




# Virtuelizacija

Računarstvo u oblaku  
(Cloud Computing)



# Sadržaj

- Motivacija
  - Šta je virtuelizacija
  - Značaj virtuelizacije za CC
  - Kako radi
  - Slojevi sistema i virtuelizacija
    - Interfejsi i prenosivost koda
  - VMM / Hypervisor
    - Virtuelizacija procesora
    - Type 1 i Type 2 Hypervisor
  - Performanse i izolacija
  - Uslovi za uspešnu virtuelizaciju
    - Dual mode operations
    - Izazovi virtuelizacije x86 arhitekture procesora
  - Tehnike virtuelizacije
- 

# Motivacija

- Funkcionisanje računarskog sistema se može opisati korišćenjem tri fundamentalne apstrakcije:
  - (1) procesori/interpreteri koda, (2) memorija, (3) komunikacione veze
- Sa rastom samog sistema i broja korisnika upravljanje resursima postaje vrlo izazovno
- Ključni problemi pri upravljanju resursima:
  - Planiranje po vršnom opterećenju (**overprovisioning**)
  - **Heterogenost** hardvera i softvera
  - Otkazi računarskih resursa
- ***Virtualizacija je tehnologija koja se smatra ključnom za omogućavanje Cloud Computing-a.*** Značajno pojednostavljuje upravljanje fizičkim resursima. Primeri:
  - VMM (virtual machine manager) na kome se izvršava određena virtuelna mašina (VM) može snimiti stanje virtuelne mašine (VM) i migrirati je na drugi fizički server kako izbalansirao opterećenje.
  - Korisnici mogu uvek da svoje aktivnosti izvršavaju na poznatim okruženjima umesto da uvek iznova podešavaju konfiguracije za svako pojedinačno okruženje.

# Šta je virtuelizacija?

- Kreiranje virtuelne, umesto realne, verzije nečega. U računarstvu ovaj postupak se koristi za kreiranje virtuelne verzije OS, servera, skladišnog prostora, mrežnih resursa...
- Virtuelizacija koristi softver kako bi simulirala funkcionalnost hardvera i time formira virtuelni računarski sistem.
  - Na ovaj način moguće je na jednom fizičkom serveru istovremeno pokrenuti više operativnih sistema, i na njima različite aplikacije – kreiraju se međusobno nezavisni virtuelni sistemi koji koriste istu hardversku platformu za izvršavanje.
- Prva primena pre nekoliko dekada kada je korišćena virtuelizacija OS-a da bi se na velikim *mainframe* računarima istovremeno pokretalo nekoliko slika OS, sa ciljem boljeg iskorišćenja računarskih resursa.

# Značaj virtuelizacije za *Cloud computing*?

- Virtuelizacija resursa “u oblaku” omogućava:
  - Izolaciju performansi (*Performance isolation*)
    - Resursi se mogu dinamički dodeljivati raznim korisnicima / aplikacijama po potrebi
  - Bezbednost sistema (*System security*):
    - Omogućava izolaciju servisa koji se pokreću na istom hardveru
  - Bolje performanse i pouzdanost (*Performance and reliability*):
    - Omogućava da se po potrebi aplikacije migriraju sa jedne na drugu hardversku platformu
  - Ponuđa resursa je u mogućnosti da ponudi servise za upravljanje virtuelizovanim resursima

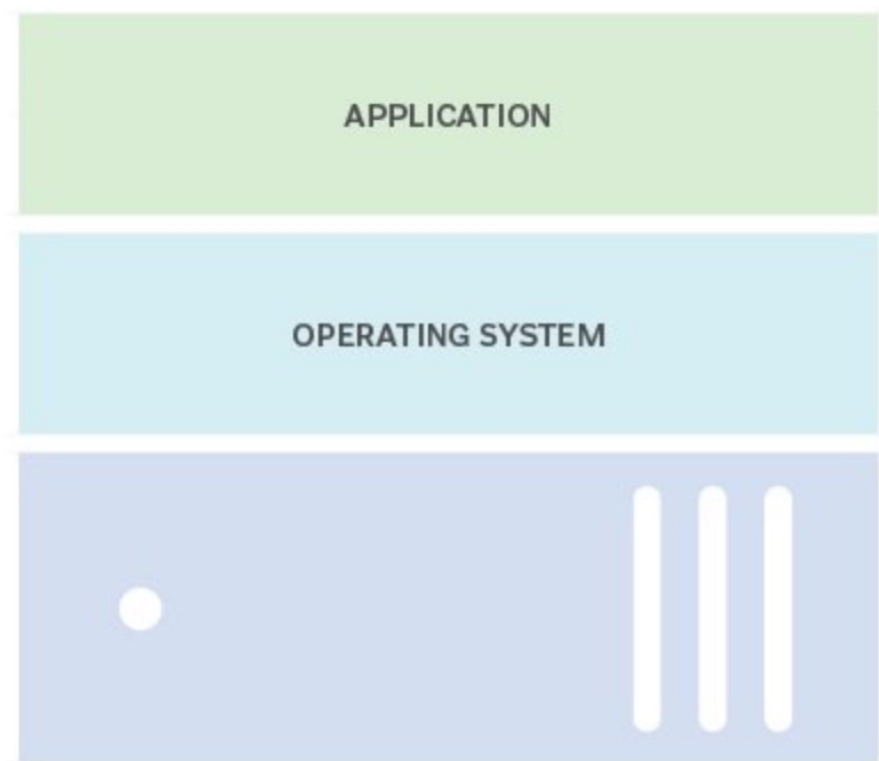
# Kako radi virtuelizacija?

