

Universidade Federal de Santa Catarina  
Programa de Pós Graduação em Ciências da Computação

Disciplina: INE651600  
Sistemas Operacionais  
Professor: Antônio Augusto Fröhlich

Estudo Dirigido sobre Bancos de Dados Distribuídos

Aluno: Renato Deggau – matrícula 200800034  
Setembro/2008

# Banco de Dados Distribuídos

## 1. Introdução

Este trabalho apresenta uma visão geral sobre Bancos de Dados Distribuídos e efetua uma descrição de pesquisas atuais na área. Apresenta as características principais da tecnologia e descreve trabalhos recentes que buscam atualizar os conceitos da tecnologia e do uso de bancos de dados distribuídos.

## 2. Bancos de Dados Distribuídos

### 2.1 Conceitos

Banco de dados distribuído (BDD) pode ser definido com uma coleção de múltiplos bancos de dados logicamente inter-relacionados e distribuídos sobre uma rede de computadores. Um sistema de gerenciamento de bancos de dados distribuídos (SGBDD) pode ser definido como um sistema que permite o gerenciamento dos BDDs e efetua a distribuição de forma transparente aos usuários.

Num SGBDD todos os pontos estão interligados entre si, compartilhando de um mesmo esquema global; cada ponto é composto de um hardware e software próprios, com sistema operacional, memória e discos próprios.

Esta tecnologia é a união de duas abordagens distintas da tecnologia da informação: sistemas de banco de dados e redes de computadores. Numa primeira impressão, parece difícil entender como estas abordagens contrastantes podem produzir uma tecnologia mais poderosa e promissora do que cada uma individualmente. Mas a chave para esta compreensão é que a tecnologia da informação busca a integração e não a centralização. E isto exatamente que a tecnologia de bancos de dados distribuídos busca obter.

### 2.2 Vantagens, Desvantagens e Características

As principais vantagens dos bancos de dados distribuídos são:

- A natureza distribuída de algumas aplicações de bancos de dados; muitas aplicações estão distribuídas naturalmente em diferentes lugares. Uma empresa pode ter a sua matriz e tem filiais espalhadas em diversos locais; é natural que os dados deste banco estejam distribuídos entre as filiais, sendo que usuários locais do sistema têm acesso aos dados da sua filial, enquanto usuários globais têm acesso ao conjunto de todos os dados.
- Maior confiabilidade e disponibilidade; estas são vantagens potenciais das bases distribuídas. Confiabilidade é a probabilidade que um sistema esteja em funcionamento em determinado momento. Disponibilidade é a probabilidade que um sistema esteja disponível continuamente durante um intervalo de tempo. Quando os dados estão distribuídos em vários pontos, se um ponto falha os demais continuam em operação, ficando indisponíveis somente os dados do ponto onde ocorreu o problema. Isto melhora tanto a confiabilidade quanto a disponibilidade. Caso se utilize replicação

de dados através de um critério adequado se obtém um índice ainda maior. Num sistema centralizado, a falha do ponto central faz com que toda a rede fique indisponível; porém numa rede com replicação, a queda de um ponto não significa a indisponibilidade daqueles dados.

- Possibilidade de compartilhamento de dados enquanto se mantém certo grau de controle local; políticas distintas de acesso e uso dos dados podem ser estabelecidas em cada ponto da rede; é possível controlar os dados e o software em cada ponto da rede, possibilitando o acesso a usuários remotos de outros pontos, permitindo um compartilhamento controlado de todos os dados de um sistema distribuído.
- Melhor performance; com a distribuição dos dados de uma grande base de dados em pontos menores, as consultas e transações que tem acesso somente a dados de um ponto ficam mais rápidas. E também, cada ponto tem um número menor de transações a gerenciar do que se todas as transações estivessem num banco de dados centralizado. Em cada ponto da rede, como existe apenas uma parte do banco de dados, a contenção de CPU e de serviços de I/O não são tão fortes como em bancos centralizados. E no caso de transações que necessitam acessar diversos pontos, o processamento pode ser efetuado em paralelo reduzindo o tempo de resposta.

