nedeljainformatike v7.0

Kako rade virtuelne mašine

Marko Grujčić

Matematička gimnazija

15. 04. 2022.

Šta je virtuelna mašina?

- Simulacija mašine - više instanci na jednom fizičkom računaru
- VM sistema i VM procesa
  - VM sistema ponaša se kao celokupan računar sa svojim OS
  - VM procesa - Programi nezavisni od platforme. Interpreteri. JVM.
- Hipervizor - softver koji upravlja virtuelnim mašinama

Zašto virtuelne mašine?

- Primer: kompanija ima Web server, FTP server, mail server...
- Jeftinije. Bolja upotrebljenost resursa nego više računara.
- Sandboxing - bolja sigurnost i robustnost nego jedan računar. Greške su uglavnom softverske, a ne hardverske.

- Legacy softver
- Testiranje softvera

Zahtevi za virtuelnu mašinu

- **Sigurnost** - Izolovanost - hipervizor ima punu kontrolu nad resursima
- **Ekvivalentnost** - Ponašanje programa identično kao na hardveru
- **Efikasnost** - Minimizovati intervencije hipervizora

Režimi rada, prekidi

- Privilegovane instrukcije. Režim rada određuje koje instrukcije se mogu izvršiti.
- Kernelski i korisnički režim rada.
- Rad sa I/O uređajima, ključne strukture operativnog sistema...

- Prekidi - mehanizam kojim se zaustavlja izvršavanje na procesoru i poziva se zadata rutina obrade prekida.
  - Hardverski - spoljni uređaj.
  - Softverski - Zahteva ih procesor pri uslovima poput: specijalne instrukcije, greške. Najčešće kontrolu preuzima kernel - trap.

Virtuelna memorija. Straničenje.

- Problemi pri smeštanju problema u memoriju - jedni izolovani od drugih, ne zna se unapred gde će program biti smešten u memoriji...
- Rešenje: svaki program ima svoj virtuelni adresni prostor.
- MMU prevodi virtuelne u fizičke adrese.
- Popularno rešenje: straničenje.

Interpretacija i direktno izvršavanje

- Sigurno i ekvivalentno rešenje - svaka instrukcija se interpretira.
- Direktno izvršavanje instrukcija
- Instrukcije koje se nikako ne mogu izvršavati direktno (isključivanje prekida, modifikovanje tabela stranica)

Kernel u korisničkom režimu

- Privilegovane instrukcije - trap u korisničkom režimu, mogu se izvršiti u kernelskom režimu
- VM u korisničkom režimu, potrebno emulirati instrukcije kernelskog režima
- “Trap and emulate” arhitektura - pri prekidu hipervizor preuzima kontrolu i emulira privilegovane instrukcije.

Teorema Popeka i Goldberga

- Osetljive instrukcije- instrukcije koje se ponašaju različito u korisničkom režimu i kernelskom režimu. Instrukcije koje rade sa I/O, prevode logičke u fizikče adrese isl.
- Popek i Goldberg: Mašina je virtuelizabilna samo ako je skup osetljivih instrukcija podskup skupa privilegovanih instrukcija.

x86

- Po kriterijumu teoreme arhitektura skupa instrukcija x86 nije virtuelizabilna.
- Primer: POPF
  - Instrukcija sa vrha steka uzima 16 bitova i smešta ih u EFLAGS registar
  - U zavisnosti od režima rada procesora menja ili ne menja flegove koji označavaju da su prekidi isključeni
- Pre 2005. Program je mogao da utvrdi iz korisničkog režima da li je u korisničkom ili kernelskom režimu.

Virtualization Technology

- Od 2005. Intelovi i AMD-ovi procesori pružaju podršku za virtuelizaciju arhitekturom “trap and emulate”. (VT/SVM)
- Ideja - kreiraju se kontejneri u kojima se pokreću VM. Hipervizor postavlja bitmapu koja označava koje instrukcije izazivaju trap.
- Prvi hipervizor za x86 lansirao je VMware 1999. godine. Kako?

