacni proces ma svoje aplikacni registry, ke kterym pristupuje i jadro OS, jadro ma zase privilegovane registry, ke kterym ma pristup jen ono samo
- situace na virtualizovanem CPU - trap and emulate
    - aplikacni proces pracuje normalne
    - jadro OS
        - pracuje v aplikacnim rezimu
        - privilegovana instrukce zpusobi SW preruseni, ktere obslouzi VMM a odemuluje instrukci, ktera preruseni zpusobila
        - privilegovane registry virtualniho CPU jsou ulozeny v pameti VMM
    - aplikacni proces ma opet svoje aplikacni registry, ke kterym pristupuje i jadro OS, to ale nyni musi na privilegovane registry fyzickeho CPU pristupovat pres VMM - je tu vlozena vrstva navic



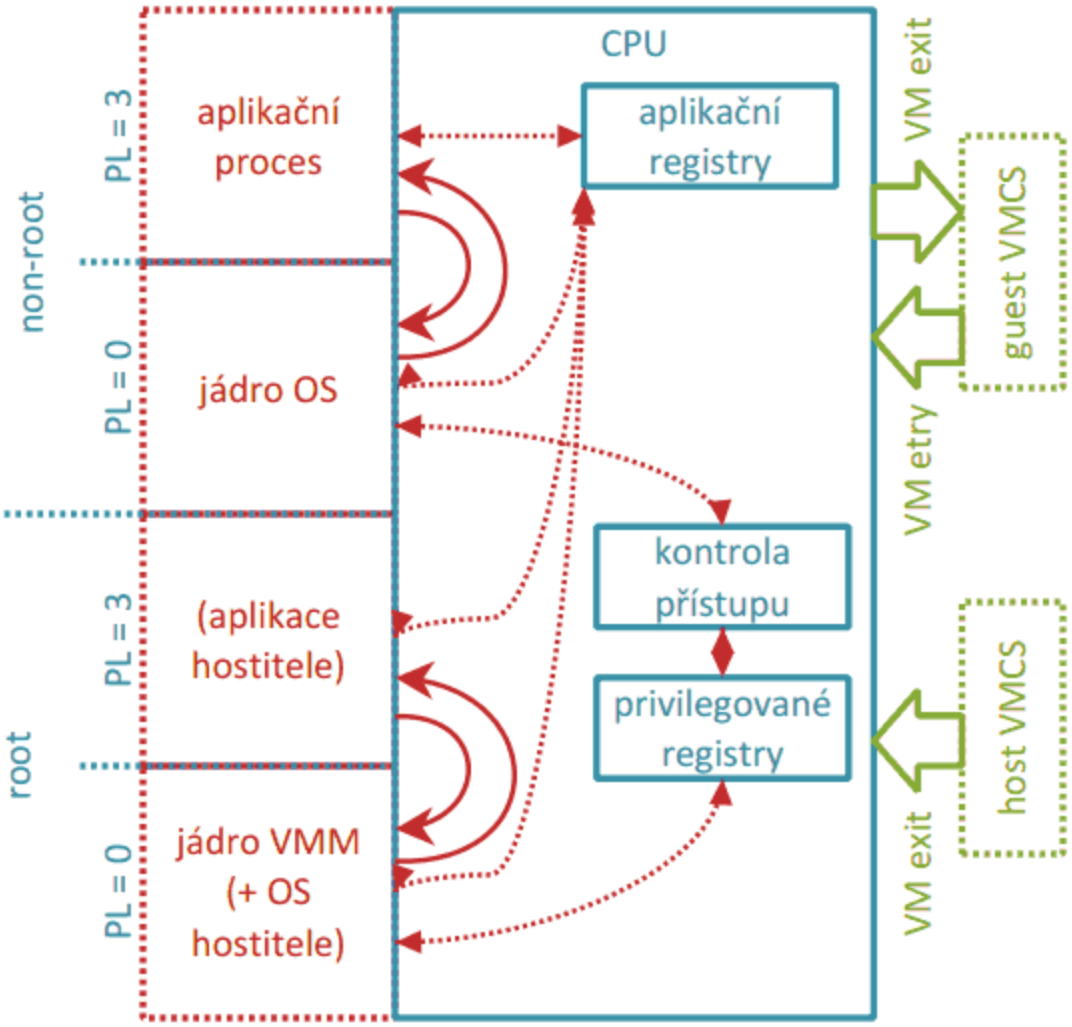
- problemy
  - jadro OS vyvolava hodne privilegovanych instrukci, pocet zavisi na dane architekture CPU, systemu a OS
  - SW emulace instrukci je pomala
    - rezie preruseni
    - rezie dekodovani
    - rezie zavisi na architekture CPU

  - vhodne u prvni ery virtualizace - IBM 370
    - I/O reseny HW kanaly → malo privilegovanych instrukci v OS
    - tenkrat miziva podpora paralelismu → levne skoky
    - jednoducha a pravidelna instrukcni sada → levne dekodovani
    - monoliticke aplikace → malo meziprocesove komunikace
  - nevhodna pro Intel x86 - typicke chyby:
    - cast privilegovanych registru je čitelná neprivilegovanou instrukci
      - current privilege level (CPL) je ulozen ve spodnich dvou bitech %cs (code segment - adresa kde bezi program) registru → instrukce typu mov %cs, %ax prozradi guest OSu, ze je to virtualni mod, coz je spatne

                - nektere instrukce se v ruznych rezimech chovaji ruzne
                    - treba nezpusobi preruseni
                - prilis mnoho instrukci jadra OS vyvolava v aplikacnim rezimu chybu
            - jadro virt. OS pracuje na jine prioritni urovni nez si mysli - “komprese privilegii”
                - nektere instrukce se tak chovaji jinak (x86)
                - resi se to velmi pracne prekladem napr. u VMWare
            - spolecny adresovy prostor
                - CPU neprepina adresovy prostor pri volani jadra
                    - ochrana OS resena privilegovanymi strankami
                - virt. OS naklada s VAS jako s vlastnim
                - nezbyva misto pro VMM - VMWare resi segmentaci
            - prilis mnoho prechodu VM-VMM
- SW emulace s prekladem
    - uprava jadra OS
        - binarni kod jadra je prekladacovymi technikami upraven tak, aby neprovadel privilegovane instrukce
        - privilegovane registry CPU si upraveny kod emuluje sam
        - VMM ale musi v nekterych pripadech zasahovat
            - navrat do aplikacniho procesu
            - akce s vyznamnymi efekty na strankovani apod.
            - I/O operace
            - system interruptu



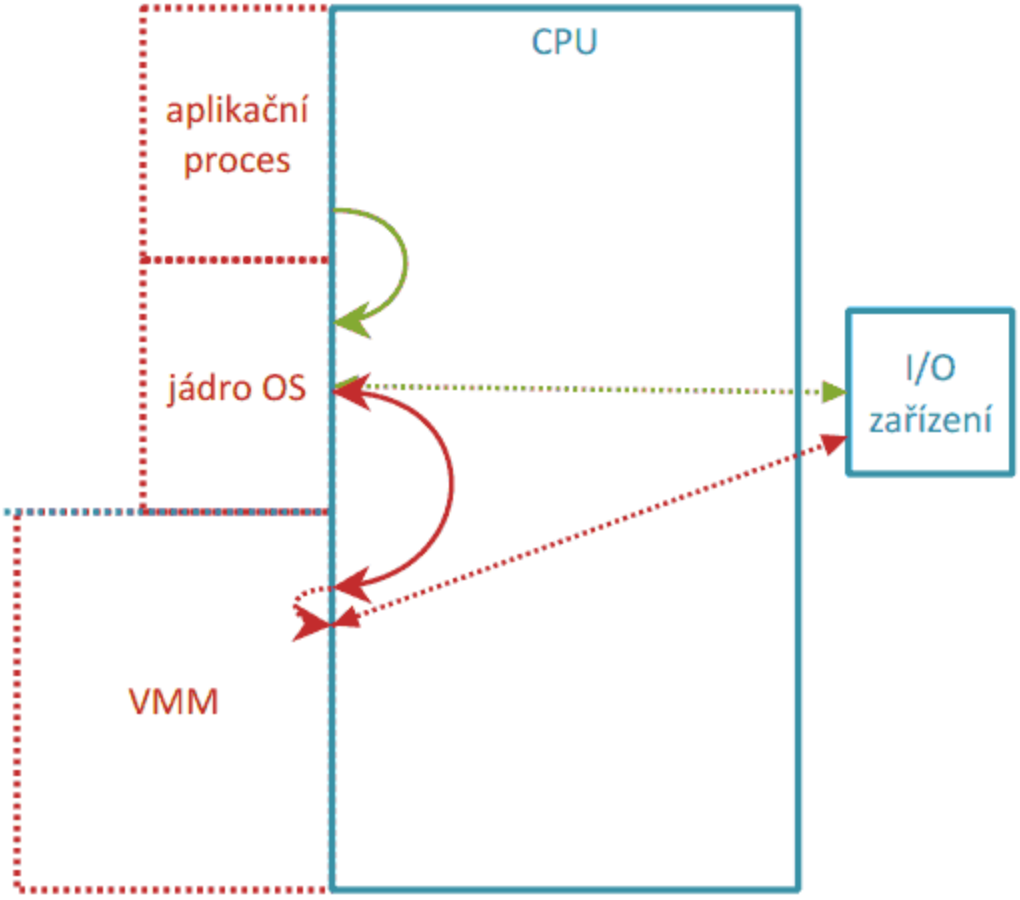
- HW podpora virtualizace
  - novy rozmer privilegovanosti
  - Intel VT-X / AMD-V, provedeni se lisi
  - VMCS - virtual machine control structures
    - HW-based pametova struktura na Intelu, ktera zrychluje operace VM entry a VM exit, ktere slouzi pro delegaci vykonavani privilegovane instrukce
  - uvazujeme root rezim CPU
    - odpovida CPU bez virtualizace
    - lze vyuzit pro beh host OS
  - non-root rezim CPU
    - ma omezeny pristup k privilegovanemu stavu
    - nezadouci akce zpusobuji VM exit (vypadek stranky napr.)
  - prepinani rezimu
    - kriticka cast stavu CPU se nacita/uklada do pameti
    - zahrnuje prepnuti strankovani



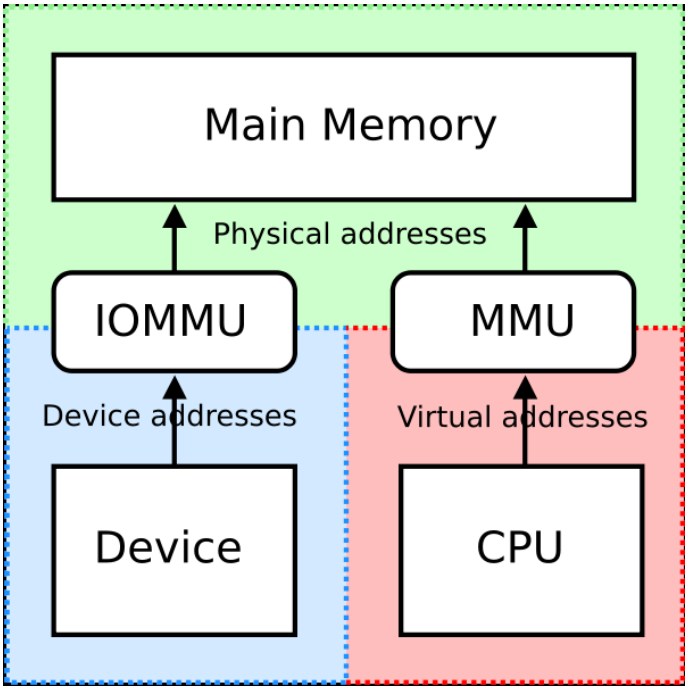
- vyhody:
  - odstranena komprese privilegii
    - za podminky ze virt. OSy samotne HW podporu virtualizace nepouzivaji
      - nelze virtualizovat virtualizator
      - na IBM 370 demonstrovano 5 urovni vnoreni virt.
  - prepinani adresoveho prostoru pri VM entry a VM exit
    - ochrana pameti VMM, plna transparence pro virt. OS
      - komplikuje vsak pristup VMM do pameti VM (emulace I/O apod.)
  - mensi pocet prechodu VM-VMM
    - lze vyladit konfiguraci HW kontroly pristupu k privilegovanemu stavu
    - demonstrovano asi 20x zrychleni nekterych uloh
      - unix fork & wait benchmark
      - kompilace rozsahlych projektu s malymi moduly

Virtualizace I/O

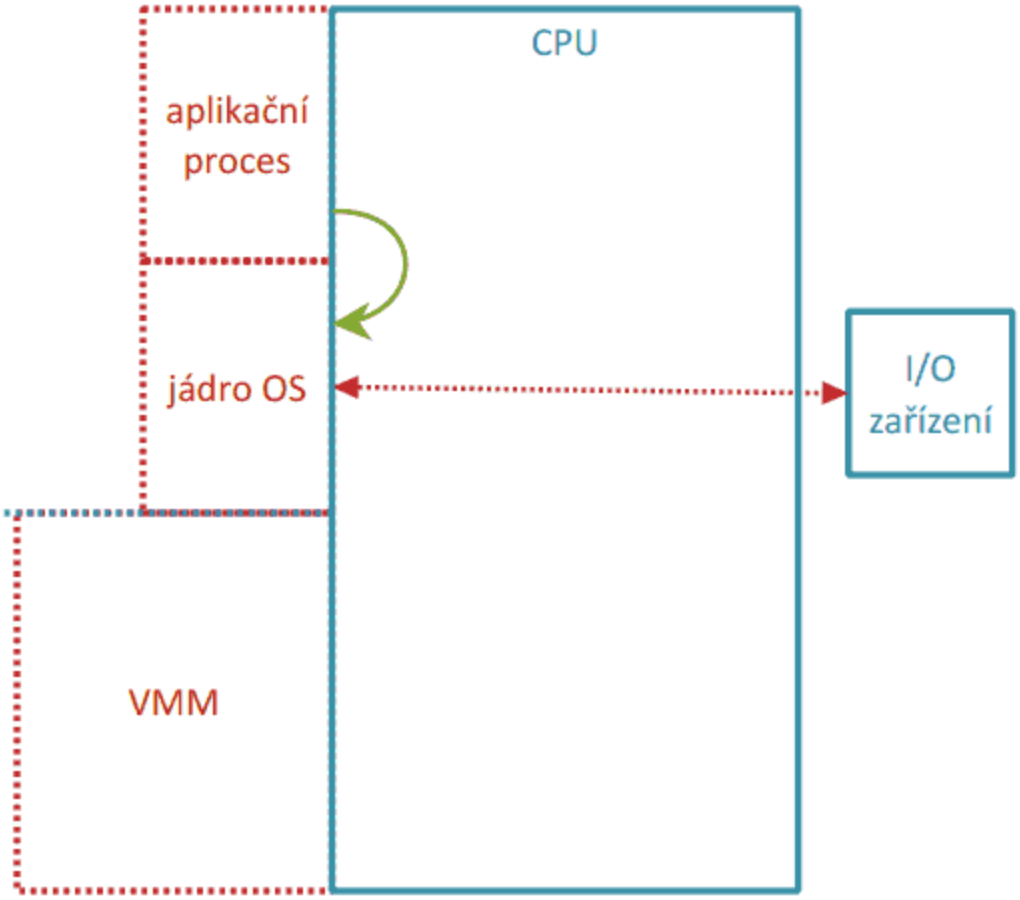
- situace na klasickem fyzickem CPU
  - aplikacni procesy realizuji veskere I/O volanim OS
  - OS komunikuje s I/O zarizenim
    - pouziva privilegovane I/O instrukce nebo
    - pametove mapovane zarizeni chranene strankovacim mechanismem
- pristup k fyzickemu IO na virtualnim CPU
  - privilegovane IO instrukce jsou provedeny emulatorem ve VMM
  - pametove mapovane zarizeni muze byt zpristupneno primo
  - exkluzivni pristup
    - k danemu zarizeni muze pristupovat jen jeden virtualni stroj
    - krome samotneho IO zarizeni je treba zpristupnit nebo virtualizovat system preruseni, pripadne DMA



