

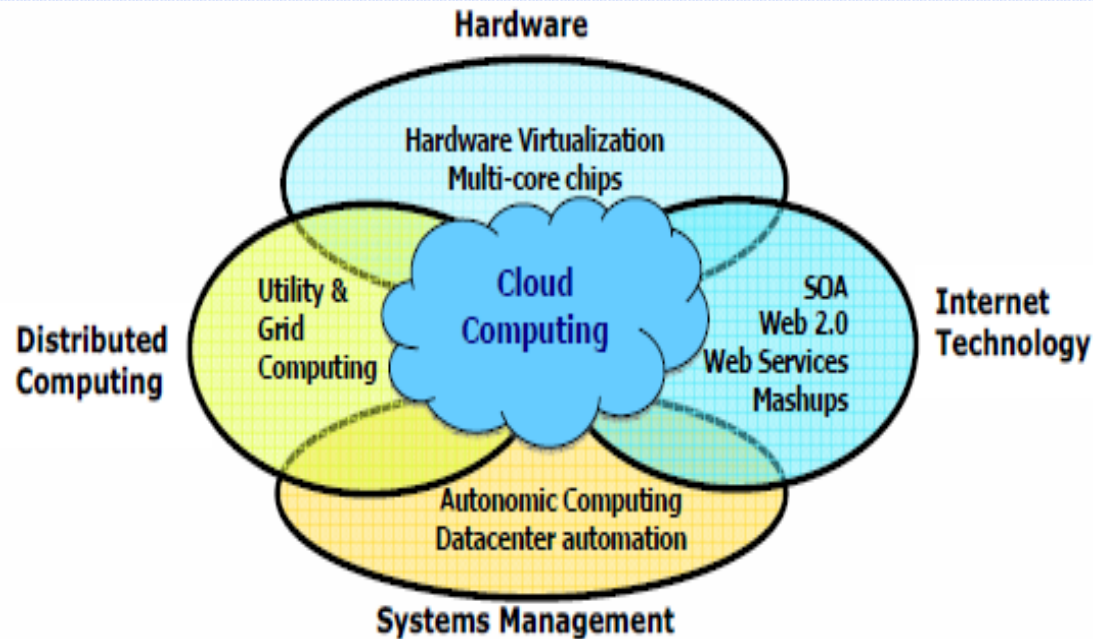
Bazne tehnologije i modeli distribuiranih sistema I

- ❖ Bazne tehnologije
- ❖ Modeli distribuiranih sistema
- ❖ Modeli usluga

Od Računara do Oblaka u 30 god.

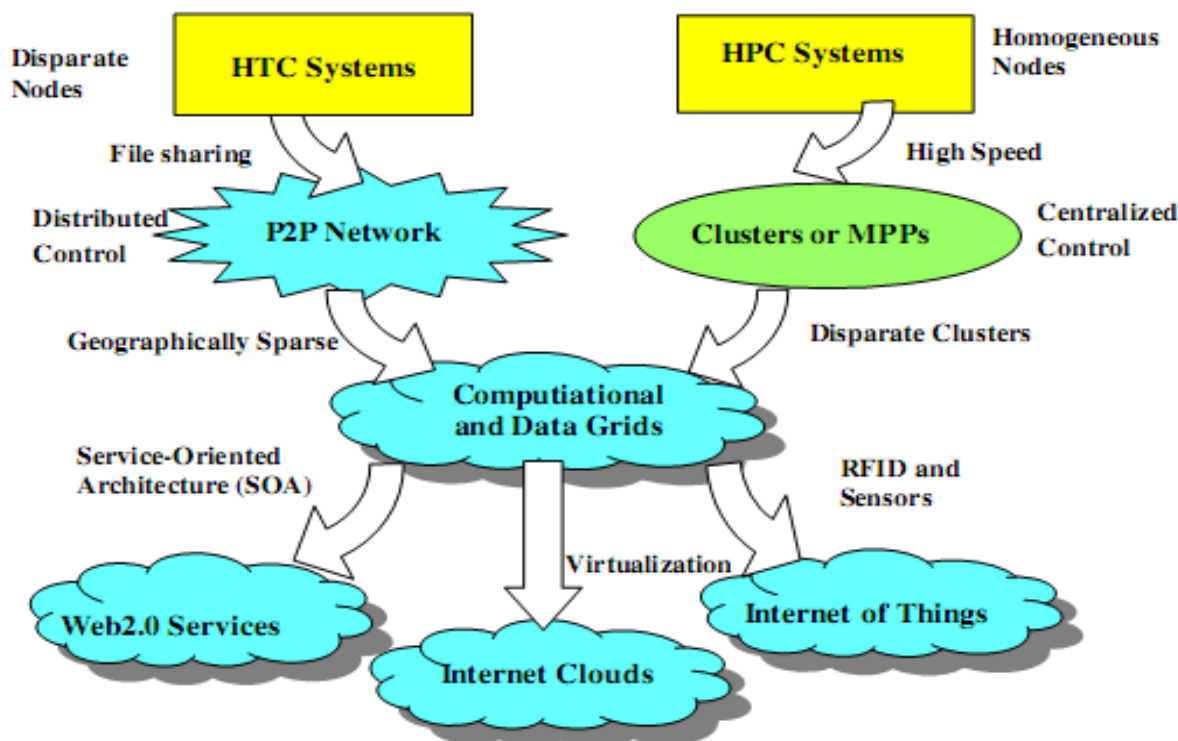
- ◆ Razvoj računanja visokih performansi (HPC)
 - HPC, eng. High Performanse Computing
 - Od centralizovanih super-računara preko geografski distribuiranih računara, klastera, i gridova, do oblaka
- ◆ Tehnologije koje su dovele do pojave oblaka
 - Klasteri, gridovi, učesničke mreže (P2P), i virtuelne mašine (VM)
- ◆ Oblaci su zasnovani na centrima podataka
 - Lociranje računarske infrastrukture u oblastima sa manjim troškovima hardvera, softvera, skupova podataka, prostora, i napajanja el. energijom

Tehnološka osnova oblaka (tj. računanja u oblaku)



- ◆ Oblak je baziran na 4 osnovne tehnologije
 - HW (HW virtuelizacija i višejezgarni procesori)
 - Distribuirano računanje (uslužno i grid računanje)
 - Rukovanje sistemom (autonomno računanje i automatizacija centara podataka)
 - Internet tehnologije (SOA, Web 2, Servisi, Mashups)

Rodoslov oblaka i internet stvari



- ◆ Koreni HTC i HPC
- ◆ HTC cilj: veliki propusni opseg (High Throughput)
- ◆ HPC cilj: velika brzina, tj. visoka performansa (High Performanse)
- ◆ P2P (Peer-to-Peer): distribuirano upravljanje
- ◆ MPP (Massively Parallel Processors): centralizovano uprav.

- ◆ Spajanjem P2P i MPP nastali Gridovi za računanja i podatke, dalje:
 - SOA dovela do Web 2 usuga
 - HW/SW virtuelizacija dovela do Internet oblaka
 - RFID i senzori doveli do Internet stvari

HPC za nauku i HTC za biznis

◆ Današnji HPC i HTC

- Nastali konvergencijom dve grupe tehnologija: paradigme računanja i aspekti sistemske arhitekture

◆ Paradigme računanja:

- Web servisi, Centri podataka, Uslužno (utility) računanje, Servisno računanje, Gid računanje, P2P računanje, računanje u oblaku

◆ Aspekti sistemske arhitekture:

- Svuda prisutnost (pouzdanost i skalabilnost)
- Autonomnost (dinamičnost i mogućnost otkrivanja)
- Mogućnost komponovanja (QoS, SLA, itd.)

Paradigme računanja (1/2)

◆ Centralizovano računanje

- Svi računarski resursi su centralizovani u jednom fizičkom sistemu.

◆ Paralelno računanje

- Svi procesori su ili čvrsto spregnuti sa centralnom deljenom memorijom ili labavo spregnuti sa distribuiranom memorijom.

Paradigme računanja (2/2)

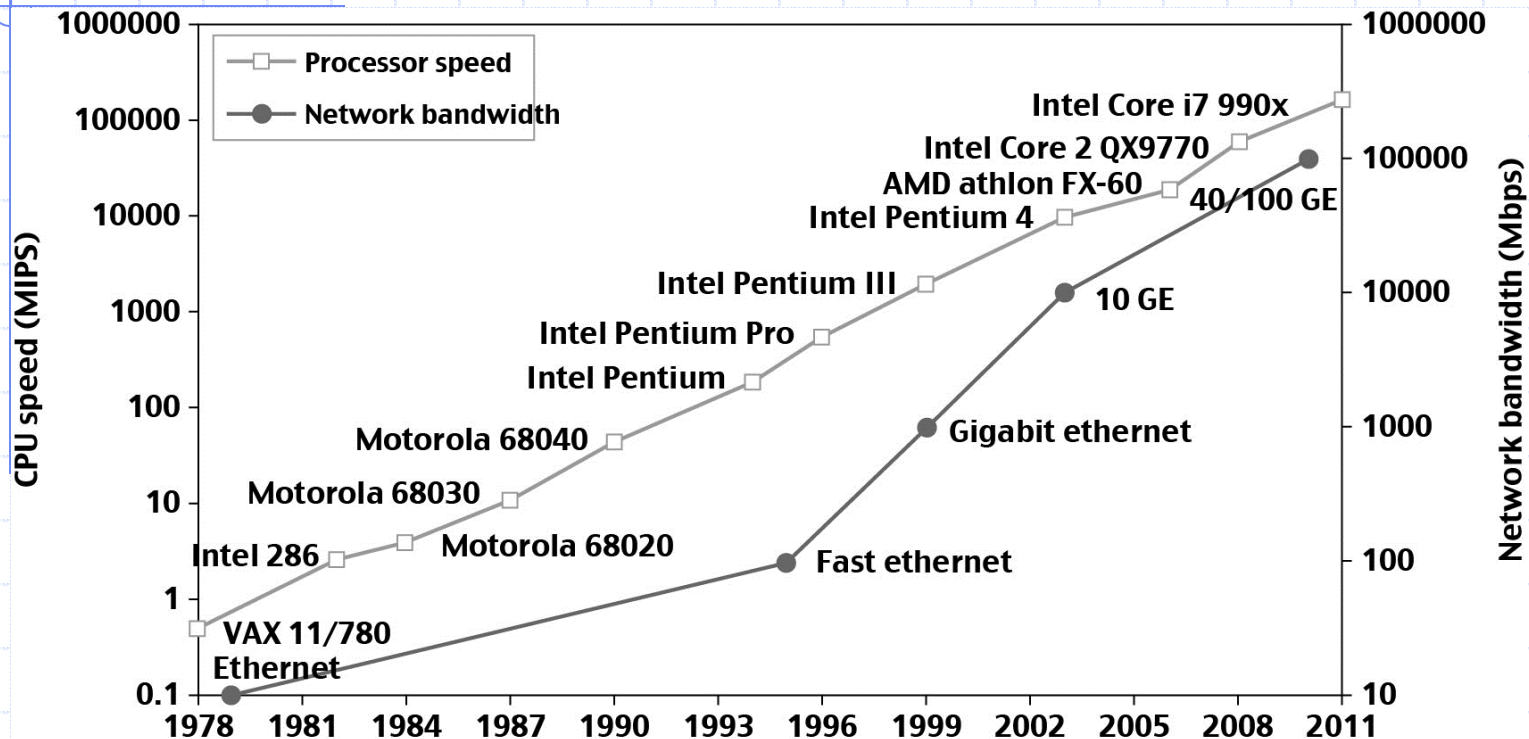
◆ Distribuirano računanje

- Distribuirani sistem se sastoji od više autonomnih računara, svaki sa svojom privatnom memorijom, i sa komunikacijom preko mreže.