As desvantagens que podem ser encontradas:

- Complexidade – alguns problemas são inerentemente mais complexos que em bancos centralizados, assim como incluem alguns novos problemas.
- Custos – a necessidade de hardware adicional, principalmente nos aspectos relacionados a serviços de infra-estrutura de comunicação, incrementa o custo, assim como a necessidade de aumento de pessoas para o gerenciamento desta nova estrutura. Entretanto este custo deve ser comparado com os benefícios advindos da nova estrutura.
- Controle – a distribuição de controle, considerada também como uma vantagem, cria problemas de sincronização e coordenação, caso políticas adequadas não sejam adotadas.
- Segurança – um dos maiores benefícios dos ambientes centralizados é o controle que proporciona para o acesso aos dados. Em ambientes distribuídos, uma rede é envolvida onde requisitos próprios de segurança são requeridos. Desta forma, os aspectos relativos à segurança em ambientes distribuídos são mais complexos que em ambientes centralizados.

Esta distribuição de dados produz um aumento da complexidade do projeto do banco de dados e da implementação do sistema. Para que as vantagens anteriormente descritas sejam obtidas é necessário que o software gerenciador do banco de dados distribuído tenha, além das funções de um banco centralizado, as seguintes características:

- A possibilidade de ter acesso a pontos remotos e transmitir dados e consultas entre os diversos pontos através da rede;
- A capacidade de seguir o caminho da distribuição e replicação dos dados através do catálogo do banco de dados distribuído;

- A capacidade de elaborar estratégias de execução para consultas e transações que tenham acesso a dados de mais de um ponto;
- A capacidade de decidir a qual cópia de um elemento de informação replicada se terá acesso;
- A capacidade de manter a consistências das copias de um elemento de informação replicado;
- A capacidade de recuperar-se de quedas de pontos individuais e de outros tipos de falhas como a falha no enlace de comunicação.

Estas funções elevam a complexidade de um SGBDD em comparação com um SGBD centralizado. Para a utilização adequada destes potenciais obtidos com a distribuição devemos obter soluções satisfatórias para estas questões.

No nível físico, hardware, os principais fatores que distinguem um sistema centralizado de um distribuído são:

- Possuem muitos computadores, chamados pontos, sítios ou nós;
- Estes pontos devem estar conectados através de uma rede de comunicação.

Estes pontos podem estar fisicamente próximos, ligadas por uma rede local (LAN), ou geograficamente distantes, ligadas através de uma rede de longa distância (WAN). A combinação desta situação também pode ser utilizada.

As redes podem possuir diferentes topologias para definir o caminho de comunicação entre elas. O tipo de topologia pode ter impacto significativo sobre o rendimento da rede.

## 2.3 Armazenamento dos dados

Uma relação ou tabela num banco de dados distribuído pode estar armazenada segundo diversos enfoques:

- Fragmentação: a tabela é particionada em fragmentos e cada fragmento é armazenado em um ponto diferente;
- Replicação: o sistema mantém cópias da tabela que são armazenadas em pontos diferentes;
- Replicação e fragmentação: a tabela é particionada em vários segmentos, e cópias destes segmentos são distribuídas pelo sistema.

A fragmentação horizontal é definida como o subconjunto de tuplas de uma relação. As tuplas que pertencem a este subconjunto são determinadas através de uma condição que envolva um ou mais atributos desta relação. Esta condição pode ser, por exemplo, um atributo de departamento ou localização geográfica. A fragmentação horizontal divide a relação horizontalmente através de algum significado lógico. E estes fragmentos podem então ser associados a diferentes pontos do sistema distribuído.

A fragmentação vertical é definida como o subconjunto de certos atributos de uma relação. Num exemplo de um cadastro, poderíamos separar as informações pessoais em um fragmento e as informações pessoais em outro fragmento. É necessária a replicação da chave primária nos segmentos para os segmentos fragmentados possam ser reagrupados ou para que se

tenha acesso às demais informações de uma tupla. A fragmentação vertical consiste na divisão em colunas de uma relação.

Um conjunto de fragmentos horizontais cuja união corresponda a todas as tuplas de uma relação R, denomina-se fragmentação horizontal completa de R; um conjunto de fragmentos verticais cuja projeção inclui todos os atributos de R denomina-se fragmentação vertical completa de R.

A mescla destas duas fragmentações produz uma fragmentação mista. Um fragmento de uma relação pode ser obtido através da combinação de operações SELECAO e PROJEÇÃO.

Um esquema de fragmentação de uma base de dados é uma definição de um conjunto de fragmentos que incluem todos os atributos e tuplas de uma base de dados e satisfaça a condição de que a base completa possa ser reconstruída através de operações de união.

