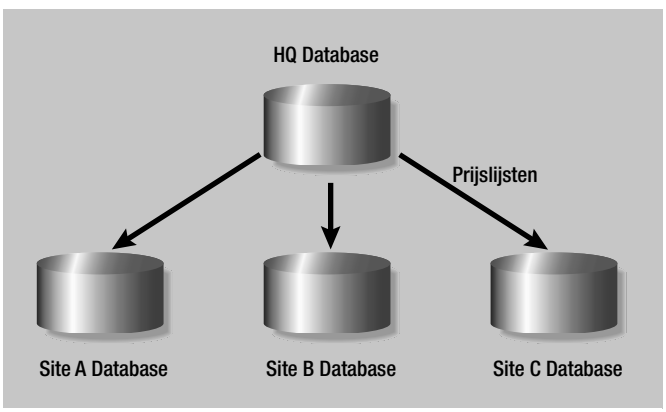
Oracle Multimaster Replicatie

Gedistribueerde databasesystemen

Overal om ons heen hebben we te maken met gegevens. Veel van deze gegevens worden opgeslagen in database systemen. Om vanuit verschillende locaties deze gegevens te kunnen gebruiken als informatie moeten de database systemen te benaderen zijn. Eén centraal databasesysteem kan in een aantal situaties niet voldoende zijn om alle gegevens te verzamelen. Om toch over alle gegevens te kunnen beschikken kunnen verschillende database systemen gebruik maken van* replicatie *om onderling gegevens uit te wisselen.

Oracle database systemen hebben de mogelijkheid om gebruik te maken van diverse vormen van replicatie: *snapshot replicatie* en *multimaster replicatie*.

- Snapshot replicatie is een vorm van replicatie waarbij de gegevens vanuit de '*master*' database gedistribueerd worden naar de '*snapshot*'database(s). De gerepliceerde gegevens in de snaphot database zijn *'read-only*' en kunnen alleen in de masterdatabase worden gemuteerd. Een voorbeeld van een toepassing met behulp van snaphot replicatie is een winkelketen, waarbij de prijslijsten vanuit het hoofdkantoor up-to-date gehouden worden. In figuur 1 is een schematische weergave van de replicatie weergegeven.



Figuur 1

- Een meer geavanceerde methode van replicatie is *multimaster replicatie*. Deze vorm van replicatie is te gebruiken wanneer alle betrokken databases ´gelijkwaardig´ zijn. De objecten in een multimaster omgeving worden in een '*replicatie groep*' opgenomen, waarmee het gedrag van de replicatie objecten gedefinieerd kan worden. Alle objecten in een replicatie groep kunnen op alle betrokken databases worden gemuteerd.

Er zijn twee typen replicatie mogelijk met multimaster replicatie: *asynchroon* en *synchroon* replicatie.

- In een asynchrone replicatie-omgeving worden mutaties in een replicatiegroep locaal bijgehouden en worden de wijzigingen in een queue geplaatst. Op een te definiëren tijdsinterval worden de wijzigingen doorgevoerd op de *remote* database(s). Met deze vorm van replicatie zijn alle betrokken databases op verschillende tijdstippen weer gelijkwaardig.
- Een synchrone replicatie-omgeving, ook wel real-time omgeving genoemd, voert mutaties uit op alle betrokken databases als één transactie. Wanneer een mutatie op één van de betrokken databases niet doorgevoerd kan worden, zal de transactie door middel van een rollback ongedaan gemaakt worden en zal een foutmelding in de error-queue worden geplaatst.

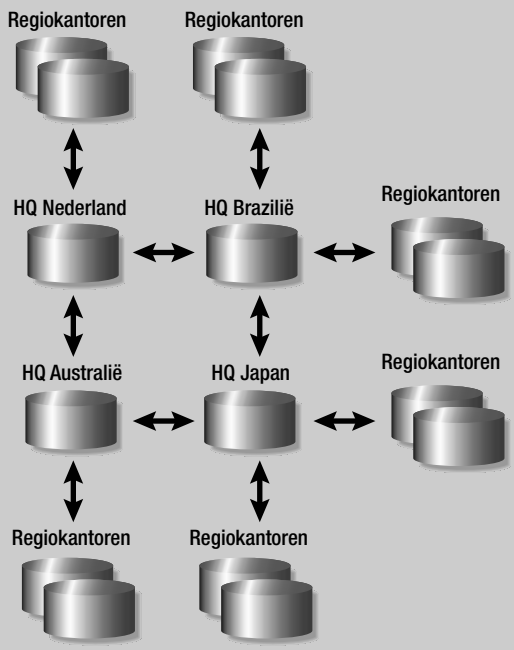
Conflictoplossing

Het gebruik van multimaster replicatie brengt conflicten met zich mee, doordat mutaties op meerdere locaties op ongeveer dezelfde tijdstip optreden. De conflicten kunnen worden gegroepeerd in drie typen: *update-*, *uniqueness-* en *delete conflicts*.