◆ Računanje u oblaku

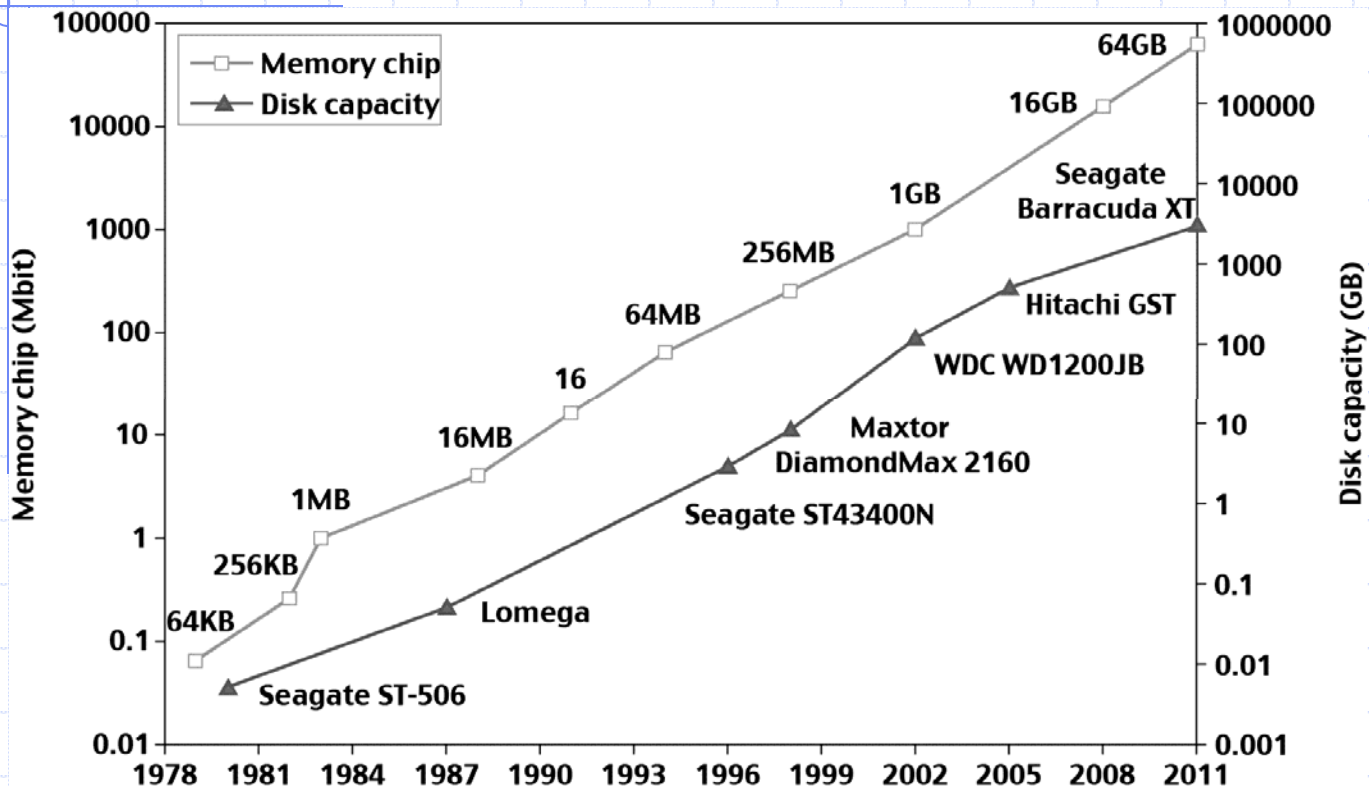
- Oblak je skup resursa (u Internetu). Može biti centralizovan ili decentralizovan. Oblak se koristi za paralelno ili distribuirano računanje ili oba. Oblaci mogu biti napravljeni od fizičkih ili virtualizovanih resursa.

Razvoj tehnologija procesora i mreža



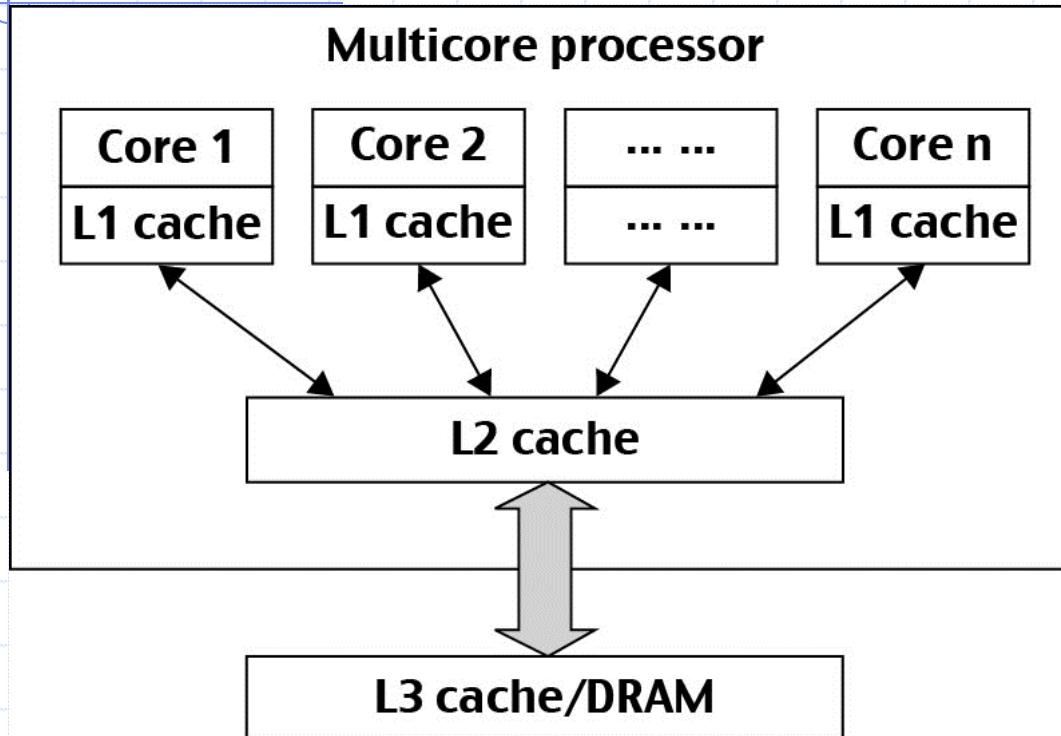
- ◆ Brzina procesora: 1 MIPS (1978) – 100 GIPS (2011)
- ◆ Mrežni propusni opseg: 10 Mb/s (1978) – 100 Gb/s (2011)

Razvoj tehnologija memorija i diskova



- ◆ Kapacitet memorijski čip: 64 KB (1980) – 64 GB (2011)
- ◆ Kapacitet diska: 0.01 GB (1980) – 10 Tb/s (2011)

Arhitektura modernog višejezgarnog CPU



- ◆ Tri nivoa skrivenih memorija
- ◆ Dva nivoa u CPU: L1 (po jezgru) i L2 (zajednički za sva jezgra)
- ◆ Nivo L3 je u operativnoj memoriji (DRAM)

Procesori sa više fizičkih niti (eng. Multi-threading) (1/2)

◆ Superskalarni (Sun Ultrasparc I)

- Implementira paralelizam na nivou instrukcija (ILP) u jednom procesoru
- Izvršava više instrukcija u taktu, slanjem tih instrukcija na redundantne funkcionalne jedinice

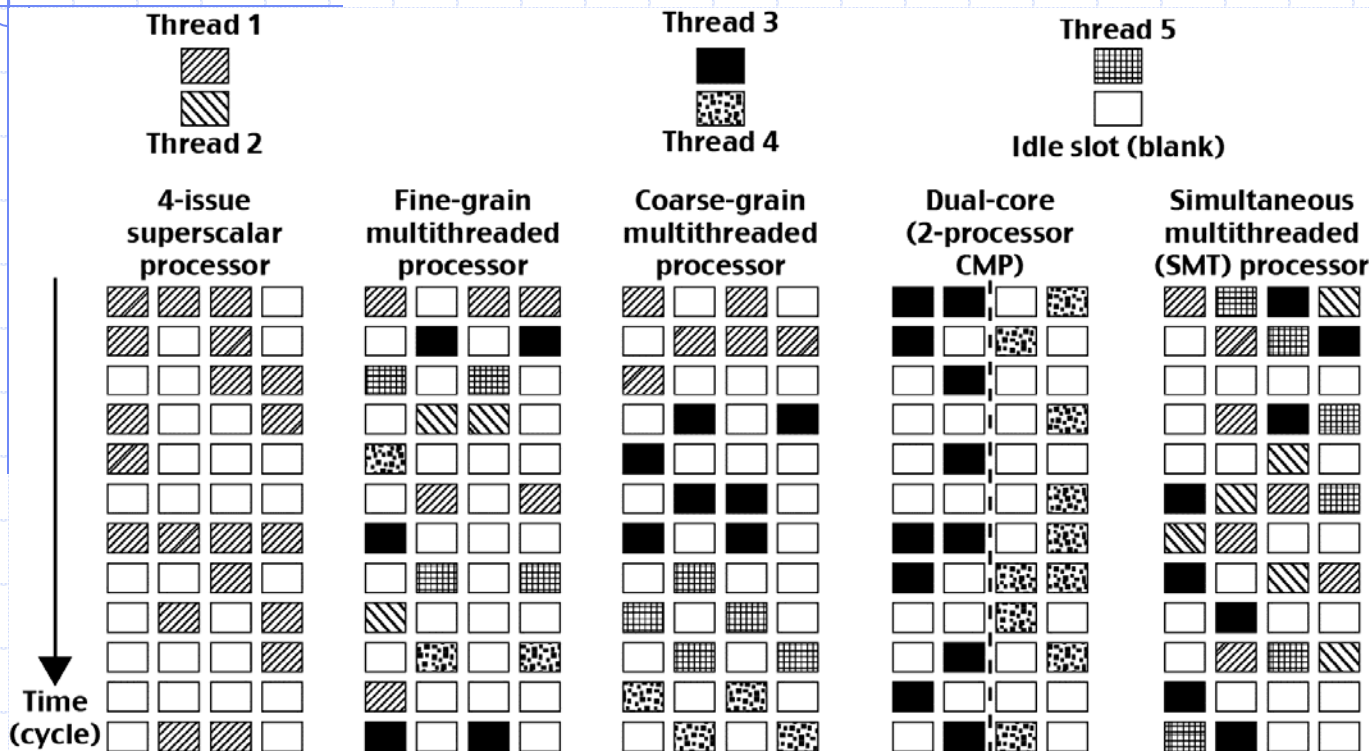
◆ Višejezgarni procesor fine granularnosti

- Prekida niti posle svakog ciklusa
- Učešljava izvršenje instrukcija
- Ako se jedna nit blokira (stall), druge nastavljaju izvršenje

Procesori sa više fizičkih niti (eng. Multi-threading) (2/2)