Um esquema de repartição descreve a divisão de fragmentos entre os pontos do banco distribuído, ou seja, é uma correspondência que especifica onde é armazenado cada fragmento. Se um fragmento é armazenado em mais de um ponto, se diz que ele está replicado.

A replicação é útil para melhorar a disponibilidade dos dados. Um banco de dados distribuído é chamado de totalmente replicado quando toda a base de dados está replicada em todos os pontos que o compõe. Isto melhora a disponibilidade, pois o sistema pode operar enquanto houver algum ponto ativo. Melhora também a performance das consultas, porque o resultado de qualquer consulta pode ser obtido localmente. A desvantagem da replicação completa é que diminui drasticamente a velocidade das operações de atualização, pois uma só atualização deve ser executada em cada uma das cópias da base para que a consistência seja mantida. O oposto da replicação completa é não ter nenhuma replicação, onde cada fragmento encontra-se em apenas um ponto da rede. Entre estes dois casos temos uma ampla gama de replicação parcial dos dados. Um esquema de replicação é uma descrição da replicação dos fragmentos de um banco de dados distribuído.

Cada fragmento, ou cada cópia de fragmento, deve estar associada a um ponto determinado do sistema distribuído. Este processo se chama distribuição dos dados. A escolha dos pontos e o grau de replicação dependem dos objetivos de performance e disponibilidade desejados.

## 2.4. Tipos de Sistemas Distribuídos

Os bancos de dados distribuídos podem ser divididos em diferentes tipos, de acordo com algumas características.

O primeiro fator é o grau de homogeneidade do sistema. Se todos os servidores locais utilizam software idêntico, se diz que o banco de dados distribuído é homogêneo. Em caso contrário se diz que é heterogêneo.

Bancos homogêneos são mais simples de administrar, tem uma abordagem que provê um crescimento incremental e permite uma performance incremental. Nos bancos heterogêneos, os pontos podem implementar seus próprios bancos de dados, sendo a integração considerada e tratada posteriormente; existe a necessidade de traduções para permitir a integração entre

os diferentes hardwares, produtos de banco de dados e a combinação destes. Uma solução típica neste caso é o uso de gateways.

O Open Group formalizou uma especificação nominada Specification Working Group (SWG) para permitir a criação de um ambiente de infra-estrutura de banco de dados onde há:

- Common SQL API – para permitir que aplicações cliente sejam escritas sem a necessidade de conhecer o fornecedor do banco de dados que esta sendo acessado;
- Common Database Protocol: permite que um banco de dados de um fabricante comunique-se com um banco de dados de outro fabricante sem a necessidade de um gateway
- Common network protocol: para permitir a comunicação entre diferentes bancos de dados

Outro fator relacionado com o grau de homogeneidade é o grau de autonomia local. Se todo o acesso ao banco se dá através de um cliente, dizemos que o sistema não tem autonomia local. Se permitir as transações locais acessarem diretamente um servidor, o sistema terá certo grau de autonomia local. Num extremo de autonomia temos um banco de dados distribuído que dá ao usuário a impressão de um banco centralizado. Só há um esquema conceitual, e todo o acesso se dá através de um cliente, sem existir autonomia local. No outro extremo encontramos com um tipo de banco de dados distribuído denominado federado (ou sistema de múltiplas bases de dados - MDBS). Num sistema assim, cada servidor é um SGBD centralizado independente e autônomo que tem seus próprios usuários, transações locais e um DBA, possuindo assim alto grau de autonomia local.

Cada servidor pode autorizar o acesso a partes específicas de acordo com as classes dos usuários. Um sistema federado é uma combinação de sistemas distribuídos e centralizados. Para usuários autônomos locais é um sistema centralizado e para usuários globais é um sistema distribuído.

Em sistemas heterogêneos um servidor pode ser um banco relacional, outro pode ser um de rede e outro um banco hierárquico. Neste caso [é necessário contar com uma linguagem e incluir tradutores de linguagem no cliente a fim de traduzir as consultas.

Um terceiro aspecto que serve para a classificação das bases de dados distribuídas é o grau de transparência da distribuição, ou grau de integração dos esquemas. Se o usuário percebe um só esquema integrado sem informação alguma relativa à fragmentação, replicação ou distribuição, se que diz que o SGBDD tem alto grau de transparência de distribuição. Se o usuário pode ver toda a fragmentação e a replicação, o SGBDD não tem transparência de distribuição.