- Virtuelizacija simulira interfejs fizičkim objektima na sledeće načine:
- Multipleksiranje (Multiplexing) – *many virtual to one physical object*: kreira više virtuelnih objekata na osnovu jedne instance fizičkog objekta. Ovi virtuelni objekti zatim „dele“ pristup fizičkom objektu.  
Primer: više procesa ili niti koriste isti procesor.
- Agregacija (Aggregation) - *One virtual object to many physical objects*: kreira jedan virtuelni objekat objedinjujući više fizičkih objekata. Primer: više diskova se udružuje u RAID disk.
- Emulacija (Emulation): konstruiše virtuelni objekat određenog tipa koristeći fizičke objekte drugog tipa. Primer: prostor na disku (danas najčešće SSD) emulira RAM.
- Kombinovanje multipleksiranja i emulacije (Multiplexing and emulation). Primer: virtuelna memorija sa *paging-om* multipleksira RAM i disk, a virtuelno adresiranje emulira stvarne memorijske adrese.

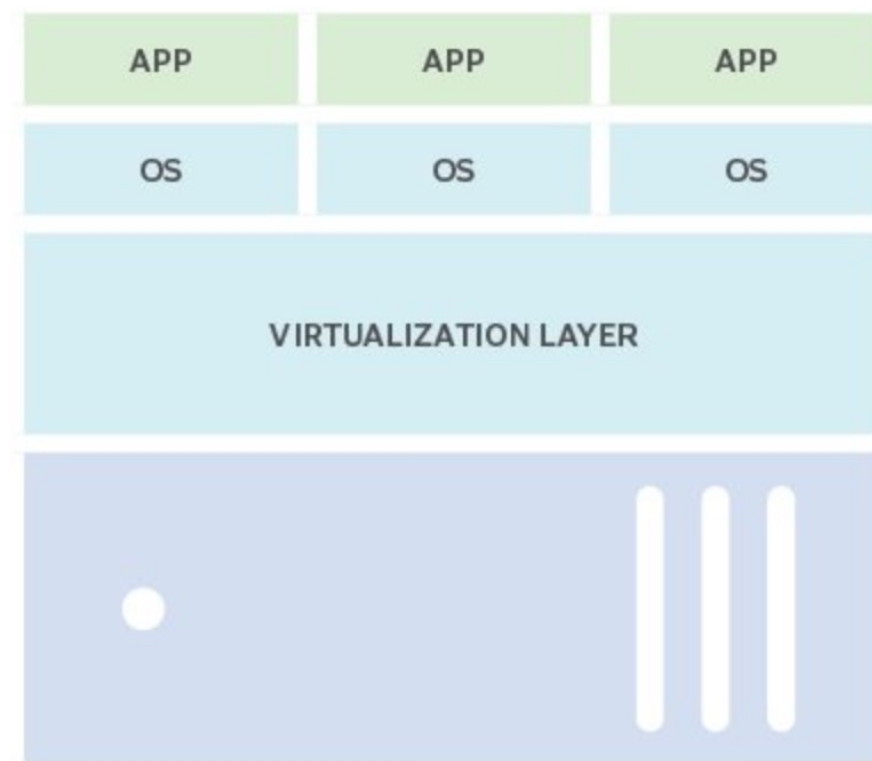
## Kako radi virtuelizacija? (2)

- Virtualizacija apstrahuje hardverske resurse, pojednostavljuje njihovo korišćenje, izoluje međusobno korisnike, omogućava replikaciju koja povećava „elastičnost“ sistema.
- Ključni način korišćenja virtuelizacije je virtuelizacija servera – softverski sloj (*hypervisor*) obezbeđuje emulaciju stvarnih hardverskih resursa (CPU, I/O, memorija, skladišni prostor, mreža... ).
- Hypervisor emulacijom omogućava da se virtuelizovanom okruženju stavi na raspolaganje deo hardverskih resursa.
- Hypervisor-i mogu biti instalirani kao aplikacija na OS, ili mogu biti realizovani tako da se oni direktno instaliraju na hardversku konfiguraciju (ovaj drugi oblik je danas češći).

