Alto Desempenho em Sistemas de Banco de Dados

Professor:

Carina Friedrich Dorneles[5]
(dorneles@upf.br)

Resumo:

Muitos sistemas modernos, como os usados em telecomunicações e em dispositivos móveis, têm necessidade de acesso a sistemas de bancos de dados. Para que este tipo de aplicação seja eficaz, é necessário que se forneça segurança e armazenamento confiável dos dados, e que seja possível recuperar e processar os dados de forma bastante eficiente. Desta forma, é importante que sistemas gerenciadores de bancos de dados estejam preparados para fornecer suporte a aplicações que necessitem alto desempenho no armazenamento e principalmente do processamento das consultas.

5 Doutora em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (jan/2006). Desenvolveu atividades de doutorado sandwich University of Washington (Seattle, USA) no ano de 2003. Atuou como pesquisadora-bolsista na UFRGS desde 1997 onde desempenhou atividades em projetos com a IBM e outros financiados pelo CNPq e Capes. Atua, desde 2001, como docente na UPF, desenvolvendo atividades de ensino nas disciplinas de Banco de Dados e de XML. Atualmente desenvolve atividades de pesquisa nas áreas de Banco de Dados, Integração de Dados em ambientes heterogêneos e em XML. Ministrou o mini-curso entitulado "XML: teoria e aplicações" em 4 edições do SBBD. Recentemente, coordenou os eventos II Escola Regional de Banco de Dados (2006), II Sessão de Demos do Simpósio Brasileiro de Banco de Dados(2004 E 2005).

3.1. Introdução

Aplicações de alto desempenho que fazem uso de Sistemas Gerenciadores de Banco de Dados devem se apoiar naqueles que provêem suporte ao processamento paralelo e distribuído [10]. Em contrapartida, bancos de dados em arquiteturas paralelas exigem aplicações desenvolvidas especificamente para este tipo de ambiente.

As aplicações com necessidade de processamento paralelo têm impulsionado a exploração de técnicas de processamento paralelo em Sistemas Gerenciadores de Banco de Dados (SGBDs), apesar de ainda identificar-se muitas questões em aberto na área [9, 4]. Muitas destas aplicações envolvem processamento de dados para suporte a decisão, com execução em tempo hábil de consultas complexas sobre grandes volumes de dados. Um Sistema Paralelo de Banco de Dados (SPBD) pode ser definido como um SGBD implementado em uma máquina com multiprocessadores, ou seja, um computador com capacidade de processamento paralelo.

Atualmente Sistemas Paralelos de Banco de Dados têm a tendência de se voltarem a arquiteturas paralelas que explorem a utilização de componentes (processadores, memória) comuns, ou seja, os SPBD não são dependentes de hardware específico. Entretanto o modelo de memória da máquina hospedeira tem influência na arquitetura do SPBD. Sistemas paralelos de banco de dados estao ganhando popularidade como uma solucao que fornece alta performance e escalabilidade em grandes bancos de dados. Um sistema sistema paralelo de banco de dados explora multiprocessamento para melhorar performance. Arquiteturas sistemas paralelos de banco de dados pode ser classificados, de forma geral, em três categorias: *shared memory*, *shared disk* e *shared nothing*. Uma importante questão, entretanto, é qual arquitetura deve ser usada para uma aplicação de banco de dados específica. Cada arquitetura tem suas qualidade e fraquezas.

O artigo está organizado da forma como segue. Na Seção 4 são apresentados aspectos importantes em um SPBD, considerando questões de paralelismo, particionamento de dados e execução de consultas. Na Seção 5 é descrito brevemente a idéia de *cluster* na área de banco de dados. Na Seção **??** algumas questões e discussões iniciais sobre Grid em Banco de dados. Finalmente, a Seção apresenta algumas conclusões.

