

Concept van grid computing voorlopig tot het rekencentrum beperkt

# Oracle 10g: trendsetter of trendvolger?

Carel-Jan Engel

**Sinds afgelopen september tikkert Oracle stevig aan de weg met de aankondiging van de nieuwe versie van haar RDBMS en Applicatie Server. De opvolger van de internetversie 9i wordt 10g genoemd, met de g van grid computing.**

Met deze stap claimt Oracle een doorbraak mogelijk te maken op het gebied van grid computing. Is dat waar; zet Oracle de piketpaaltjes uit voor een nieuwe datacenter-architectuur? Of hebben anderen dat gedaan en maakt Oracle nu met al haar marketing-power gebruik van deze mogelijkheden?

### Oergrid

Anders dan de recente opkomst van het woord grid computing doet vermoeden is grid computing zeker niet iets van de laatste tijd. Het concept is nu vooral bekend van toepassingen als SETI@home<sup>1</sup>. Op hun website staat dat het idee om de complexe wetenschappelijke berekeningen op home computers te laten uitvoeren rond kerst 1994 is ontstaan. Terecht meldt men ook dat het idee *misschien* al eerder bestond. Dat is een understatement, de historie van de oervorm van grid computing gaat nog heel wat verder terug.

**Eigenlijk is het jammer dat er in de marketing een breuk met het oude is geforceerd**

Op 18 mei 1994 presenteerde Xander Evers, tijdens het voorjaarscongres van de Nederlandse Unix Gebruikersgroep (NLUUG), een paper over Condor, waaraan hij in het kader van zijn studie had gewerkt. Condor is ontwikkeld aan de universiteit van Wisconsin, en had tot doel wetenschappelijke berekeningen gedistribueerd te laten uitvoeren op de talloze (tijdelijk) werkloze desktops en andere systemen.

Het principe is niet al te ingewikkeld, maar dat geldt voor de meeste goede ideeën. Er wordt gebruik gemaakt van Unix-werkstations. Op de werkstations draait een achtergrondproces, dat

bewaakt of het systeem bezig is met taken ten behoeve van de gebruiker. Zodra dat niet het geval is, wordt een programma gestart, dat in feite een bevroren toestand van een eerdere actieve periode bevat, inclusief kopieën van het werkgeheugen.

Het programma start met de verwerking van de rekentaken, en met een vaste frequentie worden *checkpoints* ingelast. Op die momenten wordt een nieuw programmabestand aangemaakt, dat opnieuw een bevroren toestand van dat moment bevat. Op deze manier kan de verwerking direct worden afgebroken als de gebruiker terugkeert. Daarna kan het programma worden doorgestuurd naar een volgend werkstation, om daar de verwerking te vervolgen. Hooguit het stukje werk dat na het laatste checkpoint werd uitgevoerd wordt dan opnieuw gedaan. Vanuit een centraal systeem worden de programmaatjes op reis gestuurd langs de aangemelde werkstations, en aan het eind keren ze daar weer terug met het resultaat. De werkelijkheid is wel iets ingewikkelder dan hierboven geschetst, maar het uitwerken daarvan valt een beetje buiten het kader van dit artikel, en dat van uw vakblad.

### LoadLeveler

Een uitgebreide beschrijving van Condor is te vinden in Litzkov<sup>2</sup>. Dat document dateert van 1988! In het document beschrijven Litzkov c.s. dat op basis van hun concept al zo'n vijf maanden berekeningen parallel worden uitgevoerd. Met Condor werden dus al in 1988 rekenintensieve toepassingen gedistribueerd uitgevoerd. Bij verder zoeken op internet op basis van de referenties in Litzkov<sup>2</sup> wordt er verwezen naar Litzkov<sup>3</sup> dat al in 1985 verscheen. In dat document beschrijft hij de eerste versie van het door hem gebouwde systeem, de voorloper van Condor. Het idee daarvoor kreeg hij van Wilkes, die hierover al in 1980 publiceerde. Dat betekent dus dat de ideeën rondom grid computing al weer ruim 20 jaar bestaan.

