

Aplicações de Bancos de Dados Distribuídos Utilizando a Filosofia Cliente/Servidor

Daniel Corrêa Lobato. Tecnólogo em Processamento de Dados pela Universidade da Amazônia (1996). Atua como Analista de Sistemas, consultor na área de desenvolvimento de softwares aplicativos, além de ministrar treinamento empresarial em tecnologias "ClientxServer". Atualmente (1997) está cursando Especialização em Engenharia de Software pela Universidade Federal do Pará

Samih Naif Daibes Júnior. Tecnólogo em Processamento de Dados pela Universidade da Amazônia (1996). Atua como Analista de Sistemas, e Suporte e Banco de Dados.

RESUMO: A evolução das técnicas que auxiliam no projeto de implementação de sistemas computacionais tem permitido o desenvolvimento de sistemas cada vez mais abrangentes e, dessa forma, o volume de dados está aumentando juntamente com a complexidade das estruturas de armazenamento. Entre estes sistemas nós temos o Sistema de Gerenciamento de Bancos de Dados (SGBD), desenvolvido para suprir as necessidades de administração da informação existente, e permitir que as empresas tenham o controle de seus dados de forma centralizada (hoje em dia, considerando um dos bens mais valiosos).

Entre as tecnologias existentes de banco de dados, existe o SGBD Distribuído, que constitui o assunto principal deste artigo. Esta tecnologia utiliza distribuição de dados física, isto é, dados armazenados em diferentes computadores que se comunicam uns com os outros através de um sistema de comunicação de dados, mas permitindo o controle centralizado dos dados.

Contudo, algumas dificuldades e complexidade cercam a utilização do SGDD, como manter a integridade e a consistência dos dados, além do gerenciamento da segurança. Dessa forma, há algumas vantagens e desvantagens no uso do SGBDD, considerando a área de aplicação - dentro ou fora do ambiente corporativo. Neste artigo, discute-se sobre a estrutura do SGBDD, suas dificuldades, vantagens e desvantagens.

A ARQUITETURA CLIENTE/SERVIDOR

A chave para a compreensão da arquitetura cliente/servidor é a observância de que a mesma é um conceito lógico. O cliente e o servidor podem ou não estar em máquinas físicas distintas. De maneira mais precisa, a tecnologia cliente/servidor é um modelo, ou paradigma, para a interação de processos de software em execução concorrente. Interação essa que se dá, normalmente, através de mecanismos de passagem de parâmetros, como a *Remote Procedure Call*.

Em um ambiente cliente/servidor, os processos cliente e servidor são indiferentes

se rodam ou não em uma mesma máquina. Através de uma visão teórica, o tamanho da máquina não importa também. A Figura A mostra os diferentes tipos de sistemas cliente/servidor.

Além do uso em bancos de dados SQL (o que é assumido erroneamente por algumas comunidades de usuários como sinônimo), a tecnologia cliente/servidor tem sido utilizada concorrentemente para monitores de TP (Transaction Processing - Processamento de Transações), groupware e objetos distribuídos, gerando as "três ondas da computação cliente/servidor", que podem ser vistas na Figura B.

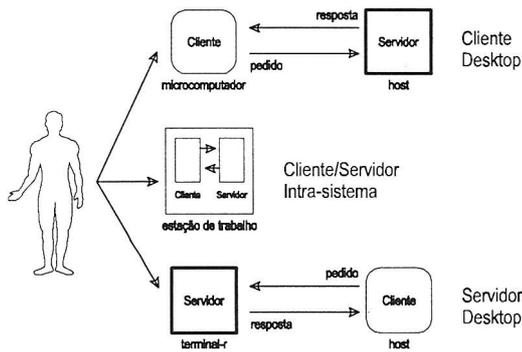


Figura A – Tipos de sistemas cliente/servidor

A tecnologia de objetos distribuídos, a terceira onda, promete sistemas cliente/servidor mais flexíveis. Isto porque ele encapsula os dados e a lógica dos negócios em objetos que podem deslocar-se para qualquer ponto das redes, rodar em diferentes plataformas, falar com aplicativos legados por meio de envoltórios de objetos e gerenciar a si mesmos e aos recursos que controlam. Os objetos podem ajudar-nos a dividir grandes aplicativos monolíticos em componentes mais gerenciáveis. Uma vez que a tecnologia de objetos distribuídos tenha decolado, ela pode englobar todas as outras formas de computação cliente/servidor, inclusive monitores de TP, bancos de dados SQL e *groupware*. Os objetos distribuídos podem fazer tudo isso, e fazê-lo melhor.

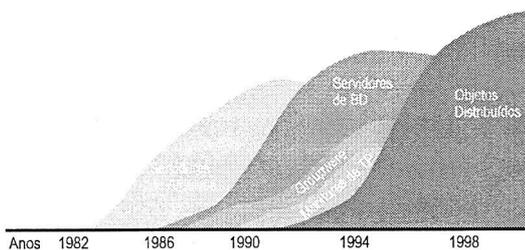


Figura B – Três ondas da computação cliente/servidor

Fundamentalmente, a tecnologia cliente/servidor é caracterizada, de uma maneira resumida, por:

- Comunicação de nível não hierárquico entre clientes e servidores;
- Estilo de interação transacional, do tipo pedido/resposta;
- Equilíbrio cooperativo do trabalho entre clientes e servidores, e;
- Transparência do local de acesso.

Os atributos de clientes e servidores estão resumidos na Tabela A [RENA94].

