

In dit artikel zal Erwin Groenendal ingaan op de ondersteuning van Java op het Oracle 9i platform. De Oracle database is het bekendste onderdeel van dit platform. Naast de Oracle 9i database bestaat het Oracle 9i platform uit de Oracle 9i Application Server en de Oracle 9i Development Suite. Ieder van deze producten bestaat in feite uit meerdere (deel)producten of onderdelen. Het Oracle 9i platform ondersteunt de ontwikkeling, het beschikbaar stellen en het beheren van Java toepassingen. De focus in dit artikel ligt op de database, application server en JDeveloper (onderdeel van de Oracle 9i Development Suite). De auteur beschrijft de Java-ondersteuning in de database, op de application server en in JDeveloper.



thema

Java op het Oracle 9i platform

Met Oracle 8i introduceerde Oracle "Java in de database". De ondersteuning van Java in de database zelf bestond aanvankelijk uit een Java Virtual Machine (JVM), Java Stored Procedures (waarvoor om verwarring met JavaServer Pages te voorkomen geen acroniem wordt gehanteerd) en J2EE ondersteuning voor onder meer Java Servlets, JavaServer Pages (JSP) en Enterprise Java Beans (EJB). In de meest recente release van de Oracle database, Oracle 9i Release 2, is de ondersteuning voor Java in de database echter in belangrijke mate teruggebracht.

OC4J Het beschikbaar stellen van Java Servlets, JSP's en EJB's wordt niet langer ondersteund vanaf deze release. Dit is opmerkelijk te noemen omdat Oracle nog niet zo lang geleden met de kreet "300% Java" aangaf dat de databaserver dé plek was voor het uitvoeren van (alle) Java code. Oracle kende echter grote problemen met de performance van de JVM in de database zelf. Met het in licentie nemen van de Orion application server van IronFlare beschikt Oracle inmiddels over uitstekende en snelle J2EE containers die onder de naam Oracle Containers for J2EE (OC4J) zijn geïntegreerd met de Oracle 9i Application Server. Klanten worden dan ook geadviseerd om hun J2EE toepassingen in de database te migreren naar OC4J.

Java Stored Procedures behelst het runnen van Java code in de database. Ontwikkelaars kunnen hierbij gebruik maken van de meeste J2SE standaard classes.

Eigen classes worden aangemaakt in de database en de aanroep vanuit PL/SQL (de standaard procedurele programmeertaal in de Oracle database) is mogelijk via wrappers. Oracle ontwikkelaars maken dankbaar gebruik van de vele mogelijkheden die hiermee verkregen worden. Voor het uitvoeren van acties buiten de database waren ontwikkelaars namelijk altijd aangewezen op ingebouwde (systeem) database packages of call

Sinds 9i release 2 streeft Oracle niet meer naar volledige J2EE ondersteuning in de database

outs naar 3GL programmatuur (external procedures). Nu kan, door gebruik te maken van de vele J2SE library's, bijvoorbeeld eenvoudig een file worden gemanipuleerd, communicatie over het netwerk worden uitgevoerd of een e-mail worden verstuurd. Voor code waarin intensieve data manipulatie plaatsvindt verdient PL/SQL echter nog steeds de voorkeur. De PL/SQL code is sneller en minder complex door de volledige integratie van SQL in PL/SQL.

Uiteraard ondersteunt Oracle JDBC. Versie 2.0 wordt volledig ondersteund en versie 3.0 in belangrijke mate. Daarnaast implementeert Oracle Java API's voor bestaande functionaliteit in de database. Bijvoorbeeld

de Java Messaging Service (JMS) API voor Oracle Advanced Queuing (AQ) en de Java Advanced Imaging (JAI) API voor Oracle InterMedia.

JAVA OP DE APPLICATION SERVER De eerste schreden van Oracle op het gebied van application servers, met de Oracle Web Application Server (OWAS) en de Oracle Application Server (OAS), waren niet succesvol te

van OAS. Maar de ondersteuning van J2EE was nog onvoldoende. Zo ondersteunde de JServ module niet de laatste versie van de Java Servlets API.