- Een update conflict kan optreden wanneer op twee verschillende locaties wijzigingen op een zelfde database record plaatsvinden in een tijdsinterval voordat de transacties in de replicatie queue daadwerkelijk is doorgevoerd naar de overige deelnemende databases.
- Een uniqueness conflict ontstaat wanneer het doorvoeren van een mutatie op een database record kan leiden tot het overtreden van de entiteit integriteit, zoals een *primary key* of *unique constraint*.

Bedrijf X

Een voorbeeld van een toepassing van (asynchroon) multimaster replicatie is het up-to-date houden van producten en diensten voor een internationaal bedrijf: Bedrijf X is gevestigd in vier landen: Brazilië, Nederland, Japan en Australië. Op iedere locatie zijn een aantal regiokantoren van bedrijf X gevestigd. Bedrijf X heeft een groot aantal andere internationale bedrijven als klant en wil daarom dat alle producten en diensten over de vier landen geregistreerd wordt. Hierdoor kan bedrijf X het totale gedrag van de klant analyseren en daarop beter op reageren en aanbiedingen op maat ontwikkelen (zie ook figuur 2).



Figuur 2. Een opzet van gedistribueerde databases systemen met (asynchroon) multimaster replicatie

De regiokantoren kunnen nieuwe producten en diensten registreren die doorgevoerd worden naar het hoofdkantoor en vervolgens worden doorgevoerd naar de overige gedistribueerde databases over de verschillende landen. Regiokantoren van andere landen kunnen deze nieuwe producten en diensten van klanten die ook in de eigen regio actief zijn aanbieden.
De initiële implementatie van een multimaster replicatie voor bedrijf X bestaat o.a. uit de volgende stappen:
Het analyseren en definiëren van replicatie objecten in de database en de daarbij betrokken master site databases.
Het definiëren van replicatie groep(en) waarin het gedrag van de replicatie objecten gedefinieerd kunnen worden. Het bepalen van de replicatie-eigenschappen, zoals de type replicatie en tijdsinterval waarin de replicatie plaats moet vinden.

- Een delete conflict kan optreden wanneer twee transacties op verschillende locaties worden uitgevoerd, waarbij de eerste transactie een database record verwijderd en de tweede transactie dezelfde database record update of verwijderd. Doordat de database record niet meer bestaat, zal de tweede transactie een melding in de error-queue opleveren.

Detectie

Hoe detecteert een Oracle database deze conflicten? De ontvangende mastersite database in een replicatie omgeving ontvangt informatie over de waarde voor de mutatie en de waarde na de mutatie vanuit de verzendende mastersite database. Hiervoor worden twee database procedures voor gebruikt:

```
DBMS_REPCAT.SEND_OLD_VALUES(
   sname            IN  VARCHAR2,
   oname            IN  VARCHAR2,
   { column_list    IN  VARCHAR2,
   | column_table   IN  DBMS_UTILITY.VARCHAR2s |
DBMS_UTILITY.LNAME_ARRAY,}
   operation        IN  VARCHAR2 := 'update | delete | * ',
   send             IN  BOOLEAN  := true | false );

DBMS_REPCAT.COMPARE_OLD_VALUES(
   sname            IN  VARCHAR2,
   oname            IN  VARCHAR2,
   { column_list    IN  VARCHAR2,
   | column_table   IN  DBMS_UTILITY.VARCHAR2s |
DBMS_UTILITY.LNAME_ARRAY,}
   operation        IN  VARCHAR2 := 'update | delete | * ',
compare          IN  BOOLEAN := true | false );
```

Parameter	Omschrijving
sname	schema naam waarin het object zich bevindt
oname	naam van de te repliceren tabel
column_list	een comma-delimited lijst van kolommen in de tabel
column_table	een procedure om een lijst van kolommen in de tabel te genereren
operation	type operatie: update, delete of '*' voor update en delete
send	true: verzenden/vergelijken van oude waarden false: niet verzenden/vergelijken van oude waarden