Atrib.	Cliente	Servidor
<i>Modo</i>	Ativo	Reativo
<i>Execução</i>	Início e final fixos	Roda o tempo todo
<i>Finalidade Principal</i>	Manter o diálogo com o usuário através de: Manipulação de tela/janela Interpretação de menu/comando Entrada de mouse/teclado Entrada de dados e validação Processamento de ajuda Recuperação de erros	Oferecer serviços funcionais através de: Compartilhamento de dados na aplicação Compartilhamento de comunicações Compartilhamento de arquivos Compartilhamento de impressoras Compartilhamento de CPU Compartilhamento de Vídeo
<i>Transpa-rência</i>	Ocultas redes e servidores	Ocultas detalhes de implementação dos serviços
<i>Inclui</i>	Comunicação com diferentes servidores	Comunicação com diferentes clientes
<i>Exclui</i>	Comunicação cliente-cliente	Comunicação servidor-servidor

Tabela A – Atributos chave de sistemas cliente/servidor

Bancos de Dados

Um banco de dados é, basicamente, um sistema de armazenamento de dados baseado em computador, cujo objetivo global é registrar e manter informações. A figura C [DATE81] mostra uma visão bastante simplificada de um sistema de banco de dados, ressaltando seus quatro componentes maiores: dados, hardware, software e usuários.

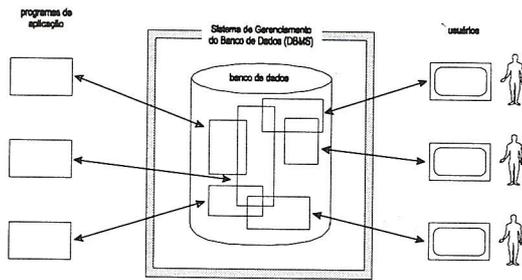


Figura C – Componentes de um SGBD

Os dados armazenados no sistema são repartidos em um ou mais bancos de dados. Por tanto, um banco de dados é um depósito de dados armazenados. Geralmente ele é tanto integrado como compartilhado.

Por “integrado” entende-se que o banco pode ser imaginado como sendo uma unificação de diversos arquivos que, de outra forma, estariam separados, eliminando parcial ou totalmente qualquer redundância entre aqueles arquivos; e por “compartilhado” entende-se que as partes individuais dos dados podem ser compartilhadas entre diversos usuários diferentes, significando que cada um dos usuários pode ter acesso à mesma parte do dado e usá-lo com finalidade diferente.

Podemos categorizar de maneira conveniente os sistemas de banco de dados de acordo com a abordagem por eles adotada para as estruturas de dados e operadores associados suportados pelo mesmo. As três abordagens, ou modelos, mais conhecidas (Hierárquico, Redes e Relacional) são explanadas a seguir.

Um BD Hierárquico consiste em uma coleção de registros que são conectados uns aos outros por meio de ligações, tomando a forma de uma árvore enraizada, onde teremos uma coleção de tais árvores, formando uma floresta, ou seja, podemos dizer que o usuário vê o BD como um conjunto de árvores. Disso

concluimos que se dois registros são unidos por um elo, um dos registros é considerado ascendente e o outro descendente, sendo que o primeiro ascendente é considerado a raiz da árvore. Em geral, a raiz pode ter qualquer quantidade de dependentes, cada um dos quais pode ter qualquer quantidade de dependentes de nível mais baixo, e assim sucessivamente, com qualquer número de níveis. Devemos ressaltar que em um BDH uma parte dos elos é explícita.

O modelo em rede apresenta uma estrutura mais geral do que a hierárquica pois uma determinada ocorrência de registro pode ter qualquer quantidade de superiores imediatos (ascendentes), assim como também podem haver qualquer quantidade de dependentes imediatos (descendentes). Podemos, então, dizer que um BD Rede consiste em uma coleção de registros que são conectados uns com os outros através de ligações (associações direta entre 2 registros).

O Modelo Relacional estabeleceu-se como o modelo de BD principal para aplicativos de processamento de dados comerciais. Neste modelo, os elos são implícitos. Criado por Edgar F. Codd, nos anos 70, começou a ser utilizado nas empresas a partir de 1987. Esta abordagem relacional aos dados está baseada na observação de que as informações em uma base de dados podem ser considerados como relações matemáticas, e conseqüentemente a teoria elementar de relações pode ser usada para lidar com vários problemas práticos com os dados dessa base. Percebemos, então, que o objetivo maior do Modelo Relacional é gerar um conjunto de esquemas/relações que nos permitam armazenar informações sem redundância desnecessária, e ainda recuperar informações facilmente.

Bancos de Dados Distribuídos

Os Sistemas Distribuídos (fragmentados, replicados ou reorganizados) estão se tornando cada vez mais importantes nesta nova fase de desenvolvimento de sistemas, devido a questões de amplitudes, localização e redes de comunicação de dados cada vez mais sofisticadas e rápidas. O interesse atual nos sistemas distribuídos deve-se à importância das razões envolvidas, tanto o alto custo de manutenção dos sistemas centralizados, como as falhas ao explorar novas oportunidades, com vistas a um desempenho mais eficiente dos equipamentos e na interação com os usuários.

Um sistema de banco de dados distribuídos (BDD) existe quando um banco de dados integrado logicamente (integração lógica significa que qualquer nó tem acesso potencial a todo o banco de dados) é fisicamente distribuído sobre diferentes nós de computação interligados por uma rede local ou formada através de vários meios de comunicação, tais como barramentos de alta velocidade ou linha telefônica. Define-se um nó de computação como um computador (mini ou microcomputador) localizado numa área da organização com certas facilidades de processamento.