Binary translation

- VMM u letu menja kod OS.
- Problematične instrukcije menjane prekidima, VMM obavlja ekvivalentnu radnju nad virtuelnom mašinom.
- Potrebno promeniti samo osetljive instrkucije koje bi trebalo obaviti u kernelskom režimu.

Binary translation

- Aktuelni operativni sistemi na x86 koriste samo prstene zaštite 3 i 0 na x86.
- Virtuelizovani operativni sistem ima svoj virtuelni kernelski prostor i virtuelni korisnički prostor.
- Gostujući operativni sistem stavlja se u prsten 1. Time je zaštićen od procesa u virtuelnom korisničkom prostoru, a privilegovane instrukcije i dalje obrađuje VMM.

Binary translation

- Kod se analizira u jedinici bloka.
- Blok je deo koda čija samo posednja instrukcija menja tok izvršavanja.
- Ako ima osetljivih instrukcija menja ih pozivom hipervizora.
- Radi boljih performansi prevedeni blokovi se keširaju.
- VMM ako je dobro konfigurisan može ignorisati virtuelni korisnički prostor i pustiti ga da se direktno izvršava.

Performanse

- Trap na savremenom sistemu nije jeftin. Uništavaju keš, TLB, i predviđanja toka.
- Često se i privilegovane osetljive instrukcije menjaju, prekidi su skupi, bolje performanse može imati binary translation.
- Binary translation može u nekim situacijama imati bolje performanse od izvornog koda. Primer: maskiranje hardverskih prekida na složenim procesorima. Jednostavnije održavati interrupt flag u strukturi koja odgovara virtuelnom procesoru.

Virtuelizacija memorije

- Gostujućem OS se ne mogu dodeliti okviri koje očekuje.
- *Shadow page table*
- Virtuelne adrese gosta mapiraju se direktno u fizičke adrese.
- Potrebno ispratiti promene koje napravi gostujući OS u tabeli.
- *Hypervisor-induced page fault* - sinhronizovanje stranica.

Hardverska podrška za virtuelizaciju memorije

- VT i SVM nisu od početka imali podršku za virtuelizaciju memorije.
- EPT (Extended Page Tables) i NPT tek nekoliko godina kasnije.
- Hipervizor i dalje održava dodatne stranice, ali procesor sada zna da ih obiđe, znatno smanjuje broj izlaza iz VM.

Vraćanje memorije

- *Overcommitment*
- *Deduplication* - Deljenje stranica između virtuelnih mašina.
- *Balooning*

Paravirtuelizacija

- Pristup virtuelizaciji koji podrazumeva saradnju OS.
- OS se modifikuje. Umesto osetljivih instrukcija izvršava hiperpozive.
- Znatno pojednostavljuje i ubrzava sistem.

Tipovi hipervizora

- Tip 1 - Jedini program u privilegovanom režimu - kao operativni sistem.
- Xen, VMware ESXi
- Mora imati svoje drajvere.

- Tip 2 - Oslanjaju se na host operativni sistem.
- VMware Workstation
- Gostujući OS instalira se na virtuelni disk, koji je suštinski fajl na host OS.

Primer: VMware Workstation

- Prvi proizvod za virtuelizaciju x86 računara.
- Virtuelizacija do tada bila primarno na mainframe računarima. Kod mainframe računara postoji vertikalna integrisanost - isti proizvođač odgovaran za hardver, OS, hipervizor i veći deo korisničkog softvera.
- x86 nije virtualizabilna, izuzetno je složena, i ima raznolike periferne uređaje.
- Uz to potrebno obezbediti i jednostavno korišćenje.