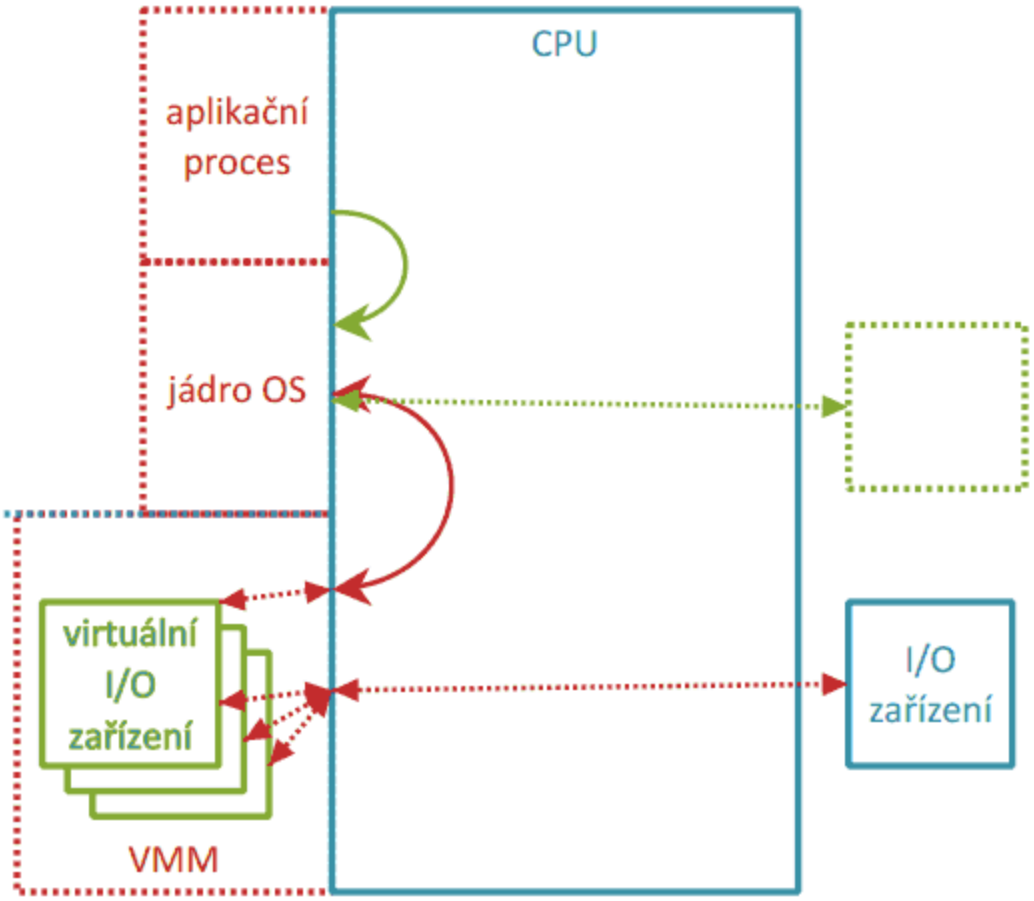
  - v pripade, ze IO instrukce k danemu zarizeni nejsou privilegovane, tak komunikuje jadro virtualniho OS primo s IO zarizenim
    - vyzaduje to konfigurovatelnost HW ochrany IO prostoru - IOMMU



- stejne podminky exkluzivniho pristupu
- moznost sdileneho pristupu
    - IO zarizeni se prezentuje vicekrat v IO adresovem prostoru
    - IO zarizeni ma pro kazdou adresu jednu kopii vnitrnich stavovych registru
    - krome samotneho IO zarizeni je treba zpristupnit nebo virtualizovat system preruseni, pripadne DMA - stejne jako u ex. pristupu



- pristup k virtualnimu IO na virtualnim CPU
    - privilegovane IO resp pristupy na memory-mapped IO jsou emulovane VMM
    - VMM pro kazdy stroj zvlast emuluje chovani HW
    - dany typ HW nemusi fyzicky existovat
    - sdileny pristup
        - VMM z emulovaneho HW extrahuje logicke akce
            - ty jsou provadene fyzickym zarizenim



- pristup k virtualnimu IO s upravou OS
  - zasah do OS
    - paravirtualizace - OS je vyrazne upraven
    - pri klasicke virtualizaci je do OS pridan ovladac virtualniho zarizeni
  - vyhody:
    - mezi OS a VMM jsou predavany logicke prikazy a nikoliv fyzicke IO
    - predani nevyzaduje emulaci IO instrukci
    - logickych prikazu je mene
    - serializace prikazu z ruznych VM je jednodussi

HW podpora virtualizace

- Intel VT-x a VT-d
  - řada rozšíření CPU i podpůrného čipsetu k podpoře virtualizace
    - stále přibývají další
  - jednotlivé úpravy jsou často použitelné nezávisle
  - vyznamné virtualizační SW je využívají téměř všechny
    - Intel spolupracuje s producenty HW
- AMD-V
  - upravy ve stejnem case podobnym smerem

        - vetsinou neni kompatibilni s Intelem
- situace zneprehlednena obchodni politikou
    - ruzne verze CPU maji ruzny uroven podpory
    - obchodni nazvy maskuji podstatu veci
        - nektera rozsireni jsou triviality, jina jsou velmi netrivialni
- Intel VT-x
    - extenze procesoru
    - root/non-root execution - reseni problemu komprese privilegii
    - extended page table (EPT) - reseni problemu virtualizace virtualni pameti
        - separatni mnozina str. tabulek preklada guestovi fyzicke adresy na hostovi fyzicke adresy → guest OS si muze modifikovat sve vlastni str. tabulky a primo obsluhovat vypadky stranek
        - umoznuje VMM vyhnout se VM exitum souvisejicich s virtualizaci str. tabulek, coz je hlavni overhead virtualizace bez EPT
    - flexpriority - virtualizace klicove casti radice preruseni
        - eliminace vetsiny VM exitu zpusobenych pristupy na privilegovane registry
    - flexmigration - virtualizace identifikace CPU a jeho schopností
        - moznost vyberu z dostupnych serveru, ktere jsou pro nas nejvhodnejsi - tim ze jsou virtualizovane jejich identifikace a dostupne schopnosti
        - migrace za behu mezi Intel servery
    - virtual processor ID (VPID) - klic zaznamu TLB obsahuje identifikator VM
        - pak neni treba invalidovat celou TLB pri prepinani VM-VMM, VM-VM
    - vguest preemption timer
        - casovac s lepsi granularitou a rychlejsi obsluhou
            - opt. na virtualizaci aplikaci s mirnymi real-time naroky
    - pause-loop exiting
        - HW podpora pro detekci spin-locků způsobující exit (preempci) do VMM
            - zlepšení výkonu
            - pro provoz více virt. procesorů na méně fyzických
    - real-mode support
        - podpora virtualizace pri startu virtualizovaneho OS (early VMM load, guest boot and resume)
- Intel VT-d
    - extenze pro podporu IO virtualizace v chipsetu
    - IOMMU
        - adresový prostor I/O má virtuální a fyzické adresy podobně jako paměť
    - interrupt-remapping support
        - částečná virtualizace řadiče přerušení
    - address translation services support
        - podpora virtualizace MMU při DMA
            - umozneni PCI-E zarizenim cachovat IOTLB polozky, ktere se pouzivaji pri DMA mapovani
        - rozšíření standardu sběrnice PCI Express

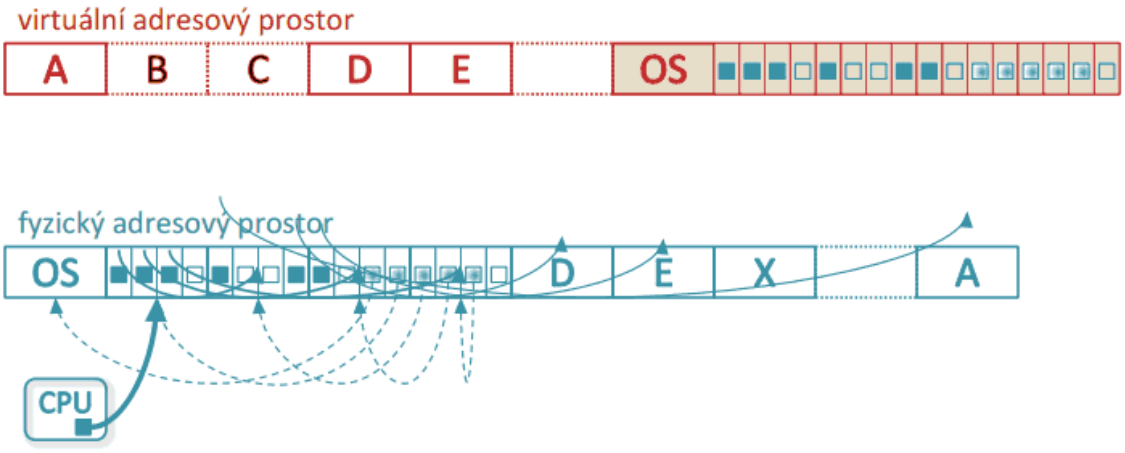
    - large Intel VT-d pages
        - podpora vetsich stranek - 2MB, 1GB
        - umožňuje sdílení CPU a DMA verzí stránkovacích tabulek
    - virtual machine device queue
        - zlepseni propustnosti, snizeni utilizace CPU
        - usporadani a grouping paketu na urovni NIC misto na urovni VMM
        - network interface card (NIC) s více stavovými prostory pro přímý přístup z VMs
    - single-root I/O virtualization (SR-IOV)
        - I/O zarizeni deklaruji sve schopnosti virtualizace (moznost sdileni ostatnimi)
        - rozšíření standardu PCI Express

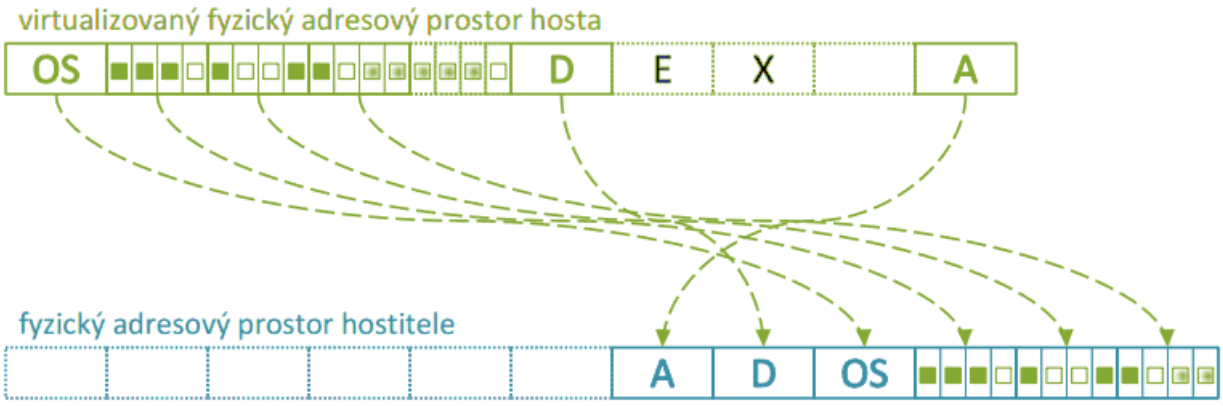
Virtualizace virtuální paměti

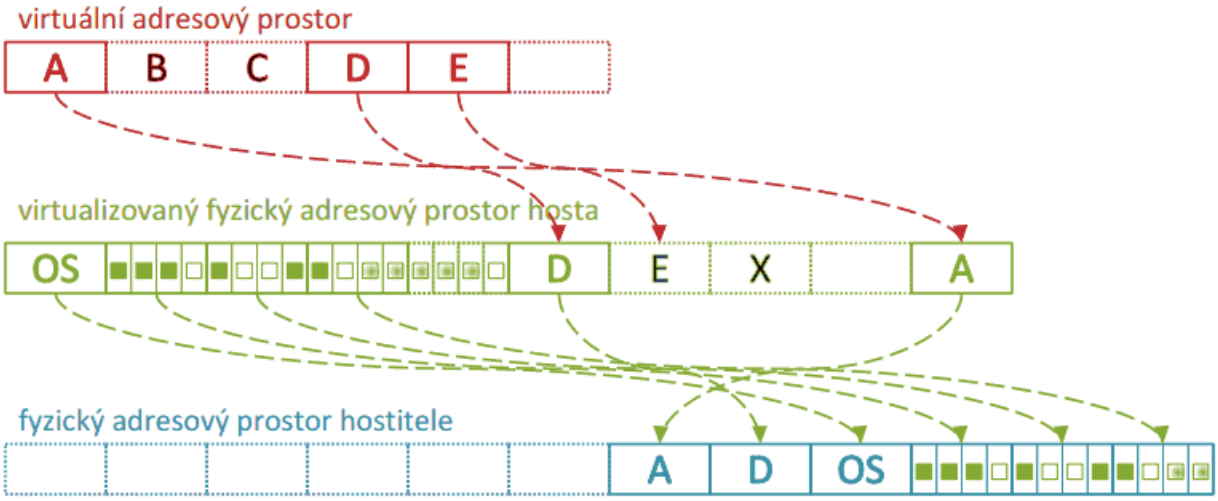
- VS ve fyzickem PC
    - adresovy prostor z pohledu procesu je jeden nebo vice souvislych useku (rozlozeni a vyznam urcen dohodou OS a aplikace)
    - deleni na stranky je neviditelne
    - abstraktni pohled na pamet
        - virtualni stranky mapovany na fyzicke ramce
        - stranky odlozene na disk mapovany nejsou
        - PAS sdilen mnoha procesy
    - 2urovnove strankovani (x86)
        - pri vypadku TLB procesoru se prochazi 2 urovne stranek
        - strankovaci tabulky v PAS
        - fyzicka adresa korene (directory page table) je v registru CPU
    - kod a data OS byvaji soucasti VAS
        - prepinani strankovani pri kazdem volani OS by bylo neefektivni
        - stranky OS pristupne pouze v privilegovanem rezimu procesoru
        - volani OS provedeno spec. instrukci, ktera zapina privilegovany rezim
    - OS plni strankovaci tabulky
        - strankovaci tabulky jsou mapovany podobne jako data OS
        - OS zapisuje do str. tabulek beznymi instrukcemi
        - zapis vetsinou musi byt nasledovan privileg. instrukci TLB flush
- VS ve virtualnim PC
    - VMM poskytuje iluzi PAS
        - mapovani PAS guesta na PAS hosta
    - VMM muze odkladat stranky na disk podobne jako OS
    - provadeny kod pracuje s VAS guesta
    - potrebne mapovani vznikne slozenim:
        - mapovani definovaneho OSem guesta
        - mapovani definovaneho virtualizací PC guesta
    - ve VAS bezi:

        - aplikacni procesy guesta
        - OS guesta
        - VMM

    → potrebujeme 3 urovne privilegii ke strankam
- realizace slozeneho mapovani
    - pomoci strankovacich tabulek v PAS hosta - obsahuji fyzicke adresy v prostoru hosta
- v systemu jsou dvoji strankovaci tabulky
    - strankovaci tabulky hosta pouzivane fyzickym CPU
        - obsahuji fyzicke adresy v prostoru hosta
    - virtualizovane strankovaci tabulky guesta pouzivane OSem guesta
        - obsahuji virtualizovane fyzicke adresy v prostoru guesta
- VMM pocita vysledne mapovani ze strankovacich tabulek guesta
    - OS guesta zapisuje do svych tabulek neprivilegovanymi instrukcemi
    - VMM musi zajistit primerenou koherenci fyzickych tabulek a tabulek guesta
        - virtualizovane tabulky guesta mohou byt mapovany read-only a zapisy emulovany
        - koherenci lze udrzovat v ramci emulace privilegovane instrukce TLB flush
- slabsi VMM neumi odkladani virtualizovane pameti na disk
    - mapovani virtualizovanych PA na PA je identicke
        - VMM pouze kontroluje, zda OS guesta nemapuje nezadouci PA
    - OS guesta se musi vyrovnat s dirami v PAS
        - pouzivano prevazne pri paravirtualizaci (Xen)







Data center HW

- motivace
  - standardizace/konsolidace (upevneni)
    - omezit pocet DC v organizaci
    - omezit pocet HW, SW platforem
    - standardizovat vypocetni, sitove a ridici platformy
  - virtualizace
    - konsolidace vice vybaveni DC
    - nizsi kapitalove a provozni vydaje
  - automatizace
    - konfigurace, zaplatovani, vzajemna koordinace, …

    - zabezpeceni
        - fyzicke, sitove, data, uzivatele
- pozadavky datovych center (serveroven)
    - historicky byly datova centra navrhovany bez zavedenych standardu, spousta sitovych administratoru tak musela celit vyberem pouzitych technologii a resenim jejich implementace v casto omezenem prostoru mistnosti
    - 2005 - TIA-942 standard
        - definovani norem, ktere musi centra splnovat z hlediska integrity a funkcnosti vypocetni techniky, ktera je v nich umistena
        - definovani 4 urovni data center podle jejich spolehlivosti
            - pozadavky pro tier 1
                - jedna neredundantni distribucni cesta pro vedeni elektriny a chlazeni
                - neredundantni kapacita s dostupnosti 99.671 %
                - umoznuje aby system nebyl dostupny max 1729 min. v roce
            - pozadavky pro tier 2
                - splnuje pozadavky (nebo prevysuje) pro tier 1
                - redundantni kapacita s dostupnosti 99.741 %
                - umoznuje aby system nebyl dostupny max 1361 min. v roce
            - pozadavky pro tier 3
                - splnuje pozadavky (nebo prevysuje) pro tier 1, 2
                - nekolik nezavislych distribucnich cest
                - vsechny IT komponenty musi mit napajeny ze dvou zdroju, navic plne kompatibilni s topologii infrastruktury site
                - konkurentně udržovatelná infrastruktura s dostupností 99.982 %
                - umoznuje aby system nebyl dostupny max 95 min. v roce
            - pozadavky pro tier 4
                - splnuje pozadavky (nebo prevysuje) pro tier 1, 2, 3
                - vsechny chladici komponenty jsou napajeny ze dvou zdroju
                - infrastruktura s odolnosti vuci chybam a dostupnosti 99.995 %
                - umoznuje aby system nebyl dostupny max 26 min v roce
- problemy
    - design
        - strojirensky design infrastruktury
            - mechanicke systemy podilejici se na udrzovani konzistentniho vnitrniho prostredi
            - topeni, ventilace, klimatizace
            - zvlhcovani/odvlhcovani, tlakovani
            - minimalizace prostoru a cen pri zachovani dostupnosti

        - elektrotechnicky design infrastruktury
            - distribuce, prepinani, UPS
            - modularne, skalovatelne
        - technologicka infrastruktura
            - kabelove propojeni pro datovou komunikaci, sprava pocitacu, klavesnice/mysi/video snimaci technika, …
        - predpoklady dostupnosti
            - pro zajisteni vetsi dostupnosti je nutny vetsi vstupni kapital a vyssi provozni naklady

        - vyber lokality
            - dostupnost rozvodnych siti, sitove sluzby, dopravni dostupnost, moznost akutnich zasahu
            - klimaticke podminky
        - modularita, flexibilita, kontrola teploty, vlhkosti (16-24°C, 40-55% vlhkost), elektricke rozvody (UPS, banky s bateriemi, naftove generatory, …), ochrana pred katastrofami (ohen, zaplavy), detektory koure, automaticke ostrikovace pri pozaru, fyzicka bezpecnost, …
    - spotreba energie
        - účinnost využití energie (power usage effectiveness) - PUE = celkovy vykon zarizeni / vykon veskereho IT vybaveni
            - nejmodernejsi DC maji PUE ~ 1.2
        - analyzovani spotreby energie, chlazeni
            - energie cini nejvetsi opakujici se naklady
            - analyza mist s extremy - hot spots, over-cooled areas
        - rozmisteni vybaveni DC
    - dalsi aspekty
        - infrastruktura site
            - routery, switche
            - 2 nebo vice poskytovatelu upstream sluzeb (aby se mohlo vysilat hodne dat)
            - firewally, VPN brany, IDS (intrusion detection system - pro odhalovani pruniku do systemu)
        - rizeni DC infrastruktury
            - monitorovani apod.
        - aplikace
            - DB, file servery, aplikacni servery, zalohy
- blade servery
    - ocesany server s modularnim designem optimizovan pro minimalizaci vyuziti fyzickeho prostoru a energie
    - narozdil od standardniho racku, kteremu staci k funkci napajeni a sitovy kabel je blade server ocesan o spoustu komponent pro usetreni mista a minimalizaci spotreby

- skrin/uzaver (enclosure) se sklada ze spousty blade serveru
  - skrin dostatecne chlazena
  - redundantni zdroje energie
  - chladici system s ridicim systemem starajicim se o regulaci otacek ventilatoru, pouziti kapalinoveho chlazeni
- servery obsahuji typicky integrovany NIC pro ethernet nebo radic pro fiber channel rozhrani (pouziti na kroucene dvojlince nebo opticke siti)
- datova uloziste typicky nejsou umisteny lokalne - pripojeni pres firewire/eSATA/SCSI/iSCSI rozhrani - SAN

- SAN
  - storage area network
  - na zaklade zajmu o oddeleni diskoveho uloziste a procesoroveho vykonu serveru vznikly site SAN
    - narozdil od DAS (directly attached storage), kde je problem se pri poruse serveru jednoduse dostat k datum, spatna rozsiritelnost
  - pro propojeni se pouziva technologie fiber channel, pres opticke kabely, umoznuji propojeni na vzdalenosti desitek az stovek metru
  - propojeni pres FC switch ke spolecnemu datovemu ulozisti - typicky diskove pole s vlastni inteligenci (radic RAID)
  - pouzivane technologie:
    - FC
      - vysoke naklady
      - vysoke naroky na obsluhu systemu
      - spis vetsi podniky
      - propustnost 800/1600/3200 MBps
      - ruzne topologie - P2P, switched fabric, arbitrated loop
    - iSCSI
      - posledni dobou se FC nahrazuje iSCSI
      - vychazi ze 2 technologii
        - SCSI rozhrani pro pripojovani disku v serverech
        - technologie TCP/IP
      - ze SCSI si bere pouze protokol, kterym spolu zarizeni komunikuji, zcela opousti jeho fyzickou vrstvu (kabely, konektory apod)
      - pri prenosu paketu SCSI se pouzije jejich zapouzdreni do protokolu TCP/IP
      - nepomerne levnejsi nez FC
      - iniciator pripojeni je klient - HW/SW, cilem je prostredek s daty
      - umoznuje network booting
    - FCoE
      - zapouzdreni FC pres Ethernet
    - iSER
      - iSCSI extension over RDMA (remote DMA)

                - InfiniBand
        - vyhody SAN
            - snadne pripojeni novych diskovych poli (u klasickych ethernet siti bez SAN technologii by se musel koupit novy server, coz by se prodrazilo)
            - sdilene uloziste (vice serveru vidi ten samy logicky svazek)
- diskova pole
    - diskovy ulozny system s nekolika diskovymi zarizenimi
    - komponenty
        - ovladace diskovych poli
        - cache - RAM, disk
        - skrine/pouzdra na disky
        - zdroje napajeni
    - co poskytuji
        - dostupnost, udrzovatelnost, elasticnost
        - redundance dat, hot swap (rychla vymena disku), RAID
    - kategorie
        - NAS, SAN, hybridni
    - enterprise diskova pole
        - vlastnosti navic
            - automaticky prevezme sluzbu pri selhani jineho disku
            - vyroba snapshotu
            - deduplikace
                - speciální technika komprese dat, která zabraňuje ukládání stejných datových bloků na jednom úložišti
            - replikace
            - tiering
                - technika, ktera presouva datove soubory, oddily nebo bloky mezi ulozistemi na zaklade definovane politiky
            - virtualni svazky
            - hot spare
                - disky, ktere umoznuji rychle nahradit funkci poskozeneho disku
    - urovne RAIDu
        - Redundant Array Of Independent Disks
        - proc
            - zvyseni dostupnosti
                - snizeni MTBF (mean time between failure)
                - snizeni MTTR (mean time to repair)
            - zvyseni vykonu
        - JBOD
            - Just a Bunch Of Disks
            - realne zadna RAID funkce
            - min # disku = 1

        - prostorova efektivita = 1
        - odolnost vuci chybam = 0
        - mira selhani diskoveho pole = 1 - (1-R)^N
            - N = # disku
            - R = mira selhani disku
            - priklad: 3 disky, kazdy z disku ma miru selhani 5%, pouzivaji JBOD, tak mira bude 1 - (1-0.05)^3 ~ 0.14 ~ 1.4%
        - vykonnost cteni = 1
        - vykonnost zapisu = 1
    - RAID 0
        - striping - segmentace logicky sekvencne ulozenych dat na ruzna diskova zarizeni
        - bez redundance
        - min # disku = 2
        - prostorova efektivita = 1
        - odolnost vuci chybam = 0
        - mira selhani diskoveho pole = 1 - (1-R)^N
        - vykonnost cteni = N
        - vykonnost zapisu = N
        - → pro zvyseni vykonu, kdyz nejde moc o integritu
            - pouziva se u nekterych hernich systemu
            - nicmene ukazalo se ze striping nemusi vzdy pridat vykon, nekdy naopak muze i snizit, navic pri testovani PC her neprineslo moc navyseni, ale u desktop systemu se muze hodit
    - RAID 1
        - zrcadleni obsahu
        - min # disku = 2
        - prostorova efektivita = 1/N
        - odolnost vuci chybam = N-1
        - mira selhani diskoveho pole = R^N
        - vykonnost cteni = N
        - vykonnost zapisu = 1
        - → pro zvyseni spolehlivosti, rychlost cteni
    - RAID 2
        - striping na urovni bitu na zaklade pouziti Hammingova kodu pro zjisteni parity
        - min # disku = 3
        - prostorova efektivita = 1 − 1/n * log2(n-1)
        - odolnost vuci chybam = 1
            - vzpamatuje se z kleknuti jednoho disku
        - mira selhani diskoveho pole = promenna
        - vykonnost cteni = promenna, vykonnost zapisu = promenna

        - → uz se nepouziva - zbytecna redundantni kontrola chyb a velka komplexita
    - RAID 3
        - striping na urovni bytu na zaklade parity
            - jeden disk je vyhrazen pro ukladani informaci o parite
                - pr. 4 disky, 4. vyhrazen pro ukladani parity, mame urovni bytu → i-ty byte na 4. disku odpovida parite i-tych bytu na discich 1-3
        - min # disku = 3
        - prostorova efektivita = 1 - 1/N
        - odolnost vuci chybam = 1
        - mira selhani diskoveho pole = N * (N-1) * R^2
        - vykonnost cteni = N-1, vykonnost zapisu = N-1
        - obecne nemuze zpracovavat vice pozadavku naraz, ale hodi se na vysokorychlosti cteni/zapis dlouhych souvislych useku dat - napr. editace nekomprimovaneho videa
    - RAID 4
        - striping na urovni bloku na zaklade parity
        - min # disku = 3
        - prostorova efektivita = 1 - 1/N
        - odolnost vuci chybam = 1
        - mira selhani diskoveho pole = N * (N-1) * R^2
        - vykonnost cteni = N-1, vykonnost zapisu = N-1
        - moc se nepouziva
    - RAID 5
        - striping na urovni bloku na zaklade distribuovane parity
            - informace o parite je narozdil od minulych pripadu distribuovana mezi disky
        - min # disku = 3
        - prostorova efektivita = 1 - 1/N
        - odolnost vuci chybam = 1
        - mira selhani diskoveho pole = N * (N-1) * R^2
        - vykonnost cteni = N-1, vykonnost zapisu = N-1

    - RAID 6
        - rozsireni RAID 5 o dodatecny blok s paritou - striping s dvema distribuovanymi paritovymi bloky
        - min # disku = 4
        - prostorova efektivita = 1 - 2/N
        - odolnost vuci chybam = 2
        - mira selhani diskoveho pole = N * (N-1) * (N-2) * R^3
        - vykonnost cteni = N-2
        - vykonnost zapisu = N-2

        - hybridni
            - RAID 0+1 - striped sets in mirrored set
            - RAID 1+0 (RAID 10) - mirrored set in striped sets, kazde zrcadleni muze ztratit disk a nic se nestane
            - RAID 5+0 (RAID 50) - block striping s distr. paritou in striped set
                - odolnost - jeden disk muze spadnou v kazdem RAID5 bloku

Virtualizační infrastruktura (VMWare)

- fyzicky stroj
    - HW
        - CPU, RAM, disky, I/O
        - nevyužitý HW naplno
    - SW
        - jeden aktivni OS
        - OS kontroluje HW
- virtualni stroj
    - abstrakce na HW urovni
        - virtualni HW ~ CPU, RAM, disky, I/O
    - virtualizacni SW
        - oddeluje HW a OS
        - multiplex fyzickeho HW napric nekolika guest VM
        - silne izolovane VM mezi sebou
        - ridi fyzicke prostredky, optimalizuje vyuziti
    - izolace
        - bezpecny multiplexing
            - nekolik VM na jednom fyzickem hostu
            - CPU HW izoluje mezi sebou VM (MMU)
        - silne zaruky
            - SW bugy, pady v ramci jedne VM nemuzou ovlivnit ostatni
        - oddeleni vykonu
            - rozdeleni systemovych prostredku
    - zapouzdreni
        - kazda VM je jeden soubor
            - OS, aplikace, data
            - pamet a stav zarizeni
        - moznost vytvareni snapshotu, klonu
            - zachytit stav VM za behu a obnovit do nejakeho zachytneho bodu
            - zalohovani, zrcadleni