Tabel 1: procedure send_old_values / compare_old_values parameter lijst

De ontvangende mastersite database detecteert een update conflict, indien de oude waarde voor de mutatie niet overeenkomt met de locale waarde in de database. Een uniqueness

conflict wordt gedetecteerd wanneer de ontvangende mastersite database een '*uniqueness constraint violation*' melding ontvangt wanneer de insert of update transacties op de lokale database worden uitgevoerd. Een delete conflict wordt gedetecteerd wanneer de ontvangende master site database geen primary key van de betreffende record kan vinden om een delete of update transactie door te voeren.

Om de genoemde conflicten op te kunnen lossen kan gebruik gemaakt worden van de *Conflict Resolution methode*. Met deze methoden kunnen regels worden opgesteld waardoor de mutaties volgens de gedefinieerde regels worden uitgevoerd indien een conflict optreedt. Enkele voorbeelden van regels waarop de conflict resolution methoden gebruikt kan worden zijn gebaseerd op:

- Minimum/maximum waarden
  indien een update conflict optreedt, waarbij de door te voeren waarde kleiner/groter is dan de huidige waarde zal de kleinste/grootste waarde worden doorgevoerd op de ontvangende master site database. In een situatie waarbij de door te voeren waarde gelijk is aan de huidige waarde, zal het conflict resolution mechanisme niet gebruikt worden en worden er geen wijzigingen in de lokale database plaatsvinden.
- Recente of oudste datum (timestamps)
  de wijziging die het eerst/laatst is uitgevoerd zal worden doorgevoerd in de ontvangende master site database.
- Additive
  De additive methode werkt uitsluitend met kolommen met numerieke waarden. Indien een conflict optreedt, wordt het verschil tussen de oude waarde en de nieuwe waarden bij de huidige waarde opgeteld.

```
current value = current value + (new value - old value)
```

- Priority groups
  Met priority groups kunnen prioriteitswaarden worden toegewezen aan mogelijke waarden die in de transacties kunnen voorkomen. Indien een conflict optreedt, zal de transactie met de grootste prioriteitswaarde worden doorgevoerd en de transactie met de laagste prioriteitswaarde ongedaan worden gemaakt.

In release 8i en 9i zijn oude beperkingen op het gebied van multimaster replicatie niet meer aanwezig

Prioriteitswaarde

Een voorbeeld van een conflict resolution methode die gedefinieerd kan worden voor bedrijf X is op basis van Priority Groups (site-name). Iedere locatie krijgt een prioriteitswaarde toegewezen, waarbij de wijzigingen vanuit het hoofdkantoor groter zijn dan de wijzigingen door de regiokantoren:

Priority Group – Site Name	Prioriteitswaarde
Brazilië	4
Japan	3
Australië	2
Nederland	1

Indien een mutatie op een zelfde product of dienst vanuit Nederland en Brazilië wordt uitgevoerd, zal de wijziging van Brazilië (prioriteitswaarde 4) worden doorgevoerd naar de overige locaties en zal de wijziging van Nederland (prioriteitswaarde 1) ongedaan worden gemaakt (Brazilië heeft een hogere prioriteitswaarde dan Nederland).

Toepassingsgebieden

Het doel van replicatie is om er zeker van te zijn dat data beschikbaar is waar en wanneer dit nodig is. Hierdoor kan het gebruik van multimaster replicatie voor de volgende doelen gebruikt worden:

Failover

- Multimaster replicatie kan gebruikt worden om de beschikbaarheid van '*mission critical*' databases te waarborgen. Indien een master database door een netwerk- of systeem storing niet beschikbaar is, kan gebruik gemaakt worden van een andere masterdatabase, omdat deze gelijkwaardig zijn. Applicaties die gebruik maken van de masterdatabase kunnen blijven functioneren.
- Oracle Net kan geconfigureerd worden om een automatische '*connect-time failover*' te gebruiken, waarmee Oracle Net automatisch overschakelt naar een andere master database site indien de eerste master database niet meer beschikbaar is. De functionaliteit van connect-time failover kan worden ingesteld door de FAILOVER parameter met de waarde ON in tnsnames.ora op te nemen en meerdere connect descriptors te specificeren.

```
sales.us.acme.com=
  (DESCRIPTION=
    (ADDRESS_LIST=
    (LOAD_BALANCE=off)
    (FAILOVER=ON)
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-server)(PORT=1521))
    (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-server)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com)))
```