Primer: VMware Workstation

- VMware Workstation - hipervizor tipa 2
- Sastoji se iz nekoliko modula
- VMM - zadužen za izvršavanje virtuelne mašine
- VMX - zadužen za interakcije sa host OS

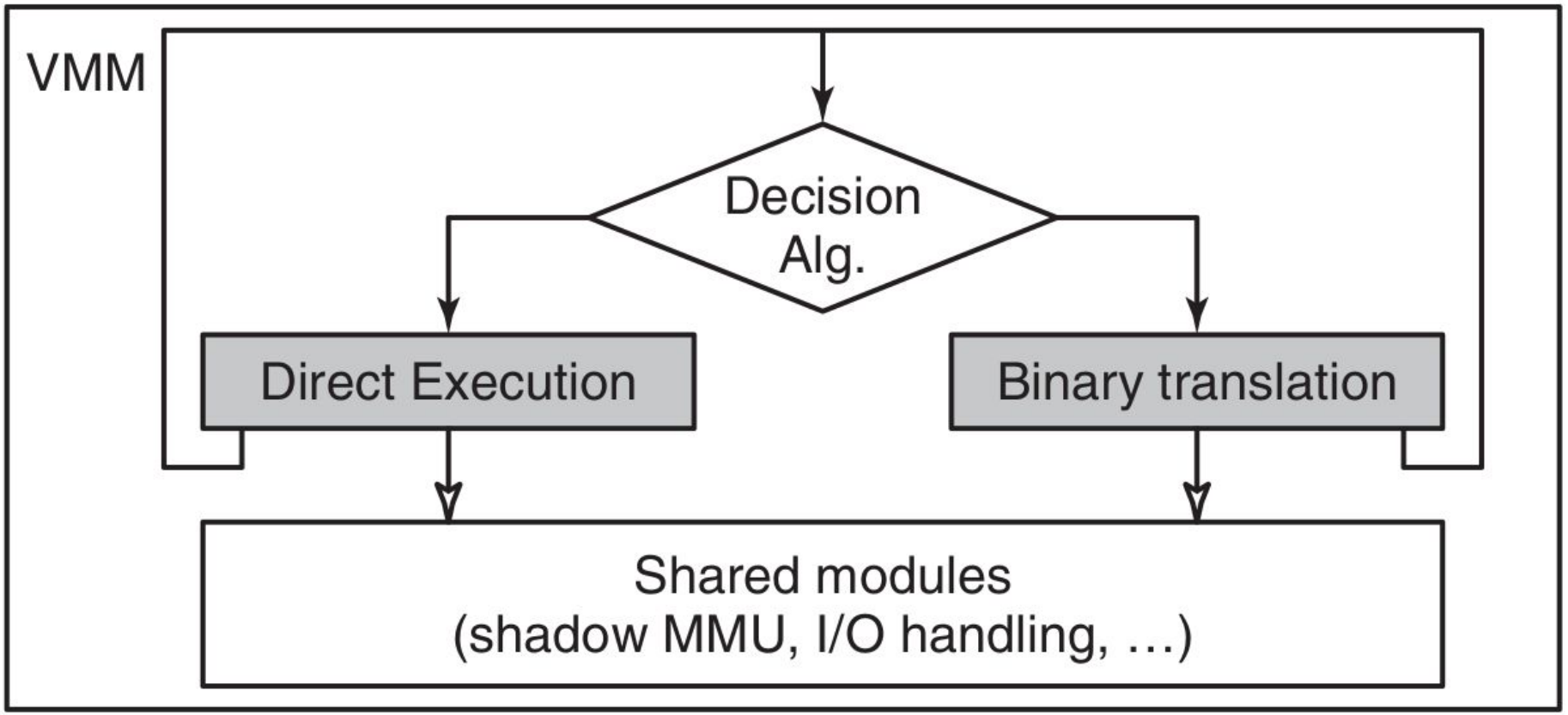
Više o virtuelizaciji x86

- Paravirtuelizacija nije moguća - izvorni kod Windows OS nije dostupan.
- Emulacija svega izuzetno sporo rešenje.
- Dinamički *binary translation* - limitira faktor usporenja do 5 puta. I dalje nije dovoljno brzo. Pošto x86 nije virtuelizabilan potrebno optimizovati ovo rešenje

Više o virtuelizaciji x86

- Ključno opažanje: Problematične instrukcije nisu uvek problematične.
- Nisu problematične u korisničkim programima koji čine većinu posla.
- Samim tim, možemo korisničke programe virtuelzivati *trap-and-emulate* pristupom.