Uiteindelijk vormde het Condor-platform de basis van het later door IBM op de markt gebrachte LoadLeveler. De ideeën over Condor leven nog steeds. Versie 6.5.5 is in september vrijgegeven. Daarnaast wordt vanuit het Condor-project ook samengewerkt in de Globus Alliance, waaraan ook bijvoorbeeld IBM en NASA deelnemen. Globus Alliance is een samenwerkingsverband dat onder andere de Globus Toolkit uitbrengt, die gebruikt kan worden voor het bouwen van applicaties die op een grid moeten kunnen draaien.

---

Jarenlang groeide het concept van grid computing in de relatief beschutte omgeving van de wetenschappelijke wereld. Pas de afgelopen paar jaar buitelen de grote software-leveranciers over elkaar heen om deze, op zich heel zinnige, ontwikkeling tot een hype te maken. Ieder geeft er zijn eigen naam aan, zoals on-demand, utility computing en, inderdaad, grid computing. Helaas is niet alleen de naam verschillend, ook de uitwerking loopt nog stevig uiteen. Praten we over servers waarin naar behoefte extra processoren kunnen worden bijgeprikt, gaat het om clusters van servers, of om flexibele financiering?

## Oracle grid

Net als de andere leveranciers is ook Oracle de laatste tijd nadrukkelijk in de weer met grid computing. Op dit moment is het nog vooral ketelmuziek, in de loop van deze maand wordt als het goed is versie 10g van het RDBMS leverbaar. Dan kunnen we zelf gaan kijken hoe het werkt.

Als we het oergrid van Condor als uitgangspunt nemen en we vergelijken dat met de vorm van grid computing die Oracle aanbiedt, dan zijn er toch wel heel grote verschillen. Voor een goede vergelijking wordt eerst een stukje historie van Oracle's groei naar grid computing toegelicht. Een tweetal jaren geleden kwam Oracle met 9i op de markt met de Real Application Cluster (RAC) feature.

RAC was nieuw, het was een absolute doorbraak op het gebied van cluster computing. RAC, RAC en nog eens RAC was de boodschap van Larry Ellison. Oracle Parallel Server (OPS) had afgedaan, het had z'n nut bewezen voor de benchmarks, maar het werd nu vervangen door RAC. De claims waren niet gering: RAC schaaft vrijwel lineair, iedere applicatie kan er ongewijzigd op worden overgezet, met RAC wordt het allemaal een stuk goedkoper. Dat klinkt allemaal heel erg mooi, maar papier is geduldig.

Is RAC helemaal nieuw? Niet echt, net als in de automobiel-industrie een model een tweede leven krijgt aangemeten is ook hier de carrosserie vernieuwd, maar de 'motor' is nog grotendeels dezelfde. Dat blijkt treffend uit Nørgaard<sup>4</sup>, die de historie beschrijft van parallelle technieken in het Oracle RDBMS: de eerste code voor OPS werd al in versie 5 ingebouwd. Nadat Oracle een verbeterde distributed lock manager had gebouwd voor VAX/VMS (waarvan de techniek en concepten later door Digital in de verbeterde VMS distributed lock manager werden overgenomen) werd uiteindelijk in Oracle 6.2 de eerste versie van OPS uitgebracht.

## Partitionering

Een applicatie die een database wil benaderen die door meerdere servers voorzien van OPS wordt beheerd, moet wel heel zorgvuldig worden ontworpen om te voorkomen dat de complexe samenwerking tussen de verschillende servers het systeem knarsend tot stilstand brengt. Het datamodel moet zorgvuldig worden gepartitioneerd. Hetzelfde geldt voor de applicatie. Als deze partitionering goed is uitgevoerd dan zal een gebruiker,

die met zijn applicatie via server 1 werkt, niet snel een datablok nodig hebben dat in de cache van server 2 staat.

Is de partitionering onzorgvuldig uitgevoerd en een applicatie wil een blok wijzigen dat door een andere server in de cache wordt gehouden, dan zal na enig netwerkverkeer over en weer de 'eigenaar' van het blok de gegevens op schijf schrijven. Daarna worden de andere servers daarvan op de hoogte gesteld en het gewijzigde blok kan weer consistent door alle servers worden ingelezen. De beschreven bewerking wordt 'pinging' genoemd en is berucht vanwege de nadelige gevolgen voor de performance.

