UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA AUTOMATIKU I UPRAVLJANJE SISTEMIMA

Distribuiran sistem

Distribuirani sistemi

Distribuirano programiranje

Distribuiran sistem (DS) - definicija

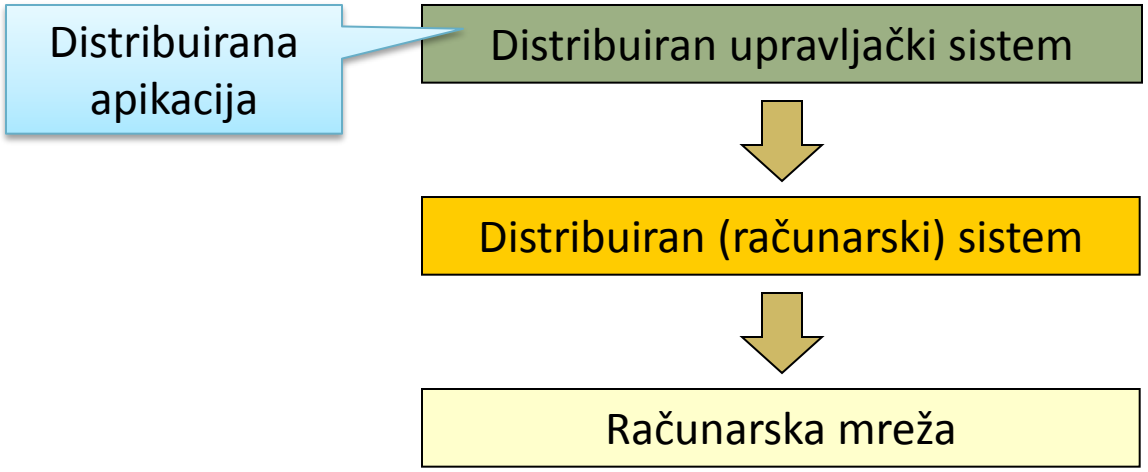
- "**DS je mnoštvo povezanih računara koje korisnik doživljava kao jedan skladan sistem**"
  - sastavljen je od hardvera – **mnoštva računara** povezanih **komunikacionom mrežom**
  - radi kao jedan sistem zahvaljujući **softveru**

- DS integriše razne aplikacije koje se izvršavaju na različitim računarima u **jedan sistem**

- DS je u suprotnosti sa centralizovanim sistemom

Primeri aplikacija u DS

- Sistem (mobilne) telefonije
- GPS sistem sa brojnim primenama
- *World Wide Web* model distribuiranih dokumenata
- Sistem elektronskog plaćanja
- Računarsko poslovanje velike kompanije
- Nadzorno-upravljački sistem u fabrici (SCADA)
- ...



Razlozi pojave DS

- Dramatično brz razvoj računara tokom poslednjih 50 godina
  - od: 100 M$ mašina za 1 IPS (instrukcija/sec)
  - do: 1 k$ mašina za 10 MIPS
  - poboljšanje 10^{12} puta

- Brze računarske mreže
  - *Local Area Networks* (LAN) – od 100 Mbps do 10 Gbps
  - *Wide Area Networks* (WAN) – od 64 Kbps do 1 Gbps

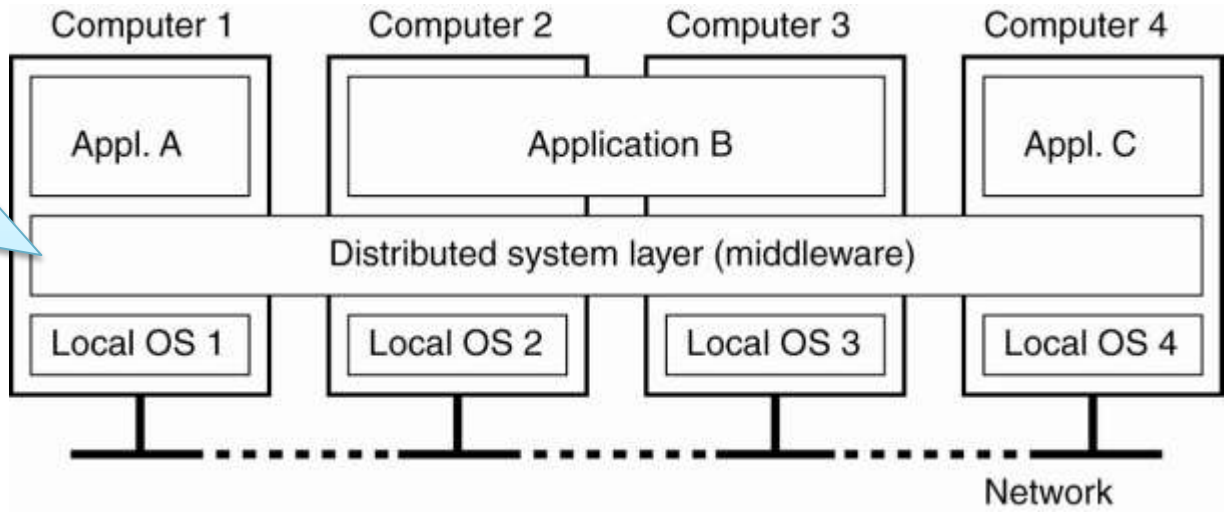
- Neophodnost distribuirane obrade informacija