Um dos fatores de fundamental importância para implementar o uso de sistemas distribuídos é a conexão entre os computadores. A duplicação de dados, sistemas e programas, devido à incompatibilidade de computadores de diversos fabricantes e à ausência de portabilidade, tem forçado muitas empresas a utilizarem o processamento centralizado. Outro fator é a diminuição dos custos de transmissão de informações digitais através da utilização de

redes compartilhadas de pacotes, que constitui grande incentivo para a implementação dos sistemas distribuídos.

A distribuição física deve ser transparente aos programas de aplicação, assim como deve prezar pela transparência de localização, esta que permite ao usuário ter acesso a dados de qualquer banco de dados residente em um ou múltiplos computadores, sem a necessidade de especificar em que máquina esses dados se encontram armazenados (cabe ao sistema localizar e recuperar o dado solicitado), ou seja, o sistema distribuído deve parecer ao usuário como um sistema não-distribuído, sendo que os problemas dos sistemas distribuídos são de implementação interna e não a nível de usuário. Esta última observação é válida pelo fato de que grandes quantidades de dados são gerados em muitos locais e utilizados em outros, permanecendo transparentes para o usuário.

Em cada um dos nós o Software do SGBDD consiste nos seguintes itens, mostrados na Figura D:

- Um Sistema Operacional em cada nó.
- O Gerenciador de Comunicação, que permite a troca mútua de informação entre programas remotos. Este gerenciador inclui três níveis de controle:
 - Controle da transmissão física dos dados;
 - Controle da ligação lógica de mensagens entre processos (i.e., programas em execução);
 - Controle das classes específicas de diálogo (i.e., transferência de arquivos)
- Um Sistema de Gerenciamento de Banco de Dados (SGBD) para fortalecer localmente todas as vantagens deste tipo de Software.

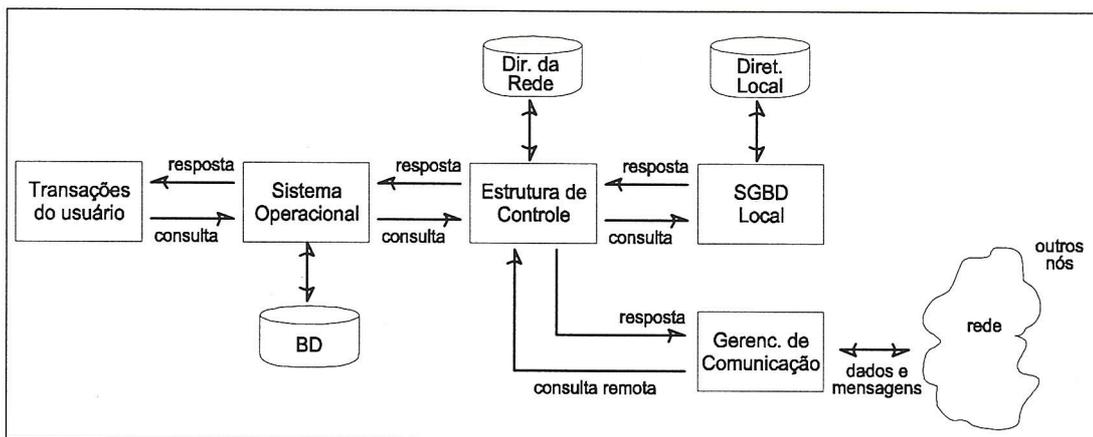


Figura D – Componentes de um nó de um SGBDD.

Podemos verificar através da figura E que existem quatro componentes básicos formadores de um SGBDD:

- Transações;
- Gerenciador de Transações (GT);
- Gerenciador de Dados (GD);
- Dados.

As transações se comunicam com os GT, que se comunicam com os GD, os quais administram os dados (Os GT não se comunicam com outros GT, assim como os GD não se comunicam com outros GD). Os GT supervisionam transações e cada transação executada no sistema é supervisionada por um simples GT. Os GD administram o banco de dados armazenado. De acordo com as exigências das transações, os GT enviam comandos aos GD especificando operações de leitura ou gravação de dados. A interface GT-GD está influenciada principalmente pela política de controle de concorrência e processamento de consultas, enquanto que na interface transação-GT são definidas quatro operações básicas: READ (x), WRITE (x), BEGIN e

END, sendo estes dois últimos denotando o campo de execução das transações.

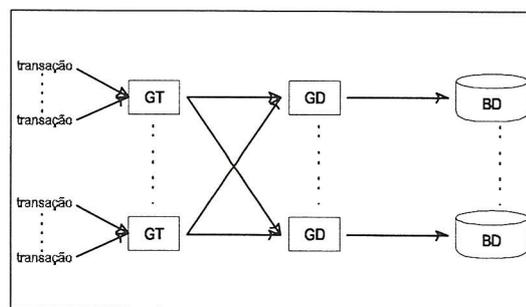


Figura E – Componentes básicos formadores de um SGBDD

LOCAL DOS DADOS

Decidir onde os dados devem ser localizados não é fácil. Parte do motivo para isso é que os dados podem ser distribuídos de várias maneiras. Não é simplesmente uma questão de colocá-los em algum lugar. A Figura F exemplifica os quatro tipos de dados distribuídos que podem existir [RENA94]:

- Replicados: múltiplas cópias idênticas
- Particionados: divididos em vários lugares
- Reorganizados: derivados ou vindos de outros locais
- Em cache: replicação parcial

