

Banco de dados distribuídos Oracle

Diogo Maron¹, José Torquato¹, Persis Castro Alves¹

¹Departamento de Ciência da Computação – Instituto de Matemática -
Universidade Federal da Bahia (UFBA) – Salvador – BA – Brasil

{dmaron, torquato, palves}@dcc.ufba.br

Abstract. *The Oracle is one of the most widely used products in the world for data storage. This article defines concepts and features exhibits of distributed database using the Oracle architecture.*

Resumo. *O Oracle é um dos produtos mais utilizados no mundo para armazenamento de dados. Este artigo define conceitos e expõe características de banco de dados distribuídos que utilizam a arquitetura Oracle.*

1. Introdução

Um banco de dados distribuído Oracle é composto de diversos bancos de dados armazenados em múltiplos computadores, porém sendo visto pelo usuário como um banco de dados local. Funcionalmente, os dados distribuídos aos diferentes computadores podem ser simultaneamente acessados e modificados por qualquer computador da rede.

Para um relacionamento eficiente entre os diversos bancos de dados, desde o início da sua concepção até sua implementação, faz-se necessário distribuí-los de forma bem planejada. Eventualmente, com a expansão das empresas, as exigências aumentam e um eficiente planejamento pode ser afetado. Porém, a relevância em ambos os casos é a capacidade de replicação ou referência de dados entre um ou mais bancos de dados remotos.

A tecnologia de banco de dados distribuídos Oracle permite a administração e manipulação de dados oferecendo um complexo conjunto de ferramentas e aplicativos. Um servidor Oracle controla cada um dos seus bancos de dados, mas mantém a coerência do banco de dados global porque possui a capacidade de interligação entre os bancos de dados distribuídos como se fosse um só. Logo, são permitidas pesquisas, inclusões, exclusões e transações distribuídas, retenção, replicação e particionamento em diferentes locais.

Usuários de bancos de dados distribuídos são beneficiados com o Oracle, um novo nível de sistemas de computação. Dados entre vários servidores podem ser compartilhados através de consultas e atualizações distribuídas com a garantia do mecanismo two-phase commit na consistência dos dados. Através da replicação de dados, os usuários podem criar cópias de leitura de tabelas com consistência transacional e garantia de integridade de dados. Dados remotos podem ser tratados como se fossem locais por meio de database links. Sendo que, havendo transferência de dados de um nó a outro não implica na recodificação dos aplicativos.

2. Arquitetura

Um sistema de banco de dados distribuídos permite que aplicações acessem dados de bancos de dados locais e remotos. Em um sistema de banco de dados distribuídos

homogêneos, cada banco de dados é um banco de dados Oracle. Enquanto que um sistema de banco de dados distribuídos heterogêneos, pelo menos um dos bancos de dados não é um banco de dados Oracle. Bancos de dados distribuídos usam arquitetura cliente/servidor para processar solicitações de informação.

2.1. Bancos de Dados Distribuídos Homogêneos

Um sistema de banco de dados distribuídos homogêneo é uma rede de dois ou mais bancos de dados Oracle que residem em uma ou mais máquinas. Considere três bancos de dados distribuídos Mn, Se, Vd (Figura1). Uma aplicação pode simultaneamente acessar ou modificar os dados em vários bancos de dados em um único ambiente distribuído. Por exemplo, uma pesquisa a partir de um único cliente Manufatura no banco de dados local Mn pode recuperar dados agregados a partir da tabela *produtos* do banco de dados local e a tabela *dept* do banco de dados remoto Se.