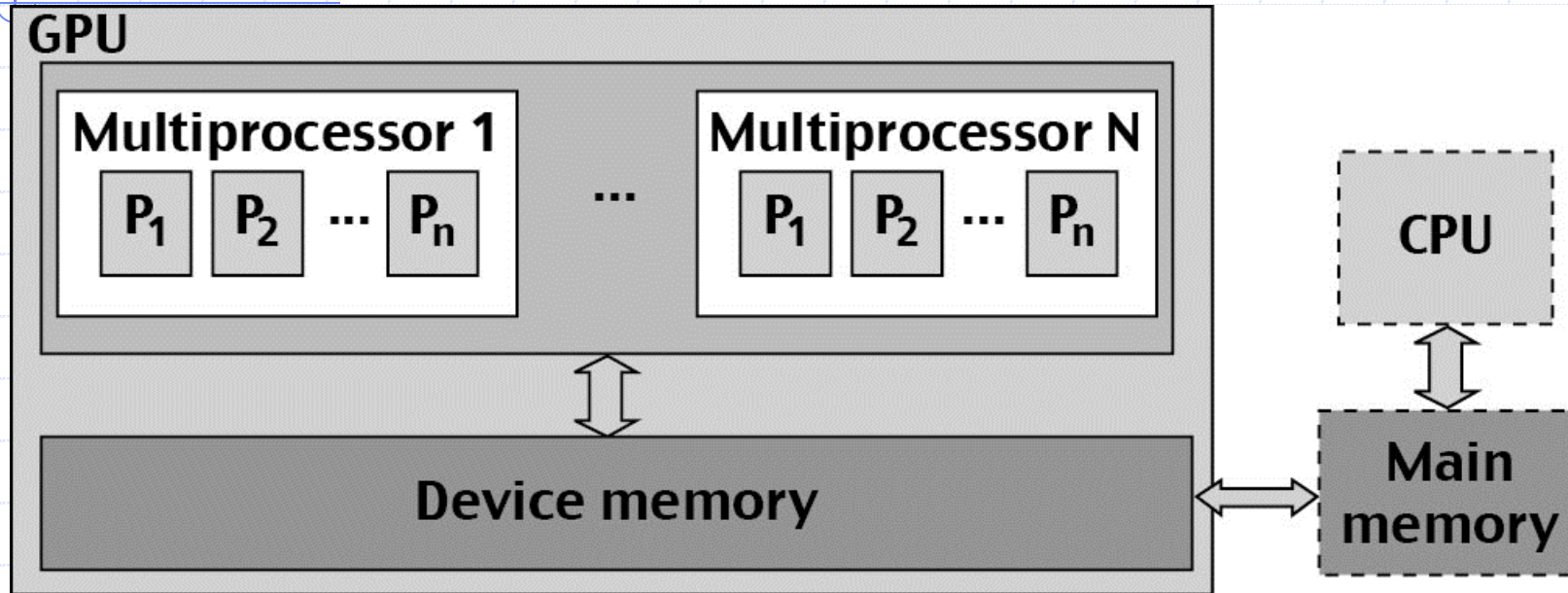
- ◆ Višejezgarni procesori grube granularnosti
 - Izvršava jednu nit dok se ne dođe do određenih situacija
- ◆ Simultani višejezgarni procesori (SMT)
 - Instrukcije iz više niti mogu da se izvršavaju u bilo kojoj zadatoj fazi protočne obrade

Pet mikro-arhitektura CPU



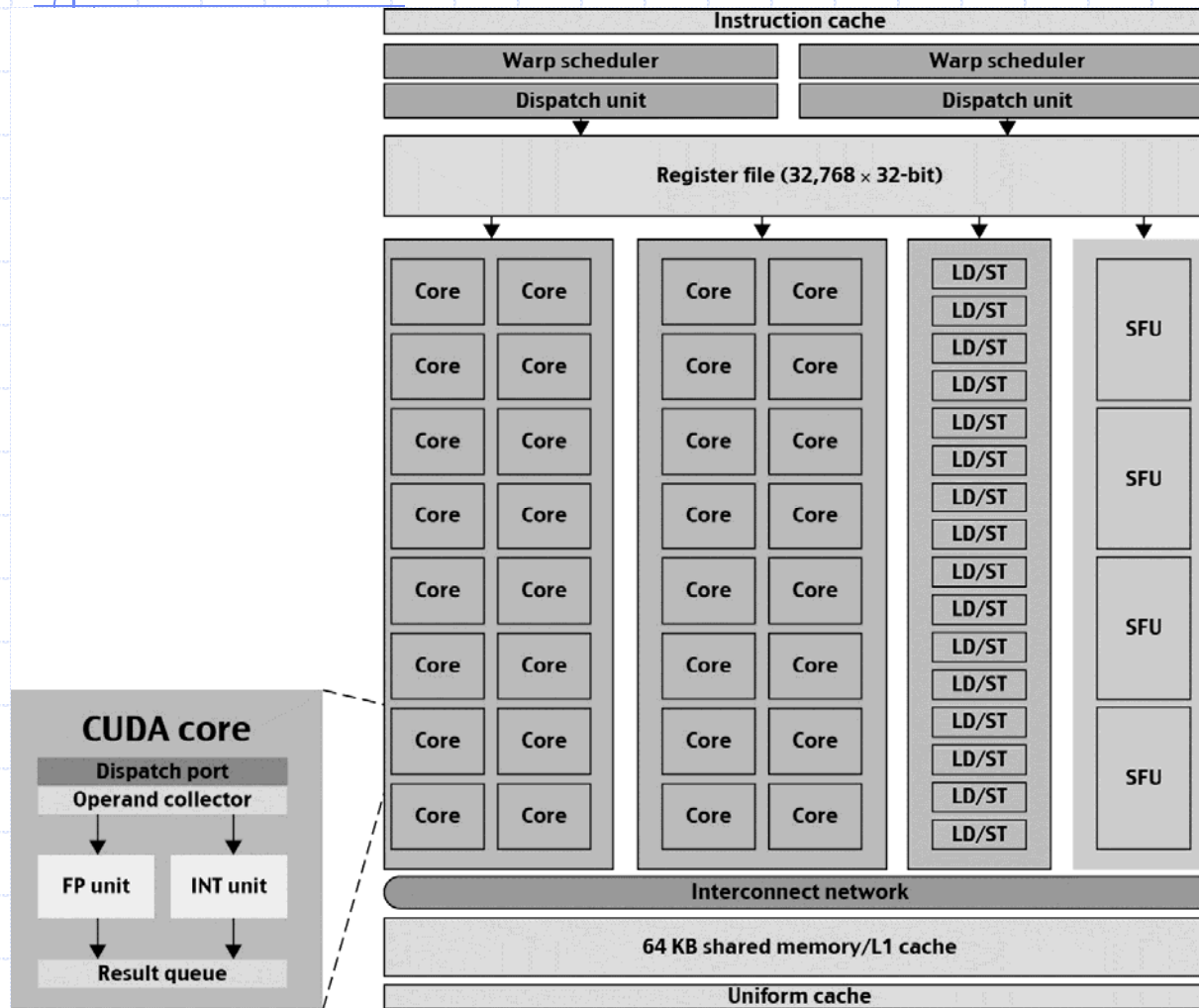
- ◆ Svaka kolona je pozicija za izdavanje inst. za jedan ciklus izvršenja:
- ◆ Popunjen simbol ukazuje da je procesor pronašao instrukciju za izvršenje u toj poziciji i u tom ciklusu
- ◆ Prazan simbol označava neiskorišćenu poziciju

Arhitektura GPU sa mnogo- jezgarnim procesorima



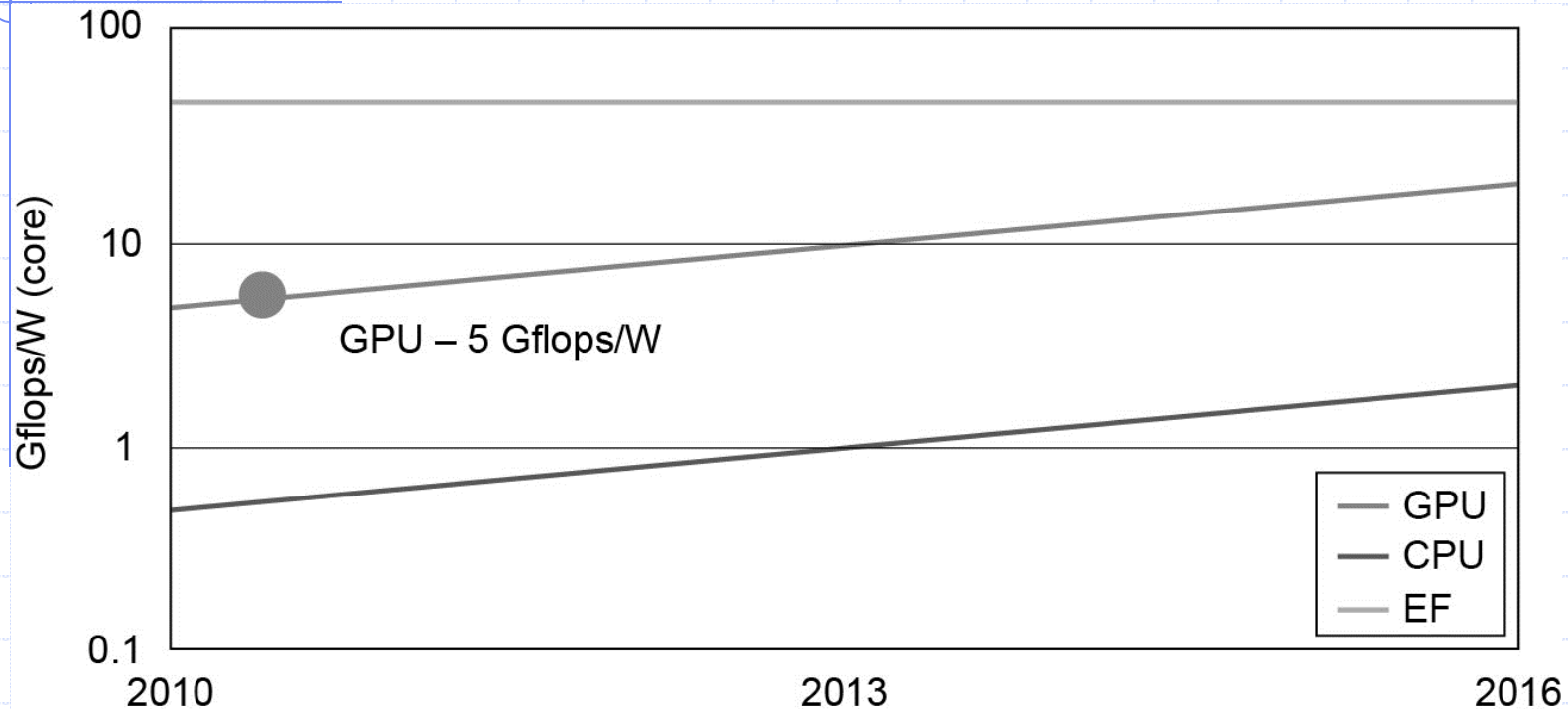
- ◆ GPU uređaj ima svoju memoriju, koja je hijerarhijski organizovana
- ◆ CPU ima svoju glavnu (operativnu) memoriju
- ◆ Obično jedan CPU upravlja sa više GPU