- Složene aplikacije
  - nastale na osnovu ospežnih zahteva korisnika

Osobine “dobrog” DS

- Razlike među računarima i način komunicije sakriven od korisnika
- Korisnici i aplikacije interaguju sa DS na konzistentan i jednobrazan način
  - bez obzira na mesto i vreme interakcije
- Lako se proširuje
- Podržava heterogene računare i mreže
  - ima softver slojevito organizovan

Middleware sloj se prostire u više mašina i svim aplikacijama nudi isti interfejs.



Loše osobine DS

- U odnosu na centralizovan sistem
  - Softver je veoma složen
  - Umanjene su performanse zadataka koji se mogu obaviti u jednom računaru (zbog trajanja komunikacije)
  - Smanjena je sigurnost (bezbednost) sistema

- Teorijski zahtevi koji se postavljaju pred DS se ne mogu u potpunosti realizovati

Ciljevi DS

1. Povezivanje korisnika i udaljenih resursa
2. Transparentnost (distribuiranosti)
3. Otvorenost
4. Skalabilnost

Cilj 1 - Povezivanje

- Povezivanje korisnika i udaljenih resursa
  - **Deljenje uređaja**: štampača, skenera, diska, ...
  - **Razmena datoteka**: dokumenti, email, audio/video zapis, ...
  - **Rad na daljinu**: telekonferencije, elektronska trgovina, ...
- Bezbednost sistema je veoma važna

Cilj 2 - Transparentnost

- DS je transparentan kada ga korisnici i aplikacije doživljavaju kao JEDAN računarski sistem
- Tipovi transparentnosti prema:
  - **Pristupu** – sakriva razlike u reprezentaciji podataka
    - Npr. *Little-big endian* format brojeva
  - **Lokaciji** – korisnik ne zna gde se resurs fizički nalazi
    - Npr. http://ccd.ns.ac.yu/aus
  - **Migraciji** – resursi se mogu premeštati bez uticaja na korisnike
  - **Relokaciji** –odnosi se na premeštanje resursa tokom upotrebe
  - **Replikaciji** – sakriva postojanje kopija resursa
    - Radi povećanja raspoloživosti ili performansi
  - **Konkurentnosti** – prividno jednovremena upotreba deljenih resursa
    - Resurs se mora ostaviti u konzistentnom stanju
  - **Otkazima** – DS se neprimetno oporavi od nepravilnog rada resursa
    - Tipično u sistemu ne sme postojati *single point of failure*
  - **Perzistenciji** – sakriva se gde je resurs pohranjen (u RAM-u ili na disku)

Stepeni transparentnosti

- Mada je svaki tip transparentnosti poželjan ima situacija gde nije dobro sakrivati aspekte distribuiranosti od klijenata
- Upotreba DS mora uzeti u obzir realnost
  - Zahtev da elektronske novine stignu u email sanduče u 7 ujutru (da bi ih čitali tokom doručka) nema smisla kada smo u vremenskoj zoni daleko od mesta “štampanja”
  - Telefonski razgovor preko satelitskog linka ima primetno kašnjenje
  - Upravljanje preko Interneta? (promenljivo kašnjenje + nepouzdano)
  - ...
- Postoji balans između visoke transparentnosti i brzine rada
  - Propagacija kopija podataka može da potraje tako da produžava poziv koji je inicirao promenu podatka.
  - Mnoge Internet aplikacije predugo pokušavaju da uspostave vezu sa udaljenim serverom pre nego se obrate drugom serveru.
  - ...

