

Projeto de Bancos de Dados Distribuídos (Parte 02)

IN1128/IF694 – Bancos de Dados Distribuídos e Móveis
 Ana Carolina Salgado – acs@cin.ufpe.br
 Bernadette Farias Lóscio – bfl@cin.ufpe.br



Cin.ufpe.br

Fragmentação Vertical

- Processo que **divide uma relação** em colunas, agrupando em cada fragmento **parte dos atributos** existentes na relação original
- **Replicação da chave** da tabela original nos fragmentos gerados
 - Fato que permite a reconstrução da relação global



Objetivo

- Particionar uma relação em um conjunto de relações menores, para que **aplicativos do usuário possam atuar apenas sobre um fragmento**
- Caso uma aplicação precise de dados presentes em mais de um fragmento: **fragmentação não benéfica** (necessidade de junções, operação muito custosa)



Complexidade

- Processo **inerentemente mais complexo** que a fragmentação horizontal;
 - Fragmentação horizontal:
 - Se temos n predicados
 - Teremos 2^n predicados minterm possíveis
 - E destes tiramos ainda os minterm que não fazem sentido (contraditórios)



Complexidade

- **Fragmentação vertical**: A quantidade de fragmentos possíveis é dada por $B(m)$, onde m são os atributos não chave e B é o número de Bell $B(m) \approx M^m$
 - Ex.:
 - $B(4) = 15$
 - $B(10) \approx 115.000$
 - $B(15) \approx 10^9$
 - $B(30) \approx 10^{23}$
- Esses valores indicam que **não vale a pena tentar obter soluções ótimas**, sendo necessário recorrer à abordagem heurística



Abordagens Heurísticas

- Há dois tipos de abordagens heurísticas pra fragmentação vertical:
 - **Agrupamento**: Começa atribuindo cada atributo a um fragmento e vai juntando outros atributos até achar os que satisfazem aos requisitos da aplicação
 - **Divisão***: Começa com uma relação e define como particionar de acordo com o comportamento de acesso aos atributos
- *se ajusta de forma mais natural dentro da metodologia de projeto top-down



Fragmentação vertical

- **Por que replicar a chave primária nos fragmentos?**
 - Permitir junções para reconstrução da relação original
 - Garantir a integridade semântica
 - Garantir a dependência funcional entre os atributos da relação particionada
 - Diminuir controle de concorrência nas atualizações
- **Uma alternativa para não replicação da chave primária seria o uso de TID (Identificadores de tupla) gerados pelo sistema**



Fragmentação vertical

- **Como a fragmentação vertical insere os atributos, que em geral são acessados em conjunto, em um único fragmento, surge a necessidade de algumas medidas que definam com maior precisão a noção de conjunto!**
- **Como identificar o conjunto de atributos que deve compor um fragmento vertical?**



Processo de Fragmentação Vertical

- **Afinidade de atributos:** indica a proximidade dos relacionamentos entre atributos
 - Como obter esse valor a partir dos dados do BD?
- **Para isso, deve-se saber o que vem a ser “valor de uso do atributo”:**
 - Seja $Q = \{q_1, q_2, \dots, q_m\}$ consultas do usuário na relação $R(A_1, A_2, \dots, A_n)$
 - Para cada consulta q_i e cada atributo A_j , associamos um valor de uso do atributo, denotado por $\text{uso}(q_i, A_j)$, definido como se segue:
 - $\text{uso}(q_i, A_j) = 1$, se o atributo A_j é referenciado pela consulta q_i
 - $\text{uso}(q_i, A_j) = 0$, caso contrário



Processo de Fragmentação Vertical

- **Os vetores de uso são definidos com facilidade se o projetista conhece os aplicativos que serão executados sobre o BD**
- **Regra 80/20**
 - 20% das consultas de usuários mais ativos respondem a 80% do total de acesso aos dados



Processo de Fragmentação Vertical

- **Relembrando a relação Projeto**

Projeto

PNO	PNome	Orçamento	Loc
P01	BI	150000	Montreal
P02	Des. de BD	135000	Nova York
P03	CAD/CAM	250000	Nova York
P04	Manutenção	310000	Paris



Processo de Fragmentação Vertical

- **Exemplificando:**
 - q1:** encontrar orçamento de um projeto pelo valor do número do projeto
“SELECT orcamen FROM proj WHERE pno = valor”
 - q2:** encontrar nomes e orçamentos dos projetos
“SELECT nome, orcamen FROM proj”
 - q3:** encontrar nomes dos projetos em uma dada cidade
“SELECT nome FROM proj WHERE loc = valor”
 - q4:** encontrar orçamentos dos projetos em uma dada cidade
“SELECT orcamen FROM proj WHERE loc = valor”



