MANIPULAÇÃO DE EXPRESSÕES DE FRAGMENTAÇÃO EM SISTEMAS DISTRIBUIDOS COM UMA MAQUINA DE INFERENCIAS

Pedro Manoel Silveira

NCE 0387

Abril, 1987

Universidade Federal do Rio de Janeiro
Núcleo de Computação Eletrônica
Caixa Postal 2324
20001 - Rio de Janeiro, RJ
BRASIL

NCE UFRJ UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

MANIPULAÇÃO DE EXPRESSOES DE FRAGMENTAÇÃO EM SISTEMAS DISTRIBUIDOS COM UMA MAQUINA DE INFERENCIAS

Pedro Manoel Silveira

Sumário

Num sistema de bancos de dados distribuídos, a definição e alocação de fragmentos de relações pode apresentar características logicamente complexas, trazendo dificuldades ao processo de escolha das estratégias de acesso aos dados distribuídos.

O presente artigo sugere uma solução para este problema baseada no uso de uma máquina de inferências. O objetivo é apresentar um método geral que permita a manipulação lógica das várias expressões envolvidas, tais como as de fragmentação, de integridade dos dados e das consultas a serem processadas.

Abstract

In a distributed database, the definition and alocation of fragments of global relations may present logically complex characteristics, causing difficulties in the selection of access strategies to the distributed data.

This paper suggests a solution to this problem based on the use of an inference machine. The aim is to have a method general enough to permit the manipulation of the several expressions envolved, such as fragmentation, integrity and queries.

1. Introdução

Num sistema de banco de dados distribuídos, a definição e alocação de fragmentos de relações pode apresentar características complexas [1]. Isto leva a uma formulação logicamente elaborada das leis que regem a fragmentação e, como consequência, acrescenta dificuldades à elaboração de estratégias de acesso às relações globais de bancos de dados distribuídos.

O presente artigo apresenta uma visão desse problema e sugere uma solução baseada no uso de métodos de manipulação de expressões lógicas [2]. Basicamente, essa solução consiste do emprego de uma máquina de inferências genérica que, através da manipulação lógica das expressões de fragmentação juntamente com as expressões das consultas, oferece alternativas simplificadas para a escolha de estratégias de acesso a dados distribuídos.

Em virtude da complexidade e extensão do assunto, este artigo apresenta-se numa versão necessariamente impressionista, na medida em que detalhes do formalismo teórico não são aqui apresentados. Ao longo do texto, entretanto, há várias referências a trabalhos fundamentando os argumentos aqui apresentados.

Na Seção 2 há um exemplo simples que caracteriza a classe de problemas aqui abordados, enquanto que na Seção 3 outras linhas de solução são ventiladas. Na Seção 4 o método aqui sugerido é aplicado ao exemplo, de modo a ilustrar seu alcance. A Seção 5 considera aspectos da aplicação a sistemas distribuídos em geral e a Seção 6 conclui o artigo.

2. Motivação

O exemplo a seguir mostra de maneira simplificada o perfil dos problemas acima mencionados. Suponha um banco de dados distribuídos onde existe uma relação global R que, logicamente, consiste da união dos fragmentos R1 e R2.

$$R = R1 \cup R2 \qquad (1)$$

Suponha que o critério de pertinência para valores de R1 e R2 seja

$$x \in R1 \rightarrow x >= 5 \qquad (2)$$

$$x \in R2 \rightarrow x > 5 \qquad (3)$$

ou seja, os valores em R1 são iguais ou maiores do que 5 e os valores em R2 são menores do que 5. As fórmulas (1) e (2) acima constituem importante informação semântica a respeito dos dados e podem, eventualmente, ser utilizadas para a escolha de estratégias de acesso. Suponha agora que a seguinte consulta deva ser processada

$$x \mid x \in R \ \& \ x < 5$$

Ou seja, os valores de x que pertencem a R e são menores do que 5. É fácil perceber que, apesar da consulta referir-se à relação global R como um todo, é necessário apenas que se lide com o fragmento R2 da mesma, uma vez que não há valores de x menores do que 5 em R1.

O método para escolha da estratégia de processamento deve garantir que referências a R1 sejam eliminadas na formulação final da consulta e que restem apenas as referências à R2. Deste modo, para a consulta acima, a formulação

$$x \quad x \in R2$$

produziria resultados equivalentes e simplificaria a estratégia de acesso ao banco de dados distribuídos.