Cilj 3 - Otvorenost

- Otvoren DS pruža servise (usluge) po standardnim pravilima
  - sintaksnim i
  - semantičkim
- Servisi se obično specificaju preko **interfejsa**
  - Sintaksa interfejsa se opisuje *Interface Definition Language*-om (IDL)
    - Spisak metoda sa opisom parametara
  - Semantika servisa se neformalno opisuje
- Implementacija interfejsa omogućava
  - da proces kome treba usluga servisa može komunicirati sa procesom koji implementira servis
  - da postoje različite implementacije servisa o čemu korisnik ne brine

Otvorenost (2)

- Dobro definisan interfejs
  - **Kompletan** – specificirano je sve što treba implementaciji
  - **Neutralan** – implementacioni detalji nisu spolja vidljivi
  - **Interoperabilnost** – delovi sistema raznih proizvođača mogu da rade zajedno i komuniciraju preko interfejsa
  - **Portabilnost** – aplikacija razvijana za sistem A se može izvršavati (bez modifikacija) u sistemu B (koji ima iste interfejse kao i A)

- Fleksibilnost otvorenog sistema
  - organizovan preko mnoštva malih komponenti koje se mogu lako zameniti ili izmeniti (prilagoditi)
  - komponente implementiraju poznate interfejse
    - Interfejse prema korisnicima i drugim aplikacijama
    - Interfejse između “malih komponenti”

Cilj 4 - Skalabilnost

- Skalabilnost se odnosi na rast:
  1. dodavanje novih korisnika i resursa – opterećenje raste
  2. geografsko proširenje sistema (pristup sa udaljenih mesta) – kašnjenja i manja pouzdanost veza
  3. očuvanje jednostavne administracije sistema iako se sistem proširuje – konfliktna pravila upotrebe resursa

Primeri: bankomati, DNS, rutiranje poruka na Internetu, ...

- Skalabilan je u suprotnosti sa centralizovan
  - **Centralizovan servis** – izvršava se na samo jednom serveru
    - Primer: jedan server za sve korisnike
  - **Centralizovani podaci** – nalaze se samo na jednom mestu
    - Primer: jedan (centralni) telefonski imenik
  - **Centralizovan algoritam** – odluka se donosi samo na osnovu kompletne informacije
    - Primer (loš): rutiranje paketa u mreži zasnovano na informacijama o svim čvorovima

Tehnike skaliranja

- Problem skalabilnosti se ispoljava kao problem performansi (zbog ograničene sposobnosti mreže i servera)
- Rešava se
  - Skrivanjem zastoja u komunikaciji – izbegavanje čekanja na odgovor – asinhrona komunikacija (čekanje odgovora u drugoj niti)
  - Distribucijom – podela posla na više malih i distribuiranih komponenti
  - Replikacijom – kopiranje komponenti povećava se raspoloživost DS (*load balancing*)
    - Keširanje podataka – odluku donosi klijent na osnovu ranije kopiranih podataka
      - Postoji problem konzistentnosti podataka u kešu