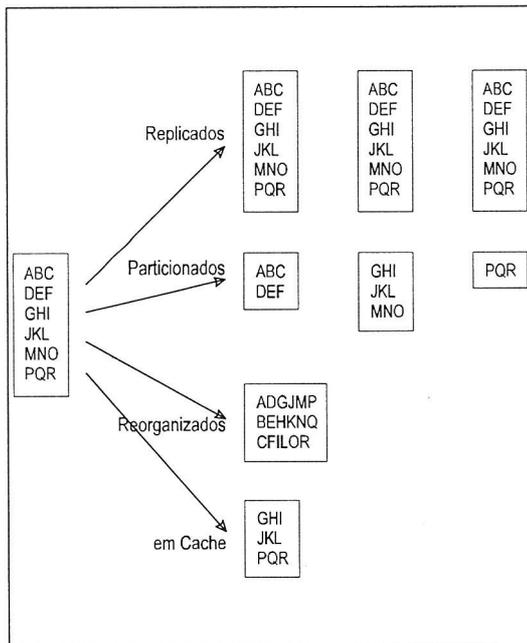


Figura F – Tipos de distribuição de dados

Se os dados são centralizados em um servidor, há uma única camada de processamento, como veremos na Figura G. Em geral, uma única camada é comum em sistemas cliente/servidor pequenos ou em ambientes onde os dados ficam armazenados em mainframes.

Em geral, faz sentido centralizar os dados se grandes quantidades forem, freqüentemente, atualizadas ou se todos os usuários acessam todos os dados por igual e precisam, sempre, dos valores mais atuais. Outro motivo para a centralização dos dados é se eles forem constantemente manipulados como um todo, como em pesquisas, classificações ou resumos. Além disso, se tornam mais fáceis de gerenciar (por exemplo, um único ponto faz o backup). O processamento cliente/servidor de única camada também é mais fácil de projetar, pois não há interação entre servidores.

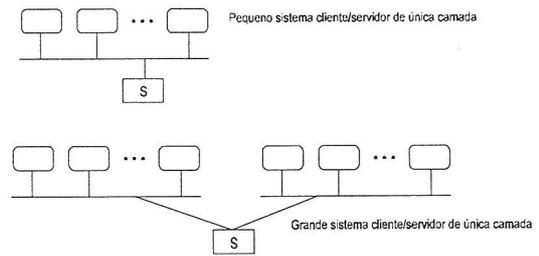


Figura G – Dados centralizados

Se os dados são distribuídos de alguma forma, há, como resultado, uma arquitetura de processamento em duas camadas, como veremos na Figura H. Os servidores A e B oferecem serviços semelhantes aos seus respectivos grupos de trabalho - essa é a primeira camada. O servidor C, por sua vez, oferece serviços globais a todos os grupos de trabalho, fazendo, assim, a segunda camada. O modo de interação desses servidores vai depender de como os dados foram distribuídos.

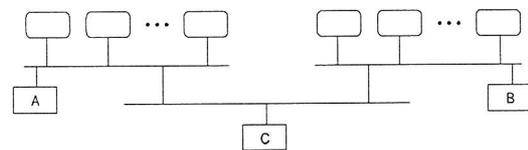


Figura H – Dados distribuídos de alguma forma

TECNOLOGIAS PARA A REDUÇÃO DE PROBLEMAS NA DISTRIBUIÇÃO DE BANCOS DE DADOS

Diversas pesquisas tentam minimizar os problemas aos quais os sistemas de bancos de dados distribuídos estão sujeitos, sendo que o principal deles é a garantia de integridade transacional entre os dados distribuídos. Como resultado, surgiram alguns algoritmos para tentar garantir a atualização de dados distribuídos.

Two-Phase Commit

O protocolo “two-phase commit” é um algoritmo distribuído que faz com que todos os nós em um sistema distribuído concordem em efetuar o commit de uma transação. O protocolo resulta, ou no commit da transação, ou no cancelamento da mesma, mesmo no caso de falhas de nós e perdas de mensagens. As duas fases do algoritmo são a “commit-request”, onde o coordenador tenta preparar todos os participantes (cohorts) para o commit, e a fase do “commit”, onde o coordenador completa a transação em todos os participantes.

O protocolo trabalha da seguinte maneira: um nó é designado para ser o coordenador, que é o nó principal, e os outros

nós na rede são chamados participantes. Outras suposições do protocolo são a existência de stable storage em cada nó e o uso de write ahead log por cada nó. O protocolo assume, também, que nenhum nó pára para sempre, e eventualmente quaisquer dois nós podem comunicar-se entre si. O último não é grande coisa, já que as comunicações nas redes de computadores podem ter suas rotas alteradas sem maiores problemas. O primeiro é uma suposição muito mais forte; imagine que um nó saia do ar!... Veremos, mais a frente, que outro protocolo tenta eliminar esta suposição.

Na Figura I, o digrama de funcionamento de uma transação bem sucedida entre dois participantes [adaptada de UNIS95].