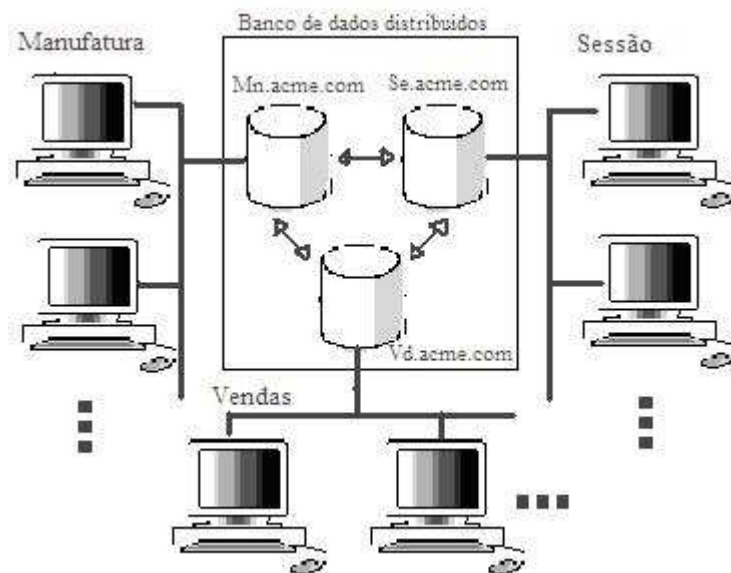


Figura 1. Bancos de dados distribuídos homogêneos

Usuários de um banco de dados distribuídos Oracle não devem saber que os dados solicitados num banco de dados local verdadeiramente pertencem a um banco de dados remoto. Desenvolvedores do sistema utilizam sinônimos para objetos remotos no sistema distribuído de modo que os usuários possam acessá-los com a mesma sintaxe de objetos locais. Por exemplo, se um usuário estiver conectado ao banco de dados Mn, mas deseja acessar dados do banco de dados Se, ao criar um sinônimo para a Mn da tabela remota, aprofundado, o sistema permitirá esta consulta:

```
Select * from Dept;
```

Desta forma, utilizando sinônimos, aparenta acessar dados próprios do banco conectado, porém é verdadeiramente um banco de dados distribuído. A localização e a plataforma dos bancos de dados são transparentes para uma aplicação cliente Mn.

Um sistema de banco de dados distribuídos Oracle pode incorporar versões diferentes de banco de dados Oracle. Para que todas as versões do Oracle participem de um sistema de banco de dados distribuído é necessário o conhecimento das funcionalidades específicas disponíveis em cada nó. Assim uma aplicação de banco de

dados distribuídos não deve esperar que a versão Oracle7 entenda extensões que somente estão disponíveis no Oracle9 i.

2.1.1. Banco de Dados Distribuídos X Processamento Distribuído

O conceito de banco de dados distribuídos define um conjunto de banco de dados em um sistema distribuído que pode aparecer para a aplicação como um único banco de dados. Enquanto, **processamento distribuído são as operações que ocorre quando uma aplicação distribui suas tarefas entre os diversos computadores da rede. Uma aplicação do sistema de banco de dados de processamento distribuído pode ser referida como uma aplicação de um sistema de banco de dados cliente/servidor. Assim, quando uma aplicação do banco de dados distribui tarefas front-end para computadores clientes permite a um banco de dados back-end gerenciar o acesso a um banco de dados compartilhado.**

Sistemas de banco de dados distribuídos Oracle empregam uma arquitetura de processamento distribuído. Por exemplo, um banco de dados Oracle servidor atua como cliente quando ele solicita dados que outro banco de dados Oracle servidor gerencia.

2.1.2. Banco de Dados Distribuídos X Banco de Dados Replicados

Em um banco de dados distribuídos puro (ou seja, não reproduzido), o sistema gera uma única cópia de todos os dados e suporta objetos de banco de dados. **Tipicamente, aplicações de dados distribuídos usam transações distribuídas para acessar os dados a nível local e remoto, e modificar o banco de dados global em tempo real.**

O termo replicação refere-se à operação de cópia e manutenção de objetos de bancos de dados em múltiplos bancos de dados pertencentes a um sistema distribuído. Embora a replicação assente na tecnologia de dados distribuídos, banco de dados replicados oferece benefícios em aplicações que não são possíveis dentro de um ambiente de banco de dados distribuídos puro.

Mais comumente, **replicação de dados locais é utilizada para melhorar o desempenho de banco de dados locais e proteger a disponibilidade de aplicações, porque existem opções alternativas de acesso aos dados.** Por exemplo, uma aplicação pode normalmente acessar um banco de dados local em vez de um servidor remoto para minimizar o tráfego de rede e obter o máximo de desempenho. Além disso, a aplicação pode continuar a funcionar até mesmo quando ocorrer falha no servidor local, pois outros servidores com dados replicados permanecem acessíveis.

2.2. Bancos de Dados Distribuídos Heterogêneos

Em um sistema de base de dados heterogêneos distribuídos, pelo menos uma das bases de dados não é um sistema Oracle. Apesar de possuir diversos bancos de dados e um ou mais não possuir sistema Oracle, para a aplicação, aparenta um único, local, banco de dados Oracle. O servidor local do banco de dados Oracle esconde a distribuição e heterogeneidade dos dados.

