Overzicht functionaliteit van SQL Server 2005 voor een SODA

Service Oriented Database Architectuur

Astrid Hackenberg en Anko Duizer

Microsoft SQL Server 2005 kan het hart vormen van een service georiënteerde architectuur. Als je de functionaliteit van dit product onder de loep neemt, wordt het duidelijk wat voor prominente rol Microsoft SQL Server 2005 kan spelen in toekomstige applicatie-ontwikkeling. En wanneer men de mogelijkheden van SQL Server 2005 kent wordt het ontwerpen van database-applicaties in de toekomst zeker anders!

In een serie van vier artikelen willen we de mogelijkheden van Microsoft SQL Server 2005 bij het realiseren van een service georiënteerde architectuur bespreken. In dit eerste artikel staat centraal wat een Service Oriented Database Architectuur (SODA) is. Daarnaast wordt een overzicht gegeven van de functionaliteit die SQL Server 2005 biedt voor het neerzetten van een SODA. In de volgende drie artikelen worden achtereenvolgens verder uitgewerkt: de rol van de SQL Server Service broker; het belang van .NET managed code; de betekenis van XML en Web Services support.

Wat is een service georiënteerde architectuur (SOA)?

Een service georiënteerde architectuur definieert hoe autonome computerapplicaties met elkaar integreren op basis van de volgende principes:

1. Grenzen zijn expliciet; functionaliteit wordt uitsluitend via voorgedefinieerde services aan andere applicaties aangeboden. De 'buitenwereld' heeft er geen enkel idee van hoe en wanneer een verzoek wordt afgehandeld;
2. Services zijn autonoom; een service-systeem wordt apart *gedeployed*, heeft eigen beveiliging en is verantwoordelijk voor de eigen data;
3. Deel de schema's en contracten; niet de classes, draag zorg voor integratie op basis van berichten (formaten) en niet op basis van objecten;
4. Asynchrone communicatie; wanneer twee applicaties autonoom zijn van elkaar, zijn er geen afhankelijkheden. Dit betekent dat geen van de applicaties het werk onderbreekt om op een ander systeem te wachten;
5. Heterogeen transport; autonome applicaties kunnen kiezen voor een eigen implementatie (technologie), dus moet het transport protocol-onafhankelijk zijn.

Waarom SODA?

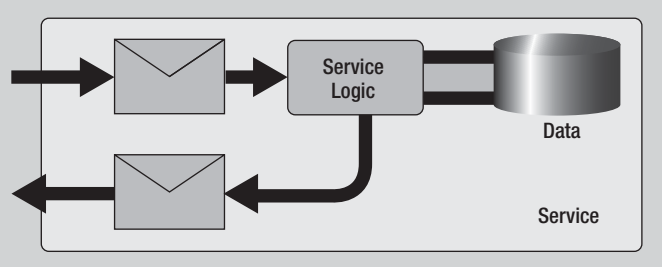
Een SODA is de natuurlijke evolutie van een Service Oriented Architecture (SOA). Bij de ontwikkeling van applicaties wordt steeds vaker gebruik gemaakt van services en de daarbij horende architectuur. Een belangrijk aspect van een SOA heeft te maken met het 'loosely coupled' zijn. Een service biedt via een afgesproken contract diensten aan. Vervolgens wordt er via 'messages' informatie uitgewisseld.

De meeste services gaan over data! Bijna alle vragen die worden gesteld leveren een set aan data op. Het verzoek (de message) is daarnaast zelf ook een gegeven dat interessant is. Kortom, een SOA draait voor een belangrijk deel rondom data zoals weergegeven in afbeelding 1. De database is van nature de plaats om de data te bewaren. Hierbij moet worden opgemerkt dat de service data meestal in een XML-formaat vastgelegd zijn. De data worden intern vaak op een traditionele relationele manier opgeslagen. De service data kunnen worden verdeeld in vier categorieën, zie tabel 1. Om met een database een SOA-architectuur direct te ondersteunen zijn de volgende functionaliteiten vereist:

- Endpoint-ondersteuning, bijvoorbeeld TCP, HTTP GET/PUT of SOAP;
- Mogelijkheid om serviceverzoeken (XML) te verwerken en vertalen naar het interne formaat;
- Een efficiënt en eenvoudig programmeermodel.