3.2. Sistemas Paralelos de Banco de Dados

Considerando de forma bastante ampla, um sistema paralelo de banco de dados pode ser considerado com um Sistema Gerenciador de Banco de dados implementado em um multiprocessador firmemente acoplado, sendo uma extensão, natural, de um sistema de banco de dados distribuídos com um servidor por nodo.

David J. DeWitt e Jim Gray [2] definiram algumas técnicas básicas que devem ser consideradas implementação, as quais são descritas sob dois aspectos: (i) métricas e objetivos do paralelismo: aceleração linear e crescimento linear; e (ii) arquitetura do hardware: *shared-memory*, *shared-disks* e *shared-nothing*. Além destes aspectos, é crucial considerar alocação dos dados e execução de consultas em tais ambientes.

3.2.1. Métricas

Como mencionado, David J. DeWitt e Jim Gray [2] afirmam que um sistema paralelo ideal deve apresentar duas propriedades chave:

1. Aceleração linear (*linear speedup*) - não leva em consideração o tamanho do problema e sim o tamanho do sistema. Se o hardware for dobrado, a tarefa deverá ser executada na metade do tempo. A aceleração linear é medida pelo ganho que é definido como a razão entre o tempo de execução com um processador e o tempo de execução com n processadores;
2. Crescimento linear (*linear scaleup*) - mede a habilidade de crescimento do sistema e do problema. Se o hardware for duplicado, o sistema será capaz de executar uma tarefa duas vezes maior no mesmo espaço de tempo.

3.2.2. Arquitetura de Máquinas Paralelas

Stonebraker [8] sugere taxonomia para o projeto de arquitetura de máquinas paralelas descrita a seguir.

1. Memória Compartilhada (*shared memory*): neste tipo de arquitetura, todos os processadores tem acesso a qualquer módulo de memória ou disco através de uma rápida rede de interconexão.
2. Disco Compartilhado (*shared disk*): no modelo de arquitetura de disco compartilhado, cada processador possui uma memória exclusiva e tem acesso a qualquer unidade de disco através da rede de interconexão.
3. Memória distribuída (*shared nothing*): aa arquitetura de memória distribuída, cada processador tem exclusivo acesso a sua memória e a(s) sua(s) unidade(s) de disco; cada nó (processador + memória + disco) pode ser visto como um SGBD local da mesma forma que os bancos de dados distribuídos.

3.2.3. Alocação dos Dados

A distribuição dos dados em SPBDs pode ser feita de duas formas: agrupada ou desagregada. No primeiro caso, cada relação ou classe é alocada como um todo a um dos nós do sistema, enquanto que no segundo, é feito um projeto de distribuição que envolve duas fases: Fragmentação e Alocação. A alocação é o processo de distribuição dos fragmentos pelos nós/processadores do sistema. Ela pode ser realizada com ou sem replicação dos fragmentos.

1. Fragmentação Horizontal: pode ser feita de forma primária ou derivada. Neste caso, a fragmentação de uma classe ou relação não é baseada nas propriedades dos seus próprios atributos mas em função da fragmentação de outra classe ou relação. Ela é usada para facilitar as operações de junção e navegação entre fragmentos. Já a fragmentação primária, utiliza três técnicas para a geração de fragmentos: faixa de valores, "hashing" e circular.
2. Fragmentação Vertical: é o processo que divide a relação/classe, agrupando em cada fragmento uma parte dos atributos existentes na relação/classe original e replicando-se em cada fragmento a chave primária/identificação do objeto da relação/classe de forma a permitir a sua reconstrução a partir da junção dos fragmentos. O seu objetivo é identificar fragmentos de tal forma que várias aplicações possam ser executadas com acesso à somente uma partição da relação/classe global. Entretanto caso uma aplicação necessite de acesso à dados presentes

em mais de um fragmento, a fragmentação vertical não é benéfica visto que será necessária a realização de junções entre os fragmentos para a obtenção do resultado.