# Tradicionalno okruženje / virtuelizovano okruženje



**Traditional architecture**



**Virtual architecture**



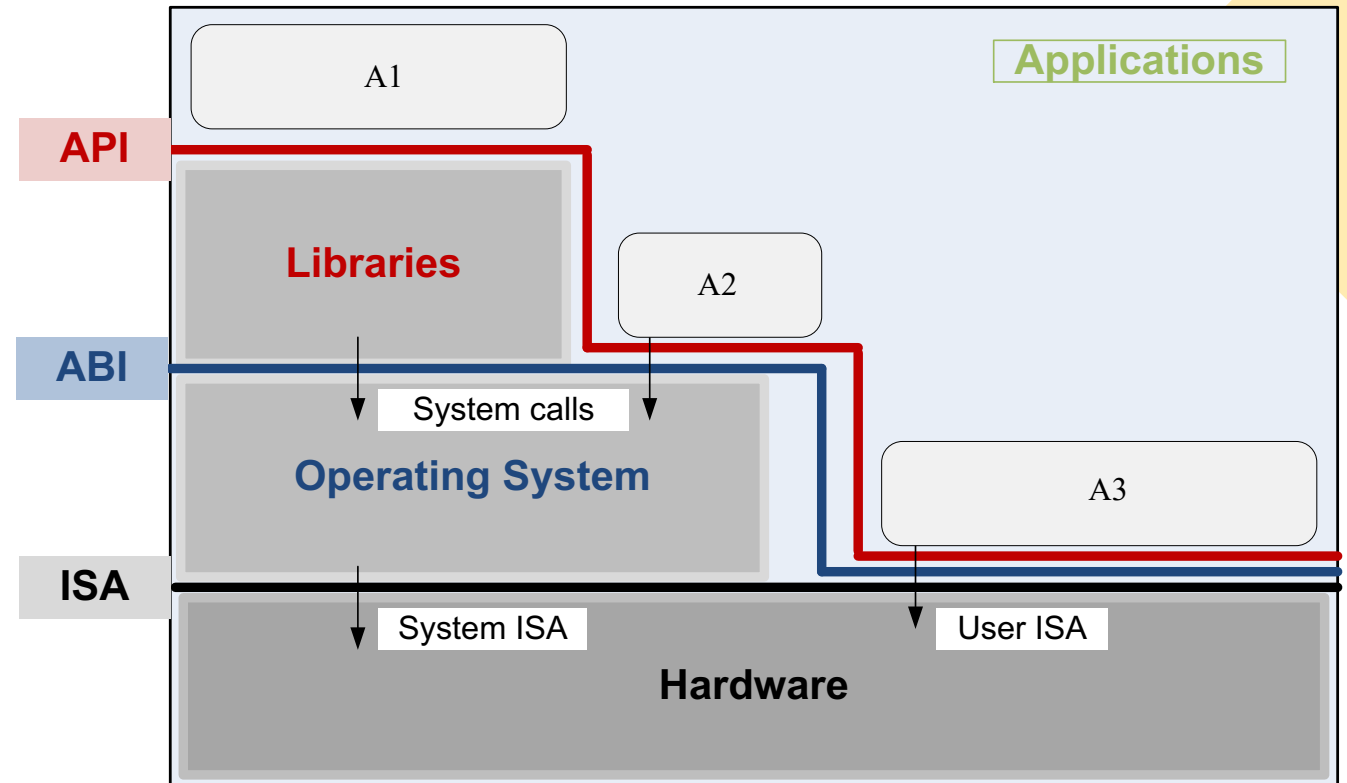
# Slojevitost sistema i virtuelizacija

- Razvoj i sagledavanje sistema kroz slojeve je standardno rešenje za kompleksne sisteme.
  - Pojednostavljuje opis podsistema. Svaki uočeni podsistem se apstrahuje putem **interfejsa** kojima se omogućava interakcija sa drugim podsistemima.
  - Minimizuju se neophodne interakcije između različitih podsistema u kompleksnom sistemu
  - Omogućava nezavisni dizajn, razvoj i održavanje podsistema (slojeva)
- Slojevitost u računarskim sistemima:
  - Hardware
  - Software
    - Operating system
    - Libraries
    - Applications

# Slojevitost sistema i interfejsi

- Application Programming Interface (API),
- Application Binary Interface (ABI),
- Instruction Set Architecture (ISA).

Tokom rada različite aplikacije mogu u određenim momentima koistiti pozive na različitim nivoima (A1 koristi API za pristup bibliotekama), (A2 poziva sistemske rutine), and (OS I A3 izvršavaju mašinske isstrukcije )



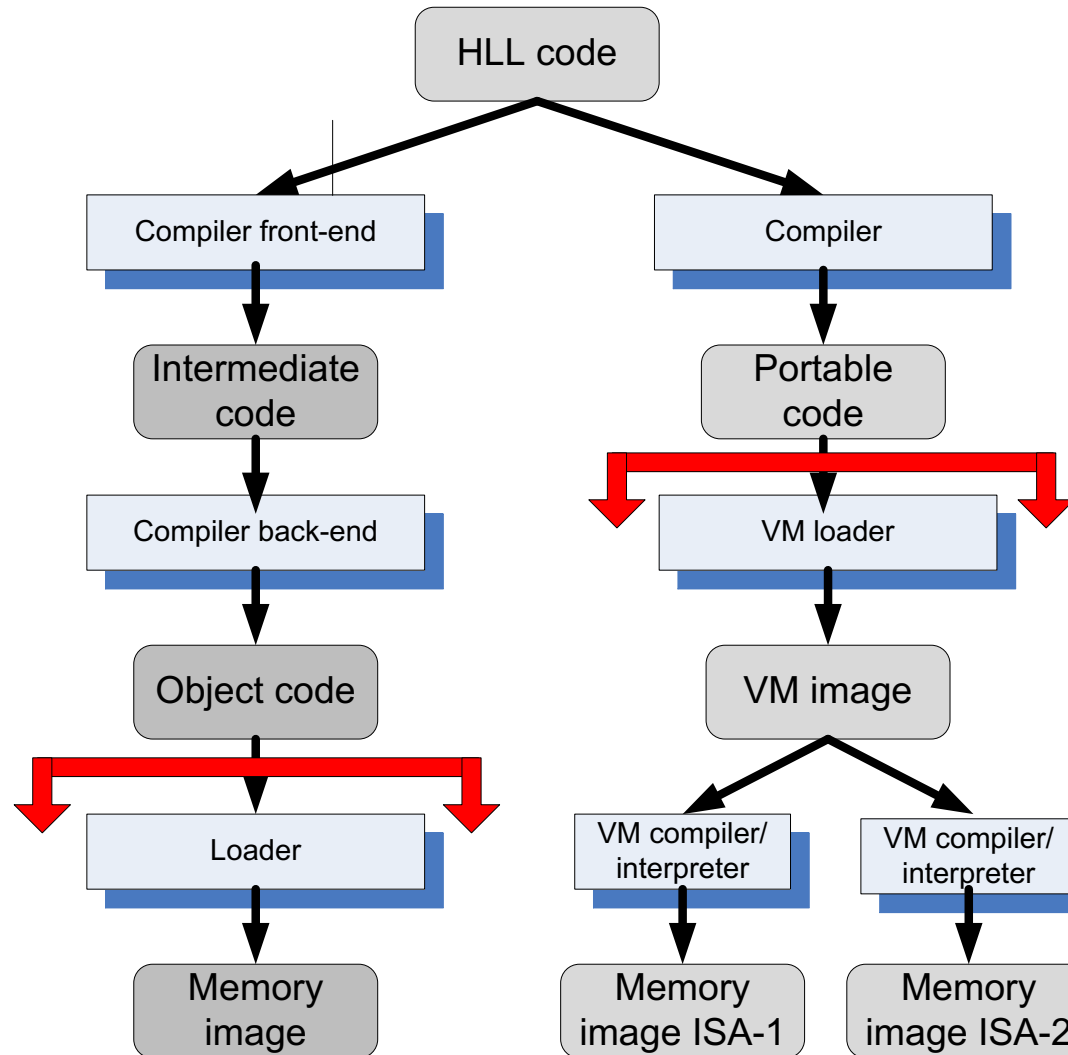
# Interfejsi

- Instruction Set Architecture (ISA) – na granici između hardvera i softvera
- Application Binary Interface (ABI) – omogućava aplikacijama i bibliotekama da koriste hardverske resurse. Ne sadrži pozive privilegovanih sistemskih instrukcija, već se pristup obavlja posredno preko sistemskih poziva.
- Application Program Interface (API) – definiše skup instrukcija koje hardver može da izvršava, i omogućava aplikacijama pristup ISA. Često sadrži biblioteke razvijene u višim programskim jezicima koje se oslanjaju na sistemske pozive da obave interakciju s hrdverom.