Er zijn diverse redenen om de SOA direct in de database te implementeren. Belangrijke redenen om een SOA te veranderen in een SODA zijn:

- Messages, de serviceverzoeken en -antwoorden, zijn data. Dit is interessante informatie die bewaard moet worden en waarover vragen gesteld kunnen worden. Bijvoorbeeld vragen rondom auditing of business-analyse;



Afbeelding 1: SOA en data.

Categorie	Eigenschappen	Voorbeeld
Service interactie-gegevens	- Communicatiegegevens tussen services (berichten) - Formaat moet door alle partijen begrepen worden - Kan nooit wijzigen	- Bestelformulier - Bevestiging
Activiteitgegevens	- Wordt alleen intern in het systeem gebruikt - Privé formaat - Levensduur is de duur van de request - Kandidaat voor Business Intelligence	- Winkelmand - Paklijst
Referentiegegevens	- Read-only voor de service consumer - Wijzigen niet vaak - Kunnen verschillende versies van bestaan - Kandidaat voor replicatie en caching	- Catalogus - Landentabel
Resource-gegevens	- Business-entiteiten - Privé formaat - Hoge concurrent toegang vereist - Transactionele verwerking	- Klanten - Producten

Tabel 1: Overzicht van de data-categorieën in een SODA.

- Wanneer de service en database samensmelten ontstaat een situatie waarin scale-out plaats kan vinden. Op basis van relatief goedkope hardware kan een goede schaalbaarheid worden bereikt. Er kunnen meerdere instanties van SQL Server worden 'gekopieerd' om de serviceverzoeken af te handelen;
- SQL Server 2005 stelt de keuze voor één of twee tiers uit tot de deployment van de service-applicatie.

Wat is SODA?

Een Service Oriented Database Architectuur is de samensmelting van de serviceconcepten met een database. Dit betekent dat de database via services functionaliteit beschikbaar gaat stellen aan de buitenwereld.
In de servicewereld is XML een belangrijke technologie. De boodschappen en contracten worden gemaakt in XML. Alle informatie die wordt uitgewisseld is in een XML-formaat. De database moet daarom in staat zijn om goed met XML om te gaan en de XML op te slaan. De database moet vervolgens in staat zijn om de XML te verwerken en verzoeken te beantwoorden. Hiervoor is het noodzakelijk om te kunnen programmeren in de database.
In de servicewereld is het belangrijk om autonoom te kunnen werken en zo min mogelijk afhankelijk te zijn van de 'ander'.
Er wordt vaak gebruikt gemaakt van queue's om deze afhankelijkheid te verkleinen. Het ondersteunen van een queuing mechanisme is een ander mogelijk aspect van een SODA.
Ten slotte moet het mogelijk zijn om serviceverzoeken te ontvangen in de database en te beantwoorden. Hiervoor moet de database dus ondersteuning bieden, bijvoorbeeld door functionaliteit naar buiten te brengen via webservices. In afbeelding 2 is de samensmelting van services en databases schematisch weergegeven.

Wat biedt SQL Server 2005?

Om een SODA te implementeren is het nodig dat de database de beschreven functionaliteit biedt. Microsoft SQL Server 2005 lijkt ontworpen met de SODA-wereld in gedachten. SQL Server 2005 biedt veel nieuwe functionaliteit, tabel 2 geeft daar een beknopt overzicht van.