3. Enfoque

A idéia aqui é usar técnicas de provadores de teoremas para a manipulação lógica das expressões. Essa idéia não é nova, pois já existe uma quantidade significativa de trabalhos explorando a associação de bancos de dados com técnicas dedutivas. A aplicação para sistemas distribuídos, entretanto, não foi suficientemente explorada.

Em [3], Ceri e Pelagatti apresentam o embasamento teórico para técnicas cujo objetivo é mostrar a equivalência de duas expressões de uma consulta. Com isso, é possível levar a efeito um processo de manipulação de expressões com a obtenção de formula-

ções alternativas no caminho. Esta Algebra de Relações Qualificadas, como é chamada, entretanto, não configura um provador de teoremas e portanto, como o autor indica, carece de refinamentos para ser utilizada como um método geral.

Outros autores que sugerem o emprego de técnicas dedutivas para consultas são King [4] e Hammer [5], embora num escopo reduzido e sem mencionar diretamente a aplicação a sistemas distribuídos.

4. Método

A metodologia aqui apresentada parte da utilização de uma máquina de inferências, trabalhando com teoremas expressos em L+ [6]. Esta é uma linguagem cuja estrutura é baseada no Cálculo de Predicados, e que permite a expressão de consultas, leis de integridade dos dados distribuídos e expressões de fragmentação.

Para a consulta utilizada como exemplo acima, nós teríamos

```
x  WHERE  x IN R  WHEN  x < 5
```

e

```
R  IS  { x  WHERE  x IN R1 OR R2 }
```

como a expressão que define a relação global R. É necessário inicialmente exprimir a consulta em forma de cláusulas lógicas e para tal nós imaginamos o predicado Q como uma relação virtual que contém o resultado da consulta.

A substituição do símbolo R pela sua expressão na consulta original produz

```
x  WHERE  x IN R1 OR R2  WHEN  x < 5
```

e sua tradução para forma de cláusulas [2, 6] resulta em

```
Q(x)     x ∈ R1     x < 5                    (c1)
Q(x)  ←  x ∈ R2     x < 5                    (c2)
```

As duas cláusulas acima podem ser lidas como "se x pertence a R1

e x é menor do que 5, então x aparece na resposta da consulta" para a primeira cláusula, e numa maneira similar para a segunda cláusula.

Quantos às restrições dos valores de R1 e R2, teríamos as fórmulas

```
CONSTRAINT  FOR x IN R1:  x >= 5
CONSTRAINT  FOR x IN R2:  x < 5
```

Estas, em forma de cláusulas resultam em

$$x \in R1 \mid x >= 5 \qquad (c3)$$

$$x \in R2 \mid x < 5 \qquad (c4)$$

Com a máquina de inferências é possível então descobrir que a cláusula (c3) produz

$$x \in R1 \quad x < 5 \qquad (c5)$$

pela axiomatização da relação ">", e que (c5) é um subconjunto de (c1), porque seus literais são equivalentes ao segundo e terceiro literais de (c1). Logo, a cláusula (c1) pode ser removida sem alterar o resultado final. Além disso, das cláusulas (c4) e (c2) é possível deduzir

$$Q(x) \quad x \in R2$$

significando que a consulta pode agora ser reformulada como

```
x  WHERE  x IN R2
```

simplesmente.

Note-se que o exemplo aqui apresentado é extremamente simples. Num sistema onde as leis de integridade, as expressões de fragmentação e as consultas sejam complicadas e complexas, tal processo de otimização pode ser efetivamente benéfico, em virtude da provável incapacidade dos usuários humanos de lidarem eficientemente com tais expressões.

5. Aplicação

A aplicação de uma metodologia dessa natureza a sistemas distribuídos tem aspectos diversos. Supõe-se que a fragmentação dos arquivos seja claramente expressa em termos lógicos. Paralelamente, a estratégia de acesso é derivada do resultado do processo de inferência e, portanto, deve estar apoiada em alguma linguagem lógica qualquer.

A vantagem decisiva no emprego de uma máquina de inferências em tal contexto deriva da generalidade e abrangência do método. Além dos resultados aqui citados em relação à estratégia de acesso aos fragmentos, vários outros graus de otimização podem ser obtidos para consultas a bancos de dados distribuídos, todos sob uma mesma estrutura básica. Ou seja, vários processos de otimização são agrupados e levados a efeito num ambiente uniforme.