## **2.5 Processamento de consultas em bancos distribuídos**

Examinaremos como o SGBDD processa e otimiza uma consulta. Num sistema distribuído, vários fatores complicam o processamento de uma consulta. O primeiro é o custo de transferir dados pela rede, quer sejam arquivos intermediários de troca entre pontos de rede ou os resultados finais de uma consulta que devem ser transferidos ao ponto original.

Por este motivo os algoritmos de otimização de consultas em SGBDD consideram o objetivo de reduzir a quantidade de dados transferidos como critério de otimização para a escolha de uma estratégia de execução de uma consulta distribuída. A solução que prover a menor taxa de transferência de dados será a escolhida como estratégia de otimização.

A junção parcial se baseia na idéia de reduzir o numero de tuplas de uma relação antes de transferi-la para outro ponto. A idéia é enviar a coluna de junção de uma relação para o ponto onde se encontra a outra relação. Depois disso, os atributos de junção, associados aos atributos de resultado são extraídos e devolvidos ao ponto original. Isto pode ser uma solução eficiente quando uma pequena fração das tuplas pertence à junção.

## 2.6 Decomposição das consultas e das atualizações

Em SGBDD sem transparência de distribuição, o usuário de consultas deve especificar a sua consulta diretamente sobre fragmentos específicos. Deve também garantir a consistência dos elementos replicados quando atualiza informações. Em SGBDD com transparência de distribuição, a alteração é especificada como num banco centralizado, e o sistema se encarrega de manter a consistência dos elementos replicados.

No caso de consultas, um módulo de decomposição de consultas devera dividir uma consulta em sub-consultas que possam ser executadas nos pontos individuais, combinando os resultados e formando o resultado da consulta. Para determinar qual replica faz referencia aos elementos da consulta, o SGBDD consulta as informações de fragmentação, replicação e distribuição em seu catálogo.

## 2.7 Controles de concorrência e recuperação em bancos de dados distribuídos

No controle de concorrência e recuperação num banco de dados distribuídos, surgem muitos problemas que são inexistentes em um banco de dados centralizado. São eles:

- Manipular múltiplas cópias dos elementos de informação – o método de controle de concorrência tem a obrigação de manter consistentes as diversas copias. O método de recuperação deve cuidar que uma copia seja consistente com as demais no caso da recuperação posterior de um ponto que tenha ocorrido falha.
- Falhas de pontos individuais – o SGBDD deve continuar operando com seus pontos ativos quando os demais falham. Quando um ponto se recupera, sua base de dados deve se atualizar com os demais pontos antes de se reincorporar ao sistema.
- Falhas nos enlaces de comunicação – o sistema deve ser capaz de tratar falhas em um ou mais dos enlaces de comunicação entre os pontos. Um caso extremo deste problema é que pode haver particionamento da rede. Isto divide os pontos em duas ou mais partições dentre as quais os pontos podem se comunicar, porém não com as outras partições.
- Confirmação de gravação distribuída – pode haver problemas para confirmar uma transação que está tendo acesso a bases armazenadas em múltiplos pontos se alguns destes pontos falham durante o processo de confirmação; um protocolo de confirmação de fases deve ser usado para resolver este problema.

- Gerenciamento do Deadlock distribuído – pode ocorrer um deadlock entre os diversos pontos, tornando necessária a extensão de técnicas de manejo de deadlocks para considerar este ponto. O controle pode ser através um coordenador de deadlock centralizado, ou por uma detecção hierárquica ou pela detecção distribuída de deadlock.

As técnicas de controle de concorrência e de recuperação distribuída devem resolver estes e outros problemas. A seguir algumas destas técnicas serão descritas.

### **2.7.1 Controle de concorrência distribuída baseada em cópia distinta de um elemento de informação**

Existem vários métodos de controle de concorrência distribuída que tratam os elementos de informação de uma base distribuída, que se baseiam numa extensão das técnicas de controle de concorrência das bases centralizadas. Serão analisadas estas técnicas no contexto de estender o bloqueio centralizado. A idéia é designar uma determinada cópia de cada elemento de informação como uma cópia distinta. Os candidatos a este elemento de informação se associam a esta cópia e todas as solicitações de bloqueio e desbloqueio se enviam ao ponto que contém esta cópia.