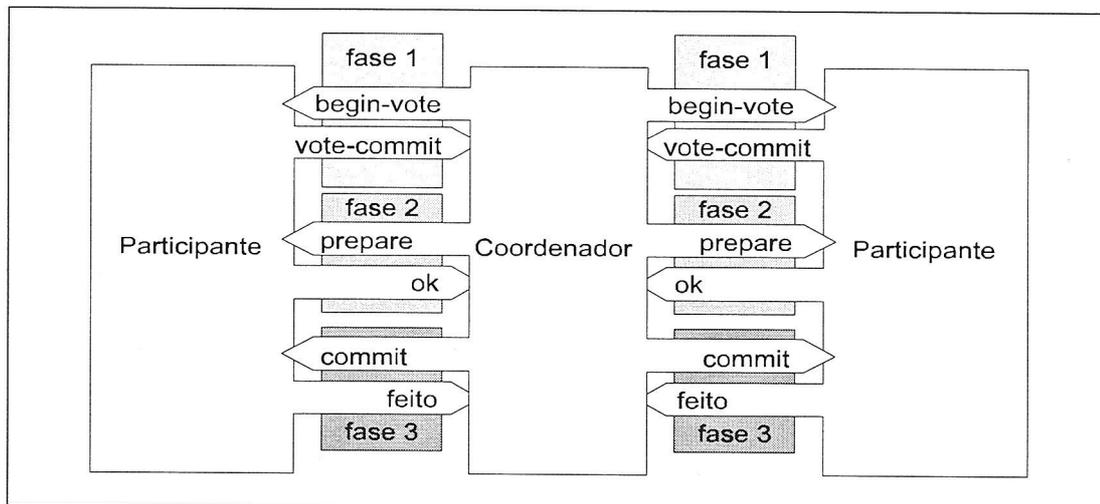


Figura J – Transação bem sucedida em Three Phase Commit

A grande desvantagem deste protocolo está no fato de ele ser um protocolo blocking. Um nó ficará bloqueado enquanto estiver esperando por uma mensagem. Isto quer dizer que outros processos competindo por recursos bloqueados pelo processo que está utilizando o protocolo, deverão ficar aguardando pela liberação dos recursos. Um único nó ficará esperando, mesmo que todos

os outros nós tenham falhado. Se o coordenador falhar permanentemente, alguns participantes nunca resolverão suas transações. Isto terá o efeito de bloquear os recursos para sempre.

Three Phase Commit

O protocolo de two-phase commit é de grande valia na ajuda da redução dos

problemas de atualização de bancos de dados distribuídos, mas apresenta alguns problemas, como vimos em suas desvantagens.

O protocolo three-phase commit, tenta reduzir ainda mais os problemas que existiam antes do two-phase commit e, ao mesmo

tempo, resolver alguns problemas do mesmo protocolo, como o caso de travamento dos recursos no caso de falha do coordenador ou participantes. O protocolo incorpora um controle de timeout, o que reduz sensivelmente a possibilidade de ocorrer um travamento de recursos.

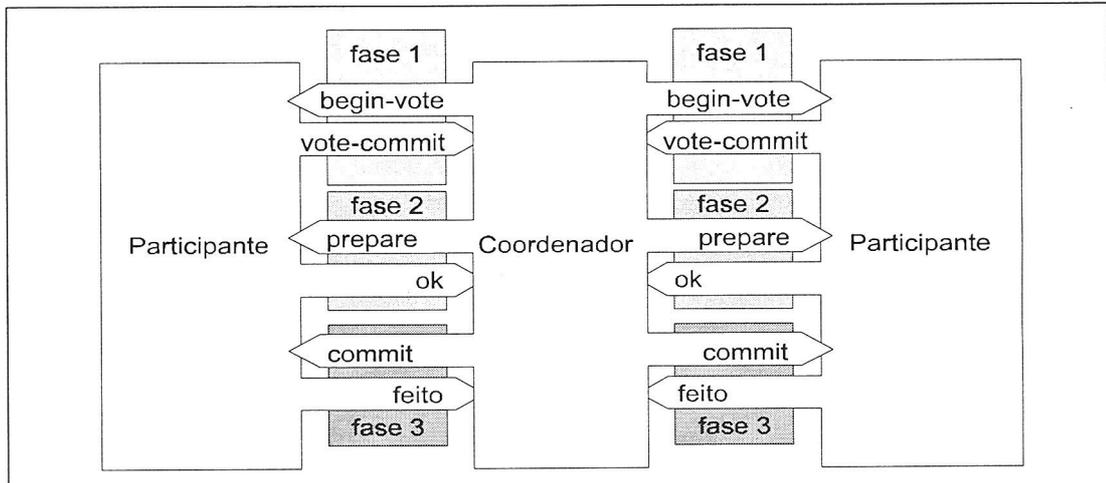


Figura J – Transação bem sucedida em Three Phase Commit

A principal desvantagem do protocolo reside no fato de haver poucas ou nenhuma implementação do mesmo a nível comercial, tornando difícil os testes e aprimoramentos do mesmo em situações reais, quer a nível corporativo quer seja fora dele.

APLICAÇÕES E TENDÊNCIAS

A tecnologia de Bancos de Dados têm evoluído constantemente e, com a Internet/ Intranet e outras redes WAN, algumas aplicações novas tem sido desenvolvidas para o acesso à Bancos de Dados Distribuídos. Além das redes WAN, outra área que tem impulsionado o desenvolvimento da área de banco de dados é a área gerencial, com o Data Warehousing e o Database Marketing.

A UTILIZAÇÃO DE RDBMS EM INTERFACES WWW

É sabido por todos que a interface WWW de navegadores como o Microsoft Internet Explorer e o Netscape Navigator é uma das mais fáceis e intuitivas de se usar. As empresas de desenvolvimento de RDBMS tem se voltado para a utilização destes navegadores como front-ends para acesso à bancos de dados. A Microsoft, por exemplo, oferece um conjunto completo de ferramentas para integrar bancos de dados com o seu servidor HTTP, o Internet Information Server. Este servidor vem integrado à nova versão do Windows NT 4.0. As soluções da Microsoft funcionam, tanto com seus próprios gerenciadores de bancos de dados - Access, FoxPro e SQL Server - como com

produtos de terceiros compatíveis com a tecnologia ODBC (Open Database Connectivity).

As páginas publicadas podem ser estáticas ou dinâmicas. As estáticas são geradas com um conteúdo fixo para representar o estado de um banco de dados em um determinado instante, ao passo que as dinâmicas são montadas em tempo real pelo servidor para atender às solicitações dos usuários.