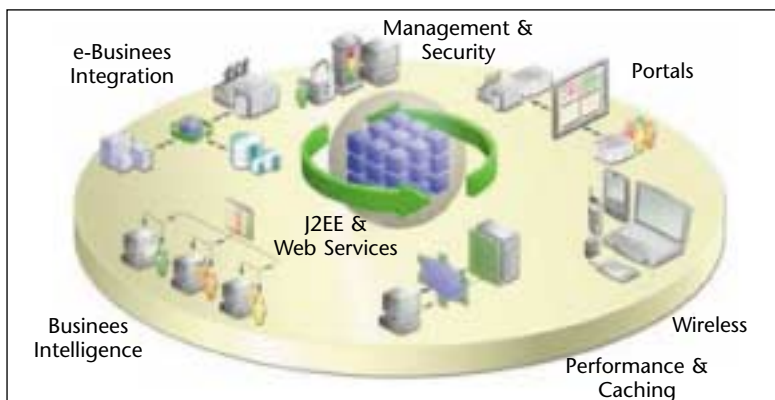
NIEUWE GENERATIE Voor de derde, en huidige, generatie van de application server nam Oracle de uitstekende J2EE containers van Orion over (en noemde deze OC4J: Oracle Containers for J2EE). De nieuwste versie van de Oracle 9i Application Server (9iAS), zoals de derde generatie van de application server heet, is J2EE 1.3 gecertificeerd en voldoet dus volledig aan de meest recente J2EE standaard. De Apache HTTP server wordt nog steeds gebruikt voor web server functies zoals security en het beschikbaar stellen van statische HTML pagina's. In presentaties geeft Oracle

Oracle biedt met User Interface XML een interessant alternatief voor een HTML user interface

toe dat het tot twee keer toe een verkeerde keuze heeft gemaakt met betrekking tot de application server. De eerste keer door met OAS zelf een application server (proberen) te bouwen. De tweede keer door, zoals hierboven beschreven, application server functionaliteit in de database te willen stoppen. Echter met de combinatie van de Apache HTTP Server, OC4J en aanvullende producten is Oracle erin geslaagd een zeer goede application server op de markt te brengen die de concurrentie met bedrijven als IBM en BEA aankan. IBM heeft BEA inmiddels ingehaald in marktaandeel. Oracle is haar grote achterstand hard aan het inlopen en is de nummer drie. Argumenten die voor een keuze voor Oracle 9iAS pleiten zijn de brede functionaliteit (zie afbeelding 1), J2EE 1.3 certificering of de kleine *foot print* (OC4J is slechts enkele Mb's groot ten opzichte van IBM met meer dan 100 Mb) en prijs. Voor veel Oracle gebruikers die aanvankelijk voor een andere application server kozen is 9iAS een aantrekkelijk alternatief. Maar ook voor niet-Oracle gebruikers is 9iAS een zeer serieuze optie. Het is precies deze groep die Oracle graag wil bereiken en met Oracle 9iAS JDeveloper beschikt Oracle over nog een belangrijke troef.

noemen. Instabiliteit, beperkte ondersteuning van standaarden en slechte performance waren er debet aan dat vele Oracle database klanten application servers van onder andere BEA en Apache gingen gebruiken. De tweede generatie application servers van Oracle, de Oracle Internet Application Server (IAS), werd gebaseerd op de populaire open source software van Apache. Oracle leverde met deze versie de HTTP server van Apache en een aantal modules van Apache, waaronder de JServ module voor het runnen van Java Servlets. Naast de standaard modules van Apache bevatte IAS een aantal eigen modules, waarvan de belangrijkste de MOD_PLSQL module is waarmee PL/SQL gebruikt kan worden voor het ontwikkelen van web applicaties. Op deze manier kunnen PL/SQL procedures in de database gebruikt worden om op een eenvoudige en productieve manier dynamische HTML pagina's te genereren. Deze procedures worden soms wel PL/SQL Servlets genoemd. De IAS application server werd gecompleteerd met een reeks aan producten die (in verbeterde vorm) ook in de meest recente versie van de application server opgenomen zijn (zoals Oracle Portal voor het ontwikkelen van portalen). IAS was een enorme verbetering ten opzichte