3.2.4. Consultas

Um dos maiores desafios em SPBD é a execução das consultas. Existem três níveis de paralelismo que podem ser obtidos no processamento de consultas em SPBDs: (i) paralelismo entre consultas (*interquery*); (ii) dentro da consulta (*intraquery, interoperator, pipelined*) e; (iii) dentro da operação (*intraoperator, partitioned*); descritas como segue:.

1. paralelismo entre consultas: caracteriza-se pela execução concorrente de múltiplas consultas de diversos usuários;
2. dentro da consulta: consiste da execução paralela de diferentes operadores dentro de uma única consulta. Se um operador envia o resultado obtido para o seguinte, ambos podem executar em paralelo, diminuindo o tempo de resposta da consulta;
3. dentro da consulta: é alcançado através da distribuição dos dados de entrada pelos diversos nós da máquina paralela e da replicação da operação que é executada simultaneamente sobre os fragmentos. Nesse nível de paralelismo encontram-se os principais algoritmos de consultas em Banco de Dados.

Os benefícios a serem obtidos com o paralelismo dentro da consulta é limitado por três fatores [2]: (i) o encadeamento entre operações geralmente não é muito longo; (ii) alguns operadores relacionais não emitem sua saída até que tenham processado toda a sua entrada; e (iii) muitas vezes o custo de execução de um operador é muito maior que o de outros.

3.3. Clusters e GRID em Sistemas de Banco de Dados

Os conceitos de GRID e *cluster* muitas vezes se confundem. No entanto, existem algumas diferenças básicas entre eles, podendo-se afirmar que *cluster* é uma tecnologia utilizada para criar uma arquitetura de GRID. *Clusters* possuem recursos estáticos utilizados por aplicativos previamente definidos por usuários, ou Administradores de BD. Em contrapartida, GRIDS podem ser constituídos por vários *clusters*, em que os recursos utilizados, configurações podem ser compartilhados no processamento de muitos aplicativos.

A tecnologia de *cluster* [6] é usada em ambientes de SGBDs como uma tecnologia de suporte. O ambiente *clusterMining*[1] permite o desenvolvimento de novas aplicações em diversas áreas estratégicas, resultando em novas tecnologias de bancos de dados e extração de conhecimento em grandes massas de dados em arquiteturas paralelas. O ambiente permite que novas técnicas, metodologias e métodos computacionais sejam usados para o aumento de desempenho de aplicações que consultam e manipulam intensivamente grandes coleções de dados multidimensionais em ambientes distribuídos e de alto desempenho.

O ParGRES [1, 3] é um projeto para processamento de consultas paralelas no SGBD PostgreSQL. O principal objetivo do projeto é desenvolver um software de código aberto para processar consultas sobre grandes bases de dados de forma eficiente. O projeto

[1] http://clusterminer.nacad.ufrj.br/

é implementado em uma arquitetura que envolve *clusters* de PC e explora técnicas de paralelismo para obter eficiência.

O consórcio GSA-DAI e OGSA-DQP, liderado pelo Norman Paton[2], da University of Manchester na Inglaterra, visa o desenvolvimento de ambientes computacionais com acesso e integração de dados via GRIDs. O *framework* baseado em DQP (Distributed Querying Processing) consiste nos serviços descritos a seguir.

1. *Grid Distributed Query Service (Coordinator) - GDQS*: é o principal ponto de interação com os usuários. Quando um GDQS é setado, obtem-se os metadados e informações computacionais que são necessárias para compilar, otimizar, particionar e agendar planos de execução de consultas distribuídas sobre múltiplos nodos no GRID.
2. *Query Evaluation Service (Evaluator) - QES*: é usado pelo GDQS para executar os planos de consultas geradas pelo compilador, otimizador e agendador. Cada QES avalia uma partição do plano de execução de consulta assinalado pelo GDQS. Um conjunto de avaliadores que participam de uma consulta forma uma árvore na qual os dados dos avaliadores folhas interagem com os serviços de GRID.