Processo de Fragmentação Vertical

- Exemplificando:

A1: pno A2: nome A3: orcamen A4: loc
- Esses valores ainda não são suficientes pra formar a base de divisão e fragmentação de atributos
- Esses valores não representam o peso das frequências dos aplicativos
- A medida de frequência pode ser incluída na definição da medida de afinidade de atributos $aff(A_i, A_j)$, que mede a ligação entre dois atributos de uma relação de acordo como eles são acessados pelos aplicativos



Processo de Fragmentação Vertical

- Matriz de uso de atributos:

	A1	A2	A3	A4
q1	1	0	1	0
q2	0	1	1	0
q3	0	1	0	1
q4	0	0	1	1



Processo de Fragmentação Vertical

- Deve-se saber o peso das frequências dos aplicativos. Para isso o conceito de **Afinidade entre Atributos** é definido
- A medida de afinidade de atributos entre dois tributos A_i e A_j de uma relação $R(A_1, A_2, \dots, A_n)$ com respeito ao conj. De consultas(aplicativos) $Q = \{q_1, q_2, \dots, q_m\}$ é definida em função de:
 - $ref(q_k)$: número de acessos dos atributos (A_i, A_j) a cada execução de q_k no site S_i
 - $acc(q_k)$: medida de frequência de acesso do aplicativo/ consulta (incluindo frequências em diferentes sites)



Processo de Fragmentação Vertical

- Supondo $ref_i(q_k) = 1$ pra todo k e pra todo i e as frequências:
 - $acc_1(q_1) = 15$
 - $acc_2(q_1) = 20$
 - $acc_3(q_1) = 10$
 - $acc_1(q_2) = 5$
 - $acc_2(q_2) = 0$
 - $acc_3(q_2) = 0$
 - $acc_1(q_3) = 25$
 - $acc_2(q_3) = 25$
 - $acc_3(q_3) = 25$
 - $acc_1(q_4) = 3$
 - $acc_2(q_4) = 0$
 - $acc_3(q_4) = 0$



Processo de Fragmentação Vertical

Assim, a afinidade entre A_1 e A_3 , por exemplo é dada por:

Matriz de Afinidade dos atributos:

	A ₁	A ₂	A ₃	A ₄
A ₁	-	0	45	0
A ₂	0	-	5	75
A ₃	45	5	-	3
A ₄	0	75	3	-

- Esta matriz é usada como base para o agrupamento em **clusters** de atributos com maior afinidade para posterior fragmentação



Processo de Fragmentação Vertical

- O processo de fragmentação envolve
 - primeiro o agrupamento em **clusters** dos atributos que têm uma grande afinidade entre si
 - seguido pela **divisão da relação** de forma adequada



Processo de Fragmentação Vertical

- Algoritmos de agrupamento em clusters
 - Tarefa fundamental da Fragmentação Vertical: agrupar atributos de uma relação em grupos
 - Agrupar os atributos com base na matriz de afinidades
 - Algoritmo de Energia de Ligação



Processo de Fragmentação Vertical

- Algoritmo de Energia de Ligação foi considerado para esse propósito:
 1. Atributos de mais afinidades em um *cluster*, atributos de menos afinidades em outro;
 2. Os agrupamentos não dependem da ordem que os itens são apresentados ao algoritmo;
 3. Complexidade de $O(n^2)$, sendo n o número de atributos;
 4. Os inter-relacionamentos secundários entre grupos de atributos reunidos em *clusters* são identificáveis.



Processo de Fragmentação Vertical

- O algoritmo de energia de ligação aceita como entrada a **matriz de afinidade** de atributos, **permuta suas linhas e colunas**, gerando uma matriz de afinidade agrupada em *clusters*
- A permutação entre linhas e colunas é feita de modo a maximizar a medida de afinidade global
- A **medida de afinidade global** é dada em função das medidas de afinidade entre cada par de atributos



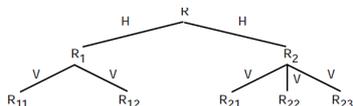
Processo de Fragmentação Vertical

- O objetivo da atividade de divisão é encontrar **conjuntos de atributos que são acessados unicamente**, ou em sua maioria, por conjuntos de aplicativos distintos
- Se for possível identificar dois atributos A_1 e A_2 , acessados apenas pelo aplicativo q_1 , e os atributos A_3 e A_4 acessados por dois aplicativos q_2 e q_3 , seria simples decidir sobre os fragmentos
- A **tarefa de divisão consiste em encontrar um método de identificação desses grupos!**