O servidor Oracle acessa o banco de dados não-Oracle utilizando Serviços Heterogêneos Oracle em conjunto com um agente. O agente é um sistema de aplicação específica ao acessar dados armazenados no sistema não-Oracle usando um Gateway Oracle Transparente. Por exemplo, se incluir um banco de dados Sybase em um banco

de dados distribuídos Oracle é necessário obter um gateway transparente específico Sybase de modo que o sistema de banco de dados Oracle possa se comunicar com ele.

Alternativamente, você pode usar conectividade genérica de acesso a dados armazenados num banco não-Oracle tão como o sistema não-Oracle suporta os protocolos ODBC ou OLE DB.

2.2.1. Serviços Heterogêneos

Serviço Heterogêneo (SH) é um componente integrado com o servidor de banco de dados Oracle que permite tecnologia para acessar sistema não-Oracle de um servidor Oracle, através de Gateways Transparentes Oracle via ODBC.

2.2.2. Agente Gateway transparente Oracle

Gateways Oracle abordam a necessidade de acessar dados de bancos de dados distribuídos heterogêneos. Gateways tornam possível a integração com qualquer número de sistemas não-Oracle a partir de uma aplicação Oracle utilizando componentes de Serviços Heterogêneos. Para cada sistema não-Oracle acessado, o Serviço Heterogêneo pode usar um agente gateway transparente de interface com o específico sistema não-Oracle. Então para cada tipo de sistema não-Oracle deve haver um agente diferente. O agente executa pedidos SQL e transacionais no sistema não-Oracle em nome do servidor do banco de dados Oracle.

2.2.3. Conectividade genérica Oracle

Conectividade Genérica permite conexão com dados armazenados em um banco de dados não-Oracle utilizando quer um agente ODBC de Serviços Heterogêneos ou um agente de Serviços Heterogêneos OLE DB. Qualquer fonte de dados compatíveis com os padrões ODBC ou OLE DB pode ser acessada através de um agente de conectividade genérico.

A vantagem para conectividade genérica é que pode não ser necessário comprar e configurar um novo sistema de agente específico. Usa-se um driver ODBC ou OLE DB que pode interagir com o agente. No entanto, é mais lento e limitado, pois algumas funcionalidades de acesso aos dados só estão disponíveis para agentes gateways transparentes Oracle.

2.3. Arquitetura de Bancos de Dados Cliente/Servidor

Cada computador pode agir como cliente e/ou servidor. Um cliente é uma aplicação solicitante de informações do servidor e o servidor é um software Oracle que fornece informações ao cliente sendo responsável por administrar um banco de dados.

O banco de dados Oracle tem suporte a ambiente cliente/servidor heterogêneo criando um Sistema Heterogêneo de SBDD, onde clientes e servidores podem não ser Oracle.

2.3.1. Conexões Diretas e Indiretas

Um cliente pode se conectar diretamente ou indiretamente a um banco de dados servidor. Uma conexão direta ocorre quando um cliente conecta um servidor e acessa informações a partir de um banco de dados contidos no servidor. Enquanto, uma

conexão indireta ocorre quando um cliente conecta a um servidor e depois acessa informações contidas em um banco de dados em um servidor diferente.

3. Database Links

O conceito central em Sistemas de Bancos de Dados distribuídos é um database link. Um database link é a conexão entre dois servidores de bancos de dados físicos que permitem que um cliente acesse-os como um banco de dados lógico.

3.1. Definição

Um database link é a forma que o Oracle permite que você se conecte de um banco de dados para outro. Um database link é um ponteiro que define um caminho de comunicação unilateral de um servidor de banco de dados Oracle para um outro servidor de banco de dados. O ponteiro do link é na verdade uma entrada em um dicionário de dados. Para acessar o link, você deve estar conectado no banco de dados local que contém esse dicionário.

Um cliente conectado ao banco de dados X pode usar um link armazenado no próprio banco para alcançar informações em um banco de dados Y de forma remota. Mas, por esse link ser unilateral, os clientes conectados em Y não poderão acessar as informações em X. Para um usuário conectado em Y acessar o banco de dados X, deve definir um link que acesse X no dicionário de dados de Y.

Uma conexão de um link permite que os usuários locais acessem dados em um banco de dados remoto. Para essa conexão ocorrer, cada banco de dados no sistema distribuído deve ter um nome global único na rede. Esse nome único identifica o banco de dados no sistema distribuído.