Primer GPU: NVIDIA Fermi GPU



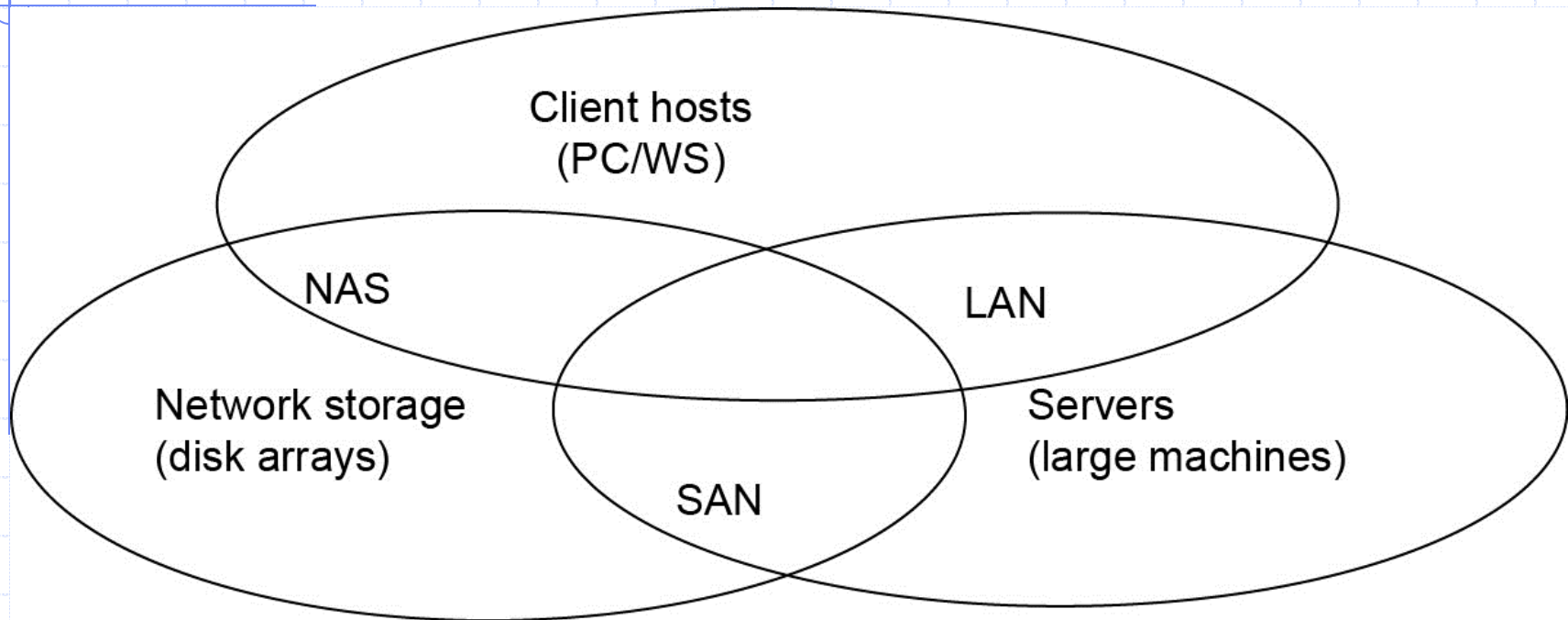
- ◆ Svako jezgro ima FP i INT jedinice
- ◆ Jezgra povezana ukrasnom mrežom dele niz registara i skrivene memorije

Poređenje performanse CPU, GPU i EF



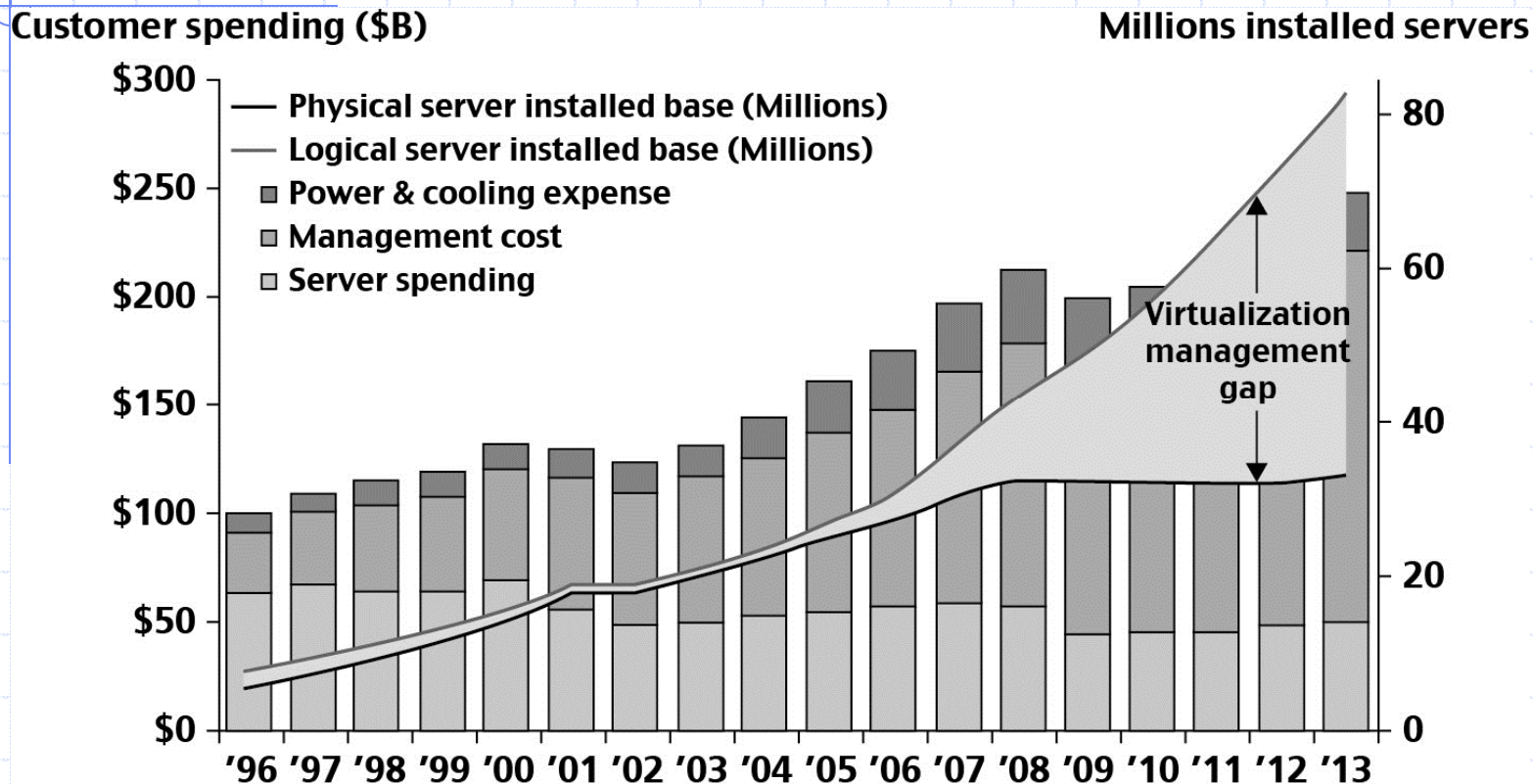
- ◆ Dno – CPU - 0.8 Gflops/W/Core (2011)
- ◆ Srednina – GPU - 5 Gflops/W/Core (2011)
- ◆ Vrh - EF – Exascale computing (10^{18} Flops)

Mreže za povezivanje



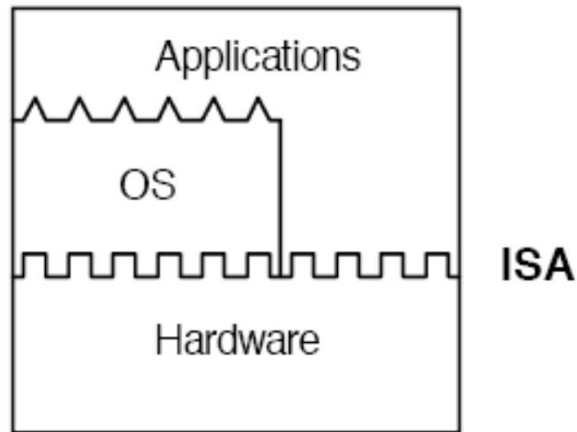
- ◆ SAN (storage area network) – povezuje servere sa nizovima diskova
- ◆ LAN (local area network) – povezuje klijente i servere
- ◆ NAS (network attached storage) – povezuje klijente sa sistemima za skladištenje velikih količina podataka

Distribucija troškova centra podataka i servera



- ◆ Uz pomoć virtuelizacije troškovi rukovanja centrima ne rastu eksponencijalno već linearno
- ◆ Rang lista trškova: 1. rukovanje, 2. serveri, 3. napajanje i hlađenje

Model sistema sa realnom mašinom



- ◆ Aplikacije pristupaju HW resursima (memorija, U-I uređaji) kroz systemske pozive OS (privilegovane instrukcije)
 - OS rukuje HW resursima
 - ISA (Instruction Set Architecture)

Prednosti i nedostaci modela sistema sa realnom mašinom

◆ Prednosti:

- Razdvojeno projektovanje OS i hardvera
- HW i OS se mogu unaprediti bez znanja aplikacionih programa

◆ Nedostaci:

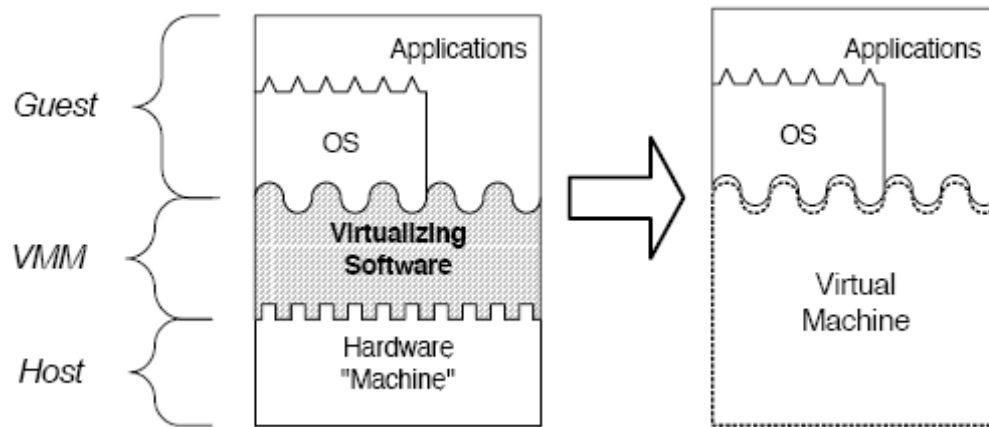
- Aplikacija kompajlirana na jednoj ISA neće raditi na drugoj ISA (npr. Mac i Windows)
- ISA mora podržavati stari softver (može negativno uticati na performansu)
- Pošto se SW razvija nezavisno od HW, SW nije obavezno optimizovan za HW

Virtuelne mašine (VM)

- ◆ Eliminiraju ograničenja realne mašine
 - Povećavaju prenosivost i prilagodivost
- ◆ VM pomoću softvera od fizičke mašine stvara pojavu druge platforme ili više platformi

- ◆ Prednosti korišćenja VM:
 - Kompatibilnost između platformi
 - Povećana zaštita
 - Poboljšana performansa
 - Pojednostavljena migracija softvera

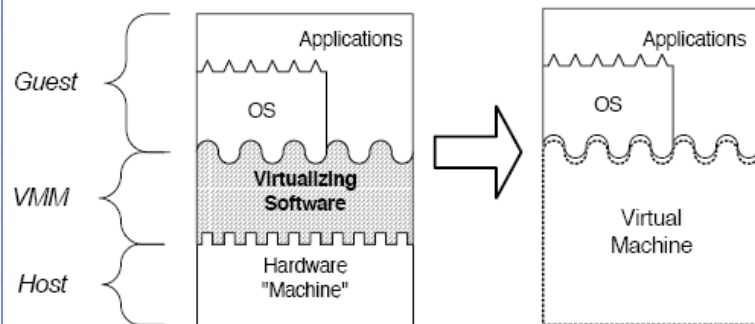
Osnove virtuelne mašine (1/2)



- ◆ Pojmovi:
- ◆ VM Monitor (VMM)
- ◆ Gost = softver (OS + aplikacije)
- ◆ Domaćin = mašina (realna ili virtuelna)

- ◆ Vitrualizujući softver (VMM) je smešten između osnovne mašine i konvencijalnog softvera
 - Konvencionalni softver vidi različitu ISA od one koju podržava hardver

Osnove virtualne mašine (2/2)