Osobine decentralizovanih algoritama

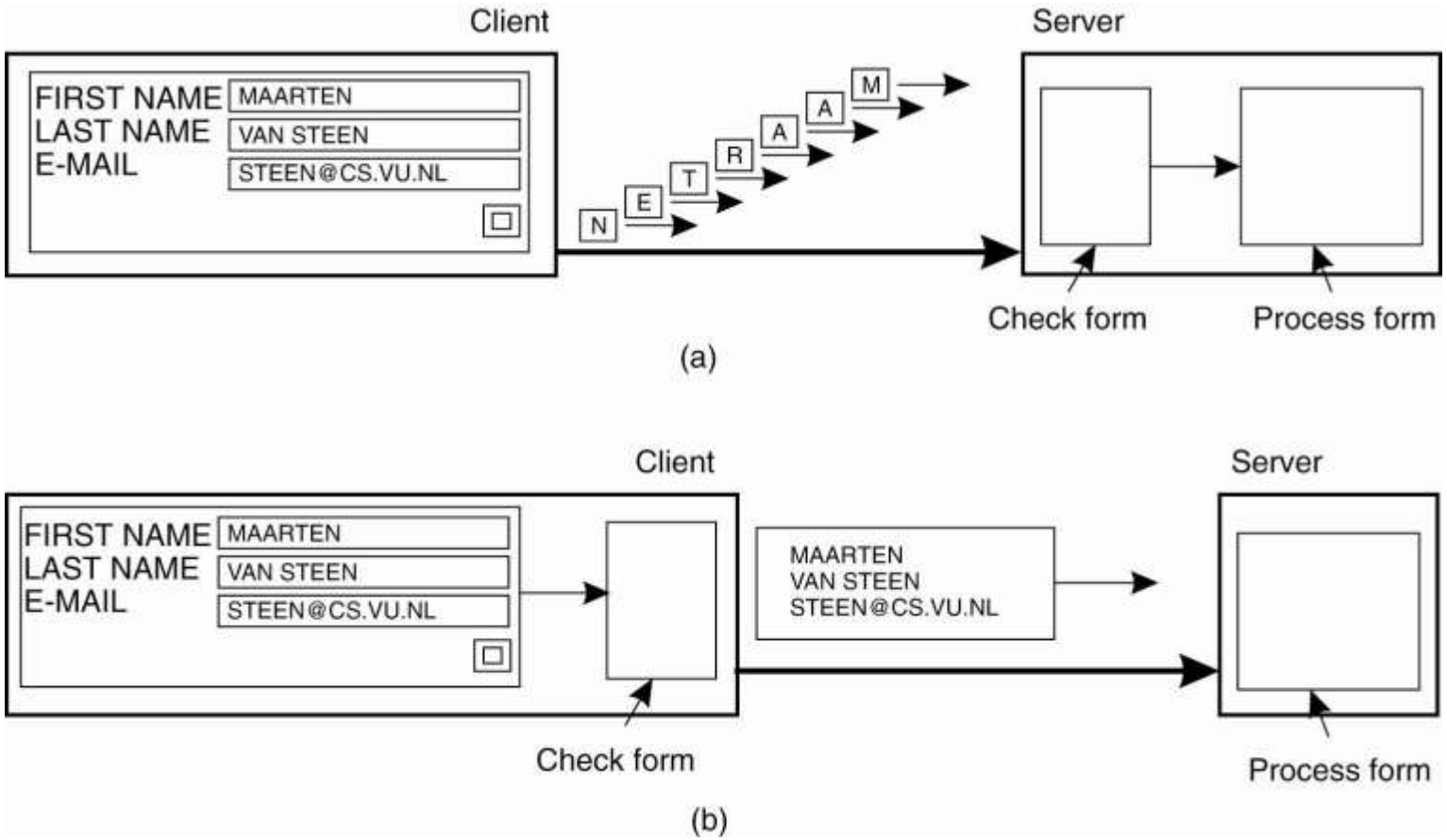
- Ni jedna mašina nema kompletnu sliku o celom sistemu (stanju sistema)
- Mašina donosi odluku samo na osnovu lokalnih informacija
- Kvar jedne mašine ne prekida rad algoritma
- Ne postoji globalan sat

Tehnika skaliranja (1)