toe dat het tot twee keer toe een verkeerde keuze heeft gemaakt met betrekking tot de application server. De eerste keer door met OAS zelf een application server (proberen) te bouwen. De tweede keer door, zoals hierboven beschreven, application server functionaliteit in de database te willen stoppen. Echter met de combinatie van de Apache HTTP Server, OC4J en aanvullende producten is Oracle erin geslaagd een zeer goede application server op de markt te brengen die de concurrentie met bedrijven als IBM en BEA aankan. IBM heeft BEA inmiddels ingehaald in marktaandeel. Oracle is haar grote achterstand hard aan het inlopen en is de nummer drie. Argumenten die voor een keuze voor Oracle 9iAS pleiten zijn de brede functionaliteit (zie afbeelding 1), J2EE 1.3 certificering of de kleine *foot print* (OC4J is slechts enkele Mb's groot ten opzichte van IBM met meer dan 100 Mb) en prijs. Voor veel Oracle gebruikers die aanvankelijk voor een andere application server kozen is 9iAS een aantrekkelijk alternatief. Maar ook voor niet-Oracle gebruikers is 9iAS een zeer serieuze optie. Het is precies deze groep die Oracle graag wil bereiken en met Oracle 9iAS JDeveloper beschikt Oracle over nog een belangrijke troef.



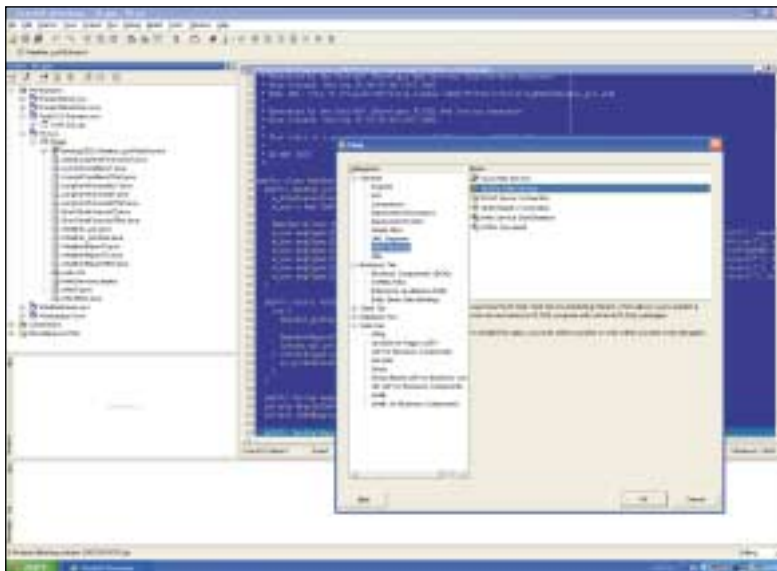
AFBEELDING 1. De brede functionaliteit, J2EE 1.3 certificering en de kleine *foot print* pleiten voor gebruik van Oracle 9iAS

JAVA ONTWIKKELING MET JDEVELOPER Oracle heeft veel ervaring met het maken van ontwikkeltools. Producten zoals de CASE tool Oracle Designer en de 4GL ontwikkeltool Oracle Developer (bestaande uit onder andere Oracle Forms voor de ontwikkeling van schermen en Oracle Reports voor het maken van rapporten) zijn zeer succesvol. In combinatie met de bijbehorende ontwikkelmethode Custom Development Method (CDM) beschikken Oracle gebruikers over krachtige en productieve middelen voor het ontwikkelen van enterprise applicaties. In het begin waren dit terminal (server)applicaties (SQL*Forms 3.0), later client-server applicaties (vanaf Oracle Forms 4.0) en tenslotte (met de Oracle Forms Server, onderdeel van 9iAS) web applicaties. Deze tools hebben een sterke aan-

hang in Nederland. De technologie is echter Oracle-eigen. In lijn met Oracle's strategie om open standaarden te ondersteunen, en Java en XML in het bijzonder, biedt Oracle sinds enige jaren JDeveloper aan als ontwikkeltool voor Java toepassingen.