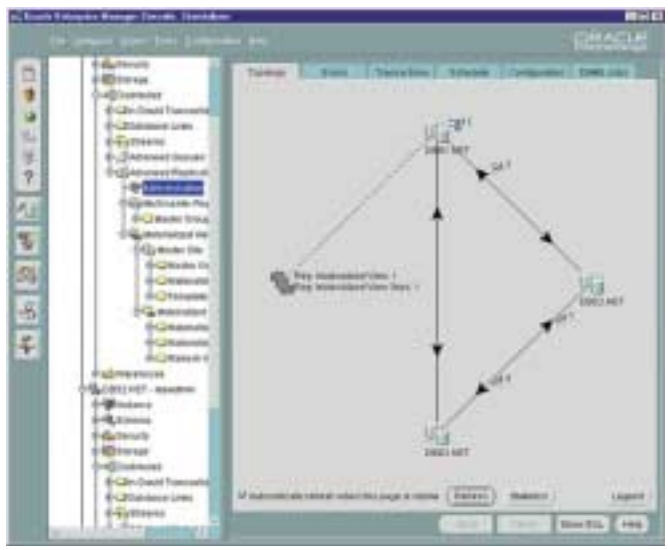
Load Balancing

Multimaster replicatie kan goed gebruikt worden voor '*transaction processing*' applicaties, waarbij de database informatie op verschillende locaties benaderd kunnen worden voor:

- Distributie van grote hoeveelheden data
- Waarborgen van continue beschikbaarheid
- Aanbieden voor gelokaliseerde data toegang

Tools

Om een multimaster replicatie omgeving op te zetten kan gebruik worden gemaakt van tools, zoals Oracle Enterprise Manager met de Replication Management Tools. In figuur 3 is een voorbeeld van een multimaster replicatie omgeving met bijbehorende topologie weergegeven.



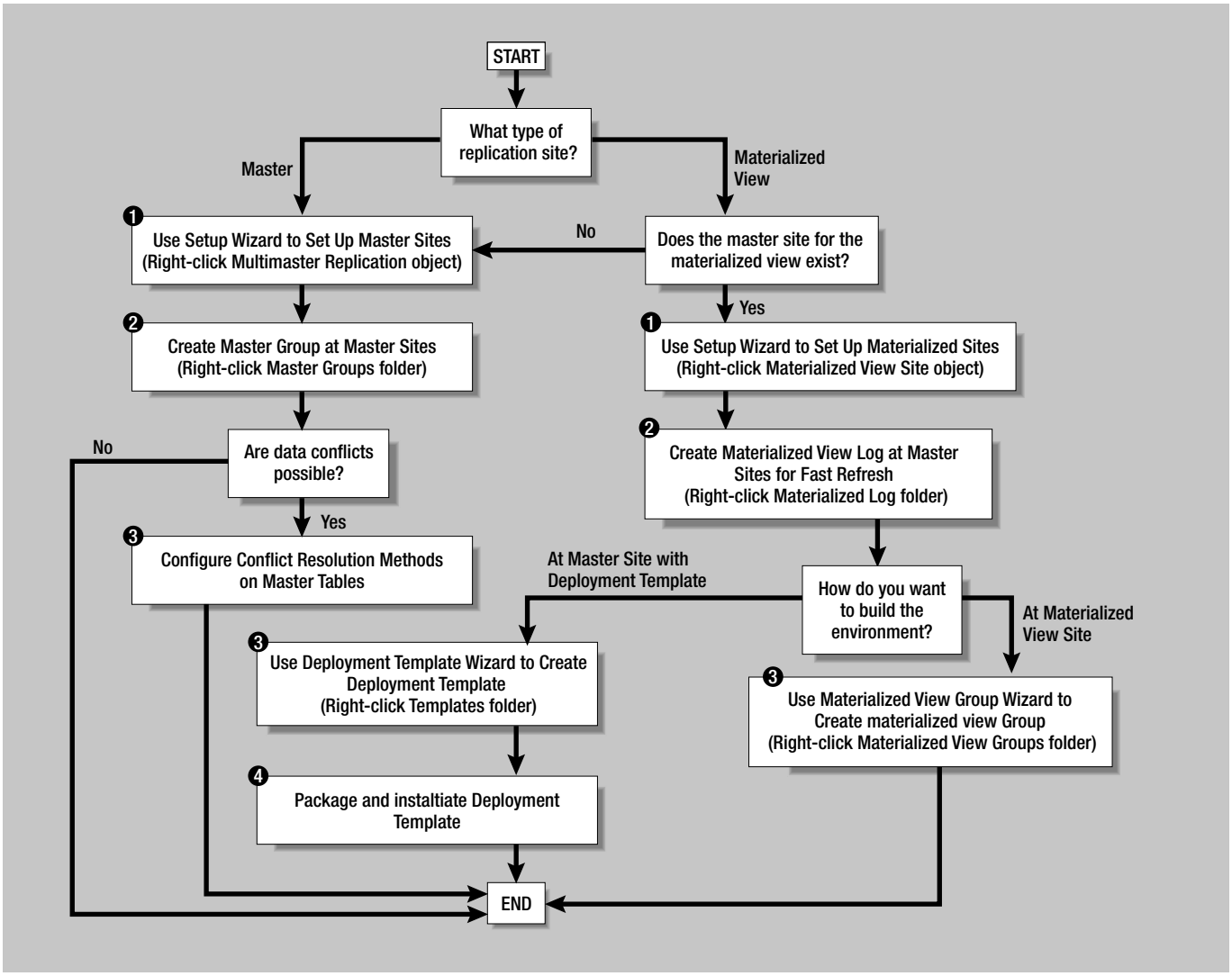
Figuur 3. Topologie van een multimaster replicatie omgeving