Primer: Provera ispravnosti unetih podataka

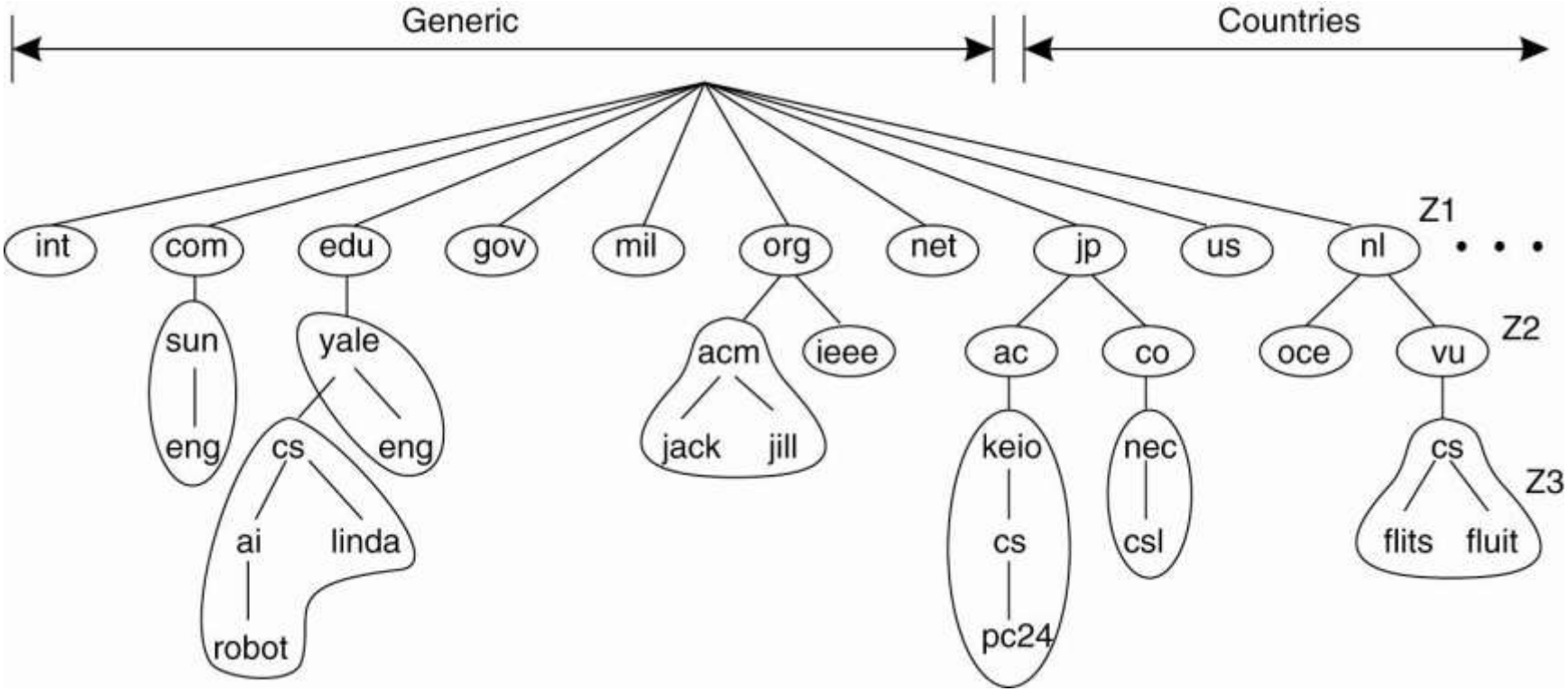
a) na server strani

b) na klijent strani



Tehnika skaliranja (2)

Primer: Traženje računara na osnovu Web adrese
(podela DNS adresnog prostora u zone)



“Zamke” kod razvoja DS

Pogrešne pretpostavke tipa:

- Mreža je pouzdana
- Podaci u mreži su sigurni
- Mreža je homogena
- Topologija mreže se ne menja
- Nema kašnjenja u prenosu
- Propustni opseg je neograničen
- Nema transportnih troškova