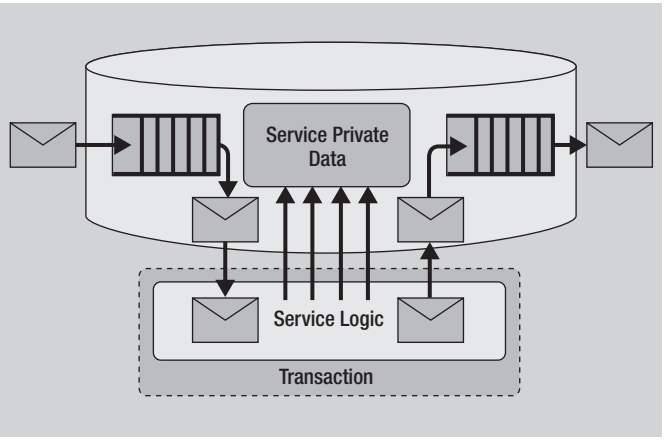
Microsoft SQL Server 2005 kent een flink aantal verbeteringen ten opzichte van de voorganger SQL Server 2000. Naast belangrijke verbeteringen voor een SODA hebben de meeste andere verbeteringen en vernieuwingen betrekking op Business Intelligence. Ze worden daarom in deze artikelenreeks buiten beschouwing gelaten.

XML support

De eerste aanzet voor de ondersteuning van XML is gemaakt met SQL Server 2000, maar de implementatie in SQL Server 2005 gaat veel verder. In deze nieuwste versie van SQL Server is XML een volwaardig onderdeel geworden van de engine. De onderstaande voorbeeldcode toont dat er een datatype XML is opgenomen. Hierbij kan XML worden opgeslagen in de database. Tabeldefinitie met een XML-kolom:

```
create table XmlTable (
id int identity not null,
xdoc xml not null,
CONSTRAINT [PK_id] PRIMARY KEY CLUSTERED ([id]) )
```

Op het eerste gezicht is dit niet echt een spectaculaire ontwikkeling maar eerder een evolutie. Bij nader onderzoek blijkt dat dit



Afbeelding 2: Overzicht van SODA in een database.

datatype in alle opzichten een 'normaal' datatype is. Zo kan er bijvoorbeeld worden geïndexeerd op basis van een XML datatype. En aan het datatype kan een XSD worden gekoppeld, zodat de inhoud van de XML wordt gevalideerd.
Een ander belangrijk aspect van de XML-technologie in SQL Server 2005 is de mogelijkheid tot het zoeken in de XML. Het traditionele SELECT statement kan worden verijkt met een XQuery statement om te zoeken in de inhoud van een XML-veld.

Functionaliteit	Omschrijving
.NET CLR ondersteuning	De SQL Server engine is een host geworden voor de .NET Common Language Runtime (CLR). Dit heeft tot gevolg dat binnen SQL Server 2005 code kan worden uitgevoerd die geprogrammeerd is in een .NET taal zoals C# of VB.NET. Hierdoor kunnen bijvoorbeeld stored procedures worden geschreven in C#.
Integration services	Een sterk vernieuwde versie van de voormalige Data Transformation Services (DTS). In integration services zijn de datatransformatie (datapipeline) en de logica (flow) van elkaar gescheiden. Beide kunnen grafisch worden gemodelleerd.
Management studio	De vernieuwde omgeving om SQL Servers mee te beheren.
HTTP support	SQL Server 2005 heeft de mogelijkheid om als HTTP endpoint te fungeren. Hiervoor is een eigen Windows HTTP service beschikbaar.
XML support	SQL Server 2005 behandelt XML in alle opzichten als een volwaardig datatype.
Reporting services	Een platform voor het maken, beheren en uitleveren van rapportages. Je kunt het integreren in je applicatie of als een op zich zelf staande omgeving gebruiken. Er zitten ook tools bij voor het maken, beheren, tonen en bezorgen van rapporten.
Service broker	Functionaliteit die het mogelijk maakt om berichten via queue's asynchroon uit te wisselen. De basis voor een SODA.
Analysis services	De bestaande analysis services zijn uitgebreid. Analysis services is onder andere uitgebreid met een aantal nieuwe algoritmes, bijvoorbeeld tijdreeksen. Daarnaast hebben cubes extra functionaliteit gekregen zoals het key performance indicator framework.

Tabel 2: Overzicht van de belangrijkste nieuwe SQL Server 2005 functionaliteit.