A figura 2 mostra o usuário *José* acessando a tabela *Aluno* no banco de dados remoto com nome global *dcc.ufba*:

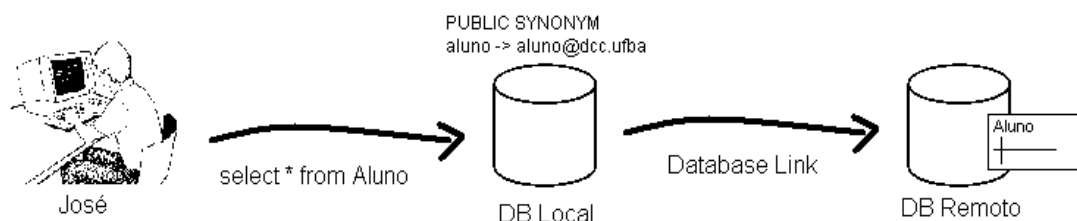


Figura 2. Database Link

3.2. Formas de acesso

Databases Links são privados ou públicos. Se eles são privados, somente o usuário que criou o link tem acesso. Se eles são públicos, todos os usuários do banco de dados têm acesso.

Os databases links podem ser diferenciados pela forma como as conexões a bancos de dados remotos ocorrem. Os usuários acessam um banco de dados remoto nos seguintes tipos de link:

- Link de usuário conectado: o usuário se conecta como ele mesmo. Ou seja, o usuário deve ter uma conta no banco de dados remoto com o mesmo nome que tem no banco de dados local. Sintaxe: `CREATE DATABASE LINK [linkname] CONNECT TO CURRENT_USER USING '[remote_database]'`.

- Link de usuário fixo: o usuário se conecta usando nome e senha referenciado no link. Por exemplo, se Daniela usa um link de usuário fixo que conecta no banco de dados *ufba* com o nome de usuário *mata060* e senha *daniela123*, então ela se conecta como *mata060*. Daniela tem todos os privilégios em *ufba* garantidos para o usuário *mata060*. Sintaxe: `CREATE DATABASE LINK [linkname] CONNECT TO [username] IDENTIFIED BY [password] USING '[remote_database]'`.

- Link de usuário corrente: o usuário se conecta como um usuário global. Um usuário local pode se conectar como um usuário global no contexto do procedimento armazenado, mesmo sem ter a senha de usuário global. Por exemplo, Daniela pode alcançar um procedimento que Persis escreveu e assim, teria os privilégios de Persis na execução deste procedimento. Link de usuário corrente é um recurso do Oracle Advanced Security. Sintaxe: `CREATE DATABASE LINK [linkname] USING '[remote_database]'`.

Depois que um link é criado, pode-se usá-lo para especificar um schema em comandos SQL.

3.3. Motivo de utilização

A grande vantagem dos databases links é permitir a usuários acessar outros objetos em um banco de dados remoto. Esses usuários estarão limitados aos privilégios definidos pelo proprietário do objeto. Em outras palavras, um usuário de um banco de dados local pode acessar um banco de dados remoto mesmo sem ser um usuário no banco de dados remoto.

Por exemplo, suponha que os empregados submetem os relatórios de despesas ao setor Financeiro, e suponha que um usuário que usa a aplicação do Financeiro precisa recuperar a informação sobre os empregados do banco de dados do setor de Recursos Humanos. Os usuários do Financeiro devem poder conectar ao banco de dados do RH e executar um procedimento armazenado no banco de dados remoto do RH que recupera a informação desejada. Os usuários do Financeiro não devem precisar ser usuários do banco de dados do RH para fazer seus trabalhos. Devem poder alcançar a informação do RH de uma maneira controlada, limitada pelo procedimento.

3.4. Nomes de bancos de dados globais

Para entender como um database link funciona, deve-se entender antes o que um nome de banco de dados é. Cada banco de dados em um banco de dados distribuído é identificado por seu nome global. O Oracle forma o nome do banco de dados pelos domínios da rede do banco de dados.

Uma tabela, por exemplo, é acessada de todos os nós que formam um banco de dados distribuído. Conseqüentemente, funciona da mesma forma que uma arquitetura de um banco de dados não distribuído que não podem fornecer nomes iguais para referenciar objetos distintos. O esquema de nomenclatura assegura que, em todo o

banco de dados distribuído, os objetos devem ser unicamente identificados e referenciados.

Para resolver as referências aos objetos como em um simples banco de dados, os administradores de banco de dados freqüentemente formulam nomes usando hierarquia. O sistema de administração de um banco de dados distribuído estende o modelo de hierarquia de nomenclatura forçando nomes únicos para os bancos de dados dentro da rede. Como resultado, um nome de objeto global é certamente único dentro de um banco de dados distribuído.