- Postoji samo jedan administrator

dovode do teških posledica.

Tipovi DS

- Distribuirani sistemi za intenzivno računanje

  Obično izvršavaju jednu aplikaciju.

  – Klasteri
  – *Grid computing*

- Distribuirani informacioni sistemi

  Brojne postojeće mrežne aplikacije se integrišu u okviru organizacije.

  – Na niskom nivou - procesiranja transakcija
  – Globalno integrisane aplikacije (u *Enterprise*-u)

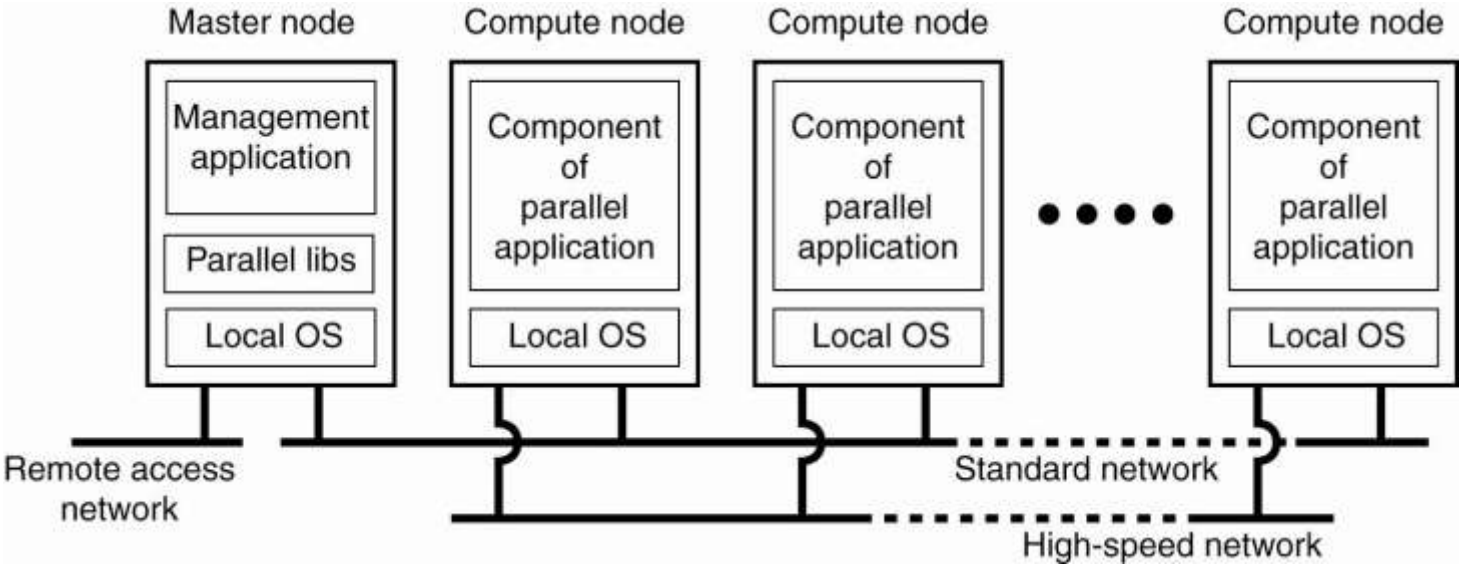
- Distribuirani rasplinuti (*pervasive*) sistemi

