NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk

Institutt for datateknikk og informasjonsvitenskap

Løsning på SIF8037 Distribuerte systemer og ytelsesvurdering
(Kun distribuerte systemer)

Onsdag 29. mai 2002, 0900-1400

Typegodkjent lommekalkulator med tomt minne tillatt
Ingen trykte eller håndskrevne hjelpemidler tillatt

Der det synes å mangle noen opplysninger, må det angis hvilke antagelser som synes å være naturlige.

Distribuerte systemer dekkes av oppgavene 1, 2 og 3, mens ytelsesvurdering dekkes av oppgavene 4, 5, 6 og 7.

Oppgave 1 – Kommunikasjon og synkronisering (Communication and synchronization) - 15%

a) Drøft kort noen ulike former for ordning (ordering) som kan kreves i forbindelse med kommunikasjon mellom grupper av prosesser i distribuerte systemer, og diskuter kort aktuelle måter å implementere minst to slike former for ordning på

SVAR:

Ulike former:

- Uordnet – Ingen intern ordning
- Kausalt ordnet – Intern ordning tilsvarende eventuelt årsak-virknings forhold
- Totalt ordnet – Intern ordning uansett
- Synkront ordnet – Ekstern, dobbel barriere

Implementasjon:

- Uordnet – Ingen spesielle tiltak
- Kausalt ordnet –Logiske klokker + nodeidentifikatorer (/ vektorklokker – egentlig unødvendig)
- Totalt ordnet – Logiske klokker + nodeidentifikatorer (/ vektorklokker – egentlig unødvendig)
- Synkront ordnet – Logiske klokker (/ vektorklokker) + Ekstra nodekom. & -nodesynkronisering

b) Angi kort overordnete krav som bør stilles til algoritmer for å foreta primas utvelgelse (secretary election) i et distribuert system, og illustrer minst en aktuell algoritme for å løse denne oppgaven

SVAR:

Overordnete krav:

- Oppnå absolutt primas utvelgelse
- Sikre at alle aktive prosesser involveres
- Sikre at ingen passiv prosess ødelegger
- Unngå vranglås og utsulting

- Unngå avhengighet av enkeltnoder eller enkeltmeldinger
- Unngå eller begrens flaskehalser i systemet (noder og lenker)
- Begrens meldingsmengder og prosesseringsforsinkelser

Tilsvarende algoritmer:

- Sentralisert algoritme (enkel)

```
Start Valg-rutine:
    1a) Send Valg-melding til global koordinator
```

```
Får Valg-melding:
    2a) Send N-1 * OK?-melding
    2b) Motta ≤ N-1 * OK!-melding
    2c) Send Sjef-melding til alle m/ Størst identifikator
```

```
Får OK?-melding:
    3a) Send OK!-melding til global koordinator
```

- Distribuert algoritme (avansert)

```
Start Valg-rutine:
    1a) Send Valg-melding til alle med høyere identifikator
```

```
Får Valg-melding (fra noen lavere):
    2a) Send OK-melding til avsender
    2b) Start Valg-prosess som over
```

Får OK-melding (fra noen høyere): 3a) Gi opp for andre

Får ingen melding (fra noen høyere): 4a) Send Sjef-melding til alle m/Egen identifikator

- Ringalgoritme (typisk)

Start Valg-rutine: 1a) Send Valg-melding m/ Egen identifikator

Får Valg-melding: 2a) Med høyere identifikator og Egen ikke inkludert / Egen inkludert: Send videre m/ Gitt identifikator 2b) Med lavere identifikator og Egen ikke inkludert: Send videre m/ Egen identifikator 2c) Med lavere identifikator og Egen inkludert: Stopp Valg-melding 2d) Med samme identifikator og Egen inkludert: Send Sjef-melding m/ Egen identifikator

Får Sjef-melding: 3a) Med høyere identifikator: Send videre m/ Gitt identifikator 3b) Med samme identifikator: Avslutt rutine

Oppgave 2 – Distribuerte databasesystemer og distribuert pålitelighet (Distributed database systems and distributed reliability) - 20%

Tofase-låsing (2PL – 2 phase lock) i databasesystemer tilsier at alle objekter som skal aksesseres må låses før noe objekt blir låst opp. 2PL er nok til å sikre seg mot gale resultater i forbindelse med transaksjoner som utføres i parallell i sentraliserte databaser.

a) Begrunn kort om 2PL-løsningen fra sentraliserte databasesystemer kan anvendes direkte i distribuerte databasesystemer med fragmentering men uten replisering (fragmented, non-replicated distributed databases). Hvis noen nye problemer må løses i tillegg, skal dette illustreres med eksempler. Tilhørende nye løsninger skal eventuelt kort beskrives.

   SVAR:

   Tilstrekkelighet:

   - Basis 2PL er direkte anvendbar.

Nye problemer:

- Kun effektivitetsutfordringer
  (Vranglåser oppdages senere, og tilhørende aborter kaster bort mer arbeid),
  ingen korrekthetsproblemer.