Um aspecto interessante a considerar é o do uso da linguagem L+, citada acima. A idéia é obter um contexto padronizado e de uma base lógica para a formulação de consultas. L+ cobre uma gama extensa de construções e é apenas uma sugestão para tal aplicação. A análise de alguns dos principais protótipos de bancos de dados distribuídos, como SIRIUS-DELTA [7], PROTEUS [8], R* [9] e outros [10], mostra que as arquiteturas preferidas para tais implementações sustentam-se na escolha de uma linguagem padrão para a expressão de consultas, algumas vezes chamadas linguagem-pivot. A escolha mais frequente recai sobre a Algebra Relacional e seus dialetos. A idéia é proporcionar um meio homogêneo de comunicação de consultas por entre os nós.

Esses fatos reforçam o uso de L+ e de nossa metodologia, pois embora a Algebra Relacional tenha aspectos procedurais, sua expressão através de lógica é de fácil conversão. Desse modo, é perfeitamente aceitável imaginar-se que a metodologia aqui apresentada possa ser aplicada a consultas e sub-consultas (resultados parciais) de uma maneira geral e uniforme por todos os nós componentes de um sistema distribuído, dada a predominância do uso e natureza de linguagens-pivot no projeto de bancos de dados distribuídos.

Outro aspecto importante a considerar refere-se à natureza exponencial dos processos de prova baseados no método da Resolução. Na metodologia aqui seguida, a máquina de inferências não

ficaria livre desses problemas e isso poderia comprometer o desempenho do processo.

A contra-argumentação neste caso baseia-se no objetivo do processo de otimização de consultas em ambientes de dados distribuídos. O uso da palavra otimização aqui é excessivamente forte, uma vez que a maior parte dos processos de escolha de estratégias de acesso para recuperação de dados fica melhor caracterizada como processos de melhoria [11]. Isto não é surpresa se considerarmos que, em sua maioria, esses processos baseiam-se em regras de reescrita e manipulação de expressões lógicas. Por outro lado, a quantidade de informações manipuladas, isto é, consultas, expressões de fragmentação e expressões de integridade normalmente é muito menor do que o volume de dados armazenados e, embora de natureza exponencial, os processos de manipulação lógica podem ser mantidos em ordens de grandeza inferiores aos processos de recuperação de dados. Isto pode dar-se seja através do uso de heurísticas, ou pelo puro e simples controle do processo, de modo a limitá-lo quando necessário. Desse modo, é razoável que se tente obter melhorias numa estratégia de acesso, mesmo com o risco de benefício zero.

6. Conclusões

Nas seções acima nós apresentamos sugestões para o tratamento de problemas advindos da escolha de estratégias de acesso a dados distribuídos, onde as expressões lógicas de fragmentação não são triviais. Tal solução baseia-se no uso de técnicas de prova de teoremas e no emprego de uma máquina de inferências. Através de um exemplo simples, nós ilustramos tal metodologia, de modo a demonstrar sua natureza.

Sistemas Distribuídos carecem de metodologias mais formais nesse aspecto. Alguns métodos existentes, embora incompletos ou inconclusivos, permanecem largamente desconhecidos, e é interessante elaborar idéias neste área.

Uma das principais vantagens do uso de metodologias baseada em provadores de teoremas é que outros métodos de otimização, ou melhoria, usados para bancos de dados distribuídos, podem ser naturalmente incorporados ao processo de uma maneira uniforme.

7. Referências

1. Ceri, S. & Pelagatti, G. Distributed Databases: Principles & Systems, McGraw-Hill, 1985

2. Nilsson, N. J. Problem Solving Methods in Artificial Intelligence, McGraw-Hill, 1971

3. Ceri, S. & Pelagatti, G. Correctness of Query Execution Strategies in Distributed Databases ACM Transactions on Database Systems Vol 8 Num 4, 1983

4. King, J. QUIST: A System for Semantic Query Optimization Proc. of the 7th Int. Conf. VLDB, 1981

5. Hammer, M. & Zdonick-Jr, S. B. Knowledge-Base Query Processing Proc. of the 6th Int. Conf. VLDB, 1980

6. Silveira, P. M. Database Design and Query Reformulation With an Inference Machine, PhD Thesis, U. of Kent, Inglaterra, 1985

7. Litwin, W. et al SIRIUS System for Distributed Data Management em Distributed Databases, H. J. Schneider (ed), North-Holland, 1982

8. Atkinson, M. P. et al The PROTEUS Distributed Database System Proc. of the 3rd British National Conf. on Databases, 1984

9. Williams, R. et al R*: An Overview of the Architecture Proc. Int. Conf. on Databases, Jerusalem, 1982

10. Silveira, P. M. Aspectos de Bancos de Dados Distribuídos, Monografia para V JAI, Congresso SBC, 1986

11. Ullman, J. D. Principles of Database Systems, Computer Science Press, 1980