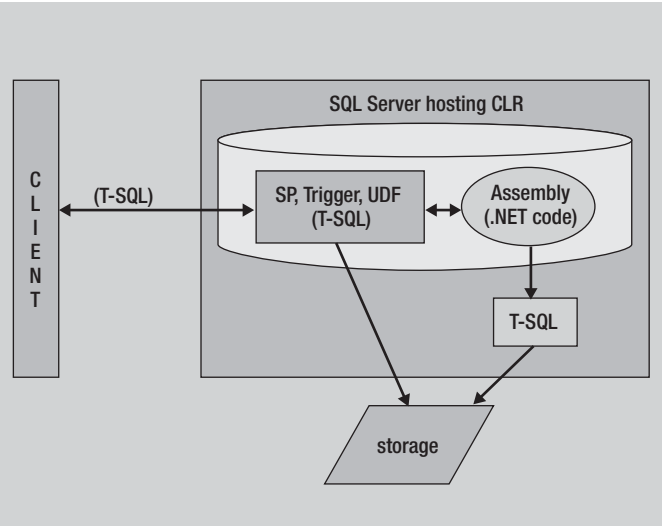
Dit kan leiden tot zeer krachtige zoekopdrachten. Bovendien kunnen de XML data op deze manier ook worden gemuteerd. In SQL Server 2005 worden de relationele data verenigd met de hiërarchische XML-variant! Voor een SODA architectuur is dit een belangrijke ontwikkeling, omdat de servicewereld beide varianten nodig heeft. De berichtuitwisseling vind plaats op basis van XML-berichten, de werkelijke data worden relationeel bewaard.

HTTP endpoint

Om een SODA architectuur te ondersteunen moet de database 'endpoint' ondersteuning hebben, zoals eerder vermeld. In combinatie met Windows Server 2003 worden HTTP endpoints ondersteund. Op basis van deze HTTP endpoints is SQL Server 2005 in staat om functionaliteit via webservices (SOAP) naar buiten te brengen. SQL Server is in staat om te luisteren naar serviceverzoeken op een bepaalde port. Aan de web service kunnen vervolgens stored procedures worden gekoppeld om de benodigde functionaliteit te implementeren.

> De relationele data worden verenigd met de hiërarchische XML-variant

Normaal gesproken moet een client een SQL Server footprint installeren om met de database engine te communiceren. Dus wanneer een stored procedure aangeroepen wordt, is er aan de client-kant een stukje logica nodig om de call werkelijk te realiseren. Deze code maakt gebruikt van het zogenaamde specifieke SQL Server TDS (Tabular Data Stream) protocol om te communiceren met de server. Bij het gebruik van een HTTP endpoint is er geen client footprint. Via een SOAP-aanroep kan een stored procedure worden uitgevoerd. De aanroep kan van ieder willekeurig platform komen.



Afbeelding 3: SQLCLR architectuur.

SQLCLR

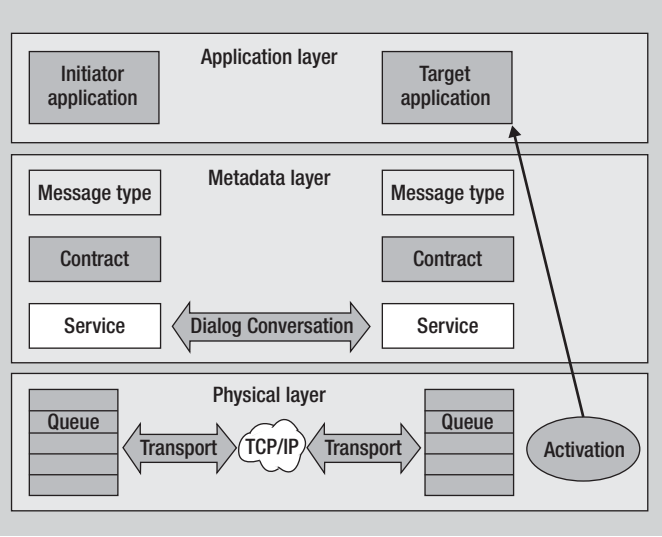
De database engine is een host geworden voor de .NET runtime (CLR). Deze integratie tussen de database- en de .NET-wereld heeft verstrekkende mogelijkheden. De CLR draagt zorgt voor een volwaardige programmeeromgeving in de database; een omgeving waarin een krachtige taal als C# gebruikt kan worden. De integratie tussen SQL Server en de CLR heeft tot gevolg dat bijvoorbeeld stored procedures en triggers geschreven kunnen worden in een .NET-taal. Dit biedt onder andere uitgebreide mogelijkheden voor de afhandeling van strings en XML in een stored procedure of trigger. Overigens is het schrijven van stored procedure in een .NET-taal een keuze. Nog steeds kunnen stored procedures en triggers worden geschreven in T-SQL.
Voor de servicewereld is het schrijven van stored procedures in een .NET-taal het meest interessant. De integratie gaat echter veel verder. Er kunnen bijvoorbeeld ook functies en eigen typen worden geschreven in .NET. Met het schrijven van een eigen type kan het bestaande SQL Server type-systeem worden uitgebreid met objecten. Er kan bijvoorbeeld een type persoon worden gedefinieerd met de bijbehorende kenmerken. Dit type kan vervolgens als object worden opgeslagen in de database. De integratie tussen SQL Server en .NET betekent programmeren in een .NET-omgeving en vervolgens deployen in de database, zodat er gebruik van kan worden gemaakt. Dit wordt weergegeven in afbeelding 3.