Diferentes métodos se baseiam nesta idéia, diferindo na forma de escolha da cópia distinta. Na técnica de ponto primário, todas as cópias ficam no mesmo ponto. Uma modificação deste enfoque é o ponto primário com ponto de backup. Outro enfoque é a cópia primária, no qual as cópias se podem armazenar em diversos pontos. Vamos detalhar estes métodos:

#### **2.7.1.1 Técnica do ponto primário**

Neste método se designa um só ponto primário como ponto coordenador para todos os elementos. A vantagem deste enfoque é que ele é uma simples extensão do enfoque centralizado e, portanto não muito complexo. Mas possui algumas desvantagens: a possível sobrecarga deste ponto gerando um gargalo no sistema. Outra desvantagem é que uma falha neste ponto primário paralisaria o sistema já que toda a informação de bloqueio se mantém neste ponto. Isto pode limitar a confiabilidade e disponibilidade do sistema. Ainda que o acesso de todos os candidatos se dá no ponto primário, o acesso aos elementos pode se dar em qualquer ponto que residam.

#### **2.7.1.2 Técnica do ponto primário com ponto de backup**

Esta técnica busca resolver a segunda desvantagem do método anterior; indicando outro ponto como backup. Toda a informação de bloqueio se mantém nos dois pontos. No caso de falha do ponto principal, o backup assume as funções. Porém este processo torna o acesso mais lento, já que os acessos devem ser efetuados tanto no ponto primário como no backup, sobrecarregando o sistema.

#### **2.7.1.3 Técnica da cópia primária**

Este método tenta distribuir a carga entre os vários pontos, mantendo cópias distintas de elementos diferentes de informação armazenadas em pontos distintos. Uma falha num ponto afeta todas as transações que estão tendo acesso a elementos cujas cópias primárias residam neste ponto, mas as demais transações não são afetadas. Este método também pode usar pontos de backup para aumentar a confiabilidade e a disponibilidade.



#### **2.7.1.4 Eleição de um novo ponto coordenador em caso de falha**

Sempre que um ponto coordenador falha em qualquer das técnicas anteriores, os pontos que seguem ativos deverão eleger um novo coordenador. Num caso de ponto primário sem ponto de backup, será preciso abortar e reiniciar todas as transações em execução. Este processo será trabalhoso. Parte do processo implica em selecionar um novo ponto primário e criar um processo gestor de bloqueios e um registro dos bloqueios deste ponto. Em métodos que usam pontos de backup, o processamento de transações é suspenso, o ponto de backup se nomeia como novo ponto primário; é então escolhido outro ponto de backup e se enviam a ele todas as informações de bloqueio do novo ponto primário. O novo ponto de backup pode ser escolhido entre os pontos ativos do sistema dentre quaisquer dos pontos ativos através de uma eleição.

#### **2.7.2 Controle de concorrência baseada em votação**

Os métodos analisados até aqui se baseiam na idéia de uma copia que mantém os bloqueios deste elemento. No método de votação não há copia distinta e cada solicitação de bloqueio é enviada a todos os pontos que tenham uma cópia do elemento de informação. Cada cópia mantém seu próprio bloqueio e pode conceder ou negar a solicitação. Se a maioria das cópias fornece um bloqueio para a transação que a solicita, esta obterá o bloqueio e informará a todas as cópias que o mesmo foi concedido. Se uma transação não recebe a maioria dos votos durante certo período de tempo pré-definido, a solicitação será cancelada e informada a todos os pontos.

O método de votação se considera um método de controle verdadeiramente distribuído, já que a responsabilidade da decisão cai sobre todos os pontos envolvidos. Estudos de simulação indicam que a votação tem um tráfego maior de mensagens entre os pontos do que o método de cópia distinta. Se o algoritmo leva em conta as possíveis falhas dos pontos durante o processo de votação, este se torna extremamente complexo.

#### **2.7.3 Recuperação distribuída**

O processo de recuperação de bases de dados distribuídas é bastante complexo. Em certos casos é bastante difícil determinar se um ponto está inativo, sem trocar um grande número de mensagens com outros pontos. Outro problema com a recuperação distribuída é a gravação distribuída. Quando uma transação está atualizando dados em diversos pontos, não é possível a gravação sem a certeza de que o efeito da transação em todos os pontos não pode se perder. Isso significa que cada ponto deve ter gravado permanentemente em disco os efeitos da transação.

