

Carel-Jan Engel over ervaringen rond het Oracle-rdbms

Alles is Productie

Een software-systeem wordt in meerdere versies actief gehouden. Als het in eigen huis wordt ontwikkeld is OTAP een goed uitgangspunt. Met een Ontwikkel-, Test-, Acceptatie en Productie-omgeving wordt een goede scheiding aangebracht tussen de verschillende ontwikkelstadia van de software. Vaak worden ook één of meerdere stadia weggelaten. In het meest extreme geval hebben we alleen een productiesysteem, en zetten we alle wijzingen direct in. Vooral bij het gebruik van software die door derden wordt geleverd is dat maar al te vaak het geval. In veel organisaties worden de wijzigingen daarop klakkeloos in productie gesmeten. Met besturingssystemen en database software is het vaak niet beter gesteld. Slechts weinigen kunnen het zich veroorloven om een aparte server beschikbaar te houden om patches eerst te beoordelen, voordat ze op de productie-omgeving worden geïnstalleerd. In grote omgevingen met vele databases voor ontwikkelaars, testers en gebruikers zitten de DBA's in een luxe positie: genoeg (ontwikkel)systemen om te testen, toch? Daar is wel enige nuance in aan te brengen.

Een tijd geleden werd ik gevraagd een dagje te komen meedenken over een backup-plan. Nu gaan bij het maken van backup-plannen bij mij meteen de alarmbellen rinkelen. Backup-plannen maak je niet, die leidt je af van een restore/recovery-procedure. Deze wordt opgezet om te kunnen voldoen aan een SLA: hoe lang mag het systeem maximaal uit de lucht, hoeveel data mogen er maximaal verloren gaan. Pas als een restore/recovery-procedure is opgesteld die aan de eisen voldoet, kan worden bepaald hoe de backup moet worden ingericht. Deze moet immers niet simpelweg de gegevens op een tape schrijven, maar ook zodanig toegankelijk maken dat de restore of recovery volgens plan wordt ondersteund. Er werd dus een recovery- en backup-plan doorgesproken. Op zich is dit een prima actie, de ontwikkeling van het systeem stond nog aan het begin. Een paar maanden later werd ik gebeld: er was een erg grote tabel weggegooid. Dat was wel de bedoeling, maar op de test-omgeving. Per ongeluk was men nog verbonden met de productie-omgeving toen de tabel werd weggegooid. Geen probleem zei ik, we hebben een recovery-plan en een backup, we kunnen nu mooi testen. Er is één probleem, was het antwoord, we maken nog geen backups... Het resultaat was dat de data opnieuw moesten worden geladen, iets dat met ettelijke miljoenen rijen even tijd ging kosten. Een paar weken later kreeg ik opnieuw een

telefoontje. Tijdens het testen van een database-reorganisatie was een table space weggegooid, tegelijk met de bijbehorende data files. Geen probleem, zei ik, na het vorige akkefietje hebben we nu toch zeker wel een backup? Er was wel een backup, maar deze was op OS-niveau gemaakt, terwijl de database vrolijk indexen stond aan te maken. Het heeft wat tijd gekost, maar met de nodige trucs waren de data gereed. Er was een verse table space voor aangemaakt en alles was weer toegankelijk. Tijdens de nachtelijke acties brachten we de tijd gedeeltelijk wachtend door, terwijl de gegevens werden gekopieerd of van tape gelezen. Hoe het zover heeft kunnen komen? Er was nog geen backup, omdat het systeem nog niet 'in productie' was.

Alles is relatief. De klant gebruikt het systeem nog niet, maar wat doen die ontwikkelaars dan? Kosten zij geen geld? In mijn optiek zijn alle systemen vanuit het perspectief van de DBA gezien productiesystemen. Of het nu een ontwikkelaar, tester of eindgebruiker is, ze gebruiken de database voor hun werk. Dat werk mag worden verondersteld enig nut voor de organisatie te hebben, anders werd het niet uitgevoerd. De resultaten van dat werk moeten dus worden beschermd, net als de werkomgeving. Het is daarom essentieel dat vanuit het perspectief van de DBA gezien naast de productie-systemen, waaronder ook ontwikkel- en testsystemen, een systeem ter beschikking staat waarmee vrij getest en gespeeld kan worden: nieuwe versies van het OS van de database, het testen van patches en nieuw te gebruiken functionaliteit. Waarom mag een ontwikkelaar wel een testsysteem hebben, en de DBA niet?

Nog even terug naar de werkelijkheid. Er kwam na een paar weken nog een telefoontje. Om aan de naamconventies te voldoen moesten de teruggehaalde gegevens opnieuw worden verplaatst. Bij het opruimen van de oude datafiles ging iets mis. Nu was wel aan de backup gedacht, maar liet de infrastructuur gezien de inmiddels bereikte omvang van het systeem nog niet toe deze goed uit te voeren. Opnieuw was er alleen een OS backup van een actieve database. Deze keer konden ze de gegevens met enige ruggespraak zelf terughalen. De bij de onvolledige backup behorende restore- en recovery-procedure is inmiddels ruimschoots getest.

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