        - jednoducha distribuce obsahu
            - predkonfigurovane aplikace
            - virtualni zarizeni
    - kompatibilita
        - HW nezavisle
            - fyzicky HW skryty virtualizacni vrstvou
            - standardni virtualni HW vystaven kazde VM
        - jednou vytvorit, pustit kdekoliv
            - zadne problemy s konfiguraci masin
            - migrace VM mezi hosty
        - podpora zastaralych systemu
            - muzeme si ho pustit na nove platforme
    - sprava prostredku (distributed resource scheduler)
        - “boj o prostredky”
        - kontrola alokace prostredku na ruznych urovnich
        - na urovni sdileni
            - specifikace relativni dulezitosti mezi VM - podilu na vykonnosti
            - naroky primo umerne podilu
                - VM ma nasdileno dvakrat vic nez jina, je oznacena, ze zere 2x vic
            - abstraktni relativni jednotky - dulezity je pouze pomer
                - low/normal/high
                    - 500/1000/2000 shares per virtual CPU
        - na zaklade rezervovani
            - dana garance minimalni alokace i kdyz je system zahlcen
                - mame 2GHz procesoru k dispozici, kdyz nastavime 1GHz garanci pro proc. 1 a 2, tak oba maji jistotu, ze vzdy dostanou aspon 1GHz kdyz chteji
                    - pokud jeden z nich ale vyuziva jen 500Mhz, druhy muze 1.5GHz
            - v konkretnich absolutnich jednotkach
            - rizeni pristupu: suma rezervaci <= kapacita
            - default = 0

        - na zaklade limitu
            - stanovuje horni mez, kolik muze VM spotrebovat, i kdyz je system zahlcen
            - v konkretnich absolutnich jednotkach
            - default = inf

- Proportional-share scheduling
  - Simple virtual-time algorithm
    - Virtual time = usage / share
    - Schedule VM with smallest virtual time
  - Example: 3 VM A, B, C with 3:2:1 share ratio

A	2	4	4	6	8	8	8	10	10
B	3	3	6	6	6	9	9	9	12
C	6	6	6	6	6	6	12	12	12

- vyuziti distribuovanych systemu
  - vyber inicialniho hosta kdyz se VM zapne
    - predtim, nez se zacne neco vykonavat nemame zadny realny odhad vyuziti prostredku → vybirame jen na zaklade nastaveni stroje (predpoklada se, ze vyuzije vse, co si nastavil)
  - migrace bezicich VM mezi fyzickymi hosty
    - rychla migrace, transparentni guest OSu i aplikacim
    - minimalni cas nedostupnosti (<= 1s)
    - pozadavky:
      - sdilene uloziste
      - na stejne podsiti
      - kompatibilni CPU
        - aby mel CPU, kam se migruje stejnou podmnozinu instrukci
          - nadefinuje administrator
          - instrukce, kterou se tazeme procesoru na jeho instrukci sadu neni volana primo, ale pouzije se mechanismus trap and emulate
  - dynamicky load balancing
    - uvazujeme dynamicky entitlement - vypocitavame na zaklade celkove kapacity clusteru, pouzivanych zdroju (prostredku) a aktualniho vytizeni CPU a pameti na jednotlivych VM
    - algoritmus:
      - mame snapshot celeho clusteru (VMs, hosts)
      - spocitame normalizovany entitlement pro kazdeho hosta ~

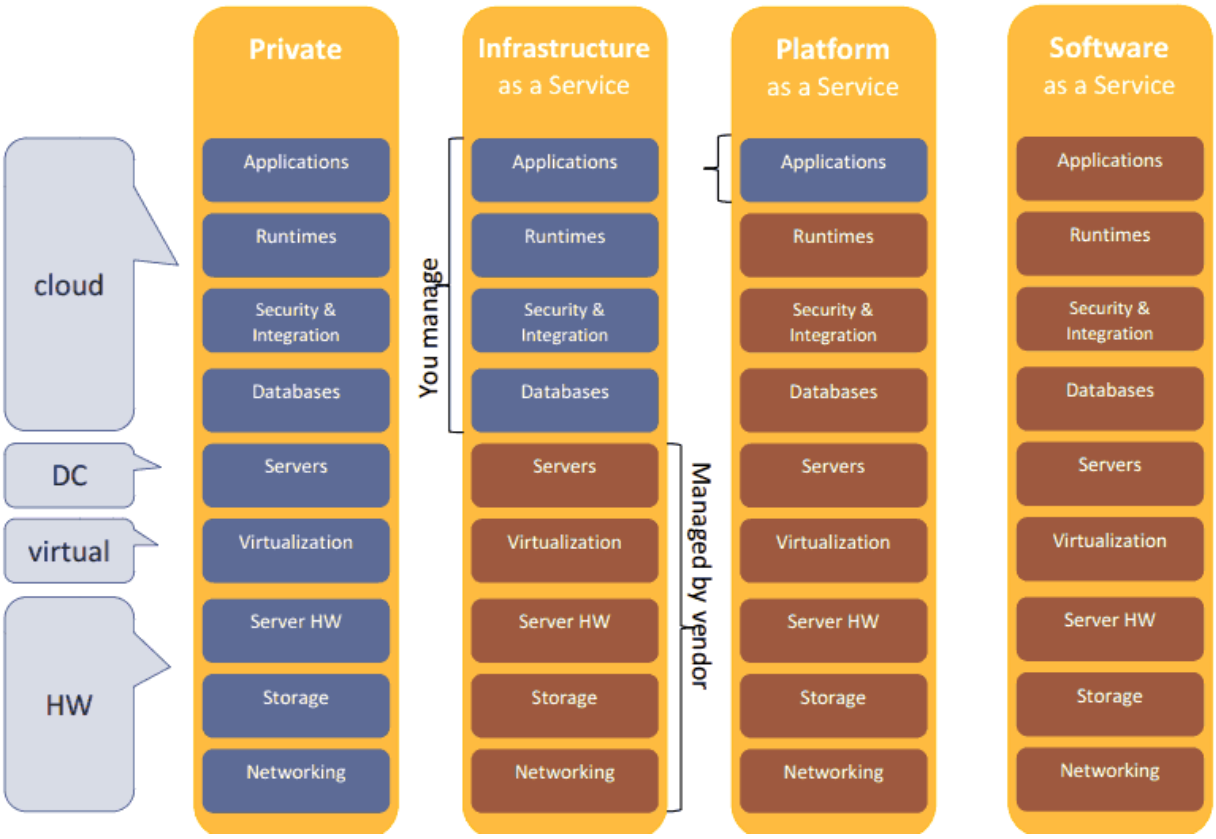
N = sum Ei/Cj, kde Ei jsou pozadavky jednotlivych VM na hostu j a Cj je kapacita hosta j
        - tedy napr. host 4GHz, bezi tam VMs A = 3GHz, B = 2GHz → entitlement = 5/4 = 1.25
    - spocitame std. odchylku nároků přes všechny hosty
    - pokud je odchylka větší než nějaký threshold (prostě jsou moc zatížené), pokračujeme dál
    - pro každou VM v clusteru spočítáme, jak je výhodný přesun na nějakého kompatibilního hosta - jak se zlepší odchylka
    - vybereme nejvyhodnejsi presun
    - takto muzeme vybrat nekolik presunu, do nejakeho limitu, ktery je nastaven (konstanta max migrations)
- distribuovany power management (DPM)
    - konsolidace VM na mene hostu a vypnuti hostu, pokud je nizka poptavka
    - zapnuti hostu znovu pokud je nutne splnit pozadavky vytizeni procesoru nebo splnit nastavene podminky, omezeni
    - pracuje ve shode s distrubovanym planovacem prostredku (DRS)
- distribuovana vysoka dostupnost
    - na zaklade pozadovane kapacity k rezervaci restartuje danou VM po selhani jejiho hosta na jinem hostu
        - definuje se pocet padu hosta, ktere se jeste toleruji
        - jake procento kapacity clusteru host zabira
        - pocet hostu, ktere jsou nachystani bokem pro prevzeti kontroly po padu
    - pravidelne a s pouzitim inteligentnich mechanismu monitoruje vyuziti dostupne kapacity
    - chytre prevzeti funkce poskozeneho stroje serverem s nejlepsimi dostupnymi prostredky
    - decentralizovana detekce poskozeni hosta
        - hosti v clusteru si navzajem posilaji informaci o jejich tiku, kdyz jeden z hostu v nejakem timeoutu neodpovi, spusti se prislusna akce, ktera restartuje dany stroj
    - funguje ve shode s distribuovanym planovacem prostredku a distribuovanym power managementem
    - DRS/DPM/HA aktivne spravuji prislusne volne prostedky
- distribuovany I/O management
- odolnost vuci chybam
    - automaticka detekce chyb stroju, okamzite vyvolani triggeru, probehne neviditelne prevzeti funkce poskozeneho stroje jinym
    - vyvolani vytvoreni sekundarniho VM na jinem hostu ihned po failoveru pro zajisteni nepretrzite ochrany aplikace, tyto pak sdili prostredky na SAN

        - primarni VM pak funguje normalne a sekundarni provadi pouze read operace
        - pokud pak nekdy host pro primarni VM spadne, sekundar ho nahradi bez jakehokoliv preruseni (~ 5 ms !)
        - zajisteni dostupnosti pri zachovani vysoke rychlosti
            - dnesni CPU mohou replikovat proud instrukci
    - snapshots/checkpoints
        - dump celeho stavu do stroje do zalozniho souboru
            - VM settings, VM disks content, VM memory content, registry CPU
        - budouci zapisy jsou presmerovany do druheho souboru, ktery pak funguje systemem copy-on-write → inkrementalni
            - proste si nejprve udelam cely snapshot a pak si uchovavam jenom nejaky zurnal zmen
    - podpora cloud-computingu, multi-tenancy

Cloud computing

Execution Models	Virtual Machines	Web Sites	Cloud Services
Cloud Storage	SQL Database	Key-Value Tables	Blobs
Data Processing	Map/Reduce	Hadoop	Reporting
Networking	Virtual Network	Connect	Traffic Manager
Messaging	Queues	Service Bus	Service Bus
Caching	Caching	Content Delivery	
Hi-Perf Computing	Scheduler	Load Balancing	
(Multi-)Media	Media Services	Streaming	
	GIS	Searching / Indexing	Marketplace
Other Services	Mobile / E-mail	Language / Translate	Collaboration
	Gaming	Prediction	
SDK	C++ .Net Java PHP Python Node.js ...		

- nabizi se ruzne urovne sluzeb
- IaaS (infrastructure as a service)
    - zakladni cloudovy model
    - poskytovatele nabizi pocitace - fyzicke/virtualni, servery a dalsi prostredky
        - mezi dalsi prostredky patri napr uloziste souboru, firewally, IP adresy, virtualni LAN site, SW baliky
    - urceno pro sitove architekty
    - Windows Azure VMs
- PaaS (platform as a service)
    - dodani pocitacove platformy typicky zahrnujici OS, prostredi pro programovaci jazyky, databazi, webovy server
    - vyvojari aplikaci mohou vyvijet a spoustet sve aplikace na cloudove platforme bez toho aby museli kupovat a zarizovat HW a SW vrstvy, ktere pod platformou lezi
    - Windows Azure, Google App Engine
    - urceno pro vyvojare aplikaci
- SaaS (software as a service)
    - pristup k aplikacnimu SW a databazim
    - “on-demand” SW, typicky se za to plati na bazi “pay-per-use” (monthly/yearly fee per user)
    - urceno pro koncove uzivatele
    - Grooveshark, Last.FM, Rapidshare, online hry apod



- execution models
  - uroven poskytovani sluzeb - nemusim se starat o nic vs muzu cokoliv
  - virtualni stroje
    - mame k dispozici obrazy/galerii svych VM
    - dalsi cloudove sluzby - DB, messaging
    - pay-per-use, pay-per-config - memory, processors, disk space
    - administrace pres webovy portal, skriptovaci konzole, API (REST)
    - bezici VM
      - virtualni disky - OS a data
      - typicky BLOBy
    - vyuziti
      - development, test environment
      - beh aplikaci, provoz sluzeb
      - rozsireni datacentra
      - disaster recovery
    - REST (representational state transfer) API a scripting
      - architektura rozhraní, navržená pro distribuované prostředí
      - jednotny pristup ke zdrojum - CRUD (create, read, update, delete)
      - pristup k sluzbam i datum
        - http://My.tables.MyCloud/<MyTable>(PartitionKey='PK',Ro

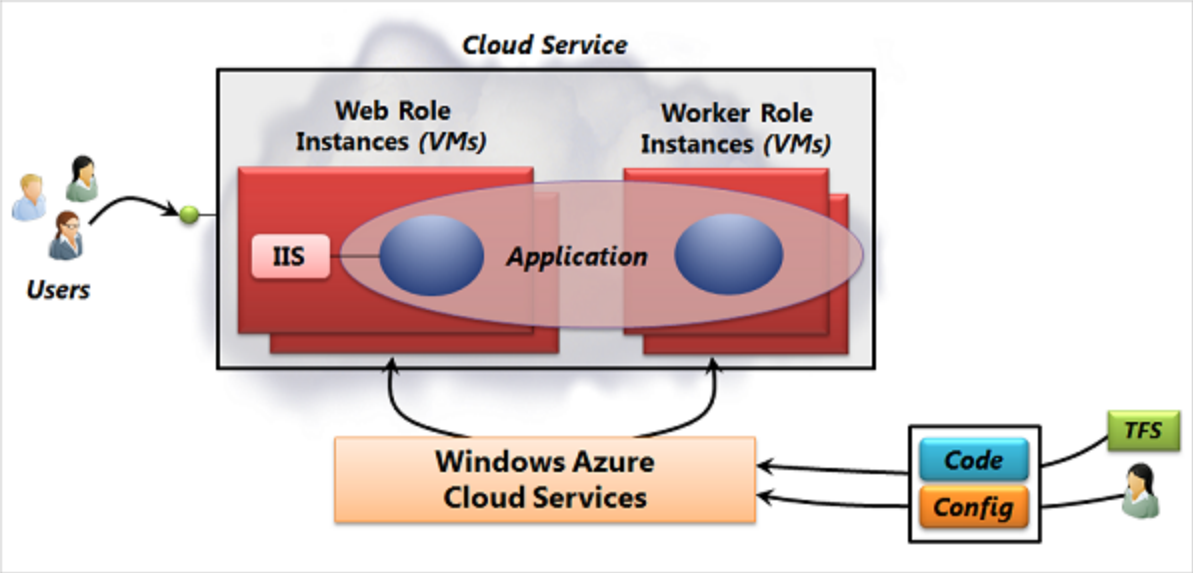
wKey='RK')?$select=<FirstProperty>

- vsechny zdroje maji vlastni identifikator URI a ke vsem jsou definovane CRUD operace
- bezestavovy protokol typu klient/server
- format vymeny dat - JSON, XML
  - pres HTTP metody get, put, post, delete

- IaaS
  - odolnost proti chybam, monitoring
    - viz virtualizace
  - replikace pres BLOBy
    - v BLOBech ulozeny i disky VM
    - zachovani persistence
  - automaticka aktualizace disku s operacnimi systemy
    - pouze u VHD, ktere poskytuje MS
    - u vlastnich obrazu si aktualizujeme sami
- grouping
  - pri vytvoreni nove VM mame moznost standalone behu nebo ho pripojit do skupiny ostatnich VMs v cloudu
  - kazdy standalone VM ma svoji vlastni verejnou IP narozdil od skupiny, ktera cela vystupuje pod jednou
  - load balancing - rozlozeni pozadavku
  - availability set - rozlozeni na ruzne uzly - ochrana proti havarii
  - komunikace v lokalni siti v ramci skupiny

