

## 7 RASPORIJELJENI SUSTAVI

- Svaki složeniji sustav sastoji se od više udaljenih, međusobno komunikacijski povezanih komponenata

### OSNOVNE KARAKTERISTIKE

- Paralelizam izvođenja elemenata sustava- koordinacija
- Nepostojanje globalnog sata- komunikacija porukama
- Neovisnost grešaka- mjesto, detekcija, oporavak

### IZAZOVI

- Heterogenost
- Otvorenost
- Sigurnost
- Skalabilnost
- Pogreške u radu
- Paralelizam
- Transparentnost
- Kvaliteta usluga

#### Heterogenost

- Na razinama:
  - o Komunikacijskih mreža
  - o Računalnog sklopovlja
  - o OS-a
  - o Programskih jezika za izgradnju aplikacija
  - o Implementacije programskih komponenti sustava
- Izazovi- poredak zapisa okteta, zapis struktura podataka, kodiranje znakova, razlike u jezicima implementacije, raspoloživa sučelja za komunikaciju
- Metode skrivanja heterogenosti:
  - o Komunikacijski protokoli
  - o Middleware

#### Otvorenost

- Određena mogućnošću dodavanja novih usluga, njihovim jednostavnim korištenjem
- Temelji- normirani komunikacijski protokoli, javno objavljena i/ili normirana sučelja za pristup API-ju
- Otvoreni raspodijeljeni sustavi građeni od heterogenih elemenata, svaki od njih je sukladan korištenim normama

#### Sigurnost

- Povjerljivost- od neovlaštenog pristupa
- Integritet- od promjena ili uništavanja
- Raspoloživost- od zapreka pristupa resursima

## Skalabilnost

- Učinkovitost sustava bez obzira na značajno povećanje posluživanih resursa i/ili broja posluživanja
- S obzirom na:
  - o Cijenu posluživanja- nadogradnja opreme, povećanje cijene posluživanja,  $O(n)$
  - o Performanse- smanjenje uz istu opremu i povećanje resursa,  $O(\log n)$
  - o Ograničenost resursa
  - o Izbjegavanje uvođenja uskih grla sustava

## Pogreške

- Djelomične- dio komponenata nastavlja obavljati funkciju, nisu svjesne tuđih grešaka
- Najčešće- u procesima sustava, mrežnoj komunikaciji
- Problematika-detekcija greške, posljedice, toleriranje, oporavak, semantika izvošenja operacija uz mehanizme oporavka od pogrešaka

## Paralelizam

- Zahtjevi za resursom mogu pristići od više klijenata istovremeno- sinkronizacija akcija

## Transparentnost

- Prikaz sustava kao cjeline umjesto kao skupa raznih samostalnih komponenata
- 
- Pristupa- jednak skup operacija za pristup lokalnim i udaljenim resursima
- Lokacije- skrivena prava lokacija
- Paralelizma- očuvanje konzistencije dijeljenog resursa
- Replikacije resursa
- Pogrešaka- skrivanje postupka oporavka
- Mobilnosti- promjena lokacije resursa ne utječe na način pristupa
- Performansi- prilagodba novom stanju zbog očuvanja performansi sustava

## MODELI

- Apstraktan, pojednostavljen i konzistentan opis nekog aspekta dizajna raspodijeljenog sustava
- Fizički- fizički uređaji i kanali kojima su spojeni
- Arhitekturni- struktura sustava- gradivne komponente sustava, uloge i suodnosi među komponentama
  - o Koji entiteti čine konkretni raspodijeljeni sustav
  - o Komunikacijska paradigma da bi komunicirali
  - o Uloge entiteta i odgovornost
  - o Kako se mapiraju na komponente fizičkog modela

## Entiteti

- perspektiva sustava:
  - o procesi
  - o dretve
  - o firmware
- perspektiva aplikacijskog sloja:
  - o objekti
  - o komponente
  - o web usluge

- prostorna sprega- entitet korisnik svjestan entiteta pružatelja usluge, najčešće i obrnuto
- vremenska sprega- oba entiteta aktivna tijekom komunikacije