# Interfejsi i prenosivost koda

- Mašinski kod koji se dobija kompajliranjem za specifične ISA i specifični OS nije prenosiv
- Moguće je kompajlirati kod sa programskoj jezika visokog nivoa za specifičnu virtuelnu mašinu. U tom slučaju se dobija prenosivi kod koji se transformiše u mašinske instrukcije host sistema tokom izvršavanja.
- Dinamičko prevođenje na mašinski kod transformiše prenosivi kod u mašinski kod u blokovima, što omogućava bolje performanse – a neki blokovi prevedenog koda se mogu i keširati i ponovo koristiti bez ponovnog prevođenja.

# Interfejsi i prenosivost koda



# VMM (Virtual Machine Monitor / Hypervisor)

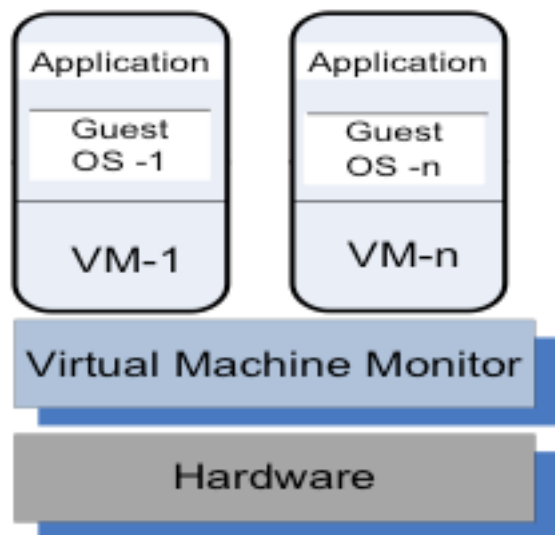
- VMM/hypervisor deli resurse računarskog sistema na više virtuelnih mašina (VM).
- Virtuelna mašina je izvršno okruženje u kome se izvršava OS
- VM – izolovano okruženje koje je prividno ceo računarski sistem, ali stvano rpistupa tek delu hardverskih resursa fizičkog sistema
- VMM/hypervisor omogućava:
  - Da više servisa dele istu hardversku platformu
  - Migraciju aktivnih VM (Live migration) – premeštanje VM sa jednog hardvera na drugi
  - Modifikaciju sistema tokom održavanja
  - Kompatibilnost „unazad“ sa originalnim sistemom
  - Forsira izolaciju između VM – time poboljšava bezbednost
- Gostujući OS je operativni sistem koji se izvršava na VM koja je pod kontrolom nekog VMM.

# VMM virtuelizacija procesora i memorije (kako?)

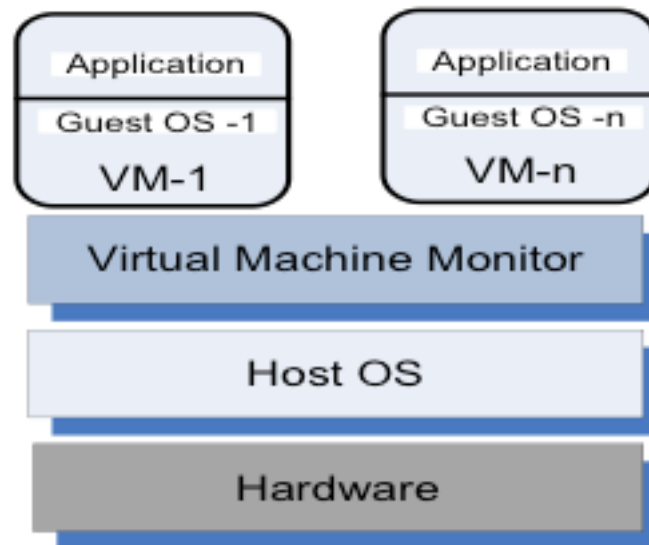
- Šta VMM obavlja da bi se postigla virtuelizacija:
- „hvata“ (*traps*) privilegovane instrukcije koje izvršava gostujući OS i obezbeđuje korektnost i bezbednost operacija
- „hvata“ interapte i prosleđuje ih gostujućim operativnim sistemima
- Kontrolise upravljanje virtuelnom memorijom
- Održava paging tabelu „u senci“ (shadow page table) za svaki gostujući OS i replicira bilo koju modifikaciju koju gostujući OS napravi u sopstvenoj paging tabeli. Ova paging tabela „u senci“ sadrži pokazivače na stvarne memorijske lokacije i koristi se za dinamičko prevođenje adresa (Memory Management Unit)
- Nadgleda performanse sistema i preuzima korektivne akcije kako bi se izbegla degradacija. Na primer, VMM može ukloniti sa sistema celu VM ako utvrdi da ona ugrožava performanse sistema iz bilo kog razloga.