Apenas no caso das páginas dinâmicas é que o navegador funciona como um front-end do banco de dados. Por meio dele, o usuário pode estabelecer critérios de consulta para as informações a ser pesquisadas. A página HTML é montada pelo servidor especialmente para atender à consulta feita pelo usuário. Uma aplicação para as páginas dinâmicas é a de consulta remota aos dados da empresa, como por exemplo, situação de estoque por um vendedor, ou mesmo o ramal telefônico de um determinado funcionário e, até mesmo, atualização remota dos dados, como, por exemplo, a venda de um produto que implica na redução de estoque, ou a alteração de dados de um funcionário.

É claro que essas vantagens não estão disponíveis apenas para o ambiente corporativo. Utilizando-se a mesma plataforma, pode-se desenvolver um banco de dados de sinopses e comentários sobre filmes e colocar disponível para acesso remoto dos cinéfilos, permitindo inclusive que os cinéfilos adicionem seus próprios comentários. Outra aplicação já existente é a compra de passagens aéreas através dessa tecnologia: o usuário informa o itinerário desejado, o sistema retorna as empresas que efetuam os vôos e as tarifas, cabendo ao usuário informar, apenas, os dados para cobrança e o local de entrega dos bilhetes.

Não ficamos limitados à apresentação de dados textuais. Nada impede que se utilize um SGBD com suporte ao armazenamento de imagens e, através de ODBC, conectemos o mesmo ao servidor HTTP para que o usuário navegue por um álbum de fotografias, por exemplo.

A solução da Microsoft para a criação de páginas dinâmicas compreende dois programas: o IDC (Internet Database Connector), que faz parte do IIS; e o dbWeb, um gateway que conecta uma fonte de dados ODBC ao servidor IIS.

A Oracle também possuiu solução para a publicação de páginas HTML dinâmicas: Designer 2000 e Developer 2000. O primeiro é uma ferramenta de modelagem e projeto que gera aplicações. O segundo é um ambiente integrado para a criação de aplicativos baseados em bancos de dados.

O Designer 2000 usa o Oracle WebServer, um servidor HTTP, que funciona integrado ao servidor de dados Oracle 7. Com ele, o WebServer recupera informações do banco de dados e gera páginas HTML empregando a linguagem PL/SQL, que é a extensão da Oracle para a linguagem SQL.

Ao utilizar o Designer 2000, o desenvolvedor precisa seguir três passos para publicar sua base de informações. A primeira é criar um módulo que define quais objetos do banco de dados serão usados e como isso será feito. Em seguida, são elaborados, com base nesse módulo, pacotes PL/SQL que são instalados no WebServer. Na etapa seguinte, esses pacotes geram páginas HTML que incorporam os formatos e dados definidos pelo Designer 2000. Para quem imagina que acessar um servidor de dados corporativo é uma operação mais complexa que acessar um servidor de dados desktop, a solução da Oracle é surpreendentemente simples.

DATA WAREHOUSING

De uma maneira simples, data warehouse é um banco de dados separado de transações, dedicado a gerenciar informação e suporte à decisão. Apesar do conceito ser aplicável a todas as organizações, o custo/benefício restringe o uso de data warehouse apenas para grandes organizações, com grandes volumes de dados. Ele se aplica, especialmente, a organizações que usam mainframes para o processamento de grande quantidade de transações. Pode-se observar, pela descrição de um data warehouse, que o método de distribuição de dados mais indicado para a implantação do mesmo é o Reorganizado.

Quando os dados são transferidos para o data warehouse, eles são traduzidos para um formato comum e integrados. Isto quer dizer que dados relacionados de fontes diferentes são combinados e concatenados. Assim, transações diferentes do mesmo cliente geradas por sistemas diferentes são combinados para oferecer um registro completo da relação do cliente com a empresa.

Um data warehouse é uma arquitetura, não um produto e nenhum software ou desenvolvedor pode fornecer todos os componentes. Apesar de data warehouse não ser um produto, existe um produto chamado "Red Brick Warehouse", que foi concebido especialmente para suporte à decisão e afirma ter uma considerável performance. O data warehouse envolve obter dados do sistema de transações, integrá-los e armazená-los em um banco de dados separado que pode ser acessado pelos usuários através de uma organização para gerenciar as informações e obter suporte à decisão. Ele não é, necessariamente, um único banco de dados, mas pode ser um subconjunto de dados

distribuídos nos servidores locais, departamentais ou de workgroup.

Os dados operacionais diferem dos dados armazenados no data warehouse, entre outras, pelas seguintes características, mostradas na Tabela B [McEL95].

<i>Dados Operacionais</i>	<i>Dados de Warehouses</i>
Vida curta, mudanças rápidas	Vida longas, estáticos
Requer acesso a nível de registro	Os dados são agregados em conjuntos (o que torna os dados dos warehouses amigáveis para RBMDS)
Transações padrão repetitivas e padrões de acesso	Consultas <i>ad hoc</i> com algum relatório periódico.
Atualizado em tempo real	Atualizado periodicamente com cargas em massa
Evento dirige: processos geram dados	Dado dirige: dados governam o processo

Tabela B – Diferenças entre dados operacionais e de warehouses

O objetivo de se montar um data warehouse é dar aos usuários acesso à informações que propiciem uma melhor tomada de decisão em um ambiente de negócios cada vez mais volátil e competitivo. Isto envolve a criação de storehouses ("casas de armazenamento") de informações comerciais, montados a partir de bancos de dados de produção da organização e múltiplas outras fontes, e o fornecimento, aos usuários, de aplicações front-end de fácil utilização para permitir a pesquisa aos dados.