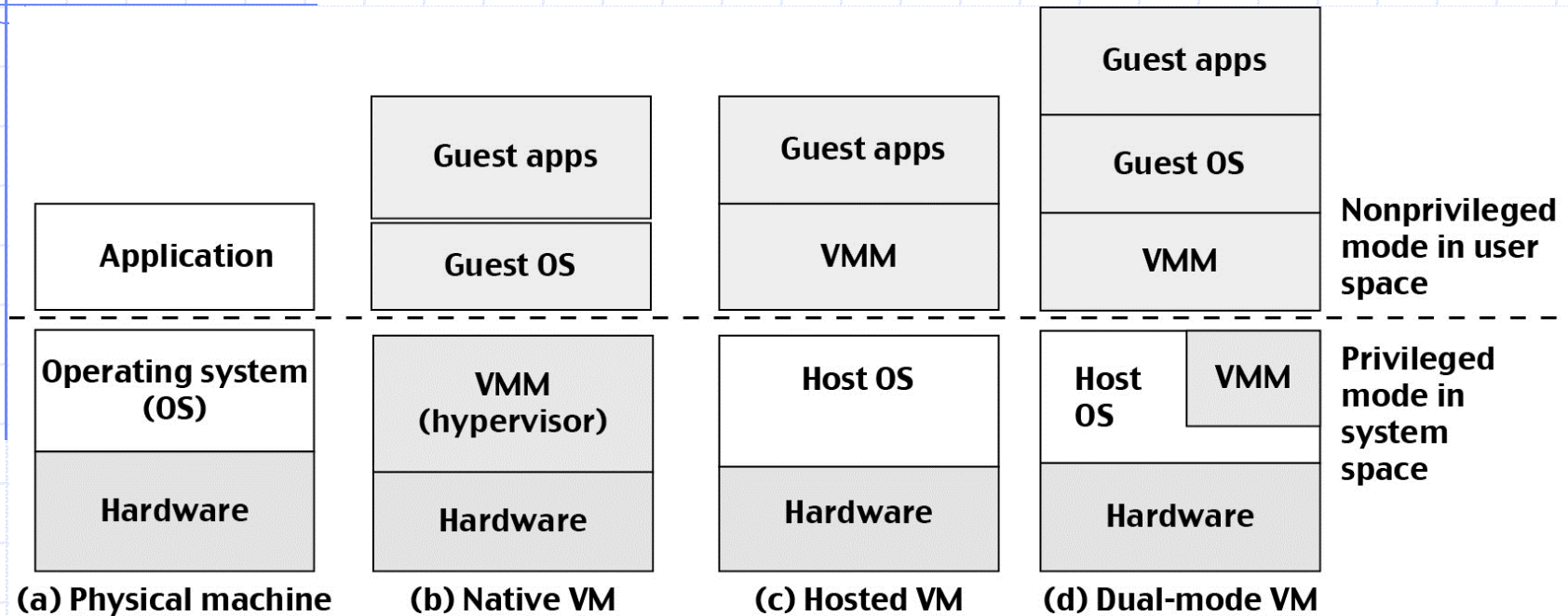


- ◆ Pojmovi:
- ◆ VM Monitor (VMM)
- ◆ Gost = softver
- ◆ Domaćin = mašina

◆ Proces virtualizacije obuhvata:

- Mapiranje virtuelnih resursa (registara i memorije) na realne fizičke resurse
- Korišćenje instrukcija realne mašine za izvođenje akcija po specifikaciji instrukcija virtualne mašine

Tri arhitekture VM



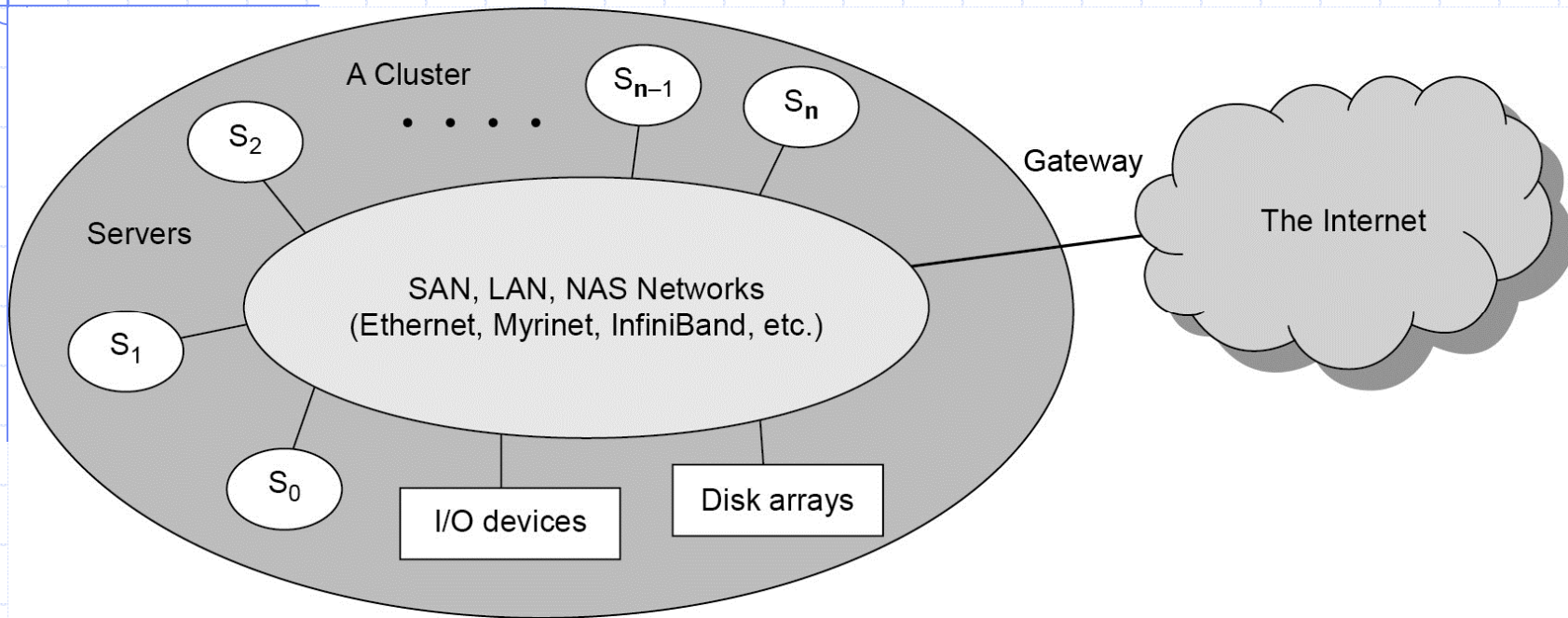
◆ Tri arhitekture VM:

- Urođena VM (Native VM)
- Gostujuća VM (Hosted VM)
- VM sa dva režima (Dual-mode VM)

Klasifikacija distribuiranih paralelnih računarskih sistema

Funkcionalnost, Aplikacije	Klasteri sa više računara	P2P mreže	Gridovi za pod. ili računanja	Platforme oblaka
Arhitektura, Umrežavanje i Veličina	Mreža računarskih čvorova povezanih sa SAN, LAN ili WAN, hijerahijski	Fleksibilna mreža klijentskih mašina povezanih prekrivačkom mrežom	Heterogeni klasteri povezani vezama velike brzine	Virtulizovani klasteri servera nad centrima podataka preko ugovora na nivou usluga
Upravljanje i Rukovanje resursima	Homogeni čvorovi sa distribuiranim upravljanjem, OS Unix ili Linux	Nezavisni čvorovi, slobodan ulaz i izlaz sa distribu. samo-organizacijom	Centralizovano upravljanje, na bazi servera, autentifikacije i statičkih resursa	Dinamičko obezbeđivanje resursa za servere, skladište i mreže nad mas. skupom pod.
Aplikacije i Mrežne usluge	HPC, pretraživači, mrežne usluge, itd.	Poslovno deljene datoteka, isporuka sadržaja, socijalne mreže	Superračunari, globalno rešavanje problema, usluge centara podataka	Bolje Web pretraživanje, uslužno računanje, i usluge račun. sa strane
Reprezentativni radni sistemi	Google search, SunBlade, IBM Road Runner, Cray XT4, itd.	Gnutella, eMule, BitTorrent, Napster, KaZaA, Skype, JXTA, .NET	TeraGrid, GriPyN, UK EGEE, D-Grid, ChinaGrid, itd.	Google App Engine, IBM Bluecloud, Amazon Web Service, MS Azure

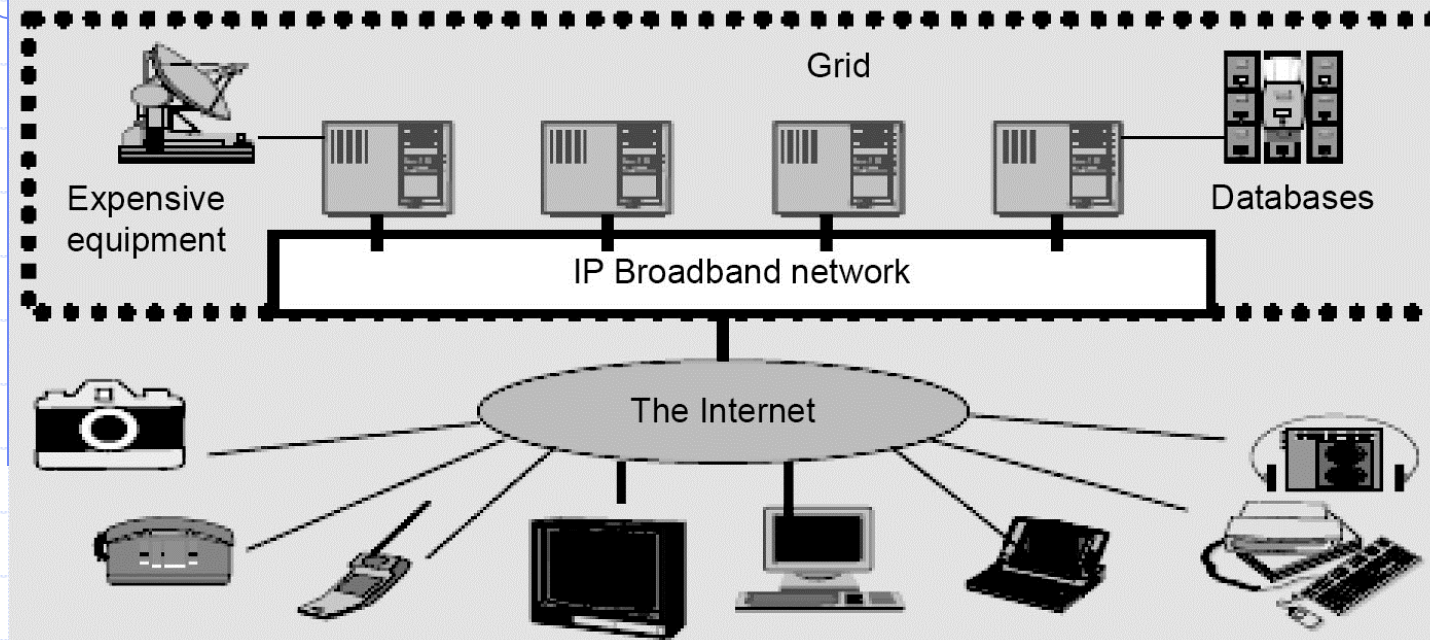
Tipična arhitektura klastera



◆ Tipična arhitektura klastera:

- Čvorovi: serveri, U/I uređaji i nizovi diskova
- Povezani SAN, LAN i NAS mrežama
- Veza preko konvertora protokola na Internet

Tipična arhitektura grida



◆ Tipična arhitektura grida:

- Čvorovi sa skupom opremom i bazama podataka
- Povezani IP mrežom širokog propusnog opsega
- Veza preko konvertora protokola na Internet sa drugim lokacijama

Mreža učesnika (1/2) (Peer-to-Peer, P2P)