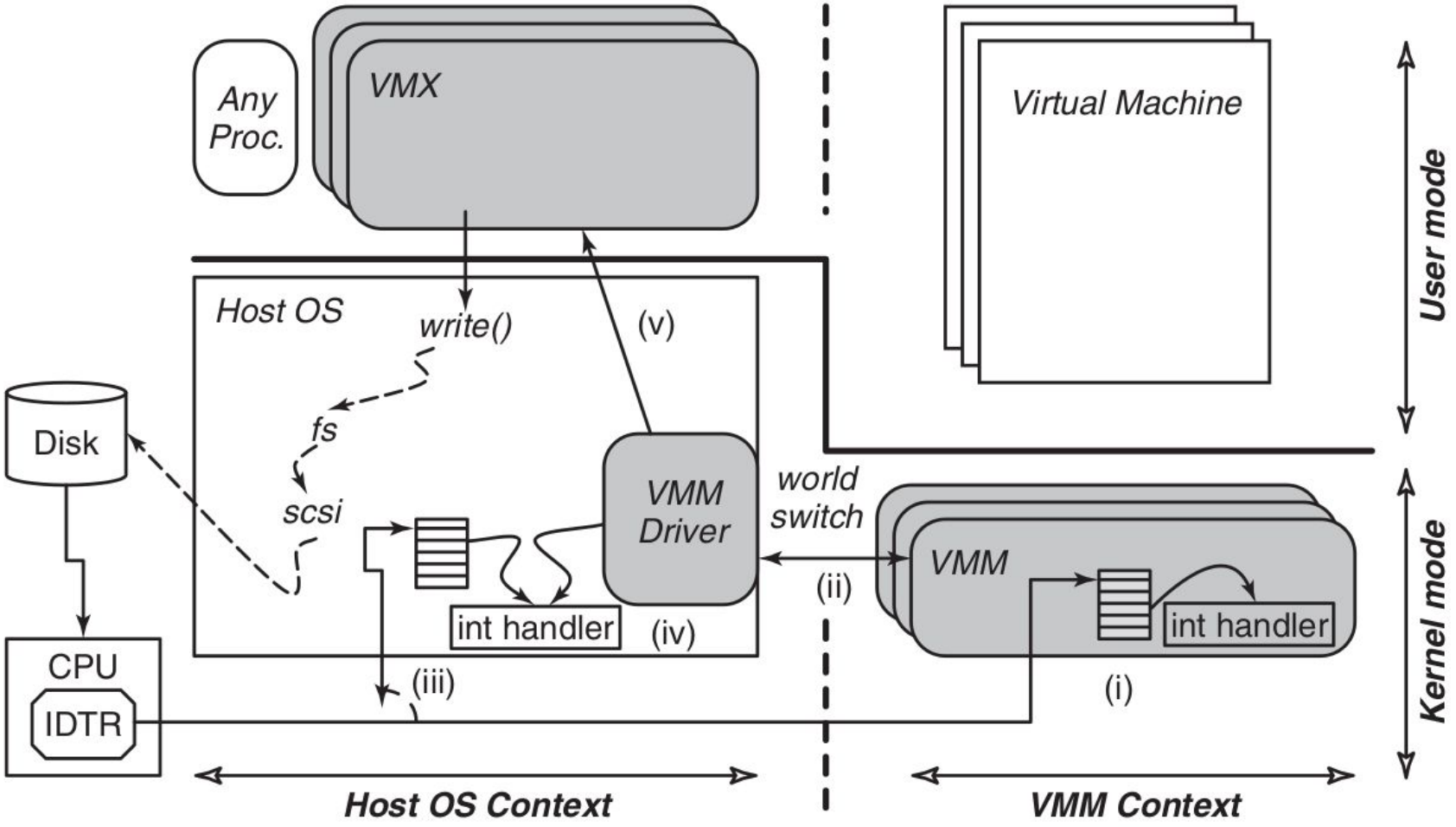
VMM


VMM
Decision
Alg.
Direct Execution
Binary translation
Shared modules
(shadow MMU, I/O handling, …)