In de onderstaande flowchart zijn (globaal) de stappen weergegeven waarmee een replicatieomgeving opgezet kan worden met Oracle Enterprise Manager en de daarbij behorende Replication Management wizards.

Naast het gebruik van grafische tools kan ook gebruik worden gemaakt van de Replication Packages in de Oracle database (DBMS_REPCAT), waarmee de volledige implementatie en beheer van Multimaster Replicatie uitgevoerd kan worden.

Nieuwe ontwikkelingen

De ontwikkelingen van replicatie hebben niet stil gestaan en zijn in de loop van de tijd verbeterd. Release 7.3 van de Oracle database heeft een beperkte mogelijkheid voor multimaster replicatie. In release 8i en 9i zijn de ontwikkelingen op het gebied van multimaster replicatie voortgeschreden en zijn de beperkingen niet meer aanwezig. Een voorbeeld van een beperking in Oracle 7.3 is het niet kunnen wijzigen van de multimaster



Figuur 4. Deze flowchart geeft een globaal overzicht van de stappen voor het opzetten van een replicatieomgeving met Oracle Enterprise Manager

replicatie groepen terwijl replicatie actief is. Wanneer een wijziging in een groep aangebracht moet worden, dient de hele replicatie gestopt te worden. Na de wijziging dient replicatie opnieuw geactiveerd te worden. In Oracle 8i en 9i kunnen deze groepen, waar geen onderhoud in plaatsvinden, actief blijven. Nadat de wijzigingen in de groep zijn doorgevoerd kan de replicatie voor de groep weer worden geactiveerd.

Replicatie versus RAC

Een andere manier om databases te koppelen, met onder meer als doel een hogere beschikbaarheid en betrouwbaarheid te verkrijgen, is RAC (Real Application Clustering). Hierbij wordt echter een geheel andere techniek gebruikt. Het grootste verschil tussen RAC en Replicatie is dat bij RAC er een virtuele eenheid van databases ontstaat en alles realtime verwerkt wordt. Bij replicatie zijn vertragingen tussen de diverse databases toegestaan en zijn de locaties als zelfstandige 'sites' herkenbaar. Daarnaast wordt replicatie veel gebruikt om een aantal database objecten synchroon te houden terwijl RAC de gehele database omvat. Meer hierover in het tweede deel uit deze artikelserie, dat in de volgende Optimize zal verschijnen.

Conclusie

Multimaster replicatie is een goede oplossing om gedistribueerde database systemen up-to-date te houden en om 'failover/load balancing' oplossingen te kunnen gebruiken voor kritieke database systemen die continue beschikbaar dienen te zijn.

Ing. Shaktiewant Adhien
Dhr. Adhien is Oracle Applications DBA bij Qualogy Applications B.V.