### **3. Aspectos atuais de pesquisa em Bancos de Dados Distribuídos**

A seguir são descritos alguns aspectos de pesquisa atuais na área de Bancos de Dados Distribuídos:

- Gorla [4] apresenta um estudo que aperfeiçoa o particionamento vertical, incluindo restrições de integridade referencial, tempos de acesso baseados em heurística, e um modelo de custo que considera outros tipos de transação como updates e joins. Os resultados obtidos apresentam ganhos de 36% a 65% sobre casos não particionados. O método aplicado em pequenos bancos de dados resultou em esquemas particionados

que são comparáveis ao ótimo. A procedure proposta pelo autor produz soluções de particionamento em poucos segundos, e pode ser aplicada em bancos de dados distribuídos e orientados a objeto.

- Wiesmann[5] apresenta uma comparação de performance das técnicas de replicação de bancos de dados baseadas na primitiva “total order broadcast”, que assegura que todas as mensagens são entregues de forma confiável na mesma ordem em todas as réplicas. É efetuada uma comparação entre estas técnicas e técnicas clássicas de replicação como cópias primárias, bloqueio distribuído e outras. As avaliações mostram que as técnicas baseadas no “total order broadcast” têm desempenho superior. E que a diferença de desempenho se amplia em casos de redes lentas e sujeitas a contenção. Isso se deve ao fato de que as técnicas de bloqueio distribuído usam muito os recursos de mensagens. Em redes mais rápidas recursos de CPU e disco são candidatos maiores a gargalo do que uma rede não indicando, porém que a comunicação não seja um ponto a considerar na replicação de bancos de dados: múltiplas cópias ainda levam a problemas como custos de sincronização e altas taxas de interrupção do processo. Mas estas características não tem relação com a performance da rede e sim com o projeto da técnica de replicação.
- Huebsch [6] aborda a questão do processamento de múltiplas queries de agregação simultânea, onde o processamento de cada query de forma individual se torna impraticável, tornando a otimização de múltiplas queries um processo crítico. A idéia do processo é diminuir a comunicação na rede. Sua idéia avança sobre o contexto de queries de agregação distribuídas que variam nos seus predicados. A partir de um número  $q$  de queries de agregação, buscam-se oportunidades de identificar  $K$  queries onde  $k \ll q$ , através da identificação de conexões entre os dados e os predicados destas queries.
- Atreya [7] apresenta uma abordagem para a identificação de predicados ‘stable’ em computação distribuída; o deadlock em sistemas distribuídos é um exemplo de ‘stable properties’. Eles apresentam um algoritmo que se mostra mais eficaz para esta identificação, através do monitoramento de mudanças de valor em variáveis relevantes: isto é feito através de mudanças na aplicação para que o algoritmo de detecção identifique modificações em variáveis relevantes ou pelo monitoramento sem alteração do programa, mas através de uma abordagem mais eficiente.
- Hachen [8] descreve o sistema Gaea, um sistema de gerenciamento de bancos de dados espaço-temporais para bancos de dados científicos. O sistema se propõe a: fornecer suporte de banco de dados para todas as fases do gerenciamento de dados e metadados científicos; estender a tecnologia de banco de dados com classes de operadores que respondam as crescentes necessidades da pesquisa científica; integrar domínios espaciais e temporais envolvendo grandes volumes de dados e permitir uma clara extensão para um ambiente de computação distribuída envolvendo bancos de dados heterogêneos e especializados com recursos computacionais. Dentro da estrutura do Gaea, o sistema de back-end consiste de um banco de dados distribuído. Este ambiente distribuído consiste de operadores de análises científicas os quais estão disponíveis em sistemas de domínio público ou comerciais.
- Gupta [9] apresenta uma proposta para otimização da verificação de integridade em bancos de dados distribuídos. Ela é baseada em um algoritmo que permite que uma

constraint global (constraint que deve ser aplicada nos vários bancos) seja analisada com o acesso aos dados em uma base única, eliminando o custo de acesso a dados remotos. Isto se dá através da identificação da restrição global e dos dados a serem inseridos numa base local. O algoritmo produz uma condição local de forma que se o dado local satisfizer esta condição, esta será aceita na restrição global. Caso contrário existe a necessidade de verificação global.

