

Institutt for datateknikk og informasjonsvitenskap

Løsningsforslag for Eksamensoppgave i TDT4190 Distribuerte systemer

Faglig kontakt under eksamen: Jon Olav Hauglid

Tlf.: 93 80 58 51

Eksamensdato: Onsdag 20. mai 2015

Eksamenstid (fra-til): 9.00 – 13.00

Hjelpemiddelkode/Tillatte hjelpemidler: D. Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Annen informasjon: Oppgavesettet inneholder 8 oppgaver. Det er angitt i prosent hvor mye hver (del-)oppgave teller ved sensur. Gjør rimelige antagelser der du mener oppgaveteksten er ufullstendig og skriv kort hva du antar. Lykke til!

Målform/språk: Bokmål

Antall sider (uten forside): 2

Antall sider vedlegg: 0

Kontrollert av:

Dato

Sign

Oppgave 1 – Karakterisering av distribuerte systemer og Google Case Study (10 %)

- a) Hvorfor er transparens en ønsket egenskap ved distribuerte systemer?

Transparens gjør at detaljer om hvordan det distribuerte systemet fungerer er skjult for brukeren og applikasjonsprogrammereren. Dette gjør f.eks. at man ikke trenger å tenke på hvor ressurser er lokalisert, om replikering blir brukt, om nettverksfeil har oppstått osv. Samtidig gjør det systemet mer fleksibelt da man f.eks. kan endre replikeringsoppsettet uten at applikasjonene må skrives om.

- b) Google File System (GFS) har ikke replikerings-transparens. Hvorfor ble det laget slik?

I GFS er klienten ansvarlig for å sende all oppdatert data til alle replikater. Dermed er ikke replikeringen transparent. Dette gjøres for å unngå at noden med primærreplikatet blir en flaskehals. Dette er greit inne i Google der de har kontroll på alle applikasjoner som bruker GFS og dermed kan endre disse enkelt.

Oppgave 2 – Likemannsnettverk («Peer-to-peer networks») (15 %)

- a) Anta at en node i et likemannsnettverk har et objekt du vil ha tak i. Forklar hvorfor du ikke er garantert å finne objektet i et ustrukturert likemannsnettverk. Forklar deretter hvorfor du er garantert å finne objektet i et strukturert likemannsnettverk. Anta at nettverket er stabilt i perioden søket utføres.

I et ustrukturert likemannsnettverk har man ingen ide om hvor objekter ligger. Dermed må man lete overalt via broadcasting. For at letingen en gang skal ta slutt, bruker man Time-To-Live for å indikere hvor mange hopp en melding skal ta før den ikke blir sendt videre. Hvis objektet ligger på en node som ikke kan nås innen TTL, blir det ikke funnet. I et strukturert likemannsnettverk er plasseringen av et objekt gitt av dets GUID. Dermed kan man hele tiden rute søkemeldingen i rett retning. Hvis noden som skulle ha objektet ikke har det, vet man at det ikke finnes.

- b) En prefiks-rutingtabell i Pastry har 16 kolonner 0-F. Hva bestemmer hvor mange rader tabellen har?

En rad per hex-siffer i GUID.

Oppgave 3 – Sikkerhet (15 %)

Alice og Bob ønsker å bruke en symmetrisk algoritme til å kryptere en stor mengde data de skal utveksle.

- a) Beskriv minst én sikker måte de kan utveksle den felles, hemmelige nøkkelen på (via nettverk).

De kan bruke asynkron kryptering der Alice har fått tak i Bob sin offentlige nøkkel via en sikker nøkkeldistribusjonstjeneste. Alice krypterer da den hemmelige nøkkelen hun har laget med Bob sin offentlige nøkkel. Denne meldingen kan bare dekrypteres med Bob sin private nøkkel (som bare Bob har). En annen mulighet er å bruke Needham–Schroeder.

- b) Hvorfor kan det være fornuftig å bruke en symmetrisk og ikke en asymmetrisk algoritme når Alice og Bob skal utveksle store mengder data?

Fordi symmetriske algoritmer gjerne er 100-1000 ganger raskere.

Oppgave 4 – Distribuerte filsystemer, distribuerte objekter og fjernkall (10 %)

Det distribuerte filsystemet NFS bruker Sun RPC til kommunikasjon mellom klient og tjener. Hvilken kallsemantikk brukes da, og hvordan påvirkes fjerngrensesnittet («remote interface») til NFS av det?

Sun RPC bruker kallsemantikken minst-én-gang, dvs. når en klient kaller en funksjon/prosedyre én gang så vil tjeneren utføre den minst én gang. Det er mulig at tjeneren utfører operasjonen flere ganger.

For tjenester som bruker Sun RPC, som f.eks. NFS, betyr det at fjerngrensesnittet må ha idempotente operasjoner, slik at resultatet er det samme om operasjonen blir utført én eller flere ganger.

Oppgave 5 – Tid og global tilstand (10 %)

Et distribuert system har to prosesser p1 og p2. Anta at et cut er slik at den nyeste hendelsen fra p1 som er med, har vektorklokkeverdien (3,4) og den nyeste hendelsen fra p2 har vektorklokkeverdien (1,2). Er dette cut'et konsistent eller inkonsistent? Eller er det umulig å si? Begrunn svaret.