Por exemplo, a figura 3 ilustra a representação de uma hierarquia de banco de dados de uma rede:

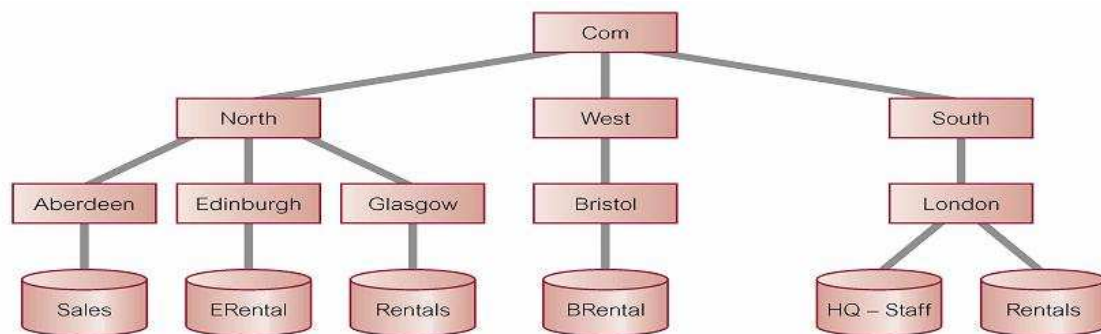


Figura 3. Hierarquia de banco de dados de uma rede

O nome de um banco de dados é definido a partir da folha da árvore e depois seguindo um trajeto até a raiz. Por exemplo, o banco de dados Sales está em Aberdeen do sub-domínio North do domínio COM. O nome global do banco de dados para Sales é criado concatenando os nós da árvore como segue:

sales.aberdeen.north.com.

3.5. Exemplo de uso de um database link:

```
CREATE DATABASE LINK vendas_us CONNECT TO daniela IDENTIFIED BY  
daniela123 USING 'sales.us.americas.acme_auto.com';
```

Selecionar *nome* na tabela *loja*:

```
SELECT nome FROM loja@vendas_us;
```

Executar um procedimento em um banco de dados remoto:

```
BEGIN
```

```
runProc@vendas_us;
```

```
END;
```

Criar uma visão usando o select abaixo:

```
CREATE OR REPLACE VIEW nome_loja_view AS
```

```
SELECT nome AS nomeLoja FROM loja@vendas_us;
```

```
SELECT * FROM nome_loja_view;
```

4. Segurança

4.1. Autenticação através de Database Links

Database links podem ser públicos ou privados, autenticados ou não autenticados. Os detalhes da autenticação de cada tipo de link estão a seguir:

4.1.1. Link privado não autenticado

O Oracle usa informações de segurança retiradas da seção local ao conectar ao banco de dados remoto. O database link é de usuário conectado e os passwords devem estar sincronizados nos dois bancos.

4.1.2. Link privado autenticado

As informações de segurança são obtidas a partir da definição do link e não são retirados da seção local. Essa configuração permite que os passwords sejam diferentes nas duas bases, mas o password do link local deve ser o mesmo do banco de dados remoto. O password é armazenado sem criptografia no sistema local, sendo um risco para a segurança.

4.1.3. Link público não autenticado

Todos os usuários podem referenciar este link para o banco de dados remoto. Nos outros aspectos, funciona da mesma maneira que um link privado não autenticado.

4.1.4. Link público autenticado

Todos os usuários no banco de dados local podem acessar o banco de dados remoto usando o mesmo login e senha. O password é armazenado sem criptografia se tornando visível para o usuário que tiver privilégios suficientes no banco de dados local.

4.2. Autenticação sem passwords

Pode-se usar fontes externas de autenticação, como o Kerberos para autenticar um database link de usuário conectado ou de usuário corrente. Nesse tipo de autenticação, credenciais são passadas de servidor para servidor e podem ser autenticadas por um servidor de banco de dados que pertença ao mesmo domínio. Por exemplo, se o usuário daniela é autenticado externamente em um banco de dados local, e quer usar um link de usuário conectado para conectar como ela mesma no banco de dados remoto, o servidor local passa as credenciais de segurança para o banco de dados remoto.