Met JDeveloper richt Oracle zich op twee groepen. Enerzijds de huidige gebruikers van de CASE- en 4GL-ontwikkeltools die willen overstappen naar Java en anderzijds Java ontwikkelaars. De tweede groep is met name interessant voor Oracle omdat dit een grotendeels nieuwe groep is voor Oracle die nog niet de Oracle application server of Oracle database gebruiken. Voor deze groep zal Oracle zoveel mogelijk standaarden en (in de Java community populaire) raamwerken ondersteunen. Voor de eerste groep, de gebruikers van de oude Oracle tools, wordt gestreefd naar aansluiting op de oude tools, waarin op een declaratieve manier werd ontwikkeld, door veel gebruik te maken van wizards en property sheets. Voor de ontwikkeling van JDeveloper heeft Oracle de compiler technologie van Borland's JBuilder in licentie genomen. De oorspronkelijke Microsoft Windows user interface is inmiddels helemaal herschreven in Java waardoor JDeveloper op vele platformen (waaronder Sun en Linux) te gebruiken is. JDeveloper ondersteunt de ontwikkeling van J2EE applicaties en webservices en de deployment van programmatuur naar de Oracle 9iAS Application Server en andere application servers. De belangrijkste J2EE standaarden voor JDeveloper zijn Java Servlets, JavaServer Pages (JSP) en Enterprise Java Beans (EJB). In JDeveloper worden ook twee Unified Modelling Language (UML) modellen ondersteund: class diagrams en activity diagrams.

ONDERSTEUNING VAN RAAMWERKEN Ontwikkelaars van Java applicaties zijn in de door hen ontwikkelde software, patronen of terugkerende structuren (patterns) gaan ontdekken. Een belangrijk overkoepelend pattern is het Model-View-Controller (MVC) pattern. MVC wordt algemeen beschouwd als de beste applicatiearchitectuur voor J2EE toepassingen. In deze architectuur zijn drie onderdelen van de applicatie min of meer onafhankelijk van elkaar: een deel dat zorgt voor de user interfaces (View), een deel dat de interactie met de database (persistence layer) voor zijn rekening neemt (Model) en een deel dat de aansturing van de view- en model delen verzorgt (Controller). In de ideale MVC architectuur zijn de drie delen volledig onafhankelijk van elkaar en kan bijvoorbeeld het model vervangen worden zonder dat de andere twee delen hoeven worden aangepast. In veel gevallen zijn de drie delen weliswaar netjes gescheiden van elkaar maar niet (geheel) onafhankelijk. Binnen JDeveloper kunnen ontwikkelaars kiezen uit verschillende technologieën voor de drie MVC onderdelen.



AFBEELDING 2. Het tool JDeveloper is op vele platformen -waaronder Sun en Linux- te gebruiken

VIEW Veel ontwikkelaars maken gebruik van JSP's voor de user interface. Een JSP is in feite een HTML pagina waarin binnen speciale tags Java code kan worden aangeroepen voor het genereren van dynamische content. Een JSP wordt door de applicatieserver omgezet in een Java Servlet. Het achterliggende idee van deze opzet is dat vormgevers de HTML ontwikkelen en Java programmeurs de code schrijven voor de dynamische content. In de praktijk bestaat de JSP echter vaak uit bijna alleen maar aanroepen van Java code. Reden hiervoor is dat door het aanroepen van dezelfde Java code voor het genereren van HTML, gemakkelijker een consistente user interface kan worden bereikt. De tol die hiervoor betaald wordt, zijn lastig te begrijpen en moeilijk te onderhouden JSP's. Oracle biedt met UIX (User Interface XML) een interessant alternatief voor het ontwikkelen

In JDeveloper worden ook twee UML-modellen ondersteund: class diagrams en activity diagrams

van een HTML user interface. UIX is een XML taal waarmee op een hoog abstractieniveau de user interface (pagina lay-out) gedefinieerd wordt. Wanneer de pagina moet worden getoond, wordt op basis van de UIX pagina definitie de juiste HTML gegenereerd. Hiermee wordt een consistente user interface gegarandeerd. In de UIX definitie is tevens aangegeven welke data moeten worden opgenomen in de pagina. Het mechanisme waarmee deze *data binding* is aangegeven is onafhankelijk van de technologie die voor het model deel (van de

MVC applicatiearchitectuur) wordt gebruikt. UIX is een technologie die door Oracle zelf is ontwikkeld en beproefd is binnen de standaard Oracle Applications. De technologie sluit echter aan op JavaServer Faces (JSF),

Voor de huidige generatie applicatieservers nam Oracle de uitstekende J2EE containers van Orion over

een toekomstige J2EE standaard. UIX is bovendien volledig geïmplementeerd in Java en past daarmee nu ook al prima binnen J2EE.