Tilhørende løsninger:

- Intet spesielt

b) Begrunn også kort om 2PL-løsningen fra sentraliserte databasesystemer kan anvendes direkte i distribuerte databasesystemer med fragmentering så vel som replisering (fragmented, replicated distributed databases). Hvis noen nye problemer må løses i tillegg, skal dette illustreres med eksempler. Tilhørende nye løsninger skal eventuelt kort beskrives.

SVAR:

Tilstrekkelighet:

- Basis 2PL er nesten direkte anvendbar.

Nye problemer:

- Både effektivitetsutfordringer
  (Vranglåser oppdages senere, og tilhørende aborter kaster bort mer arbeid)
  og korrethetsproblemer
  (Låsing av et / flere fysiske replikater må føre til låsing av tilhørende logiske objekt).

Tilhørende løsninger:

- Sentralisert låsing (På sentral node)
- Sentralisert / distribuert låsing (Av primær-kopi / master-element)
- Distribuert låsing (Spenn: W låser alle og R låser en - W og R låser begge en majoritet)

Tofase-skriving (2PW – 2 phase write) i databasesystemer tilsier at alle objekter som skal oppdateres må skrives til en logg før noe objekt blir skrevet til databasen. 2PW er nok til å sikre seg mot gale resultater i forbindelse med transaksjoner som feiler under utførelse i sentraliserte databaser.

c) Begrunn kort om 2PW-løsningen fra sentraliserte databasesystemer kan anvendes direkte i distribuerte databasesystemer med fragmentering men uten replisering (fragmented, non-replicated distributed databases). Hvis noen nye problemer må løses i tillegg, skal dette illustreres med eksempler. Tilhørende nye løsninger skal eventuelt kort beskrives.

SVAR:

Tilstrekkelighet:

- 2PW sikrer kun atomiskhet (m.h.t. feil) innen en node (alt-eller-intet).

Nye problemer:

- Trenger også å sikre atomiskhet (m.h.t. feil og autonomi) mellom flere noder (alle eller ingen). Tilstandssyn (oppfatning av hvilke logiske objekter er tilgjengelige eller ikke) er ikke et ekstra problem.

Tilhørende løsninger:

- 2PC: 1.fase (stemming), 2.fase (utføring) /
  3PC: 1.fase (stemming), 2.fase (orientering), 3.fase (utføring)

d) Begrunn også kort om 2PW-løsningen fra sentraliserte databasesystemer kan anvendes direkte i distribuerte databasesystemer med fragmentering så vel som replisering (fragmented, replicated distributed databases). Hvis noen nye problemer må løses i tillegg, skal dette illustreres med eksempler. Tilhørende nye løsninger skal eventuelt kort beskrives.

SVAR:

Tilstrekkelighet:

- 2PW sikrer kun atomiskhet (m.h.t. feil) innen en node (alt-eller-intet).

Nye problemer:

- Trenger også å sikre atomiskhet (m.h.t. feil og autonomi) mellom flere noder (alle eller ingen). Tilstandssyn (oppfatning av hvilke logiske objekter er tilgjengelige eller ikke) er dog et ekstra problem.

Tilhørende løsninger:

- 2PC: / 3PC:
  Med både nodefeil og kommunikasjonsfeil kreves egne protokoller for å utpeke partisjon for oppdatering. (Med kun nodefeil kreves intet ekstra).

Oppgave 3 – Distribuerte filsystemer og distribuerte navnetjenester (Distributed file systems and distributed name services) - 15%

a) Angi kort overordnete krav som bør stilles til et distribuert filsystem, og diskuter kort viktige komponenter som må implementeres i et slikt system

SVAR:

Overordnete krav:

- Sikre ekte globalt filsystem
- Tilby ulike former for konsistensoppnåelse
- Tilby ulike former for oppdateringssemantikk
- Håndter både feiltoleranse og mobilitetstoleranse

- Utnytt klientmaskiner
- Utnytt datacaching
- Begrens beregningsspredning
- Begrens datareplisering

Viktige komponenter:

- Filtjeneste (med grensesnitt som i sentraliserte filsystemer)
- Katalogtjeneste (med iterativ søking mot filer)
- Klienttjeneste (med mapping mot filtjeneste / katalogtjeste / filtjeneste+katalogtjeneste)

b) Angi kort overordnete krav som bør stilles til en distribuert navnetjeneste, og diskuter kort viktige komponenter som må implementeres i en slik tjeneste

SVAR:

Overordnete krav:

- Tilby hierarkisk navnerom:

  Separat for hvert enkelt organisasjonsenhet etc. – full frihet etter behov

- Tilby tilpassete navnetjenere:

  Samlet for flere organisasjonsenheter etc. – full tilpasning til last

- Anvend gradvis binding av navn:

  Åpner således opp for utnyttelse av aliaser

- Anvend iterativ binding av navn:

  Åpner således opp for utnyttelse av caching

Viktige komponenter:

- Implementer nødvendige operasjoner – for maskiner, brukere og tjenester:

  Registrer (Hvem, hva)
  Finn (Hva)
  Deregistrer (Hvem, hva)

- Implementer caching og replikasjon:

  Sikrer ytelse og tilgjengelighet