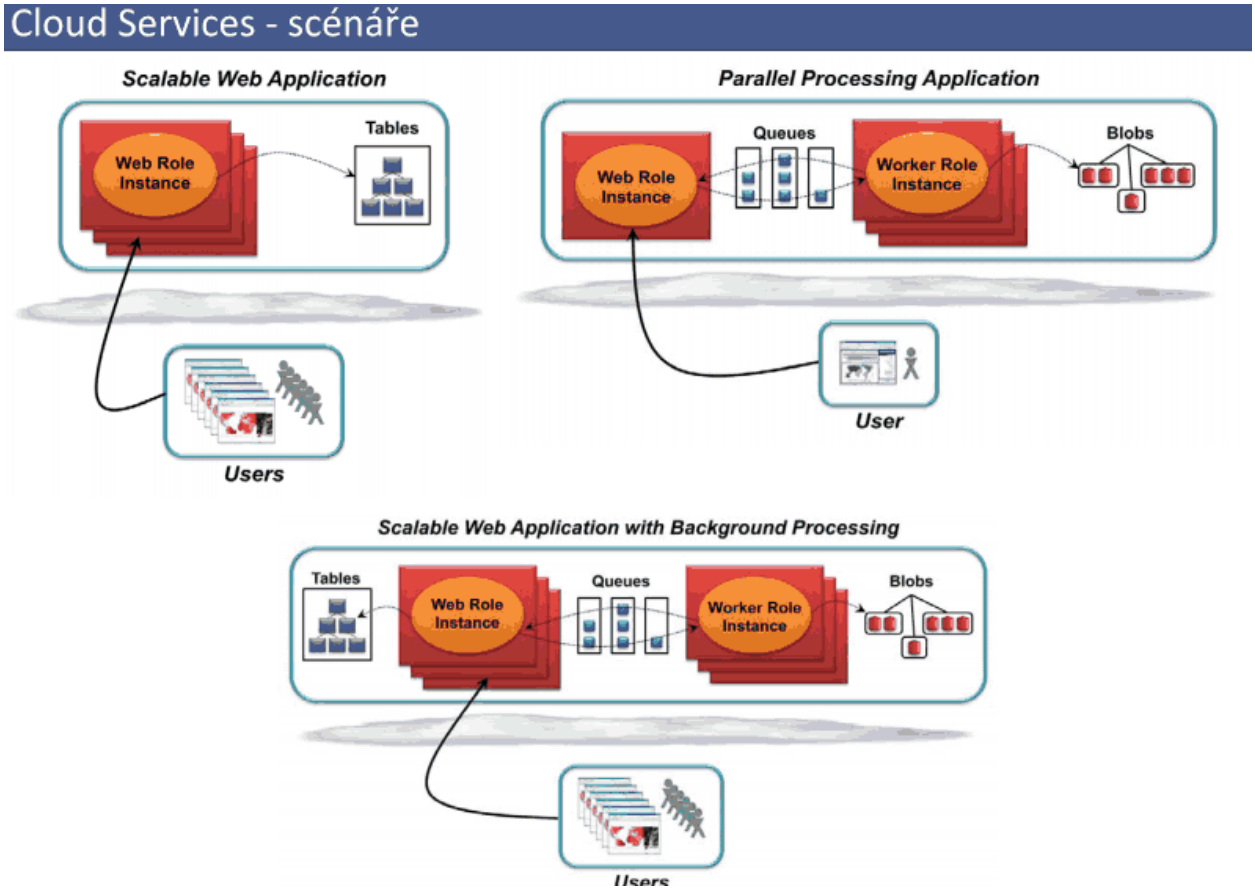
- webove stranky
  - je to jednodussi a levnejsi nez spravovat virtualni stroje
  - k dispozici je spousta frameworku, knihoven a aplikaci, na kterych se da stavet
    - JAVA, ASP.NET, MySQL, PHP, ...
    - Drupal, WordPress, Dropbox, Joomla, ...
  - nejcastejsi vyuziti cloudovych infrastruktur
    - idealni podminky pro nasazeni v cloudu
    - skalovatelnost
  - lze i pomoci VMs
    - avsak zbytecne slozite
    - nutna vlastni instalace, konfigurace, udrzba
  - PaaS
    - predkonfigurovatelne instalace
    - kompletni framework - OS, DB, web server, knihovny, …
    - administrace, aktualizace, udrzba vsech komponent
  - dynamicke pridavani instanci, load balancing
  - ruzne urovne izolace - shared / private VM
- cloudove sluzby
  - poskytovani skalovatelnych SaaS sluzeb

        - web sites - omezeny pristup, nelze nainstalovat cokoliv, nejsou dostupne vsechny sluzby
        - virtual machines - plny pristup, avsak nutnost administrace a udrzby, samo neskaluje
    - PaaS
        - skalovatelnost, dostupnost, spolehlivost, udrzba, ruzne jazyky a platformy
    - web roles / worker roles
        - technologie nabizi 2 ruzne pristupy VM
        - web roles - VMs instance, na kterych bezi varianta Windows Serveru s IIS
            - IIS - SW webovy server s velkou skalou rozsirujicich modulu, nejpouzivanejsi po Apachi
        - worker roles - VMs instance, kde bezi stejna varianta Windows jako u web roles, ale bez IIS
        - cloudove sluzby se pak sestavaji z nejake kombinace techto



    - vhodne pro vicevrstve skalovatelne aplikace
    - jedna adresa, vice web/worker roli
        - automaticky load balancing a availability set
    - vyvojove a produkcni prostredi
        - jednoduchy vyvoj/testovani a prechod na novou verzi
    - monitorovani (nejen) virtualniho HW
        - agent uvnitr web/worker role
- cloudove sluzby vs webove stranky
    - administrativni pristup do VM - instalace libovolneho potrebneho SW
    - web/worker role - vicevrstve aplikace - vlastni VM pro aplikacni logiku
    - moznost propojeni cloudove aplikace s privatnimi uzly

- remote desktop pro primy pristup k aplikacni VM

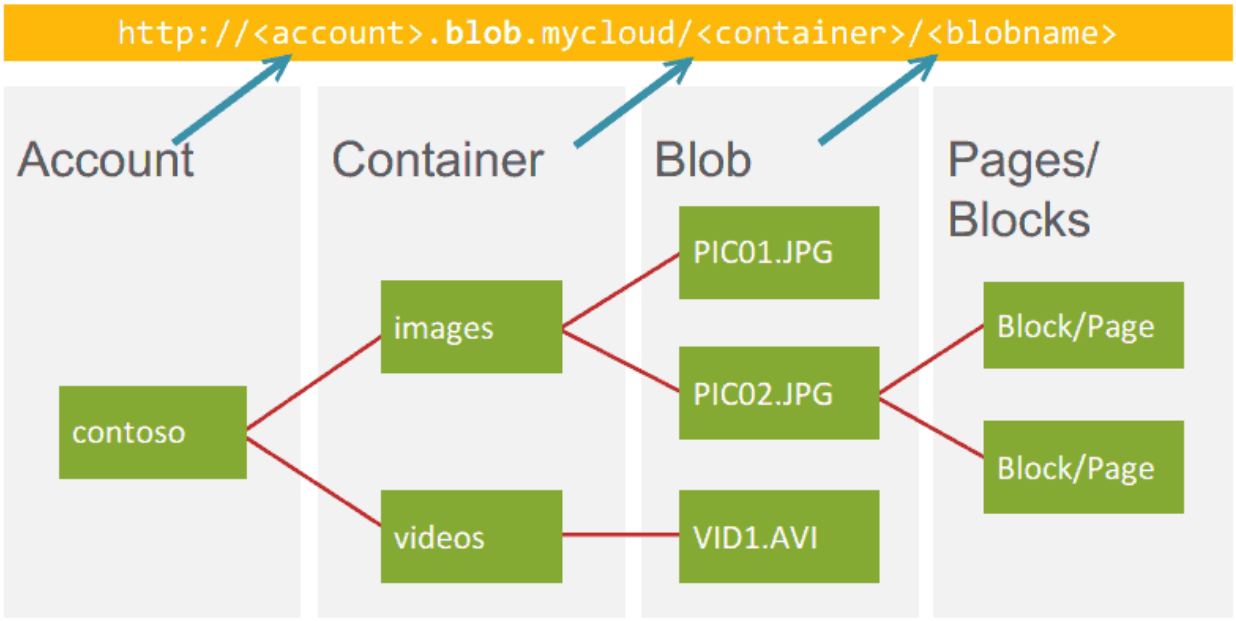


Windows Azure use cases

- high-performance computing
  - hlavni uzel prijima pracovni requesty a rozesila je worker uzlum
  - laboratorni simulace, vyzkum leku apod.
  - neocekavana / periodicka potreba vykonu
    - pravidelne chceme neco zaznamenavat
    - neocekavane checeme spocitat velky naklad dat apod
  - investice do provozu
  - paralelni problemy
  - cluster - privatni / cloud / kombinovany
  - pouziva se job scheduler (DRS), cluster manager (nejaky GUI pro konfiguraci sluzeb clusteru)
  - problemy
    - privatni a citliva data
    - zavislost na externim SW
    - prenos velkych dat
    - data management

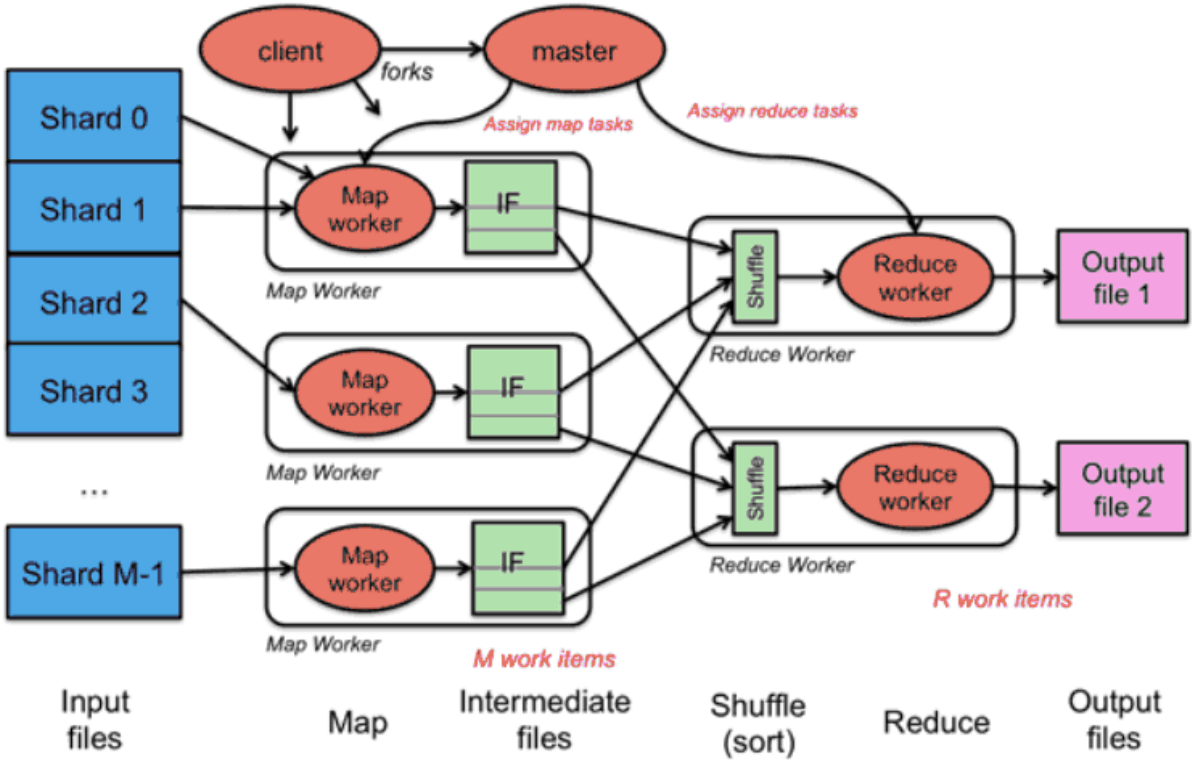
- SQL
  - ruzne implementace a knihovny - JDBC, ADO.NET, …
  - neni to jen DBMS v cloudu - je to PaaS
    - DB se stara o administrativni udrzbu - rizeni HW infrastruktury, automaticka aktualizace DB a OS SW, distribuce dat na vice serveru - replikace
  - pristupne pro cloud i externi aplikace
  - typicky distribuovane reseni - poskytuje availability a scalability
    - primarni / sekundarni replika
    - load balancing
    - cluster index
- NoSQL
  - kdyz nepotrebujeme komplexni SQL dotazovani
  - nejde o relacni databazi
  - tabulky klic-hodnota
    - velmi rychly skalovatelny pristup k velkym (rady TB) typovanym datum
    - identifikace - key, partition key/row key
    - nejdou slozite dotazy a trideni dle vice klicu
    - struktura
      - account, table, partition, entity, property (ta ma pak treba jmeno, typ, hodnotu)
    - table
      - mnozina radek
      - jednoznacne id - partition key/row key
      - partition key - rozliseni entit mezi ruzna replikovana uloziste
      - row key - jednoznacne id v ramci partition
      - trideni jen dle PK/RK
    - entity
      - mnozina properties (atributu)
      - vzdy PK/RK + timestamp
      - dalsi atributy aplikacne definovane
        - bez pevneho schematu
        - ruzne entity → ruzne atributy
    - partitions
      - operace na jedne partition jsou atomicke
      - rozlozeni dat mezi uzly rizeno aplikaci
      - volba PK/RK
        - souvisejici entity na jednom uzlu - shodny PK
        - rovnomerne rozlozeni entit mezi partitions - ruzne PK
        - efektivni dotazy - filtr pres PK
      - priklad: blogovaci server
        - temata, prispevky v tematech
        - parition key ~ tema