Desta forma, data warehouses requerem uma mistura de tecnologias e áreas de conhecimento, incluindo cliente/servidor, bancos de dados relacionais e multidimensionais, interfaces gráficas com o usuário (GUI) e modelagem de dados.

Como um data warehouse é uma completa coleção de dados sobre a empresa, organizados para um acesso facilitado, algumas empresas já tem utilizado as

facilidades de publicação de bancos de dados na Web em conjunto com os dados armazenados nos data warehouses, como é o caso da American Airlines, que permite que os viajantes chequem os horários dos vôos, e esses dados estão armazenados nos data warehouses da empresa. Empresas que produzem catálogos on-line ou provêm acesso a cotações de bolsa de valores (stock quotes) estão fazendo a mesma coisa.

Além dessas, a rede de vendas de varejo Wal-Mart permite que os fornecedores chequem as informações de estoque de mercadorias via Web. Para a Wal-Mart, a Internet e o data warehousing trabalham em conjunto para ajudar nas compras, e não nas vendas. Para outros, redes Internet internas (Intranets), são uma maneira de dar aos empregados acesso aos dados corporativos.

Mas colocar os seus dados disponíveis na Web pode ser perigoso. No mundo de hoje, a informação é uma das coisas mais valiosas. Ao colocá-la disponível para acesso externo, elas podem cair nas mãos de seu concorrente. A tecnologia disponível para segurança na Internet hoje em dia ainda não é capaz de garantir que as informações não cairão nas mãos erradas. Mas de nada adianta proteger seus dados com firewalls ou outros dispositivos de segurança, se o seu próprio pessoal não for de confiança. Muitos são os casos onde as informações da empresa vazam através de seus próprios funcionários.

DATABASE MARKETING

Os database marketing são aplicáveis a qualquer empresa que tenha alguma camada intermediária entre ela e seus clientes reais, como é o caso de companhias aéreas, hotéis, empresas de hardware e software para computadores, entre outras. A idéia básica do database marketing é voltar ao marketing

de "1 a 1" de antigamente utilizando computadores e bancos de dados como ferramentas de apoio. O marketing "1 a 1" quer dizer que a empresa vai dirigir-se a cada cliente individualmente, mas o fará através de "células de clientes", criando assim um marketing celular. Assim, a melhor forma de distribuição dos dados do banco de dados varia de caso para caso, sendo que os mais comuns são o método de replicação e o particionamento.

Banco de dados de clientes estão em constante mudança. O comportamento de compra destes muda várias vezes durante sua vida, refletindo suas fases de vida. Banco de dados de marketing integrado (integrated database marketing), dá às empresas a capacidade de aprender e medir o comportamento de compra de seus clientes, o que resulta em melhores taxas de resposta, aumenta a possibilidade de lucro e reforça a relação entre a empresa e seus clientes. Apesar disto, em algum lugar em toda a sofisticação, modelagem, planejamento e consultoria, o envio de cartas com peças de marketing direto para os possíveis clientes é tomada como certa.

No mercado de turismo [CYGL96], por exemplo, o database marketing pode ter o seu resultado visto com maior facilidade. A British Airways, por exemplo, disponibiliza um sistema de compra de passagens self service. Quando o passageiro compra, informa seus dados, e a empresa sabe o destino, podendo assim armazenar informações sobre as preferências de viagem (assento, destino, época do ano) de cada passageiro, o que lhe permite, em uma futura compra, oferecer promoções especiais para cada cliente, com base nas informações coletadas anteriormente.

Também do segmento de turismo, sabe-se que 70% do mercado de turistas que fazem cruzeiros marítimos já o fizeram antes. Estes clientes tem um alto poder aquisitivo e, se a empresa souber armazenar as preferências de cada viajante em especial, ela será capaz de customizar as viagens, tornando-as cada vez mais confortáveis para os clientes.

Por que toda essa preocupação em tentar oferecer o melhor serviço possível e de uma forma individual para cada cliente? O custo para conquistar um novo cliente é cinco vezes maior do que o custo para mantê-lo e quanto mais forte o relacionamento da empresa com o cliente, menos pesa para o cliente o custo do bem ou serviço. Quando bem utilizado, o database marketing permite a definição precisa de nichos de mercado, identificando indivíduos e grupos que pertençam a um perfil em particular.

Outra razão forte para a individualização do marketing é o volume de estímulos publicitários recebidos por uma pessoa comum em cidades como São Paulo, Nova Iorque e Tóquio: chega a 30.000 estímulos por dia. Se a empresa não souber exatamente para quem oferecer o bem ou serviço, estará desperdiçando dinheiro em publicidade e marketing.

Não é recomendada a exclusão de registros de um database marketing. Existem problemas lógicos em um banco de dados relacional - como integridade referencial, por exemplo - que impedem a eliminação pura e simples de um registro. Se o cliente não quer mais receber informações, a solução mais simples é utilizar um marcador (flag) para excluí-lo dos processos de impressão de etiquetas, mala direta e outros. Se o cliente não quer ser incomodado com uma campanha de telemarketing, basta proceder da mesma forma. Se o cliente for eliminado da base de

dados, ele poderá ser incluído novamente em uma outra ocasião e a empresa pode cometer o erro de mandar uma mala direta para o mesmo, e ele não gostará nada disso. Duas exceções à essa regra são quando o cliente sai da área geográfica de atuação da empresa, ou então quando ele morre.

Conclusões

As tecnologias atuais de distribuição de dados ainda deixam a desejar, como no caso dos protocolos de *two* e *three phase commit* que, apesar serem implementado por alguns produtos comerciais, ainda precisam ser melhorados e, especialmente, adequados aos meios de comunicação encontrados no Brasil e que serão utilizados, com certeza, na distribuição de dados por uma empresa com filiais em vários estados, por exemplo.