4.1. Suporte a contas de usuários e perfis

Em um sistema de banco de dados distribuído, pode-se planejar as contas de usuários e os perfis que serão necessários para suportar aplicações utilizadas no sistema. As contas de usuários necessárias para estabelecer conexões servidor-servidor devem estar disponíveis em todos os bancos de dados de um sistema de bancos de dados distribuído e as funções necessárias para tornar disponíveis os privilégios de aplicação para os usuários de aplicações do banco de dados distribuído devem estar presentes em todos os bancos do SBDD.

À medida que se cria database links para os nós de um sistema de banco de dados distribuídos, são determinadas quais contas de usuário e perfis que cada servidor necessita para suportar conexões com outros servidores utilizando esses links.

Em um ambiente distribuído, usuários geralmente precisam acessar muitos serviços de rede. Quando as configurações das autenticações são feitas separadamente para cada usuário acessar cada serviço de rede, a administração da segurança pode torna-se difícil, principalmente em grandes sistemas.

4.2. Criptografia de dados

A opção Oracle Advanced Security habilita o Oracle Net e produtos relacionados para usar criptografia de dados e checksumming de modo que os dados não possam ser lidos ou alterados. Isto protege os dados de visualização não autorizada utilizando o RSA Data Security RC4 ou algoritmo de criptografia Data Encryption Standard (DES). Os serviços de segurança da opção Oracle Advanced Security podem gerar mensagens criptograficamente seguras e incluí-las com cada pacote enviado pela rede, garantindo que os dados não sejam modificados, suprimidos ou repetidos durante essa transmissão.

4.3. Auditoria de Database Links

Sempre se deve realizar operações de auditoria localmente. Se um usuário utiliza um banco de dados local e, através de um database link, acessar um banco de dados remoto, as ações locais são auditoradas no banco de dados local, e as ações remotas são auditoradas no banco de dados remoto como se este usuário remoto estivesse conectado localmente. Não se pode setar as opções de auditoria local em objetos remotos. Portanto, não se pode usar a auditoria em um link, apesar de ele acessar objetos remotos que serão auditorados

5. Replicação

Replicação é o processo de cópia e manutenção de objetos de bancos de dados ou tabelas em vários bancos de dados que compõem um sistema de banco de dados distribuído. As mudanças aplicadas em um site são capturadas e armazenadas localmente para, posteriormente, serem aplicadas em cada banco de dados remoto.

Se uma tabela é replicada, uma cópia da tabela é armazenada em dois ou mais sites. No caso mais extremo, temos a replicação total, na qual cada cópia é armazenada em todos os sites do sistema.

Replicação usa tecnologia de bancos de dados distribuídos para compartilhar dados entre vários sites, mas um banco de dados replicado e um banco de dados distribuído não são a mesma coisa. Em um banco de dados distribuído, o dado está acessível em diversos locais, mas uma tabela particular reside em uma só localização. Por exemplo, a tabela *mata060* reside somente no banco de dados *dcc.ufba* em um sistema de banco de dados distribuído que também inclui os bancos de dados *mat.ufba* e o *fis.ufba*. Replicação significa que o mesmo dado está disponível em várias localizações. Por exemplo, a tabela *mata060* está disponível em *dcc.ufba*, *mat.ufba* e o *fis.ufba*.

5.1. Vantagens e desvantagens

O uso de replicação tem principalmente duas vantagens: disponibilidade e aumento do paralelismo. Se houver qualquer problema em um site com uma tabela replicada, então a tabela poderá ser encontrada em outro site. Assim, o sistema não pára nesse caso. Com relação ao aumento do paralelismo, em caso de leitura de dados, diversos sites podem processar consultas à tabela em paralelo. Com isso, quanto mais cópias houver de uma tabela, maior a chance que o dado necessário seja encontrado onde a transação está aberta, minimizando o movimento dos dados entre sites.

Porém, existem desvantagens no uso da replicação. E a principal é o aumento de overhead para atualização. O sistema deve assegurar que as cópias de uma tabela sejam consistentes. Com isso, cada atualização local deverá atualizar todas as cópias da tabela. O resultado disso é o aumento de movimento de dados de um site para outro.

5.2. Visão materializada

A replicação no Oracle é feita criando uma visão materializada de uma tabela. Essa visão é materializada, pois consiste em uma cópia completa ou parcial de uma tabela, armazenando-a em outro site.

5.2.1. Somente leitura

Em uma configuração básica, uma visão materializada fornece acesso somente leitura para os dados da tabela que foram originados na tabela original (master). As alterações na tabela master são refletidas nas tabelas replicadas assim que for requisitada uma atualização dessas tabelas. Como o acesso pelas tabelas refletidas é somente leitura, isso elimina a possibilidade de conflitos, já que não é possível atualizá-las.