# Hipervizori tipa 1 i tipa 2

Type 1 Hypervisor



Type 2 Hypervisor



## ■ Taksnomija VMM:

1. Type 1 Hypervisor (bare metal, native): podržava više VM i izvršava se direktno na hardverskoj platformi (VMware ESX , Xen, Denali)
2. Type 2 Hypervisor (hosted) VM – pokreće se na host OS-u (user-mode Linux)



# Neki od primera VMM

Name	Host ISA	Guest ISA	Host OS	Guest OS	Company
Integrity VM	x86-64	x86-64	HP-Unix	Linux, Windows HP Unix	HP
Power VM	Power	Power	No host OS	Linux, AIX	IBM
z/VM	z-ISA	z-ISA	No host OS	Linux on z-ISA	IBM
Lynx Secure	x86	x86	No host OS	Linux, Windows	LinuxWorks
Hyper-V Server	x86-64	x86-64	Windows	Windows	Microsoft
Oracle VM	x86, x86-64	x86, x86-64	No host OS	Linux, Windows	Oracle
RTS Hypervisor	x86	x86	No host OS	Linux, Windows	Real Time Systems
SUN xVM	x86, SPARC	same as host	No host OS	Linux, Windows	SUN
VMware EX Server	x86, x86-64	x86, x86-64	No host OS	Linux, Windows, Solaris, FreeBSD	VMware
VMware Fusion	x86, x86-64	x86, x86-64	Mac OS x86	Linux, Windows, Solaris, FreeBSD	VMware
VMware Server	x86, x86-64	x86, x86-64	Linux, Windows	Linux, Windows, Solaris, FreeBSD	VMware
VMware Workstation	x86, x86-64	x86, x86-64	Linux, Windows	Linux, Windows, Solaris, FreeBSD	VMware
VMware Player	x86, x86-64	x86, x86-64	Linux, Windows	Linux, Windows, Solaris, FreeBSD	VMware
Denali	x86	x86	Denali	ILVACO, NetBSD	University of Washington
Xen	x86, x86-64	x86, x86-64	Linux Solaris	Linux, Solaris NetBSD	University of Cambridge

# Performanse i izolacija

- Ponašanje aplikacija tokom izvršavanja je podložno uticaju drugih aplikacija koje se izvršavaju na istoj platformi i takmiče se za CPU, memoriju, pristup diskovima i mreži. Vreme izvršavanja je teško predvidljivo.
- Izolacija performansi (Performance isolation) – je kritični uslov za garancije za kvalitet servisa u deljenim okruženjima
- VMM je suštinski jednostavniji i bolje definisan sistem od tradicionalnih OS.
- Bezbednosti aspekt i ranjivost VMM je manji jer sistem daje pristup mnogo manjem broju privilegovanih funkcija. Xen VMM ima 28 hiperpoziva (hypercalls) dok Linux ima stotine sistemskih poziva

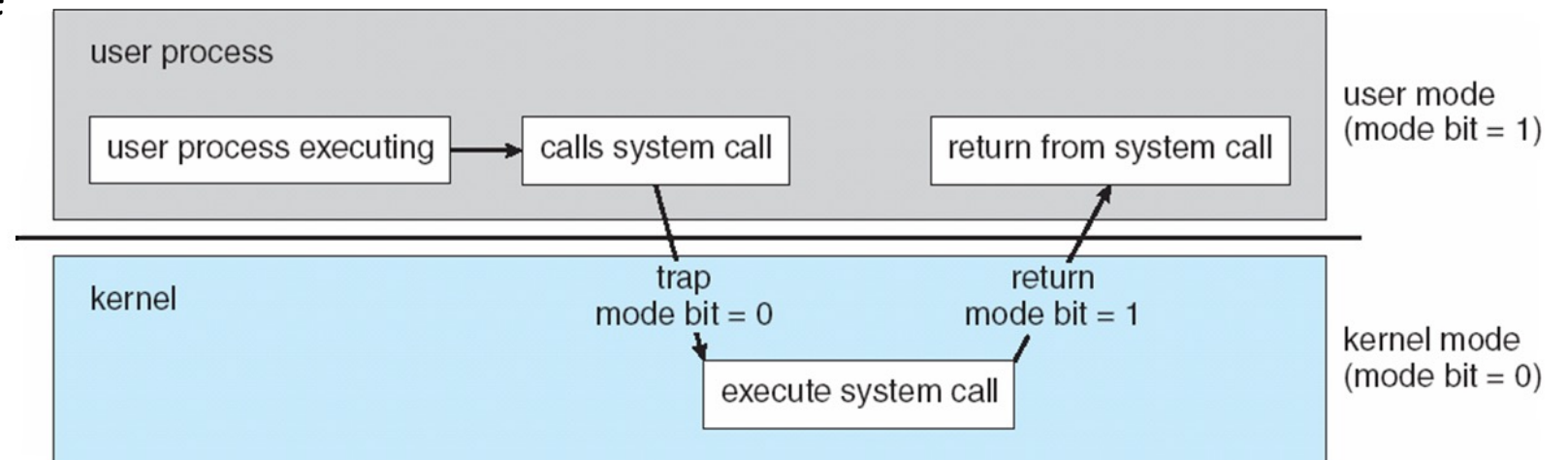
# Uslovi za uspešnu virtuelizaciju

- Uspešnost virtuelizacije počiva na sledećim uslovima<sup>1</sup>:
- Program koji se izvršava u okruženju koje kontorliše VMM trebao bi ispoljavati isto ponašanje kao kada bi se izvršavao direktno na ekvivalentnoj hardverskoj platformi.
- VMM treba da ima potpunu kontrolu nad virtuelizovanim resursima
- Statistički značajan deo mašinskih instrukcija mora se izvršavati bez potrebe da VMM izvrši intervenciju nad njima

1) Popek, G. J.; Goldberg, R. P. (July 1974). "Formal requirements for virtualizable third generation architectures". Communications of the ACM.

