

Korrekt data i Molnet

Korrekt informationslagring med korrekt teknikval



2010-02-11: Sven-Håkan Olsson

MOLNLAGRING Strukturerad informationslagring i Molnet kan numera göras på många sätt. Dessa har synnerligen olika egenskaper – där prestanda ofta ställs mot datakorrekthet.

Sedan mitten av 80-talet har programmerarna blivit helt tillvanda vid fördelarna med relationsdatabaser: ett standardiserat och interoperabelt anropsspråk (SQL), enorm flexibilitet att ställa frågor på olika sätt, god konkurrens som gör att leverantörerna står på tå, mogna och synnerligen stabila databasmotorer, bra prestanda, utmärkta säkerhetskopieringslösningar. Och kanske framförallt: Sammanhållna transaktioner (kallas ofta ACID, unit-of-work, commit-rollback, obrytbara uppdateringar etc).

Varför är nu detta så viktigt? Jo, det är det som gör att en uppdatering inte kan bli delvis gjord. Om jag flyttar 8000 kr från ett konto till ett annat i min bank så får inte pengarna försvinna ifall programmet eller servern skulle krascha mittemellan uttaget från ena kontot och insättningen i det andra. Hellre ska uttaget återföras, det som kallas rollback. Korrekt datalagring!

Sen finns det en del nackdelar med relationsdatabaserna. Ibland behövs inte finesserna de ger. Ibland skulle man hellre vilja ha direkt objektorienterad lagring. Skalbarheten om man till exempel börjar förvänta sig 100 000 simultana användare, är också diskutabel.

Målbilden för många av molnplattformarna är just 100 000 simultana användare och mycket mer därtill. Man vill snarare leverera grundstenar till applikationer som liknar Gmail, Youtube eller Facebook och har liknande – enorma – användarantal.

De molnlösningar som brukar kallas PaaS använder därför vanligen inte relationsdatabaser utan betydligt enklare strukturer. (PaaS = Platform as a Service, se den tidigare trendspaningen [Vad är vad uppe bland molnen](#)).

Många av lösningarna kan snarare liknas vid lagringsrutnät. Det finns celler, kolumner, rader. De kallas vanligen tabeller vilket kan vara förvirrande eftersom det heter så inne i relationsdatabaserna också, men lagringen är alltså mer lik hierarkiska databaser från 70-talet än SQL-motorer. Rutnäten kompletteras med så kallad blob-lagring där stora filmer, bildfiler och liknande kan lagras, ofta med en pekare ifrån rutnätet.

Den stora frågefriheten som SQL ger har behövts offras för att få denna enorma skalbarhet. Det innebär att datamodelleringen blir än mycket viktigare än med relationsdatabas eftersom endast vissa, i förväg uttänkta frågor sedan går bra att ställa.

Programmerare som är vana vid SQL kan ibland tycka att det är som att gå tillbaka till stenåldern och det är för många en kunskapströskel att övervinna att förstå hur man bäst utnyttjar "tables" i exempelvis Amazon SimpleDB, Google Bigtable eller Microsoft Azure Table.

Sämre transaktionsstöd

En mycket stor skillnad finns också vad gäller sammanhållna transaktioner, stödet är mycket sämre. I vissa av lagringslösningarna kan "närliggande" data hållas samman i uppdateringar, men inte alls på det flexibla sätt som inom en relationsdatabas. (Så kallade distribuerade transaktioner är en helt annan sak, de är sällan välfungerande, oavsett teknikbas).

Det finns alltså stor risk att man måste designa applikationer enligt helt andra principer än man varit van vid för att information ska lagras garanterat korrekt!

Vissa mönster finns för att råda bot på situationen, exempelvis:

- "Long-running transactions"
- "Deferred update" via kö med garanterad leverans
- Avstämningar
- BASE – "Basically Available, Soft state, Eventually consistent".

Gemensamt med alla dessa mönster är att de kräver klart ökad insats av modellering, design och programmering för att stödja korrekt informationslagring. Å andra sidan kan man få en enastående skalbarhet och prestanda med de här enklare lagringsmotorerna.

Relationsdatabas i molnet

Det finns molnerbjudanden även för relationsdatabaslagring. Man kan till exempel köpa Amazons infrastrukturtjänster och ovanpå det på egen hand placera en relationsdatabas som MySQL eller man kan köpa Microsoft SQL Azure. Skalbarheten blir dock som nämnts inte så enorm som i de enklare lösningarna, men det kanske heller inte är det som behövs i aktuellt fall.

Hur ska man då välja?

Sammanfattningsvis så talar följande FÖR de enklare molnlagringarna jämfört med relationsdatabas:

- Då du behöver synnerligen hög skalbarhet och prestanda
- Då datakorrekthet inte är det extremt viktiga.
- Då traditionell sökfleksibilitet inte är ett starkt krav
- Då du vill ha en billigare lagring (vanligen)
- Då du är beredd på att lägga tid och pengar på utökad modellering (för rätt datakorrekthet) och på kompetensutveckling.

Lär mer om moln (och även knytningen till SOA) på kurs

För- och nackdelar med de olika molnen, liksom anknytande SOA, integrationslösningar och migrering går igenom noggrant i tvådagarskursen [Cloud computing – migrering och integration](#) som ges igen våren 2010 med undertecknad som kursledare.



Sven-Håkan Olsson är en fristående konsult som särskilt arbetar med att kombinera verksamhetsnytta med teknikhöjd. Han har en lång karriär sedan 70-talet som it-konsult (it-arkitektur, systemdesign, programmering, reviewer, utredningar, kursledning). Sven-Håkan är också medgrundare av Know IT och var dess teknikchef 1990-2003. Han utsågs till en av "Sveriges topputvecklare" av Computer Sweden 2008.

Sven-Håkan håller regelbundet kurser åt Dataföreningen Kompetens, till exempel "Cloud Computing integration och migration". Läs gärna mer på hans blogg www.definitivus.se.

[Sven-Håkan Olsson](#)