5.2.2. Atualizável

Em uma configuração mais avançada, é possível criar uma visão materializada atualizável que permite ao usuário inserir, atualizar e deletar linhas de uma tabela master executando essas operações na visão materializada. Essa visão é similar à somente leitura exceto pelo fato de poder haver modificações e enviá-las à tabela master.

5.3. Formas de replicação

Existem duas principais formas de replicação: síncrona e assíncrona

5.3.1. Replicação Síncrona

Com a replicação síncrona, as alterações nas cópias de uma tabela fazem com que a tabela master seja imediatamente alterada. Quando se usa replicação síncrona, uma atualização de uma tabela resulta em uma imediata replicação da atualização por todas as cópias envolvidas. Conseqüentemente, se a tabela master não puder completar a transação por qualquer motivo, toda ela é cancelada. Com isso, evita-se a possibilidade de conflito.

Isso requer um ambiente muito estável para efetuar a operação. Se a comunicação de um servidor com outro não for possível por um problema de rede, por exemplo, o usuário ainda pode consultar as tabelas replicadas, mas nenhuma transação poderá ser completada até a comunicação ser reestabelecida.

Sintaxe: CREATE MATERIALIZED VIEW hr.employees AS
SELECT * FROM hr.employees@orc1.world;

5.3.2. Replicação Assíncrona

Replicação assíncrona é a forma mais comum de se implementar replicações. Quando se usa replicação assíncrona, todas as modificações feitas são armazenadas em suas visões materializadas locais ou na tabela master e a sincronização só ocorre no momento programado. As alterações são propagadas por todas as cópias da tabela master.

Usando a replicação assíncrona, significa que é possível haver conflitos de dados, já que a mesma linha pode ser alterada em duas tabelas diferentes. Entretanto, podem-se usar técnicas para evitar esses conflitos e, se os conflitos ocorrerem, o Oracle fornece mecanismos que podem ser implementados para resolver isso. As informações sobre conflitos não resolvidos são mantidas em um log de erros.

Sintaxe: CREATE MATERIALIZED VIEW hr.departments FOR UPDATE AS
SELECT * FROM hr.departments@orc1.world;

6. Fragmentação

Como o nome diz, fragmentação de dados nada mais é que dividir uma tabela em fragmentos. Esses fragmentos contêm informação suficiente para reconstruir a tabela original. Existem duas formas de se aplicar a fragmentação: de forma horizontal ou vertical.

6.1. Fragmentação Horizontal

A fragmentação horizontal divide a tabela separando as linhas em dois ou mais fragmentos. Cada fragmento pode ser definido como uma seleção da tabela global.

Por exemplo, se tivéssemos a tabela 6.1, poderíamos criar um fragmento para cada cidade:

Tabela 1. Tabela pessoa

id	nome	cidade	telefone
1	Diogo	Salvador	1234-5678
2	Persis	Salvador	9876-5432
3	Torquato	Salvador	7418-5236
4	Lula	Brasília	6541-2365
5	José Alencar	Brasília	8523-9654

Após a fragmentação, ficaríamos com as tabelas 2 e 3:

Tabela 2. Fragmento da tabela pessoa para cidade igual a Salvador

id	nome	cidade	telefone
1	Diogo	Salvador	1234-5678
2	Persis	Salvador	9876-5432
3	Torquato	Salvador	7418-5236

Tabela 3. Fragmento da tabela pessoa para cidade igual a Brasília

id	nome	cidade	telefone
----	------	--------	----------

4	Lula	Brasília	6541-2365
5	José Alencar	Brasília	8523-9654

O Oracle suporta a fragmentação horizontal:

```
CREATE TABLE pessoa
( id INT NOT NULL,
  nome char(20) NOT NULL,
  cidade char(20) NOT NULL,
  telefone char(8) NOT NULL )
PARTITION BY RANGE (nome, cidade, telefone)
( PARTITION pessoa_ssa VALUES EQUALS THAN ('Salvador')
  TABLESPACE tsa,
  PARTITION pessoa_bsb VALUES EQUALS THAN ('Brasília')
  TABLESPACE tsb );
```

6.2. Fragmentação Vertical

A fragmentação vertical divide a tabela pela decomposição das suas colunas. A fragmentação deve ser feita de modo que se permita reconstruir a tabela a partir dos fragmentos por meio de uma junção natural.