# Izvršavanje operacija u dva režima (dual-mode operation)

- Dva režima izvršavanja instrukcija omogućavaju OS da zaštiti sebe i druge komponente sistema
- **user mode / kernel mode**
- Bit koji označava režim izvršenja operacije
  - Omogućava razlikovanje situacije kada sistem izvršava neku operaciju u user modu ili kernel modu
  - Privilegovane instrukcije se mogu izvršavati samo u kernel modu
  - Sistemski pozivi menjaju režim izvršavanja u kernel mode, povratak iz sistemskog poziva vraća režim u user mode

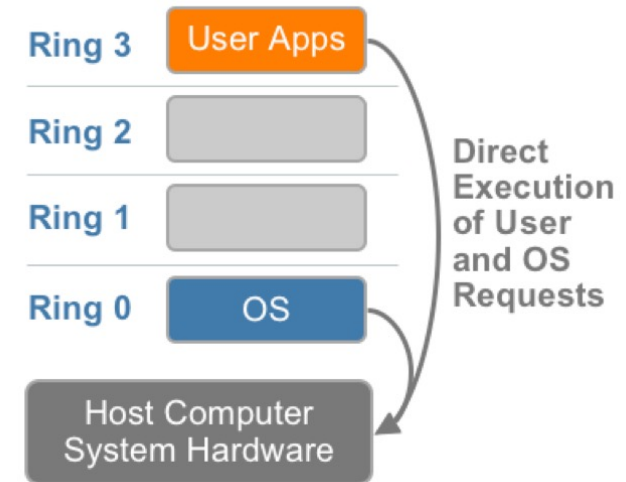


# Izvršavanje operacija u dva režima (dual-mode operation)

- Kernel kod (a posebno interrupt handler-i) se izvršavaju u kernel modu
  - U tom režimu hardveer dozvoljava izvršavanje svih mašinskih instrukcija i neograničen pristup IO i memoriji
- Sve ostalo se izvršava u korisničkom režimu (user mode)
- Svi OS-ovi se oslanjaju na ovaj hardverski mehanizam

# Izazovi virtuelizacije x86 arhitektura

- 4 sloja privilegija izvršavanja (rings - prsteni)
  - Korisničke aplikacije se izvršavaju u 3 prstenu
  - OS se izvršava na 0-tom prstenu
- Na kom nivou (u kom prstenu) izvršavati VMM?
  - Ako je na 0, onda je sa istim privilegijama kao OS -> **!OK**
  - Ako je na 1, 2 ili 3 onda OS ima veće privilegije od -> **!OK**
  - VMM na 0, a pomeriti OS na 1 -> **OK**
- Tri tipa mašinskih instrukcija:
  - **Privilegovane** – mogu se izvršavati u kernel modu. Ukoliko se pokušaju izvršiti u user modu – trap!
  - **Neprivilegovane** – mogu se izvršavati u korisničkom režimu
  - „**osetljive**“ – mogu se izvršavati u oba režima ali se ponašaju različito i zahtevaju posebnu kontrolu tokom izvršavanja.
- Neprivilegovane i osetljive se teško virtualizuju



# Tehnike virtuelizacije CPU x86 arhitekture

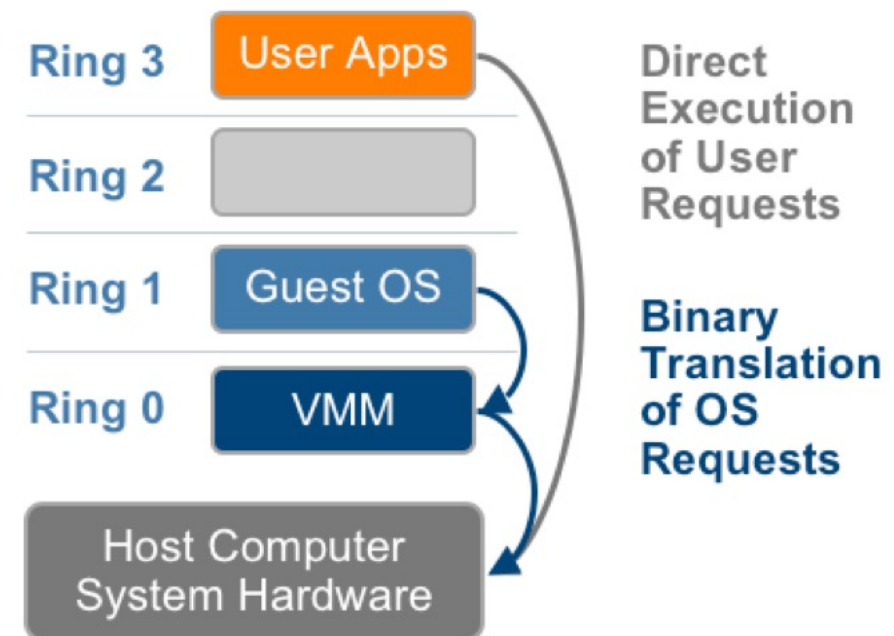
- Puna virtuelizacija sa binarnim prevođenjem
- Virtuelizacija potpomognuta od strane OS - paravirtuelizacija
- Hardverski potpomognuta virtuelizacija



# Tehnike virtuelizacije CPU x86 arhitekture

## Puna virtuelizacija

- **Puna virtuelizacija** – gostujući OS može se neizmenjen izvršavati na VM pod kontrolom VMM. Svaka VM izvršava se na identičnoj kopiji realnog hardvera.
- Binarno prevođenje dinamički (on-the-fly) prepisuje delove koda tako da menja osetljive ali neprivilegovane instrukcije za odgovarajuće privilegovane i emulira originalne instrukcije
- “Hieprvisor prevodi sve OS pozive i kešira rezultat prevođenja za naknadnu upotrebu, dok se sve user mode instrukcije prosleđuju direktno i izvršavaju istom brzinom kao na hardveru” (VMware paper)
- Primeri: VMware, Microsoft Virtual Server
- Prednosti:
  - Radi i bez hardverske podrške za virtuelizaciju,
  - Nisu potrebne modifikacije OS
  - Izolacija, bezbednost
- Nedostaci:
  - Brzina izvršavanja