MODEL Voor het model deel is JDBC de voor de hand liggende keuze. Hierbij onderschatten ontwikkelaars echter vaak de complexiteit en hoeveelheid werk die erbij komt kijken om de interactie met de (relationele) database te implementeren. In deze interactie zijn vele patterns onderkend die in de regel geïmplementeerd moeten worden. Bij het gebruik van JDBC zullen deze allemaal door de ontwikkelaar zelf moeten worden geprogrammeerd. Om de complexiteit af te schermen en het werk te verminderen biedt JDeveloper twee (elkaar aanvullende) technologieën: TopLink en Business Components for Java (BC4J).

TopLink is een product dat Oracle recent (juni 2002) heeft overgenomen van WebGain (een *venture capitalist* en eigendom van BEA). WebGain had het product in 2000 overgenomen van The Object People (wat het eerste deel van de naam van het product verklaart). Bij de acquisitie door Oracle zijn development, support en consulting afdelingen van WebGain inbegrepen. TopLink, inmiddels omgedoopt in Oracle 9iAS TopLink, heeft een behoorlijk groot marktaandeel en Oracle verkrijgt hiermee toegang tot een groot aantal klanten. Onder de huidige gebruikers van TopLink zijn veel grote namen zoals Sprint, 3M en BMW. Het product TopLink zal als separaat product blijven bestaan en klanten hoeven dus niet te migreren naar een ander (Oracle) product. TopLink sluit vooral goed aan op een top-down manier van werken waarbij een zuiver objectgeoriënteerd (OO) class model wordt gemaakt van de objecten (gegevens) waarmee de applicatie werkt zonder dat hierbij al aan de opslag van deze gegevens in de database (persistence) wordt gedacht. Bij het ontwerpen en implementeren van de gegevensopslag biedt TopLink grote toegevoegde waarde met het definiëren van de (mogelijk zeer complexe) mapping tussen objecten en relationele tabellen en de generatie van Javacode op basis van deze mapping. Voor ontwikkelaars die bottom-up werken en uitgaan van opslag van gegevens in

een relationele database is de toegevoegde waarde van TopLink veel minder. Daarnaast implementeert TopLink niet de vele patterns in het model deel van een MVC applicatiearchitectuur. Voor deze groep ontwikkelaars biedt JDeveloper Business Components for Java (BC4J). Hoewel met BC4J ook vanuit een class model richting relationele opslag gewerkt kan worden, is BC4J vooral gericht op het verkrijgen van objecten op basis van relationele opslagstructuren. Bestaande database modellen kunnen via de BC4J componenten via Java objecten benaderd en gemanipuleerd worden. BC4J implementeert bovendien een groot aantal patterns.

CONTROLLER Een populaire technologie voor het controller deel is Struts van Apache. JDeveloper ondersteunt Struts. Bij deze controller technologie wordt gebruik gemaakt van een XML file waarin staat aangegeven welke Java code class moet worden aangeroepen om bepaalde gebruikersinteractie af te handelen (waaronder aansturing van het model deel). Deze classes hebben een vaste signatuur (implementeren een bepaalde Java interface) en kunnen daardoor dynamisch door de controller worden aangeroepen. Na de aanroep van de class wordt een JSP aangeroepen om een nieuwe pagina te tonen aan de gebruiker (de aansturing van het view deel). Struts werkt eenvoudig, maar heeft daardoor ook zijn beperkingen. Herbruikbaarheid van de Java classes is gering, wat leidt tot nogal monolithische applicaties. Oracle heeft, waarschijnlijk, geïnspireerd door Struts, een veel krachtiger raamwerk ontwikkeld voor het controller deel van een MVC applicatiearchitectuur. Dit raamwerk is oorspronkelijk door de Consulting Services groep van Oracle ontwikkeld op een groot project. Het is daarna verder ontwikkeld en is nu als *Oracle 9iAS MVC framework for J2EE* onderdeel van Oracle 9i Application Server. Herbruikbaarheid en productiviteit zijn de sterke punten. Handige Java ontwikkelaars kunnen overigens prima Struts combineren met het Oracle 9iAS MVC Framework for J2EE.

CONCLUSIE Java lijkt een definitieve en logische plaats te hebben ingenomen in de Oracle database: sterke JDBC ondersteuning, mogelijkheid om (J2SE) Java code uit te voeren en Java API's voor bepaalde database functionaliteit. Ook kan men stellen dat JDeveloper de Java ontwikkelaar zeer goed ondersteunt in het ontwikkelen van J2EE applicaties en webservices.

Erwin Groenendal

Groenendal is technisch directeur van Cumquat Information Technology.