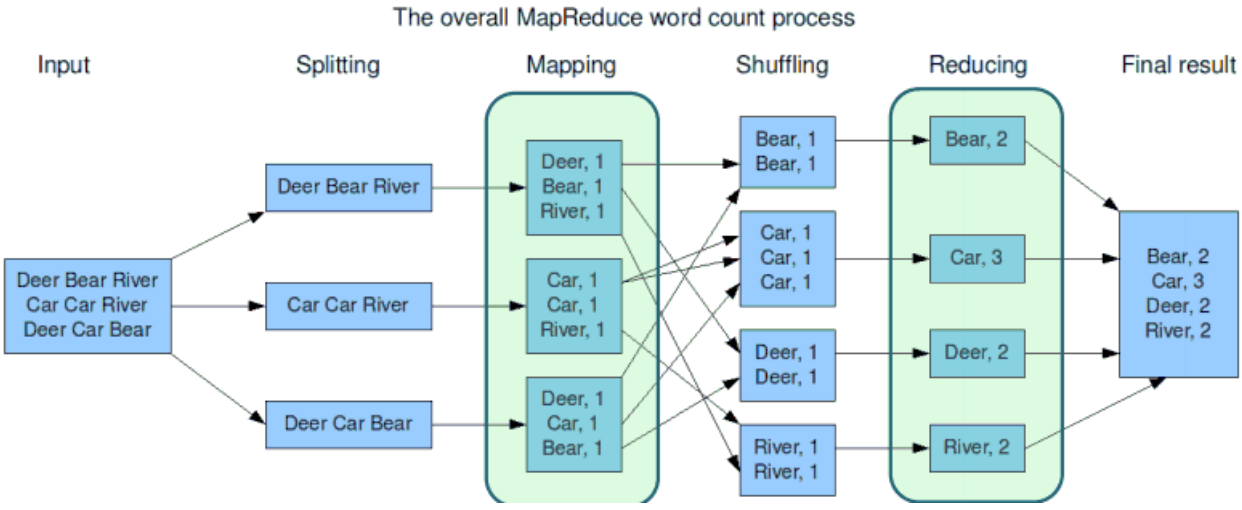
            - row key ~ datum, cas prispevku
            - dalsi atributy ~ text, hodnoceni
            - efektivni dotaz je napr. nalezeni nejaktualnejsich prispevku v tematu
        - priklad: verzovane uloziste dokumentu
            - PK: jmeno souboru
            - RK: verze souboru
            - dalsi atributy: datum, id, poznamka, …
            - vlastni data typicky v BLOB ulozisti
- column stores
    - sloupcove orientovana architektura
    - radky - velmi mnoho hodnot - radove miliony
    - adresace - column family / row / column → value
    - zadne schema - mozne ruzne pocty a typy hodnot
    - verzovani hodnot, timestamps, moznost pristupu ke starsim verzim
    - replikace, sharding (horizontalni partitioning na jednotlive shardy), rychly pristup, bez transakci, ruzne urovne konzistence dat
    - vyuziti v datovych skladech, CRM (customer relationship management)
    - Cassandra (Apache open source distributed DBMS), C-store
    - jak se udela sloupcova architektura
        - rozpad bezne radkove tabulky na sloupce
        - pak mame pro kazdy sloupec tabulku ID,value
        - udelame jeste RLE kompresi, aby se ve sloupci neopakovaly hodnoty - staci nam vedet, pro jaky rozsah ID jsou stejne
        - pak se to serializuje do souboru
- BLOBs
    - nestrukturovana, velka (rady TB) binarni data
    - sdruzena v replikovanych kontejnerech - jmeno kontejneru jako PK
    - integrace s file systemem, persistence
    - ukladani videi, virtualnich disku s VM, backup → je to levne
    - typy
        - block blob
            - sekvence blobu - az miliony
            - optimalizovane pro sekvencni pristup - streamy
            - rychly zapis (append) a sekvencni cteni
        - page blob
            - strankove orientovane
            - nahodny read/write pristup
            - vyssi rezie
    - pouziti s RESTful API - putblob, getblob, deleteblob, copyblob, …



- business analytics - SQL reporting
  - reporty, analyza dat spolecnosti
    - vydaje, prijmy v kvartale, vytizeni pracovniku, …
  - data mining nad SQL / NoSQL
    - pouziti rozhodovacich stromu, lesu
    - regresivni analyza (statisticke metody, kdy odhadujeme hodnoty nahodne veliciny na zaklade znalosti jinych velicin)
    - clustering
  - ruzne typy vystupu - grafy, sestavy, agregace, XML, PDF, …
- Hadoop - siroky system propojenych produktu (“ZOO”)
  - BigData, nezpracovatelna rozumne jednim DBMS
    - nejsou treba ani relacni data - logy, raw data apod.
  - paradigma Map / Reduce - paralelizace a distribuovanost vypoctu
  - dalsi soucasti a souvisejici moduly
    - HDFS - hadoop distributed file system
    - Pig, PigLatin - paralelizacni platforma, high-level jazyk
    - Hive, HiveQL - data warehouse & querying, analyza dat
    - ……
  - map/reduce
    - paradigma pro paralelizaci a distribuovanost
    - map: input → (key,value)
    - reduce: (key, list of values[]) → output
      - sjednoceni hodnot pro kazdy klic
      - zpracovani, vysledek
    - mezivysledky mohou byt duplicitni
      - zpracovani velmi velkych objemu dat

    - odolnost proti chybam
- aplikace - frekvence slov
    - spocitat cetnost slov v mnoha dokumentech
    - map(file,text) → (word,count)
    - reduce(word,list of counts) → count





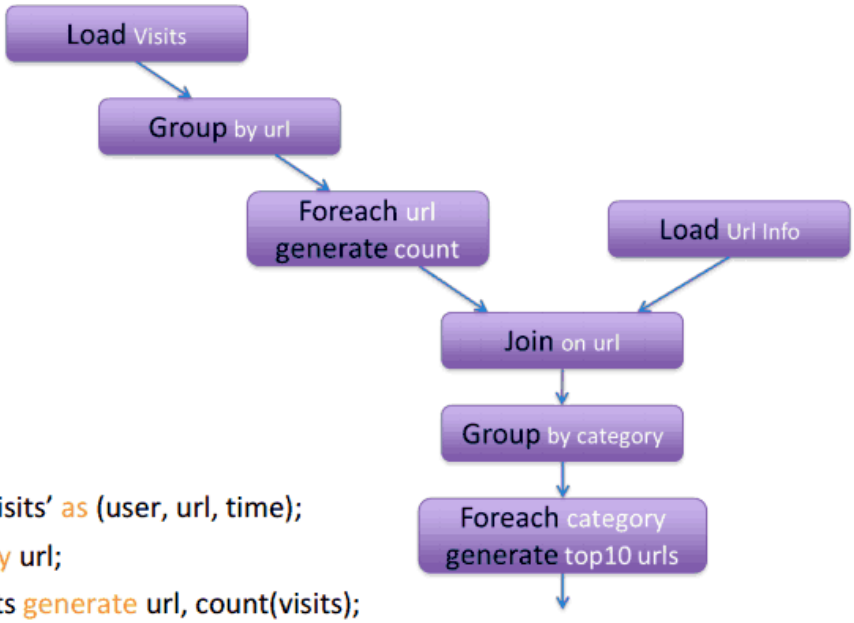
- master/slave architektura
    - jobtracker - single master server

            - rozhrani pro klienty
            - fronta jobu, zpracovani FIFO, prirazeni jobu tasktrackerum
        - tasktrackers - slave servers, jeden na kazdy uzel v clusteru
            - vykonavani tasku dle jobtrackeru
            - presuny dat mezi fazemi map, reduce
    - vyhody - velka nestrukturovana data, davkovy/offline rezim
    - nevyhody - nevhodne pro mala data, komunikace a synch. mezi uzly, vyssi latence, transakcnost
- co zahrnuje ZOO
    - HDFS - optimalizovan pro Hadoop
    - HBase - NoSQL DB, distributed, column-oriented, cilem je ukladat biliony radku, miliony sloupcu
    - Pig, PigLatin - vysokourovnovy jazyk pro analyzu dat, preklad na map/reduce joby
    - HIVE, HIVEQL - data warehouse infrastruktura, SQL-like jazyk pro dotazy nad HDFS
    - Mahout - framework pro strojove uceni a data mining
    - Hama - distribuovany (grid) vypocetni framework
    - zookeeper - high-performance koordinator distribuovanych aplikaci
        - naming, konsensus, clenstvi ve skupinach, vyber koordinatora, fronty, event notification, workflow & cluster management, …
- HDFS
    - radove 10 000 uzlu, milion souboru, PB dat
    - ulozeni strukturovanych i nestrukturovanych dat
    - vlastnosti
        - odolnost proti chybam HW
            - replikace, detekce chyb, zotaveni
            - optimalizace pro velke mnozstvi levneho HW
        - streamovani souboru
            - optimalizace propustnosti, ne pristupove doby
        - orientace na velke soubory
            - velka data ulozena v mensim mnozstvi velkych souboru
        - write-once, read-many
            - jednou zapsany soubor jiz neni mozne menit ani appendovat k nemu data
            - podpora pro append uz sice existuje, ale jen ve spec. pripadech
        - heslo: moving computation is cheaper than moving data
    - architektura
        - NameNode

            - master, spravuje metadata
            - obsahuje seznamy souboru, adresaru, jejich mapovani na bloky a umisteni bloku
        - DataNode
            - slave, uloziste bloku, operace zapisu a cteni
            - nevi nic o souborech
            - NameNode mu muze naridit blok zreplikovat nebo smazat
        - operace se souborovym systemem
            - cteni: NameNode zjisti umisteni bloku souboru, klient komunikuje primo s DataNode
            - zapis: NameNode zalozi soubor a rozhodne, jaky DataNode pouzit
        - replikace
            - v atributu souboru se uchovava info o # pozadovanych replik
            - na jednom stroji vzdy jedna replika souboru
            - odolnost proti vypadku disku i celeho stroje
            - zapis pomoci odlozene replikace
        - implementace ruznych FS na zaklade rozhrani HDFS/HCFS(hadoop cluster FS)
            - Azure BLOB storage, CassandraFS, Symantec cluster FS, …
- Pig + PigLatin
    - nevyhody Map/Reduce: prilis low-level, nepodporuje slozitejsi data-flow
    - Pig je behove prostredi (platforma) pro webově skalovane zpracovavani dat
    - provadi transformace (kompilace) na soustavu MapReduce programů/jobů
    - BigData - davkove zpracovani
    - pouziva se vysokourovnovy jazyk PigLatin pro datovou analyzu
        - paralelni zpracovavani dat
        - operace pro manipulaci s relacnimi daty
        - imperativni styl programovani (vzor Java)
    - pigs eat anything
        - umi operovat nad daty obsahujici/neobsahujici metadata, nad relacnimi/nested/nestrukturovanymi daty
    - pigs live anywhere
        - jazyk pro paralelni zpracovani dat, neni spjaty s zadnym konkretnim frameworkem
    - pigs are domestic animals
        - designovane pro jednoduche rizeni a modifikaci uzivateli

        - podpora integrace uzivatelskych kodu
            - snadna uzivatelska rozsireni, ktere mohou uzivatele psat v Jave, Pythonu, Javascriptu, Ruby, Groovy a pak volat primo z PigLatinu
- pigs fly
    - rychle zpracovani dat
    - trvale zlepsovani vykonnosti

Pig Data Flow & Pig Latin




```
visits      = load '/data/visits' as (user, url, time);
gVisits     = group visits by url;
visitCounts = foreach gVisits generate url, count(visits);
urlInfo     = load '/data/urlInfo' as (url, category, pRank);
visitCounts = join visitCounts by url, urlInfo by url;
gCategories = group visitCounts by category;
topUrls     = foreach gCategories generate top(visitCounts,10);
              store topUrls into '/data/topUrls';
```

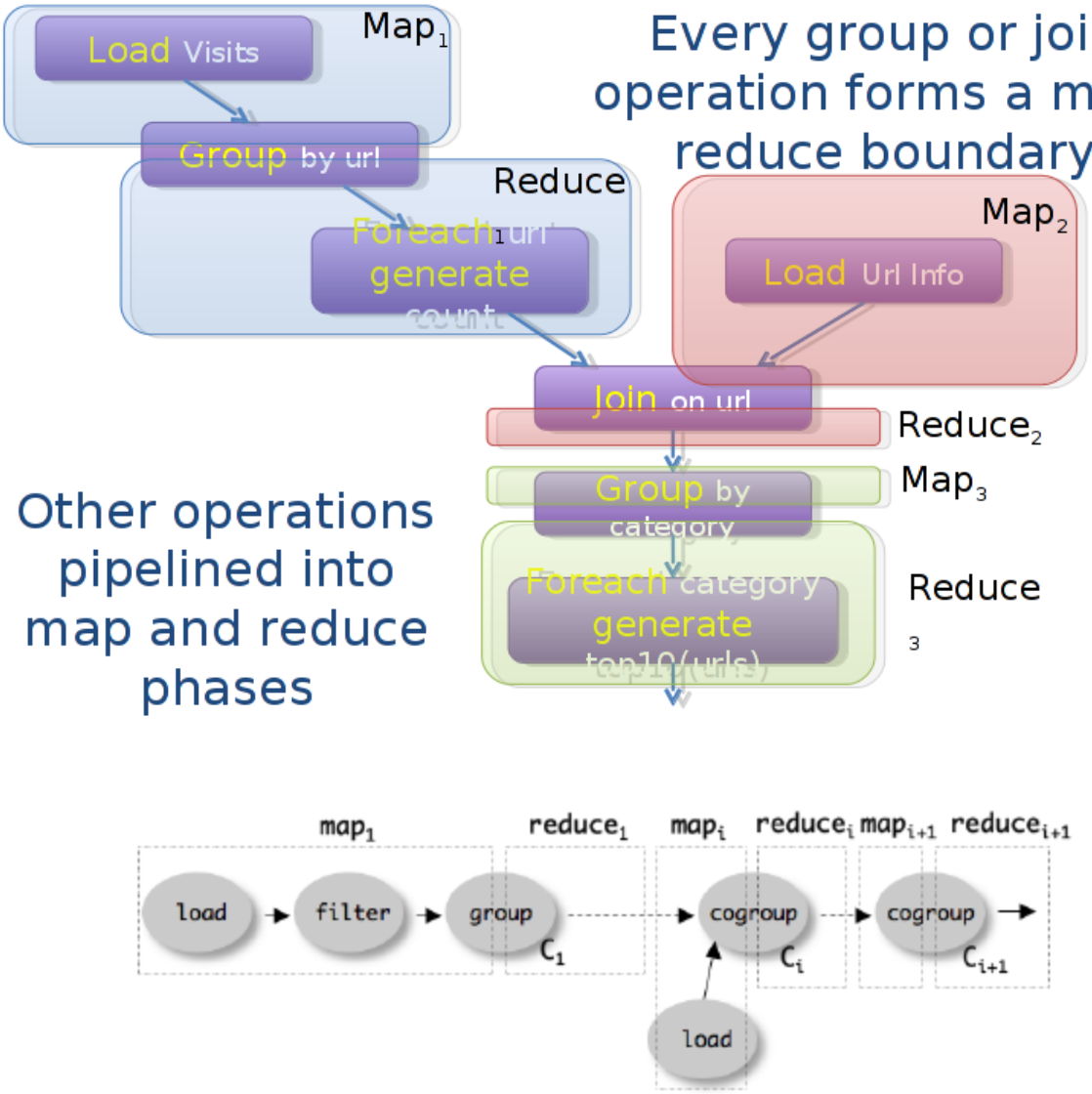
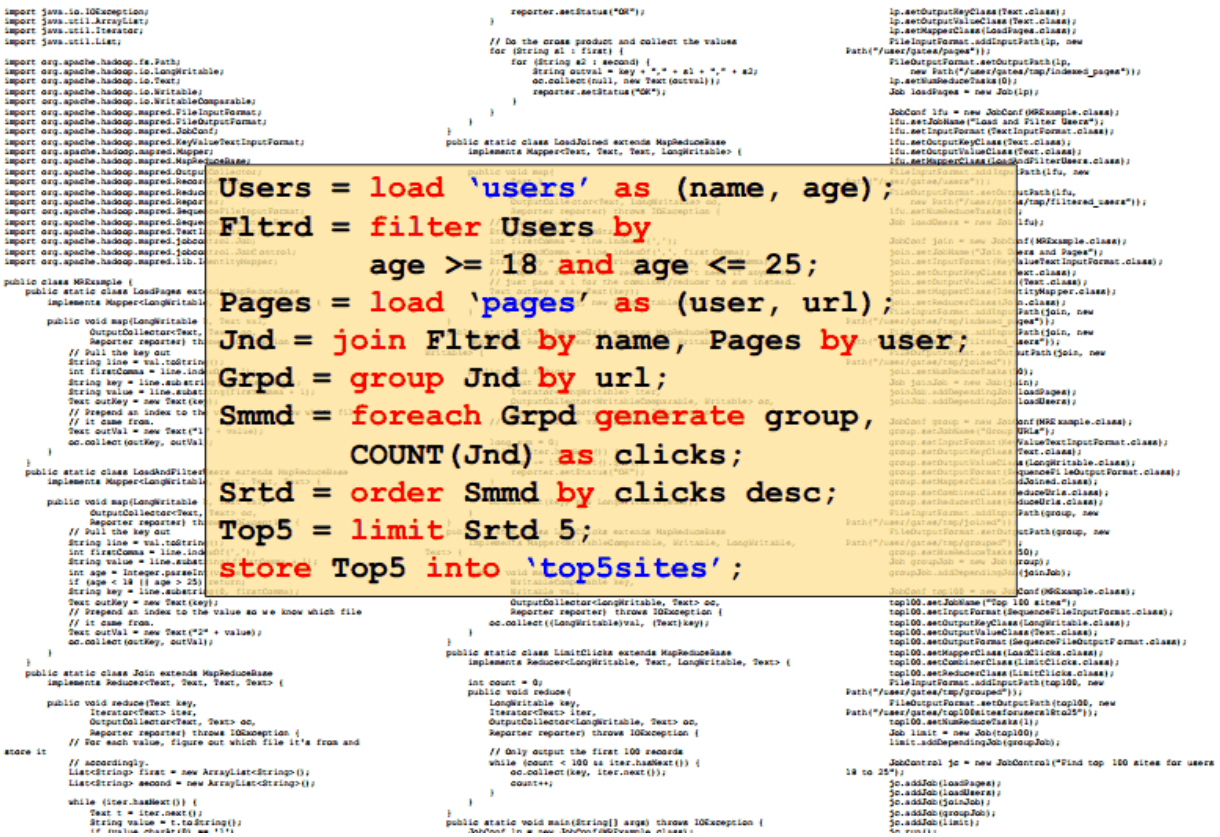