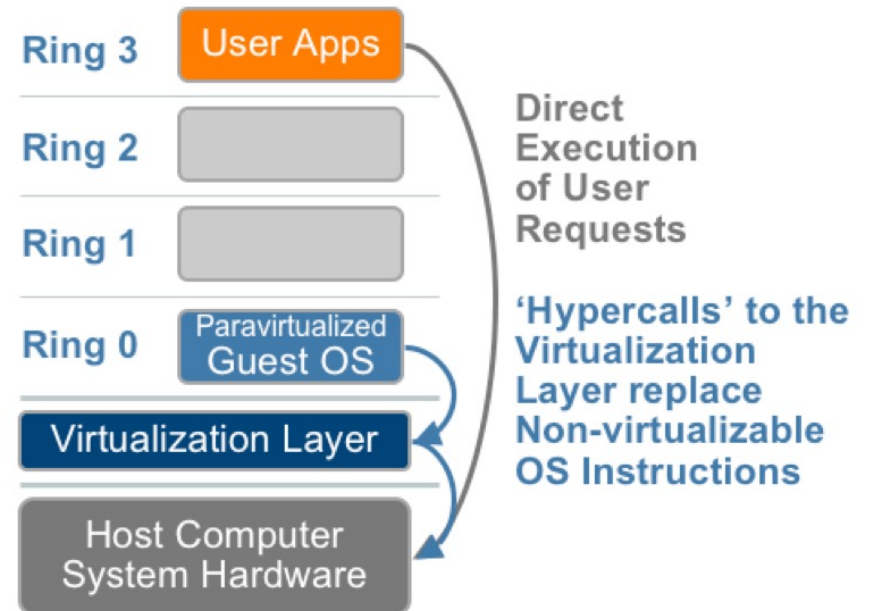




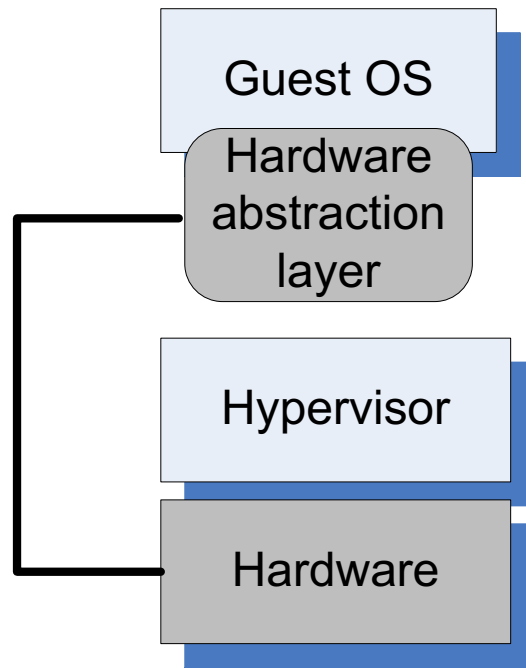
# Tehnike virtuelizacije CPU x86 arhitekture

## Paravirtuelizacija

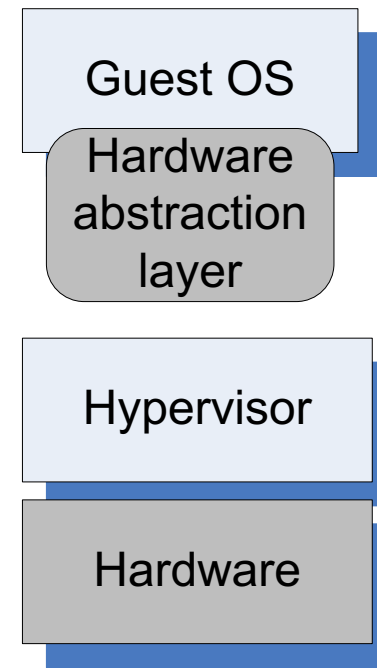
- **Paravirtualization** – “zahteva modifikciju OS kernela kako bi se instrukcije koje nisu pogodne za virtuelizaciju zamenile *hiperpozivima* koji direktno komuniciraju sa slojem za virtuelizaciju. Hipervizor obezbeđuje interfejs za hiperpozive za kritične kernel operacije kao što su upravljanje memorijom, upravljanje prekidima, evidencija vremena. “  
(VMware paper)
- Prednost: Brže izvršavanje, manji overhead virtuelizacije
- Nedostaci: Smanjena portabilnost
- Primeri: Xen, Denali



# Puna virtuelizacija - paravirtuelizacija



(a) Full virtualization



(b) Paravirtualization

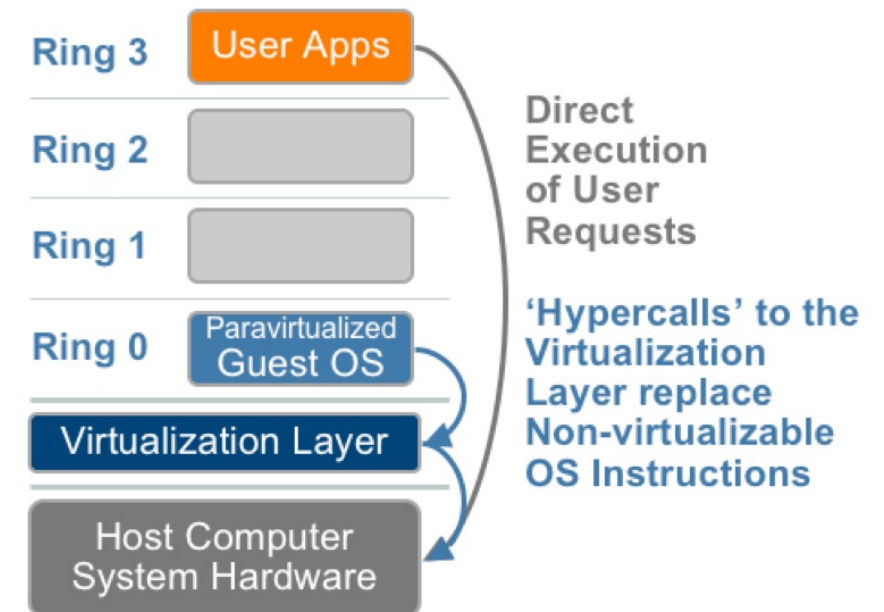
# Tehnike virtuelizacije CPU x86 arhitekture

## Hardverski podržana virtuelizacija

- **Hardware Assisted Virtualization** – “novi režim izvršavanja instrukcija na CPU koji omogućava da se VMM izvršava u root mode režimu ispod nivoa 0. Privilegovane i osetljive instrukcije se sada automatski „hvataju“ od strane hipervizora, čime se uklanja potreba bilo binarnog prevođenja bilo paravirtuelizacije “