Fragmentação híbrida

- Consiste em realizar uma **fragmentação vertical seguida de uma fragmentação horizontal** ou vice-versa



Hybrid Fragmentation

- um conjunto de fragmentos horizontais, onde cada um deles é particionado em dois fragmentos verticais



Fragmentação híbrida

- O número de níveis de aninhamento pode ser grande, mas certamente é finito
 - No caso da **fragmentação horizontal**, deve-se parar quando cada fragmento consistir em **uma única tupla**
 - O ponto de término de uma **fragmentação vertical** é atingido quando temos **um único atributo** por fragmento



Fragmentação híbrida

- Na prática não é possível executar muitas fragmentações verticais antes que o **custo das junções** se torne muito alto, uma vez que as relações globais normalizadas já possuem graus pequenos
- As **regras de correção** e as condições para a **fragmentação híbrida** decorrem naturalmente das regras para as fragmentações horizontal e vertical



O Problema da alocação

- Descrição do problema
- Dados
 - $F = \{F_1, F_2, \dots, F_n\}$ fragmentos
 - $S = \{S_1, S_2, \dots, S_m\}$ nós da rede
 - $Q = \{q_1, q_2, \dots, q_q\}$ aplicativos
- Encontre a distribuição "ótima" de F em S



O Problema da alocação

- O caráter ótimo pode ser definido com relação a duas medidas:
- **Custo mínimo**
 - **Comunicação + armazenamento** de cada F_i em um site S_j + custo de **consultar** F_i em S_j + custo de **atualização** de F_i em todos os sites onde ele está armazenado
- **Desempenho**
 - A estratégia de alocação é projetada para manter uma métrica de desempenho. Duas estratégias bem conhecidas são **minimizar o tempo de resposta** e maximizar o **throughput** em cada site



Problema de alocação – requisitos de informação

- Informações do **banco de dados**
 - **Seletividade** dos fragmentos (quant. de tuplas que precisam ser processadas para responder uma dada consulta)
 - **Tamanho** dos fragmentos
- Informações de **aplicativos**
 - **Tipos e números dos acessos** (leitura e atualização)
 - **Localidade** dos acessos
 - O **tempo de resposta máximo** permitido em cada aplicativo



Problema de alocação – requisitos de informação

- Informações sobre os **sites**
 - Para cada site é preciso conhecer seu espaço de armazenamento e sua capacidade de processamento
 - **Custo unitário de armazenamento** de um dado em um site
 - **Custo unitário de processamento** em um site
- Informações da **rede**
 - Largura de banda
 - **Overhead** de comunicação



Modelo de alocação

- Forma Geral
 - $\text{Min}(\text{custoTotal})$
- Sujeito às restrições
 - De tempo de resposta
 - De armazenamento
 - De processamento



Modelo de alocação

- **A função de custo total tem dois componentes:**
 - O **custo de armazenamento**: simples de especificar, sendo dado pelo custo total de armazenamento em todos os sites para todos os fragmentos
 - O **custo de processamento de consultas**: mais difícil de especificar
 - Deve-se procurar por um esquema de alocação que, por exemplo, responda às consultas do usuário em tempo mínimo, enquanto mantém mínimo o custo de processamento.
 - Tais modelos são difíceis de serem desenvolvidos!



Modelo de alocação

- **Problema de alocação de arquivos (PAA) x Problema de alocação de bancos de dados (PABD)**
 - Fragmentos não são arquivos individuais
 - Relacionamentos têm que ser mantidos
 - Acesso aos BD é mais complicado
 - Modelo de acesso a arquivos remotos não é aplicável
 - Relacionamento entre alocação e processamento de consultas
 - **Custos adicionais** que devem ser considerados
 - Manutenção da integridade
 - Controle de concorrência



Conclusões

- Alocação e fragmentação são tratadas de forma independente pelos algoritmos
- Entretanto, ambas as etapas têm entradas semelhantes, diferindo no fato de que **a fragmentação atua sobre relações globais e a alocação leva em consideração os fragmentos**
- As duas etapas **ignoram o modo como cada uma faz uso das informações dos aplicativos do usuário**
- Uma solução seria **formular uma metodologia que refletisse de forma mais apropriada a interdependência das decisões de fragmentação e alocação!**



Conclusões

- As estratégias apresentadas de projeto de BD Distribuído **pressupõem um ambiente estático** em que o projeto é conduzido uma única vez e esse projeto pode persistir
- Na realidade podem ocorrer **mudanças físicas e lógicas**, que requerem um novo projeto do BD!
- **Novo projeto e materialização** são etapas necessárias para refletir as mudanças no ambiente!