Figure 3: Map-reduce compilation of Pig Latin.

Pig Latin operations

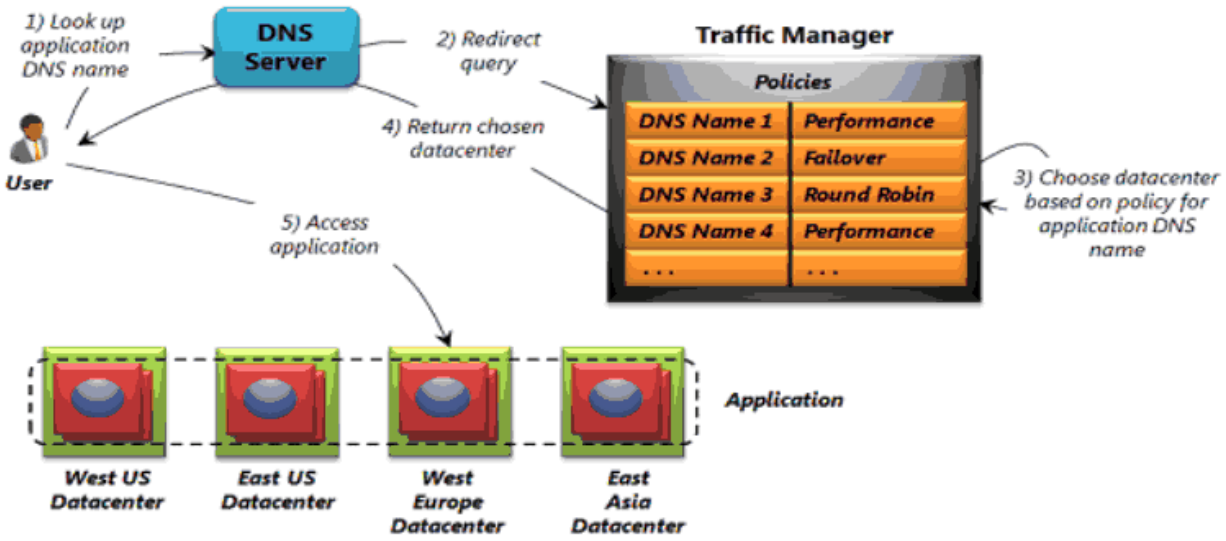
load	Read data from file system
store	Write data to file system
foreach	Apply expression to each record and output one or more records
filter	Apply predicate and remove records that do not return true
group/cogroup	Collect records with the same key from one or more inputs
join	Join two or more inputs based on a key; various join algorithms available
order	Sort records based on a key
distinct	Remove duplicate records
union	Merge two data sets
split	Split data into 2 or more sets, based on filter conditions
stream	Send all records through a user provided executable
sample	Read a random sample of the data
limit	Limit the number of records

Pig Latin vs. Map Reduce



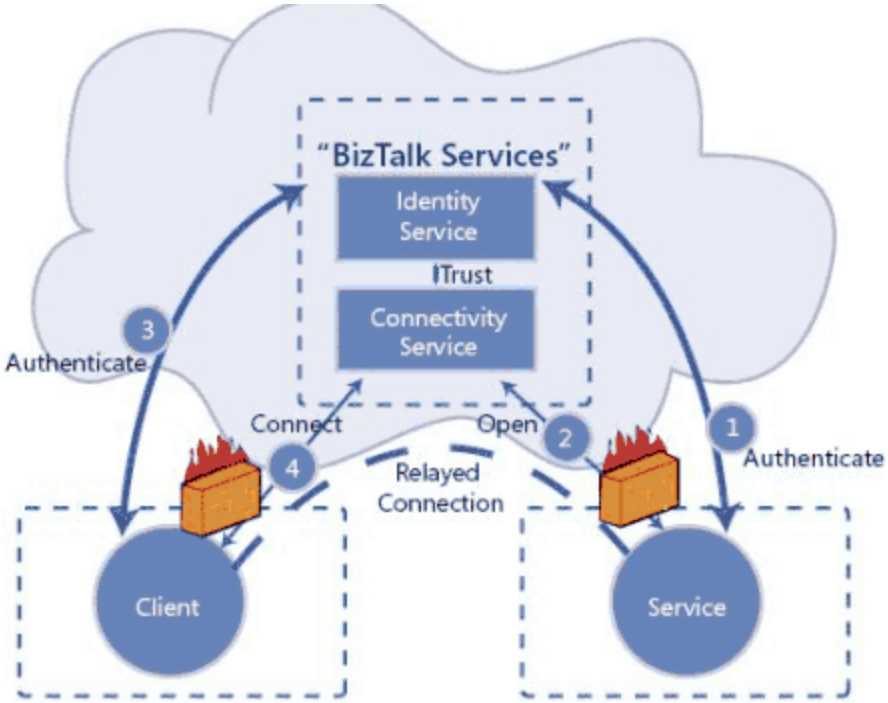
```
Users = load 'users' as (name, age);
Fltrd = filter Users by
        age >= 18 and age <= 25;
Pages = load 'pages' as (user, url);
Jnd = join Fltrd by name, Pages by user;
Grpd = group Jnd by url;
Smmd = foreach Grpd generate group,
       COUNT(Jnd) as clicks;
Srtd = order Smmd by clicks desc;
Top5 = limit Srtd 5;
store Top5 into 'top5sites';
```

        - Hive + HiveQL
            - data warehousing infrastructure
                - nastroje pro umozneni jednoduche extrakce/transformace/nacteni dat
                - mechanismus pro ulozeni struktury na ruznou skalu datovych formatu
                - umoznuje pristup k souborum ulozenych na HDFS nebo v systemech datovych ulozist jako je HBase
                - spousteni dotazu pres MapReduce
            - HiveQL je SQL-like jazyk slouzici pro dotazy nad HDFS
                - rozsiritelny o vlastni MapReduce mapovace a redukovace
- cloud networking
    - Azure bezi dnes na spouste datovych centrech v USA, Evrope a Asii
    - kdyz chceme spustit aplikaci nebo ulozit data muzeme si vybrat z techto datovych center, ktere chceme pouzit
    - existuje take nekolik zpusobu, jak se k temto centrum pripojit
        - virtualni sit - propojeni vlastni lokalni site s definovanou mnozinou Azure VMs (s nejakym cloudem)
            - rozsireni vlastni site
            - pristup k sluzbam lokalni site z cloudovych modulu
            - a naopak pristup k sluzbam cloudu z lok. site
            - take propojeni nekolika data center - geograficka skalovatelnost
        - connect, prime pripojeni - propojeni cloudove site primo s lokalnim PC
            - nutnost VPN gateway, admin site
            - klientsky SW
                - pristup ke cloudovym sluzbam z lokalnich PC
                - bez nutnosti sitovani
            - cloudove sluzby se jevi jako v lokalni siti
                - moznost individualniho nastaveni apod.
        - traffic manager
            - pokud aplikace bezi na vice datacentrech, muzeme pouzit traffic manager, ktery pouziva inteligentni smerovani uzivatelskych pozadavku mezi instance aplikace
            - load balancing pozadavku mezi data centra
            - definovana pravidla smerovani - od vyvojaru aplikace:
                - typ performance - proste nejblizsi DC
                - typ failover - prioritni
                - typ round robin - kruhova fronta
            - vhodne pro velke aplikace, skalovatelnost



- cloud messaging
    - casto je potreba interakce aplikaci
    - Azure nabizi nekolik moznosti jak komunikaci resit
    - service bus
        - zalozeno na mechanismu publish-subscribe
            - odesilatele (publishers) posilaji zpravy pro celou tridu, bez toho, aniz by vedeli, kdo do ni patri
            - prijemci (subscribers) se zapisou do trid, o ktere maji zajem a prijimaji tyto zpravy, aniz by vedeli, kdo je odesilatelem
        - volně spřažená komunikace
        - skupinová komunikace 1:N, vice prijemcu
        - API mimo cloud - letecka spolecnost, zpristupneni rezervace letenek a oznamovani zmen
        - mechanismy queues/topics/relays
        - queues
            - spolehlivy asynchronni jednosmerny kanal mezi rolemi
                - web role prijme soubor s videem
                - ulozi jej do BLOBu, posle zpravu do message queue
                - worker role zpravu vyzvedne a provede konverzi formatu
            - strukturovana/nestrukturovana cast zpravy
                - binarni data v body, key/values properties
            - skalovatelnost - nezavisly pocet read/write roli
                - odesilatelu i prijemcu muze byt vice
                - automaticky load balancing
            - vyzvednuti zpravy
                - receive/delete rezim
                    - service bus prijme pozadavek, oznaci jej ze se zpracovava a smaze zpravu z fronty
                    - pokud aplikace, ktera ma prijmout zpravu spadne, je

zprava ztracena
        - idempotentni sluzby
        - jednodussi, mozna ztrata pri vypadku
    - peek/lock rezim
        - dvoufazovy protokol pro dorucovani zprav
        - kdyz zprava dorazi do fronty, je uvedena do zamceneho stavu a setrvava v nem, dokud se stav nezmeni nebo dojde k timeoutu, kdy dojde zase ke zviditelneni zpravy ve fronte
        - kdyz je zprava v zamcenem stavu, muze aplikace vykonat pouze jedinou z techto operaci:
            - complete - zprava uspesne zpracovana
            - abandon - dorucovani selhalo (ale ne na zaklade obsahu zpravy), zprava je odemknuta a muze se opet dorucovat
            - defer - odlozene zpracovani
            - deadletter - odlozene zpracovani, kdyz dorucovani selhalo na zaklade obsahu zpravy
        - odolnost proti chybam
- topics
    - sekvence zprav tykajici se nejakeho tematu
    - predplatne (subscribing) je definovano jmenem a filtrem
        - filtr na zaklade vlastnosti zpravy (WHERE klauzule)
    - kazde predplatne ma svoji virtualni kopii zpravy
        - jedna zprava muze byt zpracovana vice prijemci
    - receive/delete, peek/lock
- relays
    - obousmerne propojeni na rozdil od queues/topics
    - motivace
        - obe aplikace za firewallem - blokovane prichozi porty
        - NAT - promenliva IP adresa
    - reseni
        - komunikace pres service bus relay service
        - obe aplikace se prihlasi
        - registr relay IDs



- caching
  - aplikace casto pristupuji na ta sama data
  - chceme uchovat kopii takovych dat co nejbliz k aplikaci, ktera s nimi pracuje
  - Azure in-memory caching - data se zachovaji primo v operacni pameti VM dane aplikace
    - key/value store, kde value je nejaky chunk dat, mozno i strukturovane
  - ruzne zpusoby - intra-app, inter-app, inter-cloud (v ramci app, mezi app, mezi cloudy)
  - CDN (content delivery network)
    - mnoho uzlu rozmistenych po cele zemi, galaxii
      - serve content to end-users with high performance & high availability
    - prvni pristup - kopie dat do uzlu CDN
    - dalsi pristup - jen k uzlu CDN
- multimedia
  - video sluzby, media sluzby
  - uploading BLOBu, encoding, konverze formatu, packaging, streaming, protection (komponenta pro digital rights management), advertising
- GIS (geographic information systems)
  - mapy - free/private/special
  - vazby na relacni data
  - API, web
- marketplace
  - aplikace, sluzby

    - prodej MP3, e-knihy
    - vydavatele - pricing, updates, ranking, evaluations, …
    - customers - identity, subscriptions, push, updates, reviews, rankings, …
    - publishing/subscription API
- mobilni sluzby
    - mobilni klienti
    - autentizace
    - pristup k DB
    - SMS, email, MMS
    - MightyText
    - dalsi sluzby: hry, prekladace, google docs, google drive