Uloga Host OS

- Velika prednost VMM tipa 2 - oslanja se na drajvere hosta.
- Na primer - disk gostujućeg OS može se emulirati fajlom, crtanjem u prozoru host OS emulira se video karta gosta.
- Druga prednost - korisnik VMM može instalirati kao obični program.
- Sa druge strane optimizacije VMM oslanjaju se na direktan pristup hardveru. Za razliku od korisničkih aplikacija, drajveri imaju znatno više privilegija.
- Problem sa drajverima je što se od njih očekuje interakcija sa OS strogo definisanim API-jem, i ne bi trebalo proizvoljno da rekonfiguriše hardver.

VMware Hosted Architecture



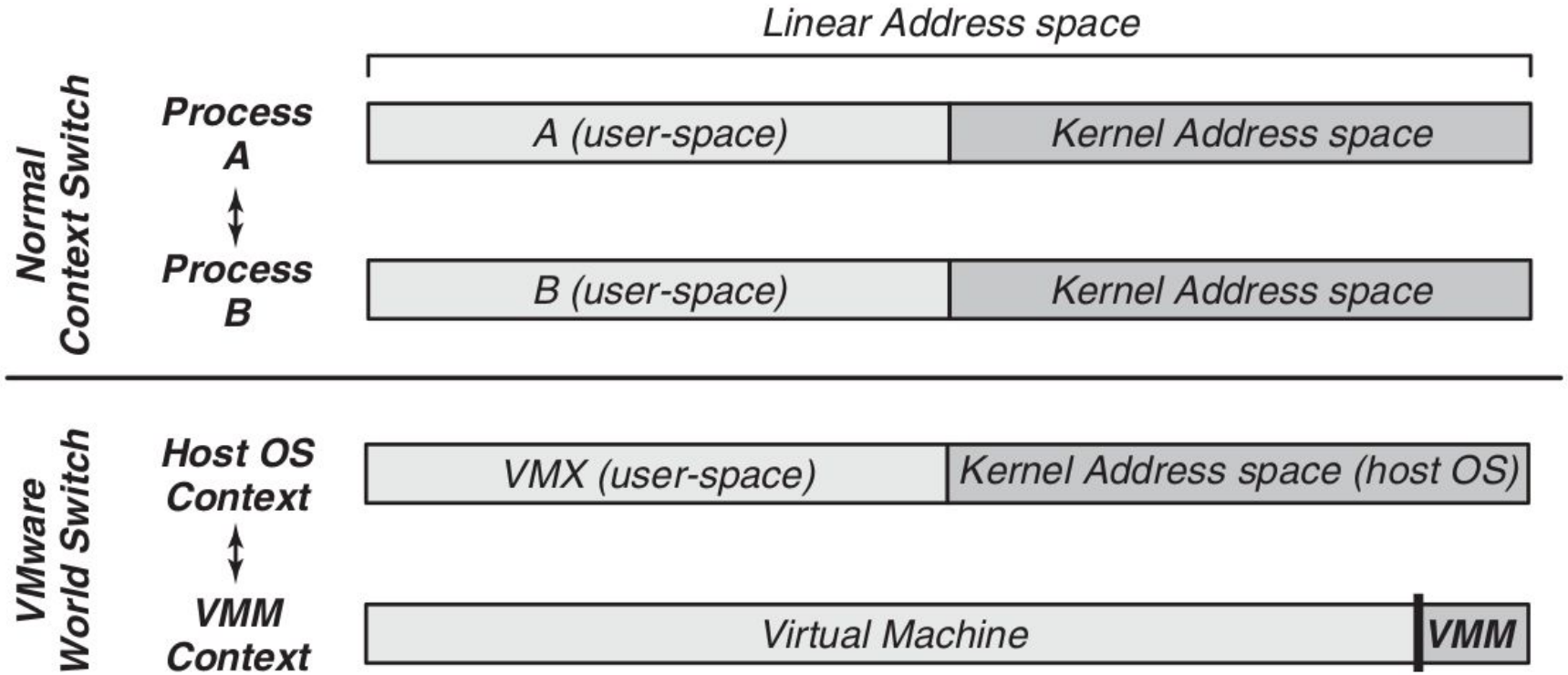
Komponente arhitekture

- VMX - program u korisničkom prostoru. Obavlja interkaciju sa korisnikom i emulaciju uređaja i obavlja sistemske pozive za interakciju sa OS. Obično postoji jedan VMX proces po virtuelnoj mašini.
- VMX driver - instalira se u OS. Uloga mu je da omogući rad VMM.
- VMM - sadrži softver za deljenje resursa, obradu prekida, *binary translator*, modul za ugnježđeno straničenje. Radi u kernelskom režimu ali van konteksta hosta, ne može se oslanjati na hosta direktno ali takođe nije ograničen hostom. Postoji instanca za svaku virtuelnu mašinu.

World switch

- Tranzicija između VMM i kernela - dva u potpunosti nezavisna konteksta na nivou sistema.
- Dok pri regularnoj promeni konteksta mnoge sistemske strukture ostaju iste, pri ovoj promeni se menja sve - obrada izuzetaka, vrednosti u privilegovanim registrima, adresni prostor...
- 45 instrukcija jezika x86 sistema.
- Kernel gostujućeg OS je deo adresnog prostora u VMM kontekstu - gostujući OS može koristiti ceo adresni prostor pa nema kolizije sa kernelom host OS.