O Oracle 10g é, segundo o *marketing* da empresa, o primeiro Sistema Gerenciado de Banco de Dados no mercado a ser totalmente desenvolvido para a *Grid Computing*. Ramalho [7] afirma que o SGBD permite o balanceamento de carga de processamento entre os diversos servidores que compõem o *pool* de equipamentos formados pelo GRID de um sistema, possibilitando melhor versão veio voltada para o gerenciamento e escalabilidade com amplo suporte a ambientes de redes ou clusters de bancos de dados distribuídos e inter-conectados. O SGBD promete oferecer, entre outras características:

1. beneficiar-se dos componentes disponíveis para o Grid através do Oracle Real Application Clusters, Automatic Storage Management (ASM) e Ultra-Large Data Files;
2. balanceamento de carga por meio de tecnologias do Oracle Real Application Clusters, Oracle Streams e dos bancos de dados distribuídos da Oracle suportam esses conceitos;
3. Compartilhar informações independentemente da localização através dos novos recursos de espaço de tabelas transportáveis, Oracle Streams (exemplo - Replication), tabelas externas e consultas SQL distribuídas;
4. Agendar recursos no Grid usando os novos recursos do Oracle Sheduler e do Oracle Database Resource Manager ajudam o DBA a efetivamente aproveitar melhor o GRID.

3.4. Conclusão

Aplicações que necessitam alto desempenham devem fazer uso de SGBDs com suporte a tecnologia paralela e distribuída. Deve-se considerar as técnicas básicas de paralelisma em sua implementação, as quais envolvem métricas e objetivos do paralelismo e arquitetura do hardware. Além disso, questões de alocação de dados e execução de consultas em ambientes distribuídos são extremamente importantes de serem observadas no desenvolvimento e administração de SGBDs que provêm suporte a aplicações de alto desempenho.

[2]http://www.cs.man.ac.uk/ norm/

O conceito de *Clusters* é usado como uma tecnologia de suporte a distribuição de diversos servidores de banco de dados em uma arquitetura distribuída [5]. A tecnologia de GRID aplicada a SGBDs é ainda bastante nova e envolve diversas questões em aberto, sendo mais aplicada a *Data Grids*, a exemplo do consórcio GSA-DAI e OGSA-DQP, do que *Database Management Grid*.

Referências

[1] Marta Mattoso (Coordenadora). Pargres - parallel query processing in the postgresql database system. (http://pargres.nacad.ufrj.br/), 2006.

[2] David DeWitt and Jim Gray. Parallel database systems: the future of high performance database systems. *Communications of the ACM*, 35(6):85–98, 1992.

[3] Marta Mattoso et al. Pargres: Middleware para processamento paralelo de consultas olap em clusters de banco de dados. In *Demos do Sinmpósio Brasileiro de Banco de Dados*, 2005.

[4] Database Group Leipzig. Parallel database systems. (http://dbs.uni-leipzig.de/en/projekte/pdbs/pdbs.html), 2006.

[5] Marta Mattoso. Desafios na gerência de dados distribuídos: Web, clusters e grid. In *SBBD*, page 2, 2005.

[6] Bernardo Miranda, Alexandre A. B. Lima, Patrick Valduriez, and Marta Mattoso. Apuama: Combining intra-query and inter-query parallelism in a database cluster. In *Current Trends in Database Technology (EDBT 2006)*, 2006.

[7] José Antônio Ramalho. *Oracle 10g*. Thomson, 2005.

[8] M. Stonebraker. The case for shared nothing. *Database Engineering*, 9(1), 1986.

[9] Patrick Valduriez. Parallel database systems: Open problems and new issues. *Journal of Distributed and Parallel Databases*, 1(2):137–165, April 1993.

[10] M. Tamer Özsu and Patrick Valduriez. *Principles of Distributed Database Systems, Second Edition*. Prentice Hall, 1999.