## Uiteindelijk vormde het Condor-platform de basis van het later door IBM op de markt gebrachte LoadLeveler

Bij de introductie van RAC mag dan het 'motorblok' grotendeels ongewijzigd zijn; de 'cilinders' zijn uitgeoord, er zit een nieuwe cilinderkop op, de vorm van de 'zuigers' is aangepast, en met elektronische brandstofinjectie zijn de prestaties sterk verbeterd. Eigenlijk is het jammer, dat er in de marketing zo'n breuk met het oude is geforceerd. Er zitten veel goede zaken in het systeem, dat inmiddels zolang is gebruikt dat de meeste bugs er wel uit zijn. De aangebrachte aanpassingen en verbeteringen maken het geheel een stuk bruikbaar.

## Clustering

OPS en de opvolger RAC zijn vooral opties die clustering mogelijk maken. Waar Oracle op dat gebied absoluut mee scoort is de ondersteuning van de verschillende cluster-architecturen.

Voor Redhat Linux ontwikkelde Oracle zelfs haar eigen clusterware, en bracht die in in de Open Source gemeenschap. Ook voor Windows ontwikkelde Oracle clusterware.

Wat maakt nu een grid van een cluster? Een cluster wordt gevormd door een aantal servers (cluster nodes), die binnen enkele meters afstand van elkaar staan. Met een aantal intelligente stukken systeem-software worden de verschillende taken verdeeld over de cluster nodes. Grids als Condor en SETI@home zijn wat dat betreft een stuk beheervriendelijker. De toepassingen zwermen uit over het grid, gebruiken de beschikbare rekenkracht waar mogelijk en verkassen net zo makkelijk naar de volgende server als dat nodig is. Dat werkt alleen bij toepassingen die niet tijdkritisch zijn. Dat is een database-toepassing bijna per definitie wél. Daarom staan de database servers ook vlak bij elkaar en worden de taken nauwgezet verdeeld.

Heeft Oracle met de naam grid nu gewoon een ander woord gelanceerd voor cluster? In feite is er immers niet zoveel veranderd, de systemen moeten nog steeds vlak bij elkaar staan.

De vertraging die in een netwerk optreedt bij 'grote' afstanden maakt het cache-management tussen verschillende systemen gewoon te traag. Bandbreedte is niet eens zozeer het probleem, latency is de bottleneck en dan kunnen enige tientallen meters al een onoverkomelijk grote afstand zijn.

Dat betekent dat het grid voorlopig tot het rekencentrum zal worden beperkt. Dat is ook wat Oracle aangeeft: bedrijven zullen het nooit accepteren dat hun programma's op reis gaan over een wereldwijd grid, zelfs als dat snel genoeg zou zijn. De toepassing zal dus toch tot het rekencentrum worden beperkt. Daarom gebruikt Oracle hiervoor ook de term *enterprise grid* en tot dusver zijn er nog geen argumenten om dat enterprise grid niet gewoon cluster te noemen.

Wat maakt dan wel een grid van dat cluster? Met 10g zet Oracle in op beheer en aanpassingsvermogen. Ook dat is een eigenschap van het oergrid: resources daar halen waar ze beschikbaar zijn, eenvoudig resources toevoegen en ook weer verwijderen. De applicatie draait intussen wel gewoon door. De nieuwe functionaliteit van Oracle Enterprise Manager moet dat mogelijk maken. Oracle 10g draait niet meer op een cluster, waarvan de nodes zorgvuldig in de juiste volgorde één voor één moeten worden gestart en waarbij het toevoegen van een extra node een forse hoeveelheid configuratiewerk betekent. De beschikbare hardware vormt een reservoir waaruit naar behoefte een servertje kan worden bijgeschakeld, als er wat meer rekenkracht nodig is. Als er meerdere databases draaien op de beschikbare hardware, dan kan deze hardware makkelijk aan de ene database worden onttrokken om een andere database van wat meer rekenkracht te voorzien. Configuratie van een nieuwe server gebeurt met een druk op de knop (nadat het besturingssysteem is geïnstalleerd moet men aannemen). Daarmee wordt een voorbeeldconfiguratie naar de betreffende server gekopieerd. Voor de opslag heeft Oracle ASM, Automatic Storage Management ontwikkeld. Schijven bijplaatsen in het NAS of SAN, aanmelden bij Oracle en het RDBMS gaat direct de gegevens op schijf herverdelen. Als er op de schijf 'hot spots' dreigen te ontstaan, worden de data verplaatst om de performance op peil te houden. Zo bezien krijgt het cluster inderdaad een reeks eigenschappen die het in de buurt brengt van het oergrid. Oracle heeft een forse inspanning geleverd met het uitbouwen van de mogelijkheden, nieuw is het allemaal niet. Vanuit de historie bezien wordt er al vanaf versie 5 in deze richting gewerkt, dat is bijna net zo lang als er aan het Condor-project wordt gewerkt. at leidt tot een aantal conclusies.