O modo mais simples de se permitir essa reconstrução é incluindo os atributos de chave em todos os fragmentos da tabela.

Para ilustração, considerando ainda a tabela 1, poderíamos fragmentar a tabela de forma que o nome estivesse em um fragmento e a cidade e o telefone em outro como nota-se nas tabelas 4 e 5, respectivamente:

Tabela 4. Fragmento da tabela pessoa

id	nome
1	Diogo
2	Persis
3	Torquato
4	Lula
5	José Alencar

Tabela 5. Fragmento da tabela pessoa

id	cidade	telefone
1	Salvador	1234-5678
2	Salvador	9876-5432
3	Salvador	7418-5236
4	Brasília	6541-2365
5	Brasília	8523-9654

6.3. Fragmentação Mista

A fragmentação mista ocorre quando se aplica uma fragmentação vertical em uma tabela e depois se aplica uma fragmentação horizontal nos fragmentos gerados, ou vice-versa.

6.4. Fragmentação e Replicação

As técnicas descritas anteriormente para replicação e fragmentação podem ser aplicadas sucessivamente a uma mesma tabela. Ou seja, um fragmento pode ser replicado. As réplicas desse fragmento podem ser fragmentadas novamente e assim por diante.

7. Transparência

Uma vantagem de SGBDD Oracle é a transparência na gerência dos dados distribuídos. O objetivo é tornar transparente um sistema de banco de dados distribuído como um simples banco de dados Oracle para usuários e aplicações. Estes, por sua vez, não precisam possuir o conhecimento da localização física dos objetos de um banco de dados, mas apenas poderão visualizar uma única imagem da base de dados logicamente integrada.

A transparência de rede requer transparência de acesso e de localização para que as aplicações dos usuários não percebam a existência da rede. Na transparência de acesso, um comando acessa um dado independente do local em que esteja. Para isto, desenvolvedores aplicam a transparência de localização utilizando sinônimos, views, para formalizar a transparência de localização para objetos e tabelas. Sendo assim, não há informações de localização nos nomes indicando um computador na rede. Logo, um banco de dados remoto é referenciado como se fosse local.

7.1. Transparência no acesso à SQL e às transações distribuídas

Assim como em um ambiente de dados não distribuídos, a arquitetura distribuída Oracle oferece os comandos padrões SQL para consultas e atualizações, adicionalmente oferece aplicações de controles de transações, assim como o Commit.

No Oracle, comandos de uma única transação podem referenciar várias tabelas locais ou remotas de forma transparente. O sistema Oracle garante que todos os nós envolvidos na transação distribuída executem todos os commit ou todos roll back da transação, mesmo em caso de falha no sistema.

Para cada transação no Oracle, um Sistema de Mudança Numérica (SNC) é associado para identificar unicamente as mudanças feitas pelos comandos da transação.

7.2. Otimização de consultas distribuídas

Consultas distribuídas utilizam a otimização de custo-base Oracle que gera expressões SQL que extraem apenas os dados necessários de tabelas remotas para reduzir a quantidade de transferência de dados entre sites, enviando os resultados para o site local para o processamento final. Esta redução do montante da transferência dos dados exigidos é notável ao comparar com o tempo que leva para transferir todos os dados da tabela para o site local de processamento.

8. Conclusão

A implantação de um banco de dados distribuídos Oracle pode tornar-se inviável, devido ao seu alto custo, pois há a necessidade de altos investimentos em software, hardware e estrutura de rede além de treinamento de pessoal. Mas o Oracle aplicado a BDD é altamente recomendado para casos em que o custo, a complexidade e o volume de informações são grandes, oferecendo uma infinidade de recursos aos seus usuários.

Referências

- Dye, C. (1999) “Oracle Distributed Systems”, <http://www.oreilly.com/catalog/ordistsys/chapter/ch01.html>, Março.
- Fanderuff, D. “Criação de Database Links: Vemos o que é um database link e como criá-los”, <http://www.criarweb.com/artigos/618.php>, Março.
- Morais, F. A. (2008) “Diferenças entre o Oracle e o SQL”, <http://www.crieseuwebsite.com/artigos/artigo.php?categoria=bds&id=1>, Março.
- Scopim, K. (2000) “Arquitetura de um banco de dados distribuídos em Oracle”, <http://www.inf.ufpr.br/sunye/mapeamento/arquitetura.htm>, Março.
- Shibayama, E. T. (2004) “Aplicações em Banco de dados distribuídos”, <http://www.dc.uel.br/nou-rau/document/?view=156>, Março.