◆ Jedna arhitektura distribuiranog sistema:

- Svaki računar u mreži može da operiše kao klijent ili server za druge računare u mreži
- Nema centralizovanog upravljanja
- Obično ima puno čvorova, ali nepouzdatih i heterogenih
- Čvorovi su simetrični u funkcionisanju

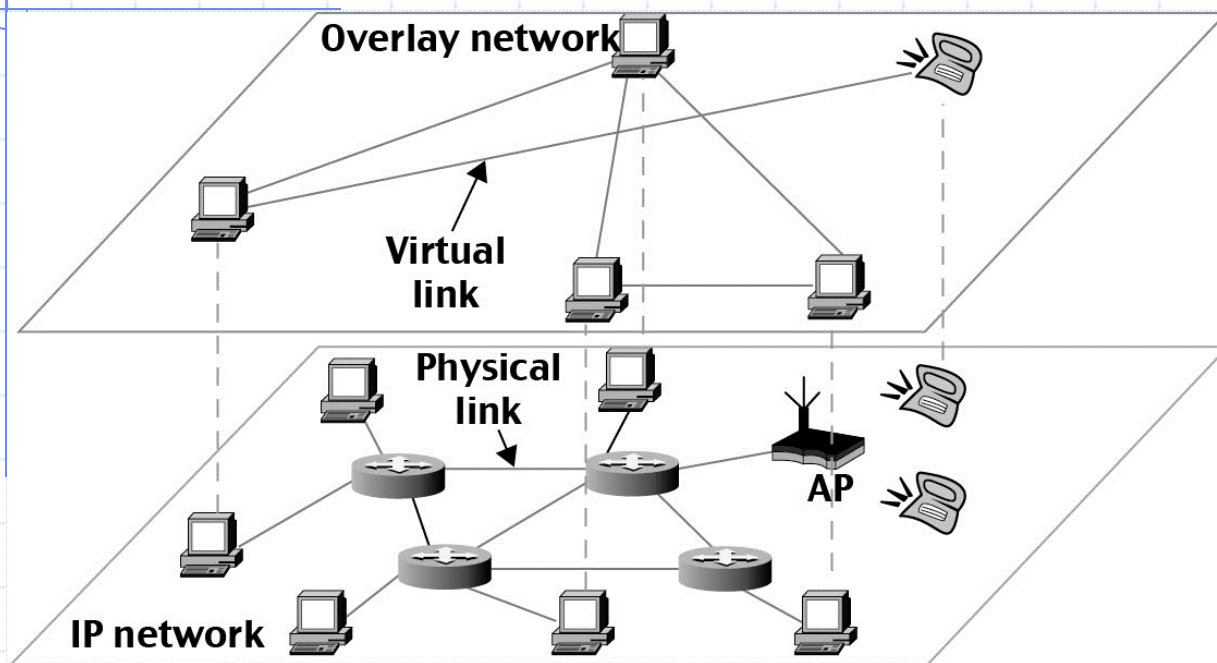
Mreža učesnika (2/2)

(Peer-to-Peer, P2P)

◆ Dobre osobine P2P mreže:

- Koristi se prednost distribuiranih, deljenih resursa na drugim čvorovima
- Otporne na otkaze, samoorganizujuće
- Operišu u dinamičkom okruženju, često pridruživanje mreži i napuštanje mreže su uobičajeni

P2P mreža: Računarska mreža izgrađena na vrhu druge mreže



- ◆ Čvorovi u prekrivačkoj mreži su povezani pomoću virtuelnih ili logičkih veza (links), od kojih svaka odgovara putanji, možda kroz mnogo fizičkih veza, u osnovnoj mreži
- ◆ Npr., distribuirani sistemi kao što su oblaci za računanja, mreže učesnika, i klijent-server aplikacije su prekrivačke mreže jer njihovi čvorovi rade na vrhu interneta

Glavne vrste P2P mrežnih familija

Osobina sistema	Distribuirano deljenje datoteka	Kolaborativna platforma	Distribuirano P2P računanje	P2P platforma
Atraktivne aplikacije	Distribucija sadržaja, MP3 muzika, video, open SW	Instant poruke, kolaborativno projektovanje i igre	Naučna istraživanja i socijalne mreže	Otvorene mreže za javne resurse
Operativni problemi	Gubitak zaštite i narušavanje prava zaštite od kopiranja	Nedostatak poverenja, spam, privatnost, zavere učesnika	Rupe u zaštiti, sebični partneri, zavere učesnika	Nedostatak standarda ili protokola zaštite
Primeri sistema	Gnutella, Napster, eMule, BitTorrent, Amister, KaZaA	ICQ, AIM, Groove, Mlagi, igre sa više učesnika, Skype	SETI@home, Genome@home, itd.	JXTA, .NET, FightingAid@home itd.

Oblak

- ◆ Istorijski koreni u savremenim IP aplikacijama

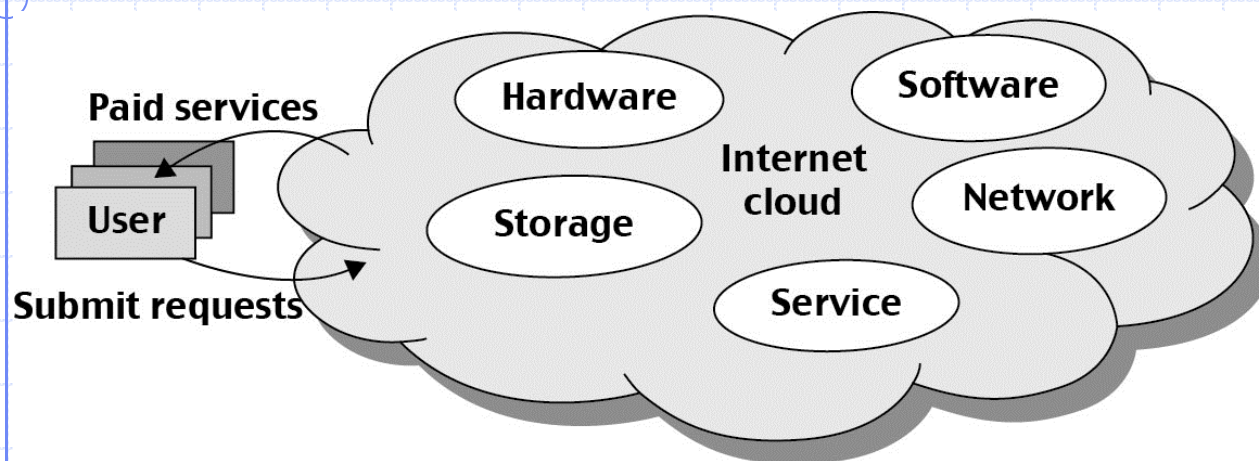
- Pretrage, email, socijalne mreže



- ◆ Neke definicije:

- Okruženje za rukovanje skalabilnim, pouzdanim, na-zahtev dostupnim aplikacijama
- Model računanja i skladišta podataka zasnovan na pristupu plaćanja prema korišćenju resursa udaljenih centara podataka

Koncept internet oblaka



- ◆ Računanje u oblaku je korišćenje računarskih resursa (HW i SW) koje se isporučuje kao usluga preko mreže
- ◆ Koren imena je simbol u obliku oblaka kao apstrakcija za složenu infrastrukturu u dijagramima sistema
- ◆ Računanju u oblaku se poveravaju udaljene usluge sa korisničkim podacima, softverom i računanjima

Sledeća revolucija u IT: Računanje u oblaku

◆ Klasično računanje:

- Kupi i poseduj HW i SW
- Instaliraj, konfiguriši, testiraj, verifikuj, evaluiraj
- Rukuj, održavaj
- Konačno koristi

◆ Računanje u oblaku:

- Predplati se
- Koristi
- Plati ono što koristiš, prema kvalitetu servisa (QoS)



Modeli usluga oblaka

- ◆ Tri osnova modela usluga oblaka, od jednostavnijih ka složenijim:
 - Infrastruktura kao usluga (IaaS)
 - Platforma kao usluga (PaaS)
 - Softver kao usluga (SaaS)
- ◆ Više o ovim modelima kasnije u toku kursa

Servisno Orijentisana Arhitektura (SOA)

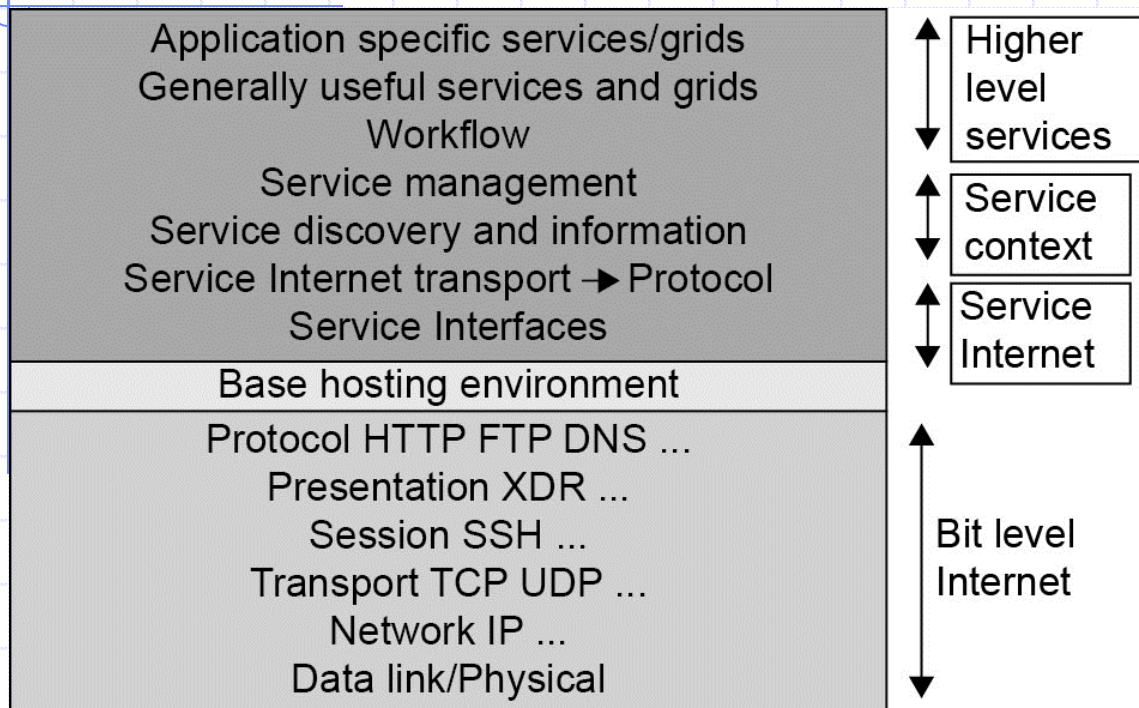
- ◆ Nastala evolucijom distribuiranog računanja
 - na osnovu projektantske paradigme zahtev/odgovor za sinhronu i asinhronu aplikaciju
- ◆ Poslovna logika aplikacije ili pojedinih funkcija se modularizuje i predstavlja kao usluga za aplik.
 - Sprema usluge je nezavisna od njene implementacije
 - Aplikacije se grade komponovanjem usluga bez poznavanja pozadinskih implementacija tih usluga
 - Npr., usluga može biti implementirana u .Net ili J2EE, dok aplikacija može biti na drugoj platformi ili jeziku

Ključne karakteristike SOA

◆ Četiri ključne osobine:

- SOA usluge imaju samo-opisujuće sprege u XML dok. nezavisnom od platforme (WSDL je standard za to)
- SOA usluge komuniciraju preko poruka formalno definisanim pomoću XML šema (XSD)
- SOA usluge se održavaju u preduzeću u registru koji funkcioniše kao direktorijum (UDDI standard)
- Svaka SOA usluga ima neki kvalitet posluživanja (QoS), koji joj je pridružen