World switch



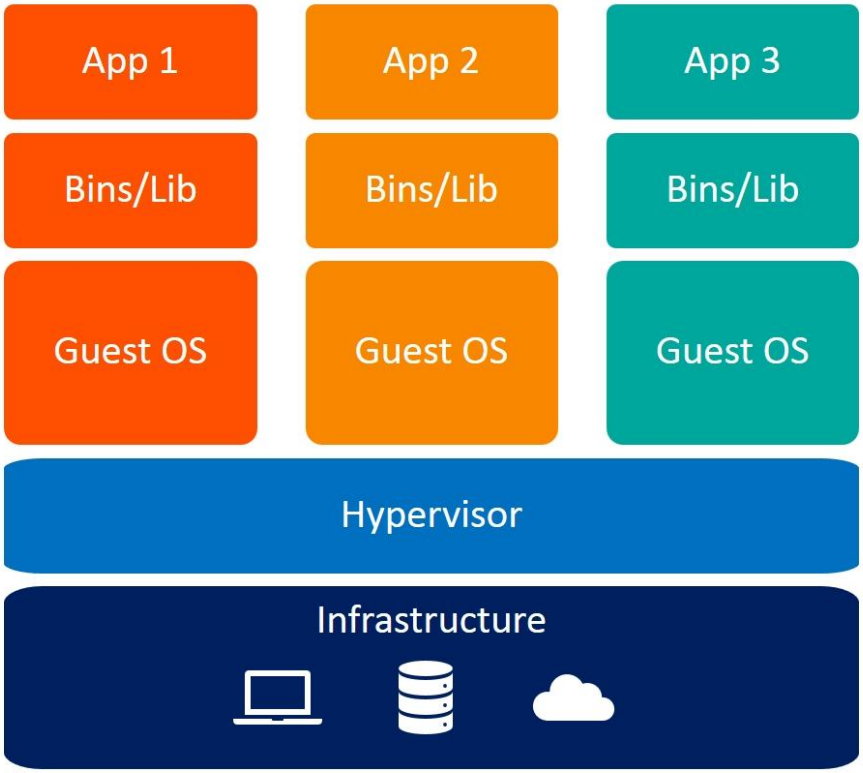
Virtuelizacija na nivou OS

- Kernel dozvoljava postojanje više izolovanih korisničkih prostora
- Posledica: mogu se ograničiti resursi računara kojima procesi mogu pristupiti.
- Različiti kontejneri mogu imati skroz drugačije poglede na sistem (stablo procesa, fajl sistem, mreže...)
- Manje resursa nego virtuelizacija sistema.
- 2007. Linux API proširen, kontejneri procesa.
- Kontejneri, zone, virtuelni privatni serveri, particije, virtuelna okruženja, *jail*

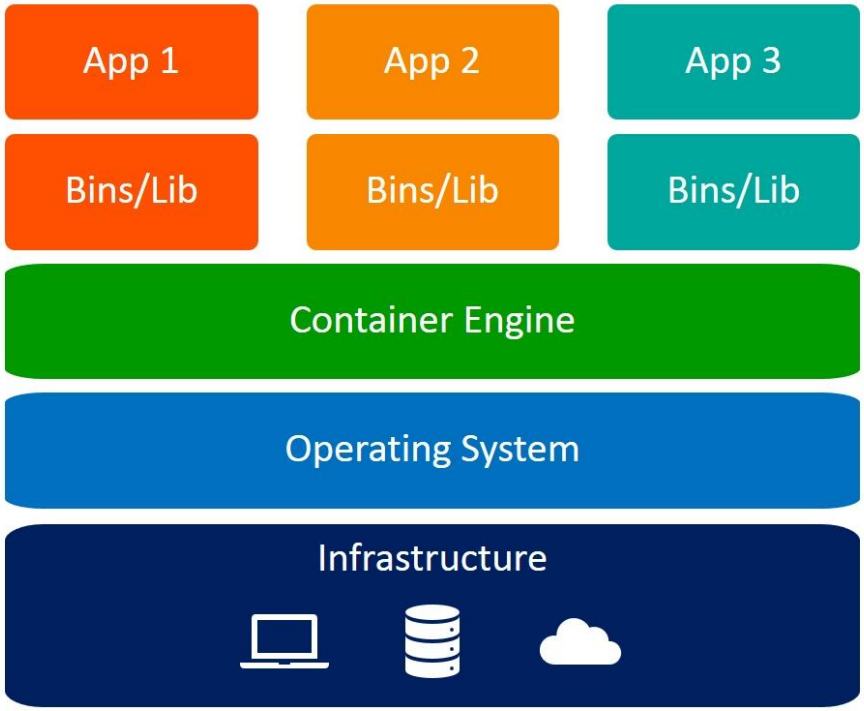
Docker

- Skup proizvoda koji koriste virtuelizaciju na nivou OS.
- Docker engine omogućava prenošenje okruženja bez obzira na platformu.
- Softver se dostavlja u paketima zajedno sa okruženjem - kontejnerima

Docker



Virtual Machines



Containers

Java virtuelna mašina

- Java programi prevode se u bajtkod koji se zatim interpretira
- Cilj JVM je izvršavanje Java bajtkoda nezavisno od platforme
- Apstraktan računar koji ima svoj skup instrukcija
- Programski brojač i 3 registra za rad sa stekom
- Stek i hip
- JIT kompajler

Cloud

- *As-a-service*
- *IaaS* - Korisnik radi sa virtuelizovanim resursima šta želi
- *PaaS* - Specifično okruženje - OS, baza podataka, Web server
- *SaaS* - Office 365, Google Apps

Virtuelizacija i računarstvo u oblaku

- Različiti OS na jednoj mašini
- Hipervizor razdvaja mašinu od hardvera. *Decoupling*
- Olakšava premeštanje. *Live mitigation*
- *Checkpointing* - čuvanje stanja mašine. Kopiranje svega. *Copy on write*

Hvala na pažnji!

Pitanja?