#### Komunikacijske paradigme

- izravna komunikacija- 1:1, nužno i vremenska i prostorna sprega
  - o međuprocena komunikacija
  - o udaljeni pozivi
- neizravna- ne treba sprega, najčešće korištenjem posredničkog entiteta
  - o grupna komunikacija
  - o objavi, pretplati
  - o redovi poruka
  - o raspodijeljena dijeljena memorija
  - o prostori podataka
- klijent-poslužitelj arhitekturni stil, klijent zahtjeva, jedan entitet može obje uloge
- svi entiteti ravnopravni- skaliranje usluge sukladno trenutnom broju korisnika, veze sukladno potrebama aplikacije, raspodjela po aktivnim članovima
- logički entiteti mapiraju se na stvarne izvedbene resurse
  - o višestruka računala poslužitelji
    - particioniranje komponenti usluge i dijelova resursa na više servera
    - repliciranje jedinstvene usluge i resursa na više servera
  - o cache
    - na klijentskoj strani
    - na posrednicima
  - o pokretni programi
    - dobavljeni kod proširuje funkcionalnost klijenta
    - lokalna ili udaljena interakcija
    - problem sigurnosti
    - java applets- dobavljanje izvršnog koda na stranu klijenta
    - java script- dobavljanje koda u web preglednik
  - o pokretni agenti
    - prenošenje i koda i stanja
    - samostalno kretanje između grupe računala i lokalno obavljanje zadataka
    - smanjenje korištenja mreže

#### Arhitekturni obrasci

- vertikalni slojevi
- horizontalni slojevi
- zastupnici
- posrednici
- slojevi- razine apstrakcije, rješavaju neovisne zadatke
- konektori- protokoli interakcije među susjednim slojevima
- **prednosti:**
  - o sloj obavlja točno određenu ulogu
  - o slojevi slabo povezani konektorima

- neovisnost o implementaciji, jednostavno zamjenjivi
- protokoli interakcije se moraju strogo poštivati
- **nedostatci:**
  - smanjena učinkovitost sustava
  - skupa promjena protokola interakcije/sučelja
  - nekad teško identificirati slojeve

Izvedba slojeva i konektora

Monolitna aplikacija

- jedan proces izvođen u okviru os-a
- slojevi- logički odvojene biblioteke funkcija, čine jedinstveni izvedbeni kod aplikacije
- konektori- skup fja vidljivih iz susjednog sloja, komunikacija pozivima fja susjednog sloja

Višeprocena aplikacija

- izvođenje na više računala- raspodijeljena aplikacija
- slojevi izolirani unutar zasebnih procesa
- konektori- mehanizmi međuprocene komunikacije, komunikacijski protokol

Slojevi

- prezentacije – GUI
- aplikacijska logika
- podataka
- 
- monolitna:
  - svi funkcionalni slojevi aplikacije unutar procesa izvođenog na 1 računalu
  - potrebni dodatni alati za omogućavanje grupnog rada
- dvoslojna:
  - funkcionalni slojevi grupirani u 2 zasebna sloja, procesa
  - klijentska aplikacija(prezentacijska+aplikacijska logika)+ poslužiteljska aplikacija(sloj podataka)
- fat client- prezentacija+ aplikacijska logika- veća snaga obrade računala domaćina i veća količina podataka putem mreže
- thin client- prezentacija, manja snaga obrade, manje prenešenih podataka

višeslojna arhitektura sadrži višestruke aplikacijske poslužitelje i /ili baze podataka

- ravnomjernija raspodjela opterećenja, potrebna veća propusnost komunikacijske infrastrukture

Zastupnici

- entitet korisnik želi komunicirati s udaljenim entitetom unutar odvojenog procesa na istom ili drugom računalu
- korisnik komunicira korištenjem lokalnog zastupnika udaljenog objekta
  - komunikacija korisnika i zastupnika lokalna
  - kompleksnost mrežne komunikacije skrivena od korisnika- mrežna transparentnost

Posrednici

- sadrži popis usluga i njihovih svojstava, pružatelji usluga ih oglašavaju, a korisnici traže