Slojevita arhitektura Web servisa



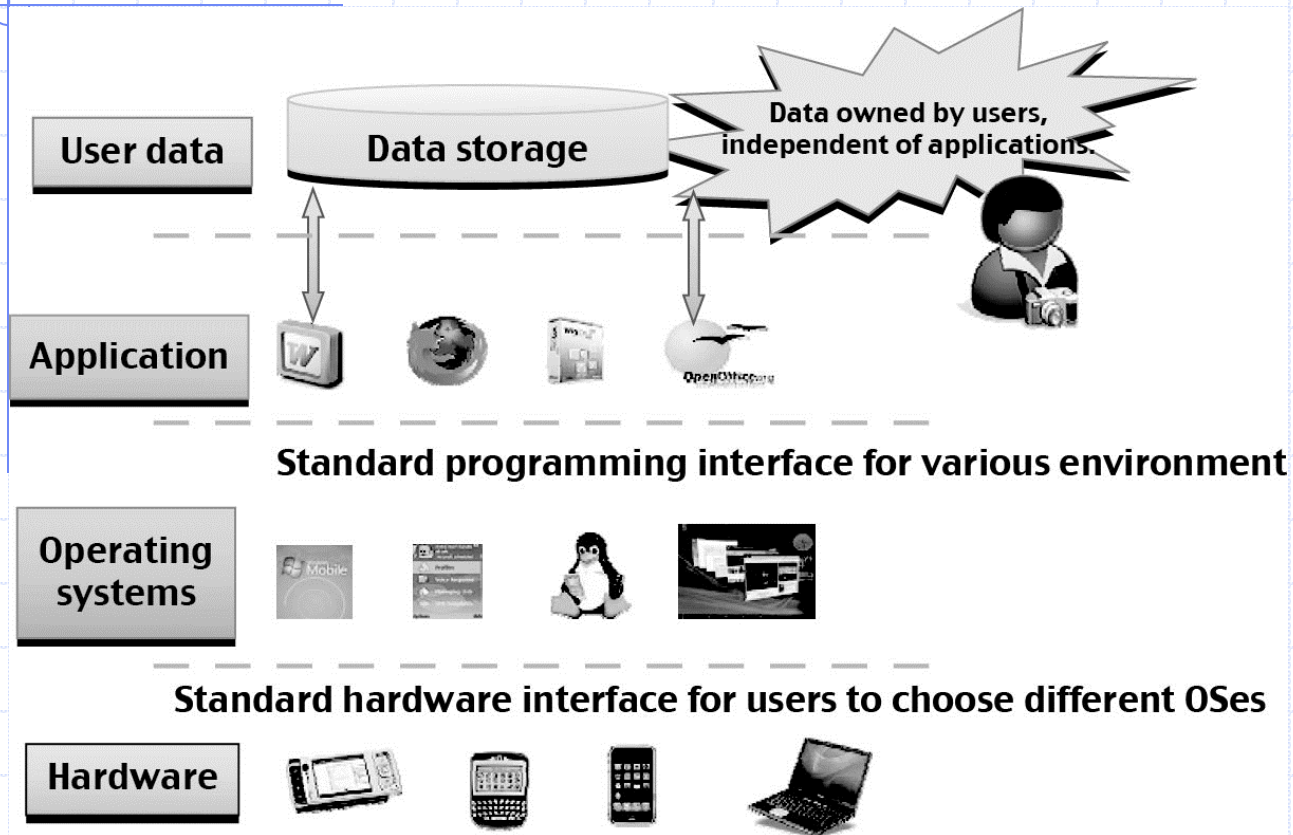
◆ Tri nivoa servisa:

- Viši nivo servisa
- Kontekst servisa
- Servis Interneta

Poređenje osobina tri distribuirana OS

Osobina distribuiranog operativnog sistema	AMOEBA razvijena na Univerzitetu Vrije	DCE kao OSF/1 od Open Software Foundation	MOSIX za Linux klastere na Hebrew Univerzitetu
Istorija i tekuće stanje	Napisan u C i testiran u EU; verzija 5 puštena 1995	Korisničko proširenje na vrhu UNIX, VMS, Windows, OS/2, itd.	Od 1977, sad MOSIX2. Koristi se u HPC Linux i GPU klasterima
Arhitektura distribuiranog OS	Zasnovan na mikro jezgri, koristi mnoge servere: datoteke, replikacija, izvršenje, boot i TCP/IP usluge	To je midlver OS za distribuirane aplikacije. Podržava RPC, zaštitu i niti (threads)	Distribuirani OS sa otkrivanjem resursa, migracijom procesa, balansiranjem opte., kontrolom plavljenja, konfiguracijom, itd.
Jezgro OS, midlver i podrška virtuelizaciji	Mikro jezgro rukuje procesima niskog nivoa, memorijom, U-I, mrežnom komunikacijom	DCE paketi rukuju datotekama, vremenom, zaštitom, RPC i autentifikacijom u prostoru korisnika ili midlvera	MOSIX2 radi sa Linux 2.6; proširenja za više klastera i oblaka sa obezbeđenim VM
Komunikacioni mehanizmi	Koristi FLIP protokol mrežnog nivoa i RPC za P2P i grupnu komunikaciju	RPC podržava autentifikovanu komunikaciju i druge usluge zaštite u korisničkim programima	Koristi PVM i MPI za kolektivne komunikacije, kontrolu prioriteta i usluge redova čekanja ³⁹

Transparentno okruženje za računanje u oblaku



- ◆ Računanje u oblaku razdvaja :
- korisničke podatke
 - aplikacije
 - OS
 - HW

Modeli programiranja za paralelno i distribuirano programiranje

Model	Opis	Osobine
MPI	Biblioteka podprograma, koja se može pozvati iz C ili Fortrana, za pisanje paralelnih programa na distribuiranim računarskim sistemima	Specificira sinhronu, ili asinhronu, P2P i kolektivne komunikacione komande i U-I operacije, na bazi sistema sa slanjem poruka (message-passing)
MapReduce	Mrežni (Web) programski model za skalabilnu obradu podataka na velikim klasterima sa velikim skupovima podataka ili u mrežnim pretragama	Funkcija Map kao međurezultat generiše skup parova ključ/vrednost; Funkcija Reduce spaja sve vrednosti sa istim ključem
Hadoop	Programska biblioteka za pisanje velikih korisničkih aplikacija na ogromnim skupovima podataka u poslovnim aplikacijama (http://hadoop.apache.org)	Skalabilan, ekonomičan, efikasan i pouzdan alat za jednostavan korisnički pristup komercijalnim klasterima

Grid standardi i alati za naučne i inženjerske aplikacije

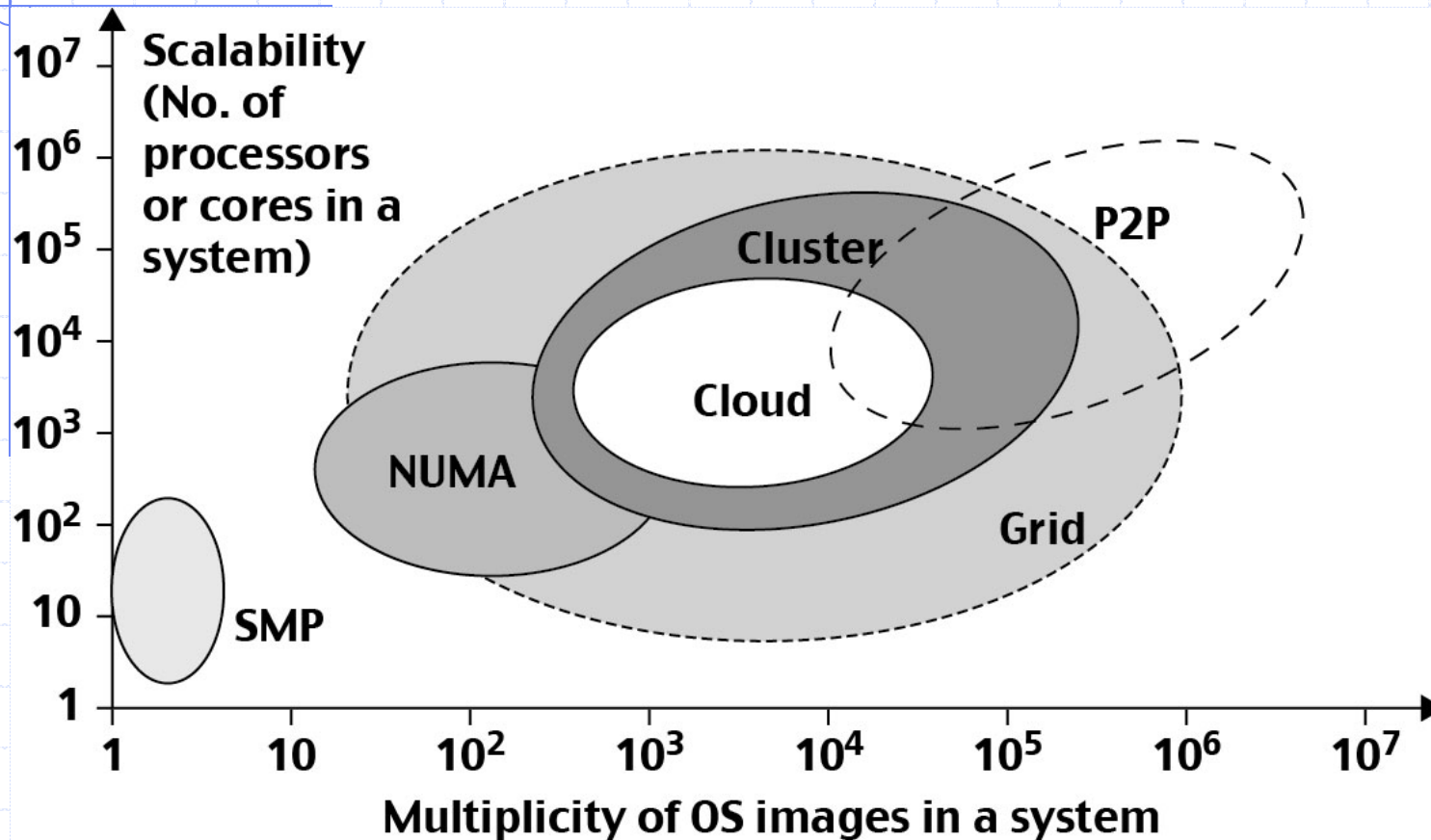
Grid standard	Glavne funkcionalnosti Grid usluga	Ključne osobine i infrastruktura za zaštitu
OGSA standard	OGSA (Open Grid Service Architecture) nudi standardne Grid usluge za opštu javnu upotrebu	Podržava heterogena distribuirana okruženja, premošćavanje CA (Certificate Aut.), više poverljivih posrednika, dinamičke politike, višestruke zaštitne mehanizme, itd.
Globus Toolkits	Dodela resursa, GSI (Globus Security Infrastructure) i generički API za usluge zaštite	Prijavljivanje na više sajtova sa autentifikacijom pomoću PKI, Kerberos, SSL, Proxy, delegiranja, i GSS API za integritet i poverljivost poruka
IBM Grid Toolbox	AIX i Linux Gridovi izgrađeni na vrhu Globus Toolkit, autonomno računanje, usluge repliciranja	Jednostavna CA, kontrola za davanje pristupa, Grid usluge (ReGS), podržava Grid aplikacije u Java (GAF4J), GridMap u IntraGrid za ažuriranje zaštite

