Arkitektur for feil-tolerante og åpne distribuerte systemer

- **Kilde: *Architectural Support for Designing Fault-Tolerant Open Distributed Systems*, Salim Hariri, Alok Choudhary og Behcet Sarikaya, IEEE Computer, juni 1992, side 50-62.**
- **Distribuerte systemer**
  - autonome datamaskiner
    - » gjerne heterogene
  - meldingsutveksling
  - økte muligheter
    - » ytelse
    - » ressurs deling og utnyttelse
    - » utvidbarhet
    - » feiltoleranse
  - økte problemer
    - » heterogene maskiner/systemer
- **Åpne systemer**
  - samkjøring av utstyr fra ulike leverandører
  - ISO OSI modell (International Standards Organization)
  - langsomt
- **Feil-tolerante systemer**
  - avveining mellom kostnad ved feil og kostnad ved bruk av redundans og andre feil-toleranse teknikker

Fault-Tolerant Open Distributed System (FTODS)

- **Hvordan kan datamaskinarkitektur gi muligheter for mer pålitelig og hurtigere utførelse av distribuerte applikasjoner som bruker OSI-standarder?**
- **OSI = Open Systems Interconnection**
  - ISO-OSI
  - 7lags modell
  - (Fig 1: Applikasjons-laget)
- **Fig 2: Foreslått arkitektur for FTODS**

FTODS beregnings-moduler --- ønskede egenskaper

- **Pålitelig kringkasting**
  - kringkasting til tross for feil
  - hastighet kritisk
- **Selv-reparasjon / "recovery"**
  - vanskelig / mange teknikker
    - » f.eks. statisk redundans,diagnose og maskering
    - » hierarkisk
- **Selektiv feil-toleranse**
- **Node-arkitektur; Fig. 3**
  - "vanlige" prosessorer
  - # prosessorer i en node avgjøres av pålitelighetskrav
  - en koordinator-prosessor velges dynamisk blant prosessorer uten feil
    - » leder "avstemmings-algoritmer"
    - » besørger kommunikasjon med andre noder
- **Hierarkisk permanent lager; Fig. 4**

Distribuert algoritme for avstemming

Fase-1) Hver prosessor beregner et resultat (stemme)og kringkaster dette til alle andre prosessorer i noden

Fase-2) Hver prosessor mottar stemme fra alle andre prosessorer og sammenlikner disse med egen stemme

a) Hvis alle er enige fastsetter en resultatet ("commit")

b) Hvis det finnes et flertall som er enige
- resultatet defineres som flertallets mening
- hvis koordinator-prosessor ikke er i flertallet velges en ny koordinator fra flertallet
- koordinator igangsetter oppretting ("recovery") av mindretallet

c) Hvis det ikke finnes flertall av enige prosessorer
- alle gjør lokal selvdiagnose
- hvis en gruppe prosessorer "finner seg selv i orden" velger de ny koordinator og foretar ny avstemming innenfor den nye gruppen
- hvis ikke, har hele noden feilet og det kreves ekstern feiloppretting ("recovery")

Litt pålitelighetsanalyse for FTODS

r = en prosessors pålitelighet

N_n = # prosessorer i en node

f = # prosessorer med feil

R_{bus} = pålitelighet til systembuss

$$R_n(N_n) = C \times R_{bus} \sum_{f=0}^{N_n-1} \binom{N_n}{f} r^{N_n-f}(1-r)^f$$

- **Antakelser**
  - prosessorfeil er eksponensialfordelt
- **En nodes pålitelighet R_n er gitt ved sannsynligheten for at minst 1 prosessor virker**
  - Likning (1) i artikkelen (gjengitt over)
    - » sum av sannsynlighet for alle mulige kombinasjoner av *f* prosessorer med feil og resten uten feil, der *f* er mindre enn N_n
  - C = Dekningsgrad (Eng. "Coverage")
- **RIF (Reliability Improvement Factor)**
  - relativ forbedring ved bruk av redundans i forhold til hva en maksimalt kan oppnå

Transaksjoner

- **Pålitelighetsanalyse av atomiske transaksjoner**
  - en transaksjon kan bestå av del-oppgaver som fordeles utover systemet
    - » del-oppgavene kan utføres redundant
    - » teori for analyse av pålitelighet
  - Fig. 10
  - Fig. 11
    - » pålitelighet av eksempel i Fig.10 for tre ulike tilfeller
- **Samtidighetskontroll**
  - (Faktarute side 52)
  - (Eng: Concurrency control)
  - Gi transaksjonene en rekkefølge
  - Identifisere konflikter mellom transaksjoner
  - Synkronisere transaksjoner for å fjerne konflikter