  Povezivanje malih, baterijski napajanih, mobilnih, bežičnih uređaja.

  – Kućne automatike
  – Elektronske brige o zdravlju
  – Mreže senzora

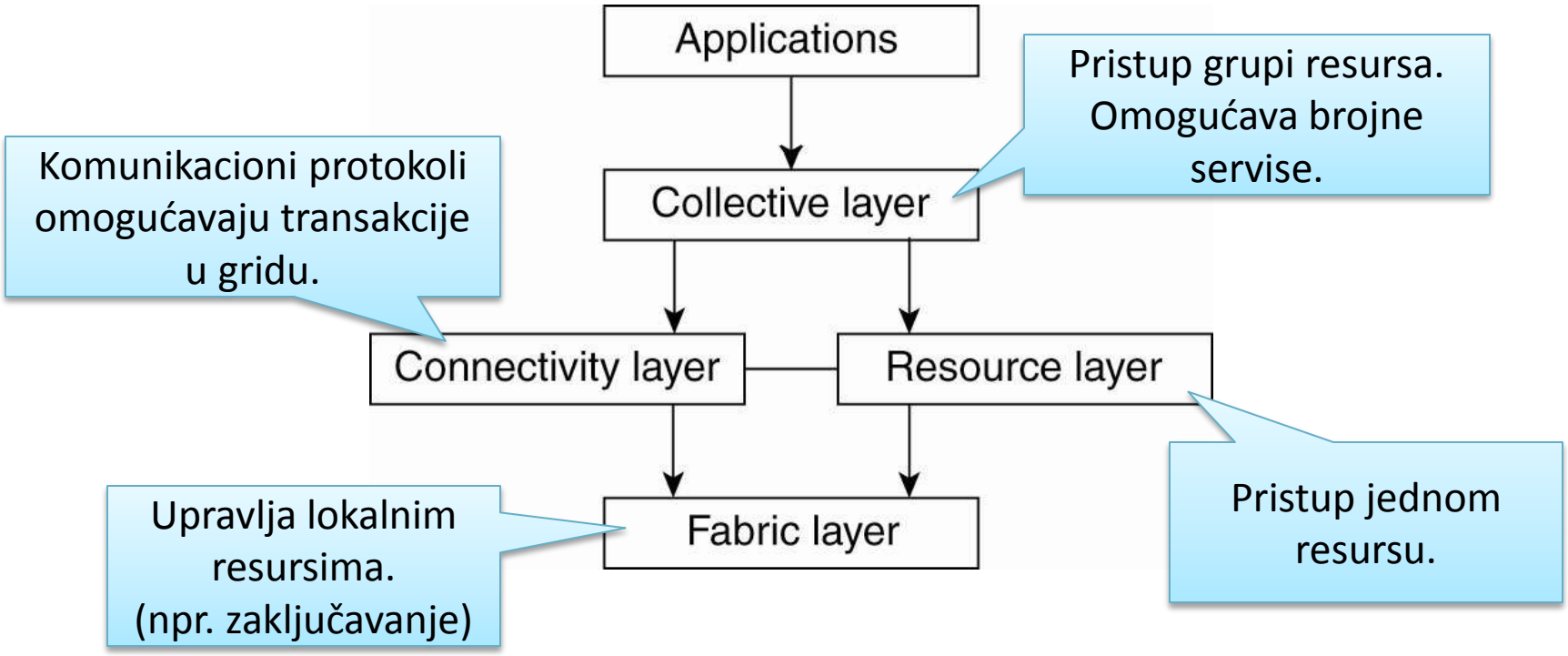
Klaster

- Tipična upotreba u sistema gde postoji potreba za intenzivnim računanjem – više računara radi u paraleli
  - Jedan računar nema dovoljnu snagu
  - Klaster čini nekoliko sličnih (često identičnih) računara smeštenih na jednoj lokaciji
    - Izvršavaju isti operativni sistem (OS)
    - Povezanih u istu mrežu
  - Ceo klaster se ponaša (logički) kao jedan računar



Grid computing

- Veći broj dislociranih računara učestvuje u računanju
  - Razlikuju se u hardveru, OS, mreži, administraciji, bezbednosti, ...
  - Organizovani su u "vitrualnu organizaciju" mada fizički pripadaju raznim organizacijama
- Arhitektura *Grid computing* sistema je slojevita



Procesiranje transakcija

- Tipično prisutno u aplikacijama koje rade sa bazama podataka
- Tipičan scenario integracije
  - Mrežna aplikacija se sastoji od **servera** (1 ili više)
  - Udaljeni programi (**klijenti**) upućuju zahteve serveru(ima)
  - Nekoliko zahteva klijenta se objedinjava u jedan veći zahtev koji se izvršava kao **distribuirana transakcija**

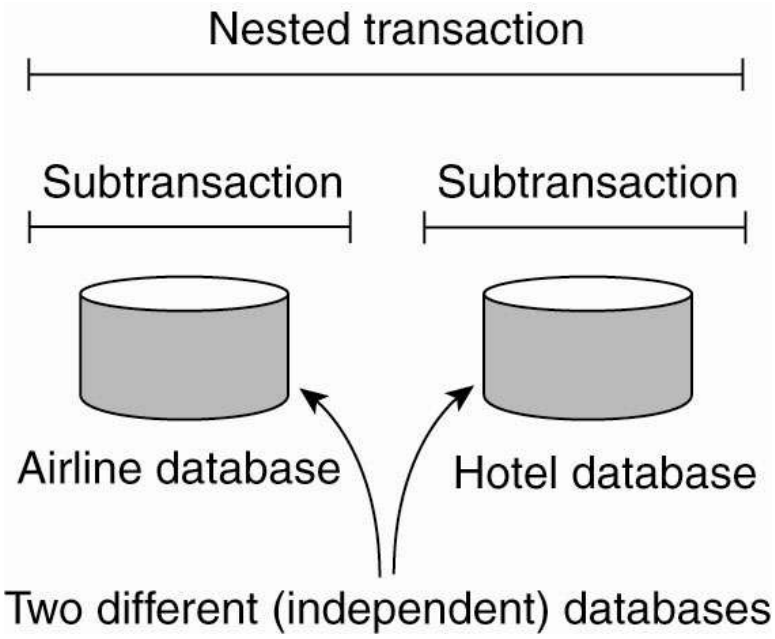
- Zasniva se na nekoliko osnovnih operacija (primitiva):
  - **BEGIN_TRANSACTION** – označava početak transakcije
  - **END_TRANSACTION** – označava kraj transakcije i pokušava da promene učini trajnim (*commit*)
  - **ABORT_TRANSACTION** – prekida transakciju i restaurira stare vrednosti
  - **READ** – čita podatke iz datoteke, tabele, ...
  - **WRITE** – zapisuje podatke u datoteku, tabelu, ...

Procesiranje transakcija (2)

- Osobine transakcija:
  - **Atomičnost** - prema spoljašnjem svetu transakcija je nedeljiva.
  - **Konzistentnost** - transakcija ne narušava ispravnost podataka.
  - **Izolovanost** - jednovremene transakcije ne utiču jedna na drugu.
  - **Postojanost** - kada se transakcija uspešno završi promene koje je izazvala postaju trajne.