Dimenzije skalabilnosti

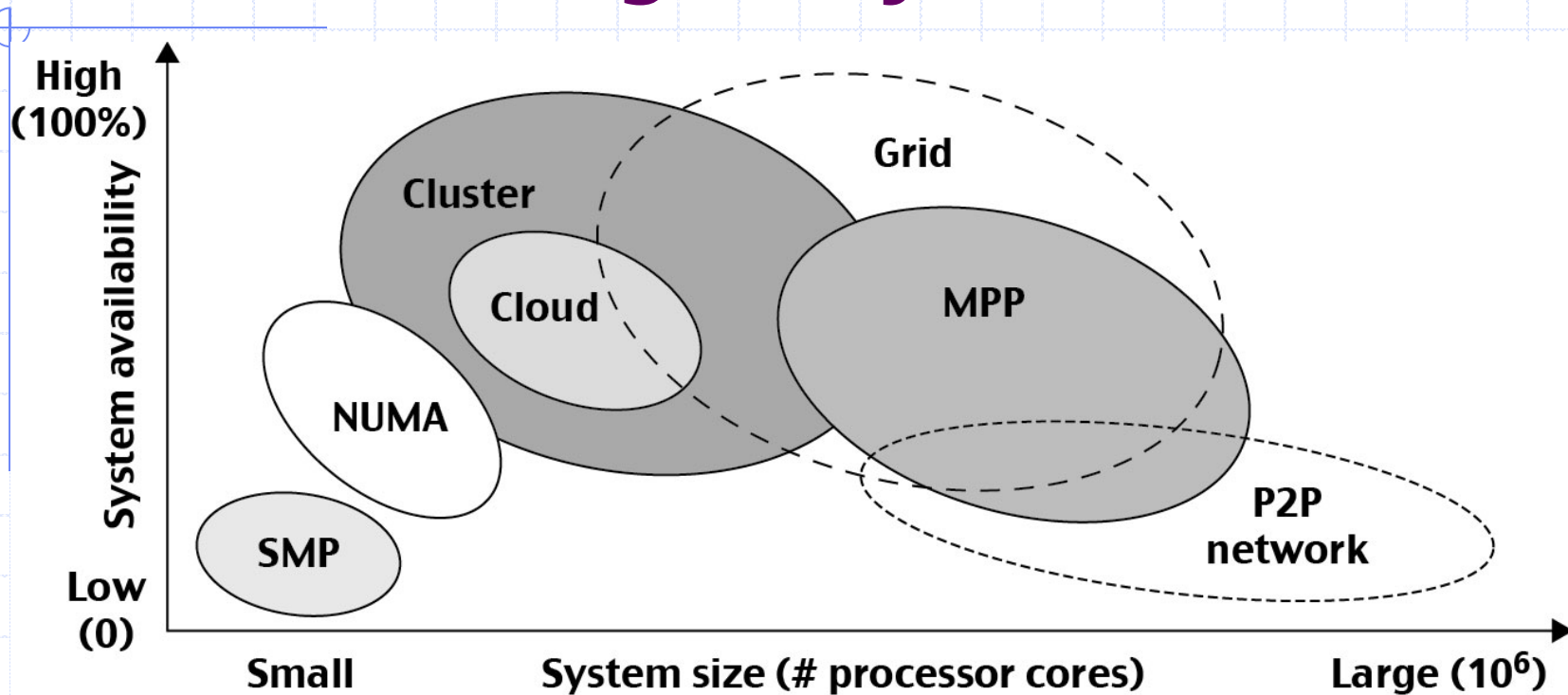
◆ Četiri dimenzije skalabilnosti:

- Veličina – povećanje performanse sa povećanjem veličine mašine
- Softver – unapređenja OS, biblioteka, nove aplikacije
- Aplikacija – poklapanje veličine problema sa veličinom mašine
- Tehnologija – prilagođavanje sistema novim tehnologijama

Skalabilnost sistema naspram Multipliciranja OS



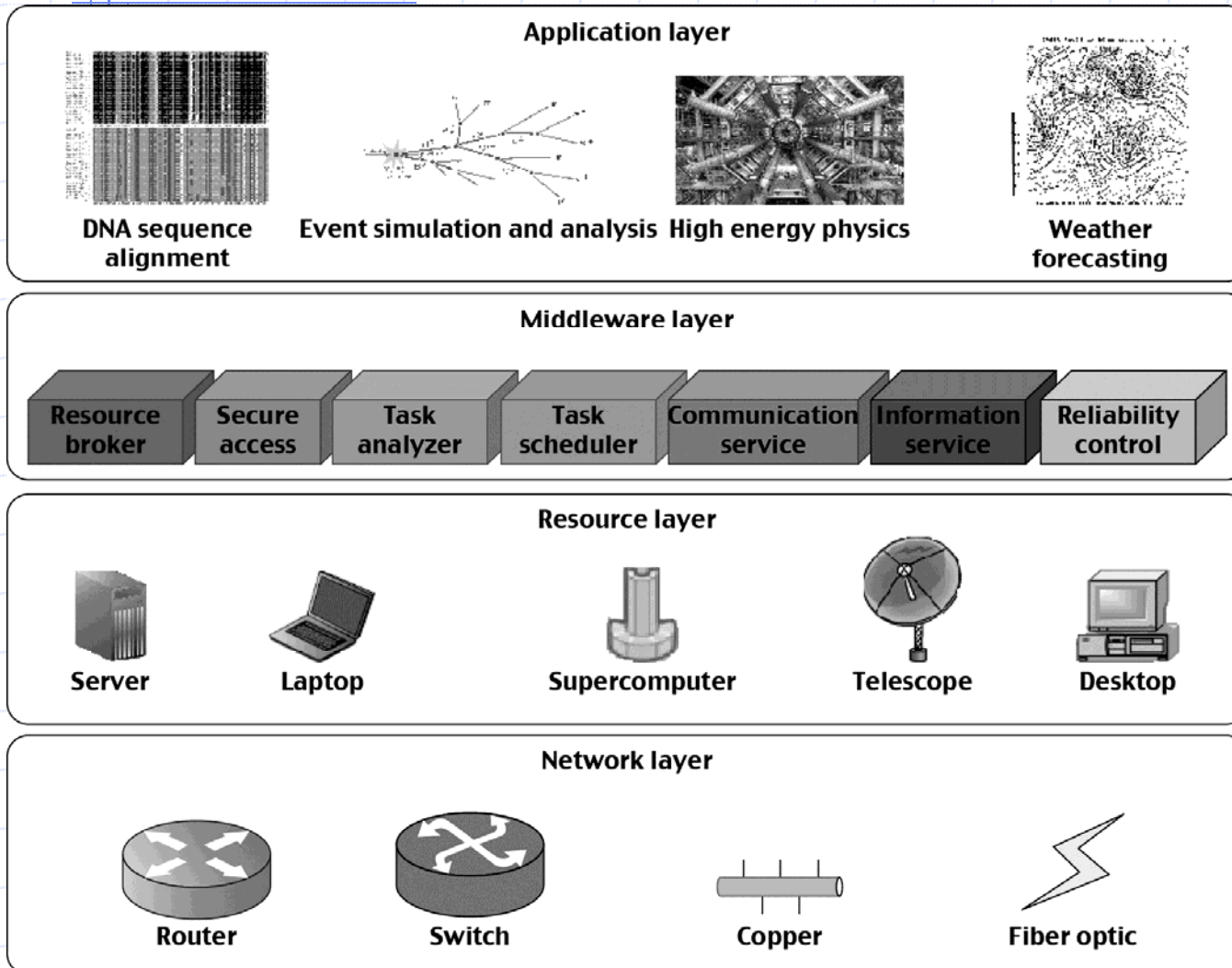
Raspoloživost sistema spram Veličine konfiguracije



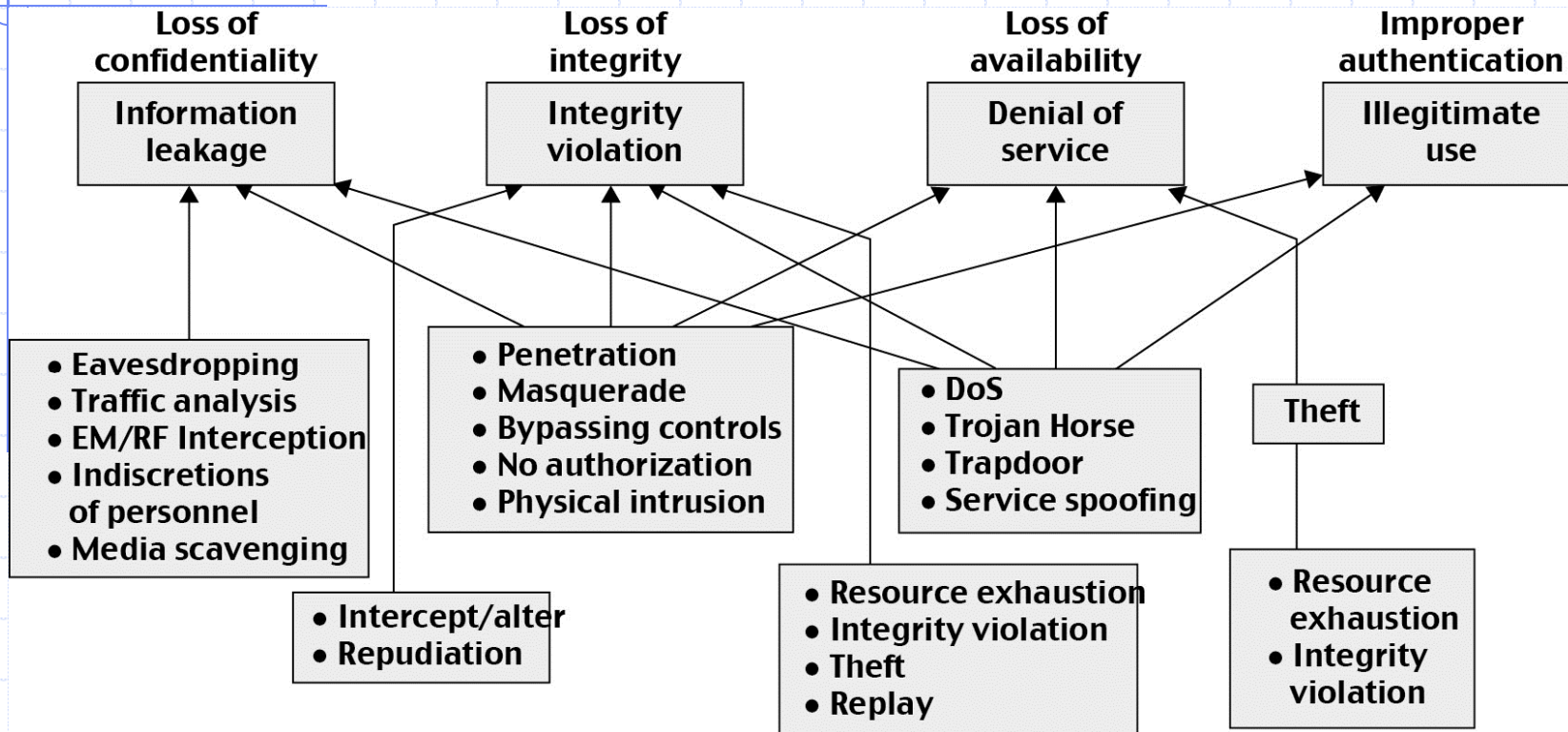
Slojevi distribuiranog sistema i komponente Midvera

◆ 7 komponenti Midvera:

- Broker resursa
- Zaštita pristupa
- Analizator zadataka
- Raspoređivač zadataka
- Komunikacioni servisi
- Informacioni servisi
- Kontrola pouzdanosti



Zaštita: Napadi na sistem i Pretnje za mrežu



- ◆ Na vrhu su prikazane 4 vrste pretnji za mrežu
- ◆ U donjem delu je prikazano 6 vrsta napada na sistem