(3,4) betyr at p1 har blitt påvirket av 4 hendelser hos p2 men (1,2) betyr at bare 2 hendelser har skjedd hos p2. Det betyr at p1 må ha mottatt en melding fra p2 som p2 ennå ikke har sendt. Dermed er cut'et inkonsistent.

Oppgave 6 – Koordinering og enighet (10 %)

Anta at avstemnings-algoritmen («Maekawa's voting algorithm») skal brukes for distribuert gjensidig utelukkelse («mutual exclusion») i et system med 5 noder. Lag avstemningssett som gjør at resultatet blir riktig og som samtidig er effektive og rettferdige.

Alle noder har hvert sitt avstemningssett.

Riktig – alle må overlappe.

Effektiv – så små som mulig og noden er selv med i sitt eget sett.

Rettferdig – alle noder med i like mange sett.

En mulig løsning blir da:

Sett for n1: (n1, n2, n3)

Sett for n2: (n2, n3, n4)

Sett for n3: (n3, n4, n5)

Sett for n4: (n4, n5, n1)

Sett for n5: (n5, n1, n2)

Oppgave 7 – Distribuerte transaksjoner (15 %)

- a) Er det nok at en 2PC-koordinator spør litt over halvparten av deltakerne om det skal gjøres commit eller abort? Begrunn svaret.

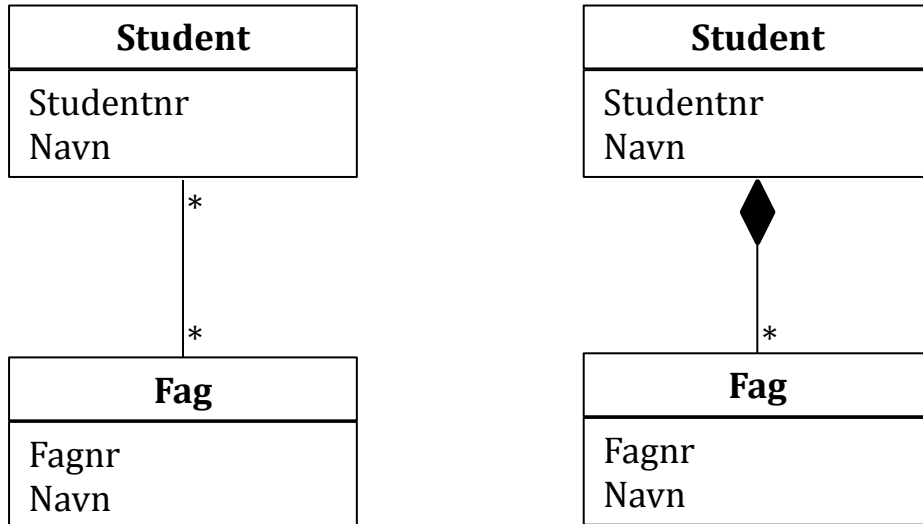
For at resultatet skal blir konsistent må alle enten utføre transaksjonen eller ingen må utføre den. Dermed er en abort nok til at alle må ta abort og man er nødt til å spørre alle deltakere.

- b) Et distribuert system bruker kant-jakt («edge chasing») til å detektere vranglås. Node X har objekt A som er låst av transaksjon U. Node X mottar en forespørsel fra transaksjon W om å låse objekt A. Hva gjør node X da?

Node X må kontakte noden som er transaksjonskoordinatoren til transaksjon U slik at den kan finne ut om U venter på en lås. Hvis U venter på en lås, vil meldingen bli sendt videre til den noden som har objektet som U venter på. Denne noden vil sende meldingen til noden som er transaksjonskoordinator for transaksjonen som har lås på dette objektet. Osv.

Oppgave 8 – NoSQL (15 %)

Anta at det skal lages en database om hvilke studenter som tar hvilke fag. En mulig løsning er en relasjonsdatabase med en mange-til-mange relasjon mellom en Student-tabell og en Fag-tabell (venstre figur). En annen mulig løsning er en NoSQL-database der Fag er aggregert inn i Student (høyre figur).



- a) Hvilken løsning er mest effektiv for å finne hvilke fag en student tar? Begrunn svaret.

Med NoSQL-løsningen blir det slik at når man har funnet posten med info om studenten, er alle fag listet opp der. For relasjonsdatabasen blir mange-til-mange typisk implementert med en egen tabell. Dermed må man utføre 1-2 kostbare join for å finne hvilke fag en student har tatt. NoSQL-løsningen blir derfor mer effektiv.

- b) Hvilken løsning er mest effektiv hvis man skal bytte navn på et fag? Begrunn svaret.

Med relasjonsdatabaseløsningen trenger man bare å oppdatere en rad i Fag. Med NoSQL-løsningen må man lete gjennom alle studenter og oppdatere alle studenter som tar faget. Relasjonsdatabase blir dermed mer effektiv i dette tilfellet.