## Trendvolger en trendsetter

Oracle is trendvolger, er wordt immers overduidelijk gebruik gemaakt van architecturen en oplossingen die eerder door anderen zijn opgezet. Is dat fout? Eind jaren zeventig zagen Larry Ellison c.s. de potentie van het relationele model, zoals dat inmiddels wel was beschreven maar nog niet commercieel verkrijgbaar. Met die stap hebben ze ontegenzeggelijk een belangrij-

ke, zo niet de belangrijkste, bijdrage geleverd aan de doorbraak van het RDBMS in zakelijke toepassingen.

Het grote Oracle ontdekt in grid computing opnieuw een veelbelovende techniek, die in de wetenschappelijke wereld haar sporen al gedurende zo'n 15 jaar heeft verdiend, maar in de commerciële wereld nog niet is doorgebroken. Na de eerste vingeroefeningen met OPS, en daarna vanaf versie 9 met RAC, wordt door Oracle het grid nu in het rekencentrum geïntroduceerd. Grid computing is zeker geen Haarlemmer-olie; grote MMP- of SMP-machines blijven in het voordeel als het gaat om systemen met zeer grote transactie-volumes. Als performance afhankelijk is van het aantal I/O-bewerkingen dat een opslagsysteem kan verwerken, zal dat met het inzetten van een cluster niet verbeteren. De winst zal in dat geval eerst in een verbeterd opslagsysteem moeten worden behaald.

Voor organisaties met tientallen applicaties, vaak ondergebracht in evenveel losse databases op een groot aantal losse servers, biedt 10g beslist mogelijkheden op het gebied van server-consolidatie en data(base)-rationalisatie. Grid computing is niet nieuw, met de adoptie van grid computing is Oracle trendvolger. Met het isoleren van het grid in het rekencentrum wordt de definitie van grid computing geweld aangedaan. Binnen het rekencentrum is grid computing wel nieuw. Op het gebied van zakelijke toepassingen mag grid computing dan ook best als trendsettend worden gezien.

## Conclusie

Ik heb de software nog niet zelf kunnen testen, ik heb er alleen in de Hands-on Labs tijdens Oracle World in Parijs even mee kunnen spelen. Dat is in de verste verte niet genoeg voor een oordeel. Er zijn situaties waar een cluster, of zo u wilt een grid, kan helpen. Er zijn er ook, die ondanks alle beloften, beter af zijn zonder deze extra complexiteit. Vast staat dat Oracle niet over één nacht ijs is gegaan, Oracle werkt al bijna net zolang naar grid computing toe als er aan het Condor-project wordt gewerkt. De namen zijn nieuw, de techniek is vernieuwd en uitgebreid. Aan te raden is om de eigen situatie goed te analyseren en een duidelijke op maat gesneden afweging te maken alvorens een definitieve keuze te maken.

**Carel-Jan Engel** (cjpengel.dbalert@xs4all.nl) is onafhankelijk Oracle-specialist op het gebied van database-ontwerp, trouble-shooting en tuning.

## Literatuur

1. SETI@home: <http://setiathome.ssl.berkeley.edu>
2. M. Litzkow, M. Livny, & M. Mutka: *Condor, a hunter of idle workstations. In Proceedings of the Eighth Conference on Distributed Computing Systems, San Jose, California, June 1988.*  
<http://www.ece.rutgers.edu/~parashar/Classes/02-03/ece572/grid-readings/condor.pdf>
3. M.V. Wilkes: *The Cambridge model distributed system, Operating Systems Review, 1980.*
4. M. Nørgaard, *You probably don't need RAC, Måløv, Denmark, 2002.*  
<http://miracleas.dk/WritingsFromMogens/YouProbablyDontNeedRACUSVersion.pdf>