

VAN DER LANS

# NoSQL: Alweer een déjà vu



Een nieuwe generatie database servers met hippe namen als Hadoop, CloudKit, CouchDB en Dynamite, is opgestaan. *NoSQL-database servers* worden ze genoemd. Ze vormen geen homogene groep, want de een is ontwikkeld voor het efficiënt opslaan en manipuleren van documenten, de ander voor het opbouwen van Petabyte grote databases, de derde om razendsnel gegevens te zoeken en de vierde om in de 'Cloud' te opereren. Wat ze gemeenschappelijk hebben is dat ze niet SQL maar een eigen taal als databasetaal ondersteunen, en dat ze pretenderen iets te kunnen dat vandaag de dag niet met een van de klassieke SQL-database servers mogelijk is.

Maar als je de verhalen over deze producten leest en aanhoort, ontstaat wel een déjà vu gevoel. We gaan daarom terug in de tijd om dit toe te lichten. Begin jaren tachtig kwamen de eerste SQL-producten op de markt. Het duurde even voordat ze door de markt geaccepteerd werden, maar zeker aan het einde van de jaren tachtig waren ze de dominante database servers. Zo begon de hegemonie van SQL in de databasewereld, die tot op de dag van vandaag voortduurt.

De eerste applicaties die ontwikkeld werden, waren voornamelijk administratieve applicaties. Sommigen wilden echter meer technische applicaties ontwikkelen. Karakteristiek aan dat soort applicaties is dat de objecten die gemanipuleerd worden zeer complex zijn. Technische applicaties ontwikkelen op een SQL-database server was toentertijd een uitdaging. In diezelfde periode kwamen ook de eerste object oriented programmeertalen op de markt, die zeer snel door de technische wereld werden geadopteerd. Maar even snel werd duidelijk dat het huwelijk tussen OO-talen en SQL niet ideaal was. Sommige leveranciers brachten dan ook speciale OO-database servers op de markt. Deze producten sloegen de gegevens niet als tabellen en kolommen op maar als objecten, en ondersteunden geen SQL maar vaak eigen *proprietary* talen. Hun marketing pitch was dat met deze database servers applicaties ontwikkeld konden worden die met SQL nauwelijks mogelijk waren. In het begin hadden zij zeker gelijk, maar uiteindelijk werden de klassieke SQL-producten zodanig verbeterd, dat ze de technische applicaties wel konden ondersteunen. Dit leidde er toe dat de nieuwere OO-database servers toch uit de markt verdrongen werden. Momenteel kom je ze alleen nog in kleine nichemarkten tegen. Hetzelfde gebeurde met datawarehousing in de jaren negentig. In het begin hadden de klassieke SQL-producten moeite om de zware en typisch datawarehouse-gerichte query's snel te verwerken. Dus kwamen er database servers op de markt die gegevens

op een andere wijze opsloegen (dus niet meer in tabellen en kolommen) en andere databasetalen boden. Dit worden ook wel OLAP- of multidimensionale database servers genoemd. Ondertussen zijn de SQL-producten uitgebreid met mogelijkheden om ook die zware query's snel te verwerken. Zeker de zogenaamde datawarehouse appliances (die allemaal SQL gebaseerd zijn) leveren soms indrukwekkende prestaties. Hiermee is de behoefte aan speciale, niet-SQL gebaseerde database servers sterk afgenomen. Tenslotte hebben we in de afgelopen jaren wederom een nieuw type workload gekregen: het opslaan, bewerken en terugzoeken van XML-documenten. Ook dat ging in het begin lastig met SQL. Dus werden er weer speciale database servers gebouwd, speciaal ontwikkeld voor XML. Echter, in de huidige versies van DB2, Oracle en SQL Server, is ondersteuning voor XML prima geïntegreerd met SQL. Wederom werd een groep niet-SQL database servers langzaam verdrongen. Kortom, in de historie van SQL-database servers hebben we nu al minstens drie keer gezien dat deze producten een nieuw type workload initieel niet aan kunnen. Er worden dan database servers ontwikkeld die speciaal voor dat type workload bestemd zijn. Ze zijn niet gebaseerd op SQL en slaan hun gegevens niet in tabellen en kolommen op. Iedereen is daar dan in het begin laaiend enthousiast over. Wat logisch is, want eindelijk kunnen bepaalde organisaties dan hun gewenste applicatie ontwikkelen. De producten krijgen dan enkele jaren volop aandacht, totdat de leveranciers van de SQL-producten een acceptabele oplossing bedenken, waardoor de niet-SQL producten langzaam uit de markt gedrukt worden.

De grote vraag is nu: gaat dit nu ook weer met de NoSQL-stroming gebeuren, is ook deze een kort leven in de *spotlights* beschoren en worden ook zij straks weer uit de markt verdrongen? Als we naar de geschiedenis kijken, dan is het antwoord duidelijk: ja. Of vormen zij de bekende uitzondering? Over twee of drie jaar zullen we het waarschijnlijk wel weten. Ondertussen zou ik organisaties aanraden voorzichtig te zijn met het adopteren van deze technologie. Want als u er nu in investeert en over enkele jaren blijken ze toch verdrongen te worden, dan heeft u applicaties ontwikkeld die niet *portable* zijn, want ze gebruiken een *proprietary* databasetaal. Maar ik zou ze niet negeren, want er zijn nu applicaties die we momenteel niet met de huidige SQL-producten kunnen ontwikkelen en dan is een dergelijke NoSQL-oplossing het enige antwoord.

**Rick van der Lans** is zelfstandig IT-consultant.