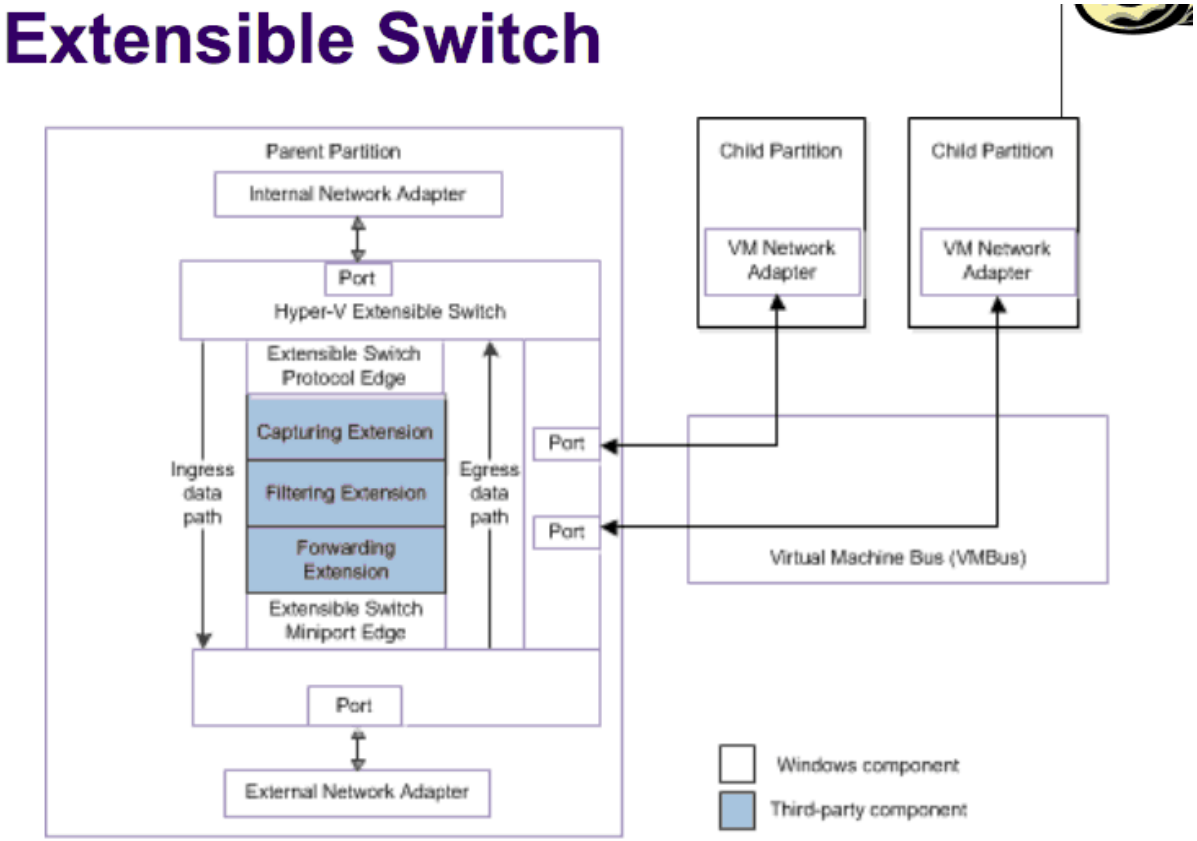
Security

Bezpečnostní rizika virtualizace

- blue pill attack
    - chyti do pasti instanci beziciho OS tim, ze nastartuje hypervizora a virtualizuje stroj, na kterem bezi
        - potrebuje vsak opravneni ke spousteni privilegovanych instrukci
    - predchozi OS bude sice stale udrzovat sve existujici reference na otevrene soubory a zarizeni, ale skoro cokoliv jineho se muze zachytit - skoro libovolna zprava - a posilat falesne odpovedi (HW interrupty, pozdavky na data, systemovy cas apod.)
    - predvedla Rutkowska 2006, oznaceno za 100% nedetekovatelne
    - zverejnila i kod, ktery zjisti, ze system bezi virtualizovany (jsme v Matrixu) - red pill
        - idea je, ze na virtualizovanem systemu, kde takto bezi hypervizor a puvodni OS se musi sdilet prostredky, VMM musi relokovat strukturu s vektorem preruseni, aby nekolidovala s hostovou strukturou
        - na x86 existuje instrukce v neprivilegovanem modu, ktera uklada do operandu prave adresu tabulky
        - muzeme otestovat zda je tabulka na svem miste nebo ne
    - dalsi kod detekujici beh ve virtualizovanem prostredi: timing attack
        - trap and emulate trva mnohem dele nez provadeni nativni instrukce
        - je potreba pouzivat externich zdroju casu (NTP), protoze lokalni cas muze byt podvrzeny
    - programy ale detekuji pouze pritomnost virtualizace, ne blue pill malware
        - spousta programu pouziva virtualizaci - velky pocet falesnych odhalenich
        - navic jsou detekce ne uplne presne
- zranitelnost VMM
    - problem je, ze utokem na VMM muze utocnik napadnout vice serveru naraz
- fyzicke, virtualni firewally
    - pri pouziti virtualizace servery z ruznych trust zone vetsinou sdili stejne fyzicke

prostredky (pamet, sitove karty, apod)

- webove rozhrani pro pristup k DC
    - spousta datacenter mohou byt administrovany pres webove rozhrani - pres nejakou centralni konzoli, ta pak musi byt dobre zabezpecena
- dalsi nebezpeci
    - spolehani na tradicni bezpecnostni bariery
        - fyzicke firewally, apod
        - nefunguji efektivne pri dynamicke povaze virtualnich instanci
    - aktualne rychle se rozsirujici poskytovani techto sluzeb klientum
    - bezpecnost prenechana netradicnim zamestnancum
    - …
- reseni
    - virtualni firewally
        - bezi plne ve virtualizovanem prostredi
        - nabizi bezne sluzby jako fyzicky firewall
        - dodavan napr jako tradicni SW na guest VM, jako ucelove navrzeny system pro virtualni sit, virtualni switch s dalsimi bezpecnostnimi schopnostmi nebo rizeny kernelovy proces bezici v ramci hypervizoru hosta
        - podpora migrace za behu
        - pouziti u stretched clusters
            - 2 a vice fyzickych hostu nainstalovanych na separatnich mistech mene nez 100 kilometru od sebe, ktere jsou soucasti stejneho vSphere clusteru
    - agentless antivirus
        - standardni agent-based bezpecnost je pro virtualni prostredi a jeho dynamickou povahu slaba
        - takto se daji ochranit vsechny VM na hostovi bez nutnosti instalace agenta na kazde VM
        - je to nejaky virtualni host, ktery se stara o detekci
        - s agenty ma detekce velky dopad na vykon systemu - AV storm, tady se temto bourim predchazi
    - extensible switches
        - virtualni switche umoznuji komunikaci mezi VM, inteligentni smerovani paketu a jejich inspekci
        - extensible switche umoznuji pridavani pluginu - doplnujici funkcionality od vyvojaru
            - prizpusobeni switche pro dane prostredi, ve kterem pracuje



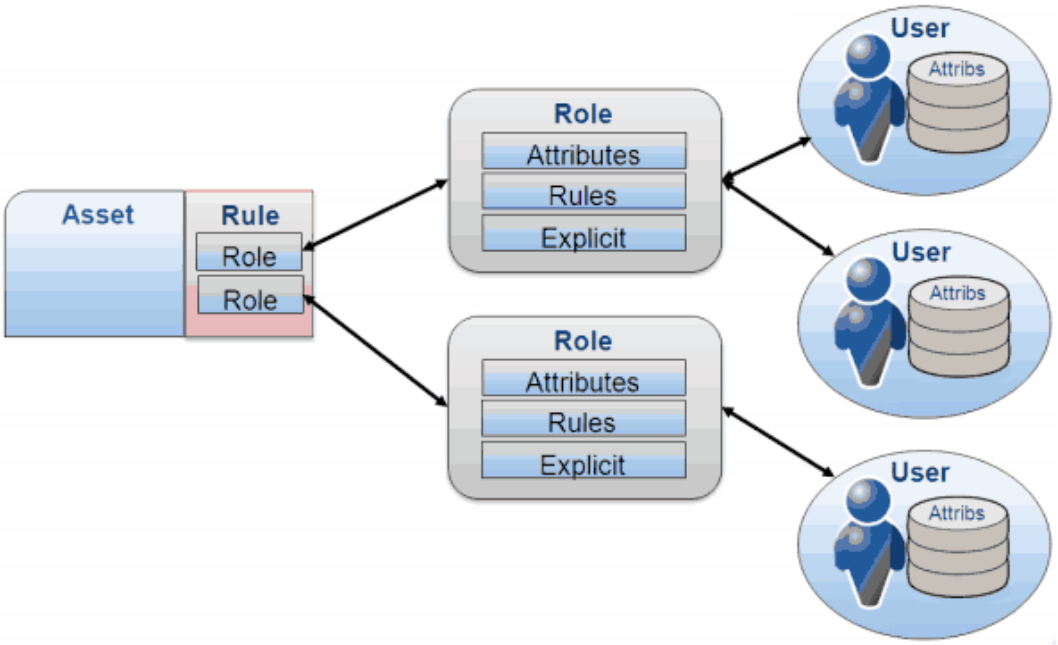
Bezpečnostní rizika cloud computingu

- cizi pristup k nasim datum
- dostatecne fyzicke zabezpeceni
- nepouzivane, poskozene pevne disky sesrotovat v drtickach
- nejista lokace ulozenych dat
- dodrzovani pravnich predpisu
- nedostatek investigativnich mechanismu
- zotaveni po havarii
- dlouhodoba zivotaschopnost

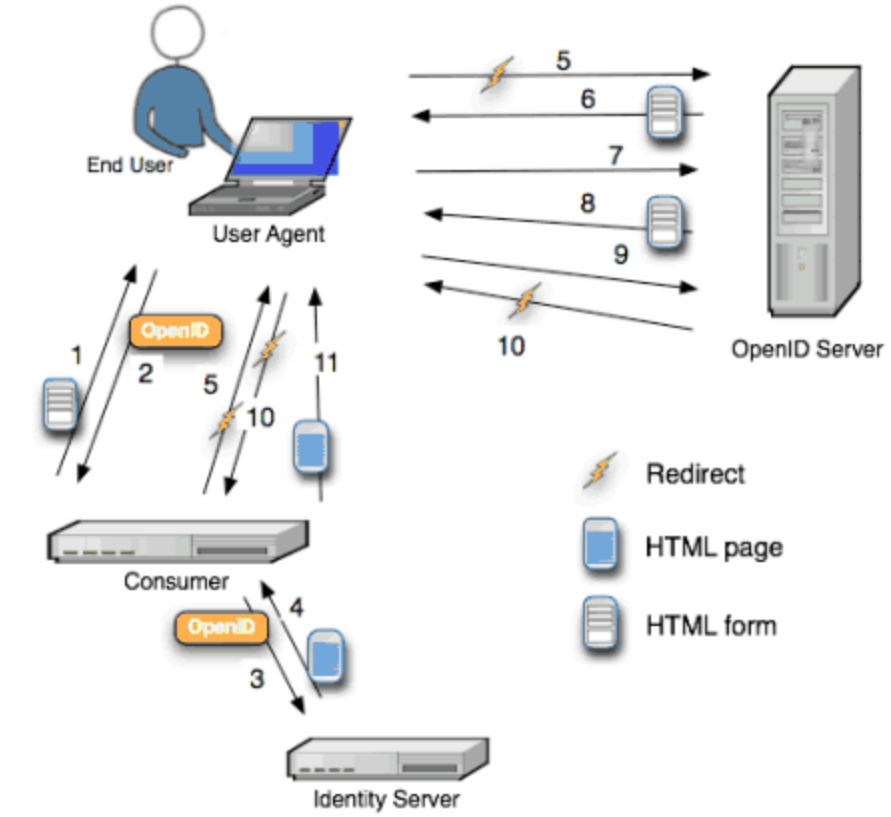
Identity management

- jde nam o autentizaci a autorizaci
    - prokazani, zda jde o spravneho uzivatele
    - prokazani, zda ma dany uzivatel pravomoce pro pristup k datum
- externi uzivatelske databaze
    - active directory / LDAP protokol pouzity jako centralni autentizacni mechanismus
        - zaznamy ve stromove strukture, informace o uzivatelich, do jakych skupin patri, jake ma privilegia
- dvoufazova autentizace
    - dve ruzne overovaci metody pro zajisteni dostatecne bezpecnosti

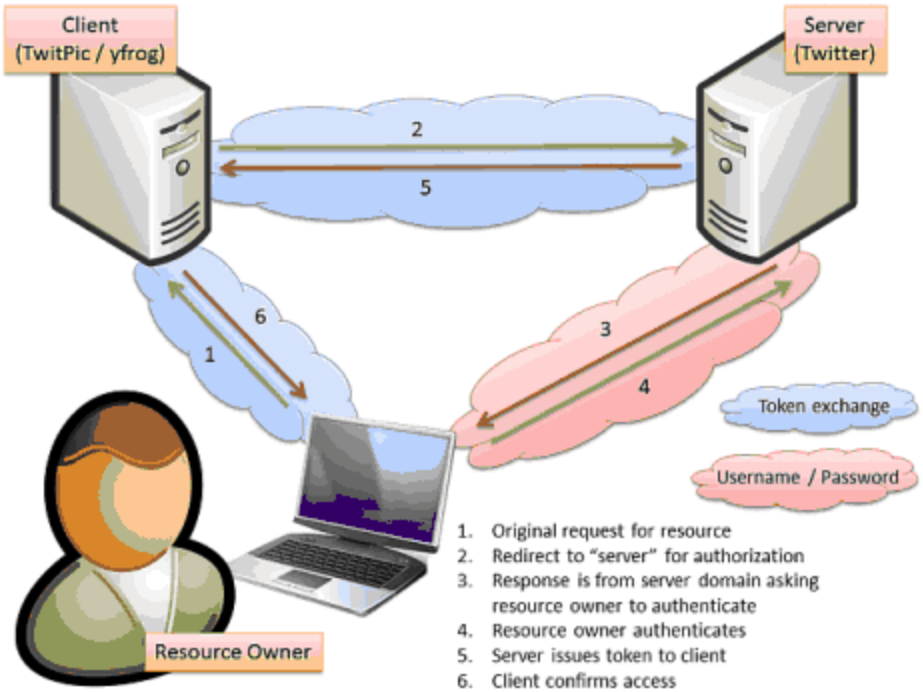
    - pristupova karta + USB flash
    - PIN + overovaci kod z mobilu
    - …
    - dnes jiz bezne - google, facebook, dropbox, paypal, …
- role-based access control
    - motivaci je specifikovat a vynutit bezpecnostni politiku specifickou pro cely system VMs, ktera prirozene reflektuje strukturu uzivatelu systemu
    - pro ruzne funkce v ramci firmy jsou vytvoreny ruzne role
    - povoleni na provedeni specificke akce se pak ridi podle role, kterou uzivatel ma/vykonava
        - prirazeni roli k uzivatelskym uctum



- OpenID
    - otevreny standard popisujici decentralizovany zpusob autentizace uzivatelu, ktery odstranuje potrebu na strane provozovatele sluzby poskytovat a vyvijet vlastni systemy pro autentizaci a ktery rovnez samotnym uzivatelum sluzby umoznuje konsolidaci jejich digitalnich identit
    - má tvar unikátního URL, ke kterému je přiřazeno heslo
    - služba, která uživatelům autentizaci pomocí OpenID nabízí, při přihlašování uživatele přesměruje požadavek na ověření identity na správce daného OpenID účtu a ten vrátí informaci o povolení či zamítnutí žádosti o autentizaci
    - pouziva Google, IBM, seznam.cz



- OAuth
    - pouziva se delegovani uzivatelske autorizace treti strane



- SAML
  - podobne jako OpenID, ale cilene na podniky
  - security assertion markup language
  - XML-based
  - podpora single sign-on (SSO)
    - uzivatel se autentizuje na serveru A, pak se chce prihlasit na server B, B zjisti se vstupuje ze serveru A, zepta se serveru, jestli uz se neautentizoval u nej a muze rovnou povolit vstup
  - nutnost vzajemne duvery mezi identity providerem a service providerem
  - podpora mapovani protokolu na jine nez HTTP
  - MS Active Directory Federation services
    - komponenta na Windows serveru umoznujici SSO → SAML-based
  - CAS (central authentication service)
    - SSO protokol
    - postaveny na open-source projektu Shibboleth
- Eduroam
  - pouziva na autentizaci hierarchickou strukturu RADIUS serveru