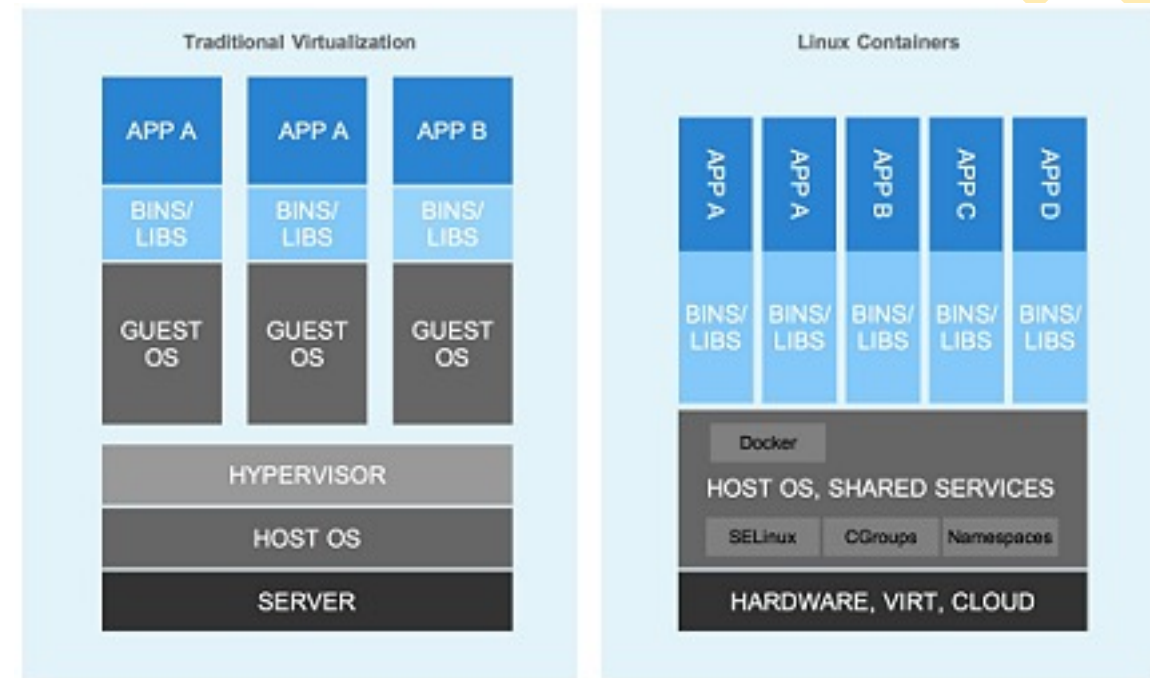
(VMware paper)

- Prednost: još brže izvršavanja
- Primeri: Intel VT-x, Xen 3.x
  - U ranim 2000 postalo je jasno da je za uspešnu virtuelizaciju neophodna podrška na nivou hardvera. Intel i AMD razvijaju procesore sa podrškom za virtuelizaciju koji postaju dostupni oko 2005 (Intel VT-x, 2006 AMD-V i Athlon 64 modeli)



# Linux kontejneri

- Linux Container je Linux proces koji se izvršava u sopstvenom virtuelnom okruženju i koji ima sopstveni mrežni (adresni) prostor. Predstavlja metoda „lake“ virtuelizacije procesa.
- Kontejneri dele deo host OS-a
- Kontejneri koriste izolaciju OS resursa na nivou procesa



# Tamna strana virtuelizacije

- U slojevitom sistemu zaštitni mehanizmi na jednom sloju potencijalno mogu biti ugroženi/isključeni malware-om koji se izvršava u sloju ispod.
- Maliciozni VMM, Virtual-Machine Based Rootkit (VMBR) se može umetnuti između fizičkog harvera i OS-a.
  - Rootkit - *malware* sa privilegovanim pristupom sistemu.
- VMBR može naknadno obezbediti pokretanje odvojenog, malicioznog OS-a, koji bi bio nevidljiv gostujućem OS-u i njegovim aplikacijama.
- Pod „zaštitom“ VMBR, maliciozni OS bi mogao:
  - Ima pristup podacima, događajima i stanju posmatranog sistema
  - Izvršava neželjene servise (spam relays / distributed denial-of-service napade).
  - Ometa ili utiče na rad aplikacija.

# Tipovi virtualizacije - šta se sve virtualizuje

- **Network virtualization** – kombinovanje raspoloživih mrežnih resursa u virtuelnu mrežu podelom propusnog opsega fizičke mreže. Mreža se na ovaj način i segmentira u delove kojima je lakše upravljati, a svako viirtuelizovano okruženje je nezavisno od ostalih.
- **Storage virtualization** – objedinjavanje fizičkih kapaciteta za skladištenje podataka u prividno jedinstven smeštajni kapacitet koji se zatim može deliti i popotrebi dodeljivati korisnicima.
- **Server virtualization** – virtualizacija serverskih resursa – skriva od korisnika kompleksnost upravljanja serverskim instancama, a istovremeno omogućava efikasno deljenje hardverskih resursa.

# Tipovi virtualizacije - šta se sve virtualizuje

- **Data virtualization** – apstrahovanje tehničkih detalja upravljanja podacima – lokacija, formati zapisa, način pristupa – povećava dostupnost i (zbog replikacije) otpornost sistema na gubitak podataka.
- **Desktop virtualization** is – virtualizacija radnih stanica koje mogu da se pokreću na udaljenim mašinama a pristupa im se preko „tankih klijenata“. Kako se radne stanice (desktop mašine) zapravo pokrežu na serverima u dana centrima ovo omogućava i bolju administraciju i zaštitu pristupa.
- **Application virtualization** - apstrakcija aplikativnog sloja od operativnog sistema. Na ovaj način aplikacije mogu da se izvršavaju u „zatvorenom“ okruženju nezavisno od OS-a na kome se pokreću.

• ...

# Zaključak

- Pitanja?