## Podatci

- pull- zahtjev entitetu koji posjeduje podatke, povremena veza
- push-
  - o klijent- slanje emaila, messengeri- privremena veza
  - o server- HTTP push- trajnija veza
- polling(prozivanje)- simuliranje guranja prozivanjem
  - o klijent periodički uspostavlja vezu s poslužiteljem i provjerava dostupnost podataka
  - o podatci se ne dohvaćaju pullom
  - o nema trajne veze
  - o značajno opterećenje mreže
  - o guranje emaila na klijenta (pop, imap), rss feeds

## arhitektura usluga klijent-poslužitelj

- www- jasni slojevi prezentacije i podataka
- dns, ldap- uloga klijenta ovisna o konkretnoj primjeni
- svn- poslužitelj kao sloj podataka, klijent (ni)je sloj prezentacije
- .
- Telenet,ssh- rad na udaljenom računalu- klijent prezentacijski sloj, poslužitelj nije skladište podataka

## Email

- Klijent
  - o slanje, dohvat iz sandučića
  - o Prezentacijski, djelomično i podatkovni
- Server
  - o Podatkovni sloj prema klijentima
  - o P2p organizacija proslijeđivanja

## FTP

- Prijenos datoteka između 2 računala
- Komunikacijski kanal za prijenos naredbi i odgovora
- Za svaki prijenos se otvara poseban komunikacijski kanal- aktivni(server) i pasivni(klijent) mode

## MEĐUPROCESNA KOMUNIKACIJA

### Metode:

- Niska razina apstrakcije- poruke ili tokovi podataka
  - o Send/receive primitive
  - o Sockets
  - o Višesmjerno odašiljanje
  - o Prekrivne mreže
- Komunikacija između procesa i računala korištenjme mehanizma i protokola za mrežnu komunikaciju
  - o Međuprocesna-tokovi podataka, dijeljena memorija, signali
  - o Mrežna- tcp/udp, ip

### Send/receive primitive

- Send šalje u odlazni međuspremnik- sloj os i komunikacijskih protokola, poruka ne mora biti odmah poslana

- Receive- dohvat, sloj os-a i komunikacijskih protokola, spremnik može biti prazan
- Primitive konceptualne naravi- implementacija nad stvarnim protokolima mrežne komunikacije transportnog sloja
- Sinkrone operacije- blokiran do okončanja
  - o Send do čitanja, receive do dohvaćanja poruke, timeout
- Asinkrone- send poruka u međuspremniku, receive povratak neovisno o postojanju poruke

#### Komunikacija udaljenih procesa

- Locirati procese
- Ostvariti komunikacijski kanal
- Koristiti zajednički jezik komunikacije

#### TCP/IP stog protokola

- Slojevi veze, interneta, transporta ugrađeni u os
- Sloj aplikacije implementiran unutar pojedinih aplikacija ili kao dodatna biblioteka funkcija

#### Transportni sloj- TCP, UDP

##### Udp

- razmjena datagrama
- send neblokirajući
- receive ne/blokirajući

##### TCP

- dvosmjerni cjevovod
- send i receive oboje
- stvaranje veze- 3 syn/ack paketa
- prijenos male količine podataka tijekom trajanja veze

#### Sockets

- programska apstrakcija krajnje točke komunikacije
- id = Ip adresa, port, protokol
- .
- Klijent
  - o TCP inicira vezu prema serveru, razmjenjuje 8b putem cjevovoda(r,w)
  - o UDP izravno šalje datagrame, zaprima odgovore
  - o Klijent mora prethodno znati adresu servera
- Server
  - o TCP- osluškuje zahtjeve, ostvaruje komunikacijski kanal s klijentom i razmjenjuje nizove 8b dvosmjernim kanalom
  - o UDP zaprima datagrame zahtjeva i odgovara klijentu
- Portovi <1024 dostupni procesima s posebnim ovlastima, veći svim
- Berkeley Sockets API
  - o Općenit mehanizam međuprocenjske komunikacije na istom/različitim računalima
- Create-connect-write-read-close()
- Create-bind-listen-accept-read-write-close()