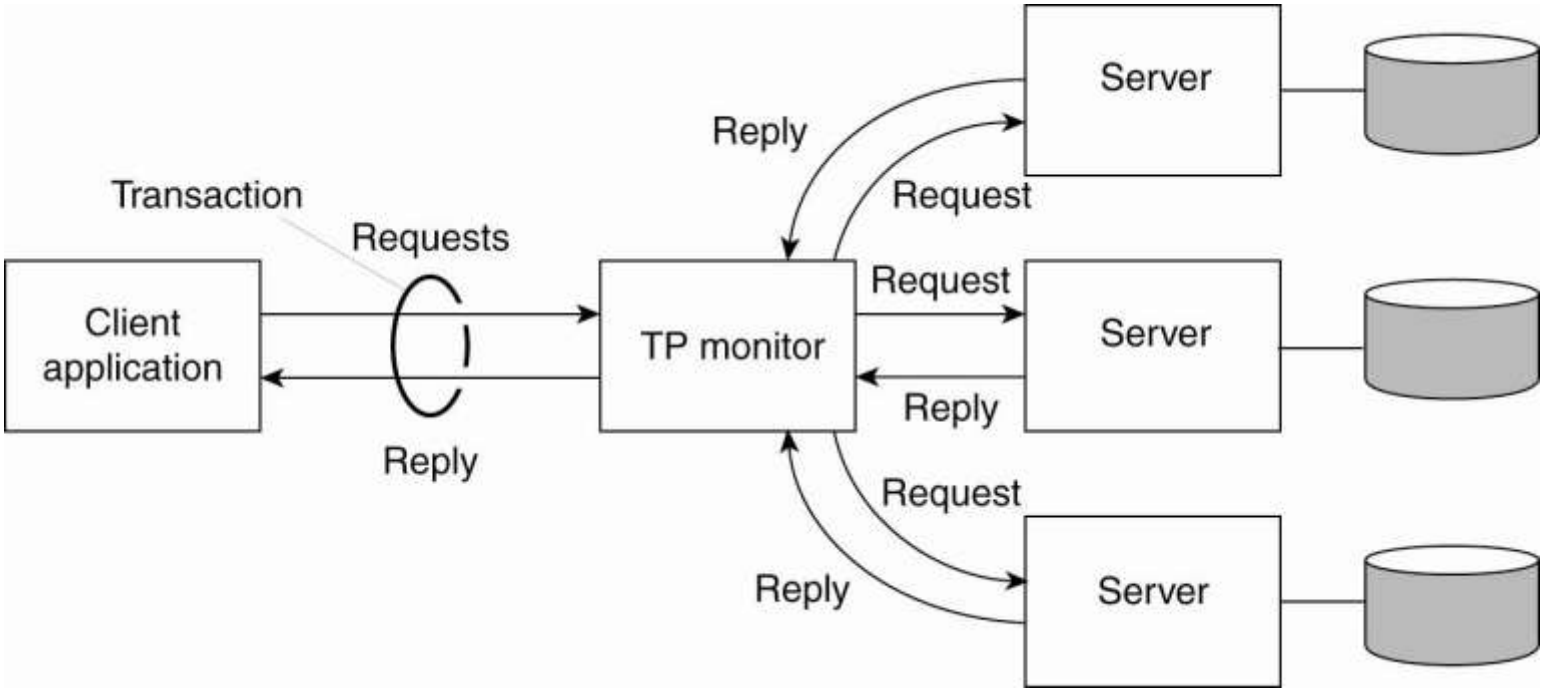
Procesiranje transakcija (3)

- Omogućene su ugnježdene transakcije
  - Osnovna transakcija sadrži više pod-transakcija
    - Prirodan način podele posla u DS
    - Pod-transakcije mogu postojati u više nivoa
  - Prekidanje pod-transakcije restaurira stare vrednosti u svim drugim pod-transakcijama (iako su se izvršile bez grešaka)



Procesiranje transakcija (4)

- Monitor procesiranja transakcije
  - Je komponenta koja izvršava distribuiranu transakciju



Globalna integracija aplikacija

- *Middleware* posreduje u povezivanju aplikacija
  - Softverska magistrala - *Enterprise Service Bus* (ESB)
- *Service Oriented Architecture* (SOA) je savremen koncept gde:
  - Serverske aplikacije pružaju usluge – servise
  - Klijent aplikacije su korisnici usluga
  - Servisi i korisnici usluga su “slabo povezani”
    - Koristi se komunikacija zasnovana na porukama (*Message Oriented Middleware*)