DIS - Transaktioner 2 - Byzantine's Generals Problem & Highly-Available Transactions

Byzantine's Generals Problem

Der skal være en, der styrer slagets gang. Det skal ikke være den samme altid - turen skal gå på skift. Der er aldrig kun én server der skal stå for alle ting

Det kaldes blandt andet for en **coordinator role**, en **log leader** eller generelt set **log election**

Husk at transaktioner skal gemmes og kunne genskabes - det gøres via en log over hvad der er sket.

For at koordinere har vi skabt det, der hedder en **two-phase commit protocol**

Det nytter ikke hvis en general (server) sender samtlige beskeder til alle andre uden videre, for beskederne kan jo forfalskes(tampered) på vejen.

Hovedregel
for hver besked fra en loyal general (anerkendt server), der skal alle andre loyale generaler bruge samme besked

Det kaldes også et **trust scheme**, når vi laver regler for hvem vi kan stole på.

Interactive consistency

er når hver proces har sin egen værdi, og når alle processer kommunikerer med hinanden for at afgøre hvilken værdi, der er den korrekte.

Der er tale om en række private værdier, der udveksles mellem processer (servere) for at finde en offentlig værdi i den givne **consensus vector**.

Majoritetsalgoritme
Se billede

En majoritetsalgoritme er et generelt begreb, der dækker over at afgøre hvilken værdi, der er korrekt.
majority(v1, ..., vn-1)

Kriterier
Ofte implementeres majority-algoritmer med udgangspunkt i antallet af ens forekomster.

Forekomster (typisk valgt)
Valg af den forekomst af en værdi, der optræder flest af.

Sandsynlighed
Valg af den værdi, der forekommer mest sandsynlig

Tillid
Valg af den værdi, der optræder hos flest, der er tillid til

Type
Valg af den værdi(er det en int eller double), som passer med den type, der er vlag

Hvad hvis 10 forskellige værdier?
Man kan sige, at ingen har ret eller man kan kombinere de andre algoritmekriterier

Hvis der ingen forrædere er, så sender generalen sin besked til alle løjtnanter, som læser beskeden og handler derefter.
Hvis en løjtnant ingen besked modtager, så falder hun tilbage og angriber ikke (fallback policy).
Master - slave #
Se på Transaktioner 1

Det betyder at konsensus i et distribueret system er dyrt, tager tid og kræver mange ressourcer.

Signed messages

afsløre når noget ikke stemmer overens. Det kan være simple *sets* med tal, der udgør en signatur

Billede

Der er tale om signering via v : i : j og så fremdeles, som afslører hvem der snyder hvem.

Fejlhåndtering

Generelt siger man at fejl i et distribueret system kan være fail-stops eller byzantine.

Fail stop
Server går bare ud

Byzantisk fejl
serveren virker, men den opfører sig ikke som resten af serverne i et cluster. Kaldes byzantisk fejl

Achieving reliability in the face of arbitrary malfunctioning is a difficult problem, and its solutions seems to be inherently expensive.

Den eneste måde at reducere hvor dyrt det er, er at gøre sig antagelser.

Highly-Available Transactions

Indeed, serializable transactions—the gold standard of traditional ACID databases—are not achievable with high availability in the presence of network partitions.

Serialized

Det betyder at den er udtrykt på en form, som kan gemmes på en harddisk. Typisk gemmer vi jo ikke det, vi har i hukommelsen, på samme måde på harddisken

Flattening objects
JSON.stringify = serialized til en string

Speed of light er en begrænsning
Fundamentally, the speed at which two servers can communicate is (according to modern physics) bounded by the speed of light.

Høj tilgængelighed eller stærk semantik

Man kan ikke få begge dele. Enten er noget meget tilgængeligt eller meget rigtigt.

Semantik

- et begreb vi bruger om sproget. Det dækker over ordenes betydning.
- forestil enten så kan du spørge en server om noget rigtig mange gange eller rigtig hurtigt, og så kan den svarer noget tilbage, som ikke altid er det samme. Ellers også skal den have noget betydning. Man tillægger noget en betydning. Semantik kan være meget beskrivende. Bjørne-tjeneste – flere betydninger fx.
- Sproget forfalder jo ikke. Et sprog er et område indenfor videnskaben som det vi siger og taler, som naturligvis redefinererer sig selv som vi bruger det. Mens matematik fx ikke kan ændres.
- Give nogle løfter om at de returnere noget der er konsistent = semantik

Stærk semantik kunne fx være at jeg altid kan huske al ting jeg har lavet. Hvis jeg skal huske alt det jeg har lavet, så skal man virkelig huske meget. Naturen har besluttet, at man ikke kan huske alt. Tilgængeld kan de ting vi kan huske huske med det samme, sådan nogenlunde. Ikke muligt at skrive dagbog over alt i vores hverdag

Man vælger nogle ting som er tilgængelige, bevidst eller ubevidst i vores hukommelse, og resten smider vi væk.

Man kan ikke beskrive et system til fulde uden at lave det.

Meget tilgængelige transaktioner er baseret på idéen om at servere ikke behøver koordinere med hinanden før *en af dem* besvarer en forespørgsel.

Last-Write-Wins (LLW)

Det sidste, der bliver sendt til en server og gemt i en database, er det som der bruges.

Det giver lidt sig selv. Men det er jo ikke sikkert at det, der sidst er ankommet, er det rigtige.

Kæmpe stort trade-off. Virkelig udbredt. Det er ikke nødvendigvis rigtigt – det kan være det første write er bedre. En antagelse – man antager at den sidste der skriver noget til en database vinder

Dirty reads/writes

A dirty read (**uncommitted dependency**) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

Det vil sige når en transaktion kan læse noget, som ikke er gemt på harddisken endnu.

når en transaktion godt må læse data fra en anden transkation som der er i gang med at skrive i det.

I et meget tilgængeligt system er **consistency begrænset** - det kan ikke garanteres. Det tager tid for alle servere at blive enige om en værdi, og det kan man ikke vente på.

1. Consistency
2. Eventual consistency
3. Availability/tilgængelighed
4. Consensus - et udtryk for at vi fx kan sige et demokrati – demokrati er en konsensus algoritme – på en eller anden måde søger at blive enige – hvordan servere bliver enige – fx majoritetsalgmoritmer
5. Convergence - Faktum at værdier ligger mange forskellige steder, og samler sig et sted
6. Partitioning (burde hedde network faults) - Så er partitionering et udtryk for, at nogen af serverne går ud. Når noget går ud, så har man network partitionering
7. Fault-tolerance
8. Replication - spejling af database - backup er et eks.

I et meget tilgængeligt system konvergerer alle værdier mod den samme værdi. hvis der nu er 8 servere, der mener at en værdi er 1, og 2 servere, der mener at værdien er 2. Eventual consistency? Altid forsinkelse i DIS