SQL Service Broker

De SQL Service Broker (SSB) implementeert de combinatie van queue's en services in de database. Voor de gemiddelde programmeur is het eenvoudiger om te programmeren met services dan met berichten en queue's. Daarom zorgt SSB bovenop het queue-model voor een abstractie door middel van services. Dit past uitstekend bij de SODA-gedachtewereld.
SSB zorgt voor betrouwbare dialogen die tussen services worden opgezet. Achter een service kan een programma worden aangeroepen (automatisch of handmatig). Een serviceprogramma wordt geïmplementeerd in een stored procedure. Een dergelijke stored procedure kan eventueel draaien in de CLR. Hier komt de integratie met .NET weer om de hoek kijken.
De dialogen kunnen zowel lokaal als 'remote' zijn. Op deze manier kan SSB een onderdeel zijn van een scale-out strategie. SSB draagt zorg voor de basisvereisten van een service-oplossing. De berichten worden gegarandeerd slechts eenmaal verzonden. Dit kan onderdeel zijn van een SQL Server transactie.
SSB introduceert een aantal nieuwe database-objecten en uitbreidingen aan T-SQL. De nieuwe objecten zijn getoond in afbeelding 4. In het volgende artikel in deze SODA-reeks behandelen we de SQL Service Broker en zijn rol binnen een SODA verder.

Conclusie

De wereld van service-oriëntatie is inmiddels niet meer weg te denken. De meeste services zijn sterk gekoppeld aan databases. Op basis van deze gedachte is een alternatieve architectuur ontstaan: SODA. Microsoft SQL Server 2005 is een van de eerste



Afbeelding 4: SSB architectuur.

databases die voldoende functionaliteit biedt om een SODA compleet in te richten.
Met de XML- en Webservices-functionaliteit in de database is het fundament gelegd voor een service georiënteerde database-architectuur. Het huwelijk tussen .NET en SQL Server draagt zorg voor de broodnodige programmeermogelijkheden. De komst van de CLR in SQL Server introduceert een rijke en betrouwbare omgeving die geprogrammeerd kan worden met vertrouwde tools zoals Visual Studio.NET. Tenslotte zorgt SSB voor de asynchroniteit en schaalbaarheid van de SODA-oplossing in SQL Server 2005.
De komst van SQL Server 2005 geeft nieuwe mogelijkheden voor een applicatie-architectuur. Er zijn veel applicaties in de wereld die roepen om een SODA-oplossing; Microsoft SQL Server 2005 biedt concreet de mogelijkheden.

> De database engine is een host geworden voor de .NET runtime

Astrid Hackenberg en **Anko Duizer** zijn beiden werkzaam bij Class-A als trainer/coach en medeoprichter. Class-A is een Microsoft kenniscentrum (www.class-a.nl).

Meer weten?

http://research.microsoft.com/research/pubs/view.aspx?tr_id=983
http://research.microsoft.com/~Gray/QueueIsDB.doc
www.pathelland.com
www.ankoduizer.nl
www.class-a.nl