

## Hajussüsteemide põhiomadused

- Ressursside jagamine
- Avatus
- Paralleelsus
- Skaleeritavus
- Tõrkekindlus
- Transparentsus

## Ressursside jagamine

- Mis on ressurss?
- Riistvara
  - Kõik ressursid pole jagatavad
- Andmed
  - Andmebaasid
  - Kommertstarkvara
  - Tarkvara tootmine
  - Grupitöö

## Ressursside jagamine: ühine osa

- Erinevad ressursid nõuavad reeglina erinevat käsitlemist, aga on ühiseid vajadusi:
  - Nimeskeem identifitseerimiseks
  - Nimede tõlkimine võrguaadressideks
  - Samaaegse kasutuse koordineerimine

## Ressursside jagamine: klient-server

- Ressursse haldavad serverid
- Ressursse küsivad kliendid
- Teenus — abstraktne olem, mida toodab teatud kogus serverprotsesse, mis töötavad eri masinatel ja mis suhtlevad üle võrgu
- Ressursnimed on enamasti serverispetsiifilised
- Klient kasutab teenust ja server osutab teenust
- Näide: WWW
- Klient-server mudeli miinused
  - Kõiki ressursse ei saa sel meetodil jagada
  - Arvutisisese jagamise jaoks sageli ebaefektiivne
  - Veel?

## Ressursside jagamine: objektmudel

- Objektid (ressursid)
  - unikaalselt identifitseeritavad
  - võib võrgus ringi tõsta
- Objektide kasutajad
- Teated objektidele
  - vajadusel vastused
- Objektmudeli miinused?

## Avatus

- Riistvaraline avatus
  - Lisaseadmete lisamine, komponentide sobivus
  - Riistvara programmeerimisinfo
- Tarkvaraline avatus
  - Lisaomaduste tekitamine, teenuste lisamine, suhtlus muu tarkvaraga
  - Avalikud programmeerimisliidesed ja protokollid
- Avatud süsteemi saab kokku panna heterogeensest riistvarast ja erinevate tootjate tarkvarast
- Iga komponendi saab asendada vastavalt liidestele
- Avatuse kontroll – testime komponendi vastavust avatud standarditele

## Paralleelsus

- Tsentraliseeritud süsteem  $n$  protsessoriga annab  $n$  kordse jõudluse kasvu, võrreldes ühe protsessoriga süsteemiga
- Hajussüsteem  $m$  ühe protsessoriga arvutiga annab meile kuni  $m$  kordse jõudluse kasvu, kui iga protsess töötab eri arvutis
- Kasutajad käivitavad programme või suhtlevad programmidega samaaegselt
- Paljud serverprotsessid töötavad samaaegselt, igaüks reageerides klientprotsessidelt saabuvatele erinevatele päringutele
- Paralleelsusest tulenevad omad probleemid (peamiselt sünkroniseerimine)

## Skaleeritavus

- Hajussüsteem peab olema koostatud selliselt, et süsteemi oleks võimalik vajadusel laiendada
- Süsteemi kasvamine ei tohiks kaasa tuua tarkvara muutmise vajadust
- Süsteem peab olema efektiivselt kasutatav ka komponentide suure arvu juures
- Skaleeritavust mõõdetakse selle järgi, missugune on süsteemi jõudlus keskmiselt ühe protsessori kohta (kogujõudlus jagatud protsessorite arvuga)
- Süsteemi disainimisel ei tohi eeldada ühegi ressursi piiratust



## Skaleerimist takistavad tegurid

- Tsentraalsed teenused — üks server kõigi jaoks
- Tsentraalsed andmed — üks keskne andmekogu
- Tsentraliseeritud algoritmid — otsuste tegemine täieliku informatsiooni järgi

## Algoritmi skaleeruvuse eeldused

- Üski masin ei sõltu täielikust infost kogu hajusa süsteemi kohta
- Masinad teevad otsuseid ainult kohaliku informatsiooni alusel
- Ühe masina tõrge ei blokeeri algoritmi tööd
- Ei eeldata globaalse kella olemasolu

## Skaleerimismeetodeid

- Asünkroonne suhtlus geograafilise skaleeruvuse saavutamiseks
- Replitseerimine, puhverdamine (*caching*)
- Vertikaalne skaleerimine — eraldame eraldi sõlmed süsteemi erinevate kihtide jaoks
- Horisontaalne skaleerimine — eraldame palju sõlmi ühe kihi teenuste realiseerimiseks

## Tõrkekindlus

- Tõrkekindluse tagamine
  - Riistvara dubleeritus: dubleeritud riistvarakomponentide kasutamine
  - Tarkvaraline taastumine: programmid peavad olema vigade korral suutelised taastama stabiilset seisundit
- Süsteemi käideldavuse (*availability*) mõiste: töövalmis oleku aeg jagatud kogu ajaga

# Transparentsus 1

- Kasutaja peab nägema süsteemi kui tervikut, tema eest tuleb "peita" konkreetsed komponendid
- Juurdepääsu transparentsus — peidab andmete esituse ja juurdepääsumetodid
- Asukoha transparentsus — peidab ressursi asukoha
- Migratsiooni transparentsus — peidab selle, et ressurss võib kolida
- Replikatsiooni transparentsus — peidab selle, et ressurss võib kasutamise ajal kolida

## Transparentsus 2

- Tõrke transparentsus — peidab lihtsamad tõrked kasutaja eest
- Samaaegsuse transparentsus — peidab selle, et sama ressursi võivad korraga mitu osapoolt kasutada
- Jõudluse transparentsus — süsteem ei pidurda kunstlikult suure jõudlusega kasutamist
- Skaleerimise transparentsus — peidab skaleeruvusprobleemid kasutaja eest

## Transparentsuse määr

- Täielik transparentsus ei tarvitse mõistlik olla
  - Geograafilised eripärad (signaali leviku aeg!)
  - Asukohaspetsiifilised teenused
  - Veast üle saamisel pika aja järel on sageli kasulikum veaga leppida ja selle vastu midagi rakenduse tasemel teha

## Mädad eeldused keskkonna kohta

- Võrk on töökindel
- Võrk on turvaline
- Võrk on homogeenne
- Topoloogia ei muutu
- Latents puudub
- Võrgu läbilaskevõime on lõpmatu
- Transpordi hind on null
- Kogu süsteemi administreerib üksainus administraator