- Baião [10] apresenta a aplicação da teoria da revisão no projeto de bancos de dados distribuídos através de algoritmo que decide a técnica de fragmentação a ser utilizada em cada classe do banco de dados. Esta técnica encontra esquemas de fragmentação melhores e aumenta a performance.
- Lin [11] apresenta um estudo que mostra uma nova abordagem para o processo de checkpoint em bancos de dados distribuídos.
- Wildemberg [12] trata do problema da alocação de dados em bancos de dados distribuídos. Algoritmos de alocação são usados para descobrir uma distribuição de dados ideal. Ele propõe um novo algoritmo que busca um custo de execução mínimo, mantendo a complexidade do algoritmo original. O algoritmo apresenta resultados melhores que o algoritmo original.
- Baião [13] indica que os principais problemas em bancos de dados distribuídos concentram-se em: distribuição dos dados (particionamento e replicação); controle de concorrência distribuído onde se busca um equilíbrio entre manutenção da consistência e o alto nível de concorrência; o processamento de consultas distribuído onde deve ser considerado o custo de otimização e o custo de execução. Destaca como pontos em aberto: problemas de crescimento da rede; o processamento de transações distribuído e a integração com sistemas operacionais distribuídos já que os bancos de dados distribuídos possuem características específicas tais como suporte a transações distribuídas com controle de concorrência e recuperação e que não são estão perfeitamente integradas com sistemas operacionais distribuídos. Levanta também a questão da integração da convergência de bancos de dados distribuídos com a gerência de dados na Web, existindo a necessidade de convergência de duas culturas distintas, com necessidades de interoperabilidade e mudanças de tecnologia, como arquiteturas multi-camadas e serviços Web

#### **4. Conclusões e utilização no meu trabalho**

A tecnologia de bancos de dados distribuídos tem evoluído e seus conceitos utilizados ou aplicados em novas tecnologias. O trabalho identificou vários pontos de pesquisa que visam aperfeiçoar este processo. A Web exigiu novos esforços para a integração dos bancos de dados distribuídos, onde os processos são compostos por sistemas não tão fortemente acoplados que são ligados por tecnologias como Web Services. E o controle de transações sobre estes sistemas precisa continuar garantindo as propriedades das transações, garantindo a consistência e a confiabilidade do banco de dados.

Em minha dissertação, que busca a integração e novas formas de representação de informações através de data-warehouses e sistemas de informação geográfica, não tenho identificado ainda a necessidade de uso de bancos de dados distribuídos. Porém esta

tecnologia pode ser utilizada e considerada como um aspecto na busca e integração de informações provenientes de diversas fontes de dados que poderiam estar num banco de dados distribuído.

## **Bibliografia**

- [1] ELMASRI, NAVATHE – Sistemas de Bases de Datos, segunda Edición, Addison-Wesley Iberoamericana
- [2] Özsu, Valduriez – Principles of Distributed Database Systems – Prentice Hall, 1991
- [3] Clack. C. DatabasesIIIIdistributedDBMSs. Disponível em <http://www.cs.ucl.ac.uk/staff/C.Clack/3C13/powerpoints/databasesIIIIdistributedDBMSs.ppt>. Acesso em 24/08/2008.
- [4] Gorla, Narasimhaiah. A Methodology for Vertically Partitioning in a Multi-Relation Database Environment. JCS&T Vol. 7 No. 3. 217-227
- [5] Wiesmann, Shiper. Comparison of database replication techniques based on total order broadcast. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 4, APRIL 2005. 551-566
- [6] Huebsch et al. Sharing Aggregate Computation for Distributed Queries. SIGMOD'07, June 11–14, 2007, Beijing, China
- [7] Atreya et al. Efficient detection of a locally stable predicate in a distributed system. J. Parallel Distrib. Comput. 67 (2007) 369– 385
- [8] Hachen et al. Distributed Database Management for Scientific Data Analysis. Worcester Polytec Institute.USA
- [9] Gupta. A. Local verification of global integrity constraints in distributed databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1993
- [10] Baiao. Applying Theory Revision to the Design of Distributed Databases. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.261>
- [11] Lin J. Dunham M. A survey of distributed database checkpointing. Distributed and Parallel Databases Journal. 1997. Vol 5 p 31.
- [12] Wildemberg et al. Alocação de Dados em Bancos de Dados Distribuídos. SBBD 2003
- [13] Baiao, F. Banco de Dados Distribuído. Apresentação. Disponível em [www.cos.ufrj.br/~baiao/ISExpert2003/BDDPublicado.pdf](http://www.cos.ufrj.br/~baiao/ISExpert2003/BDDPublicado.pdf). Acesso em 29/08/2008