Apesar de todos os problemas que cercam a distribuição de dados, tornando-a de certa forma ainda inviável, algumas empresas já começam a utilizar outras estratégias de distribuição. A Centura, representada no Brasil pela Centura Software do Brasil, por exemplo, já oferece algumas alternativas, como a distribuição em plataformas heterogêneas, onde cada ponto de distribuição tenha a capacidade de distribuir e sincronizar seus dados com outros pontos, tornando os Bancos de Dados menores, mais fáceis de instalar, configurar e gerenciar.

As aplicações expostas são bastante promissoras e algumas delas já estão sendo implementadas com sucesso, principalmente o caso de acesso à bancos de dados via Internet e *database marketing* mas, como qualquer outra transação que trafega pela Internet, os dados ficam exposto e isso pode ser um problema dependendo dos dados que são transportados do servidor ao cliente.

Existem algumas soluções para o problema da segurança, passando desde a simples criptografia por uma chave comum aos dois extremos, chegando até a utilização de criptografia por chave pública e servidores de senha no estilo “kérberos” - apologia ao cão de três cabeças da mitologia grega que guarda a porta do inferno, só deixando entrar as pessoas certas e na hora certa.

Os *data warehouses* também têm evoluído e melhorado bastante, valendo-se de plataformas (hardware) cada vez mais poderosas, com maior poder de processamento, e menor custo por MIPS, mas apesar disto, a tecnologia de manipulação de grandes massas de dados, como é o caso dos *data warehouses* com mais de 1Tb, precisa evoluir.

Podemos prever que a viabilidade comercial dos sistemas de gerência de bancos de dados distribuídos, em especial no nosso país, só se dará com a melhoria dos mecanismos de garantia de integridade transacional, bem como dos meios de telecomunicação que interligam os diversos nós, garantindo meios de transmissão de dados rápidos, confiáveis e baratos, tornando a tecnologia, desta forma, acessível a uma gama muito maior de usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BYTE95] BYTE BRASIL - São Paulo, Rever. v. 4, n. 5, mai/1995
- [CAMA93] CAMARÃO, Paulo C. B. - Glossário de Informática - 1a. edição, 4a. reimpressão, São Paulo, LTC, 1993.
- [CERÍ95] CERÍCOLA, Vicent O. - Oracle: Banco de Dados Relacional e Distribuído - São Paulo, Makron Books, 1995.
- [COUC84] COUCEIRO, Luiz A. C. & BARRENECHA, Hugo, F. S - Sistemas de Gerência de Banco de Dados Distribuídos - Rio de Janeiro, LTC, 1984
- [CYGL96] CYGLER, Jimmy - DBM: Aplicações no Campo Turístico - X SEPAI, Belém, out/1996.
- [DATA96] DATA WAREHOUSING & THE NET BECOMING A TEAM - Newsbytes News Network, out/1996.
- [DATE81] DATE, C. J. - Introdução a Sistemas de Bancos de Dados - 4a. edição - Rio de Janeiro, Campus, 1981
- [DUCK95] DUCKETT, Mike - The Two-Phase Commit Protocol - <http://ei.cs.vt.edu/~cs5204/distributedDBMS/duckett/tpcp.html>
- [EDWA95] EDWARDS, Morris - How to build and profit from data warehousing - Communications News, v. 32, nov/1995, p. 49.
- [EMIG95] EMIGH, Jacqueline - Bull Plans “Three-In-One” Data Warehouse - Newsbytes News Network, mai/1995.
- [EURO96] EUROPEAN DATA WAREHOUSING SURVEY - Newsbytes News Network, agosto/1996.
- [INFO96] INFORMÁTICA EXAME ESPECIAL - São Paulo, Abril. Ano 11, n. 06, set/1996.
- [McEL95] McELREATH, Jack - Data Warehouses: An Architectural Perspective - CSC's Consulting & Systems Integration, 1995
- [McFA95a] McFALLRY, Ryan - Coordinator in Three-Phase Commit Protocol - <http://www.cs.hope.edu/~mcfallry/390/ThreePhaseCoord.html>
- [McFA95b] McFALLRY, Ryan - Participant in Three-Phase Commit Protocol - <http://www.cs.hope.edu/~mcfallry/390/ThreePhasePart.html>
- [MORE96] MORENO, Alfonso - Implementação de uma Aplicação Cliente/Servidor sobre Bancos de Dados Distribuídos - Encontro Nacional de Usuários Oracle, Grupo de Usuários Oracle Brasil, 1996

- [NEWI96] NEWING, Rod - Data warehousing - Management Accounting, v. 74, mar/1996, p. 21.
- [RENA94] RENAUD, Paul E. - Introdução aos Sistemas Cliente/Servidor - Guia Prático para Profissionais de Sistemas - Rio de Janeiro, Livraria e Editora Infobook S/A, 1994.
- [RUFF94] RUFFIN, Marshall - The importance of data warehouses for physician executives - Physician Executive, v. 20, nov/1994, p. 45.
- [SETZ86] SETZER, Valdemar, W. - Bancos de Dados - Conceitos, Modelos, Gerenciadores, Projeto Lógico, Projeto Físico - São Paulo, E. Blücher, 1986.
- [TANE92] TANEMBAUM, Andrew S. - Sistemas Operacionais Modernos - Rio de Janeiro, Prentice-Hall do Brasil, 1992
- [UNIS95] UNISYS Computers - What Is Two-Phase Commit? - <http://www.unisys.com/softstor/2200/2pc.html>