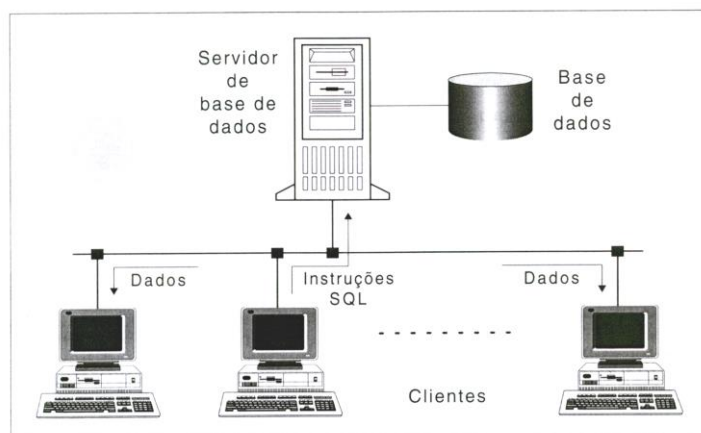


Arquitetura e Administração de Bases de Dados

João Pedro Costa
jcosta@isec.pt

Baseado nos slides da prof Fernanda Correia

Arquitetura Cliente/Servidor



Arquitetura cliente/servidor de bases de dados.

Arquitectura Cliente/Servidor

O processamento das aplicações pode distribuída pelos clientes e servidores. Parte do código pode residir e ser executado no servidor(*triggers, procedures,...*)

Vantagens:

- uma abordagem muito simples
- usa processamento dos servidores (mais potentes do que as máquinas clientes)
- diminui-se o tráfego na rede (código é executado na máquina que tem os dados)
- melhoria da fiabilidade das aplicações (há código partilhado por todas elas)

Desvantagens:

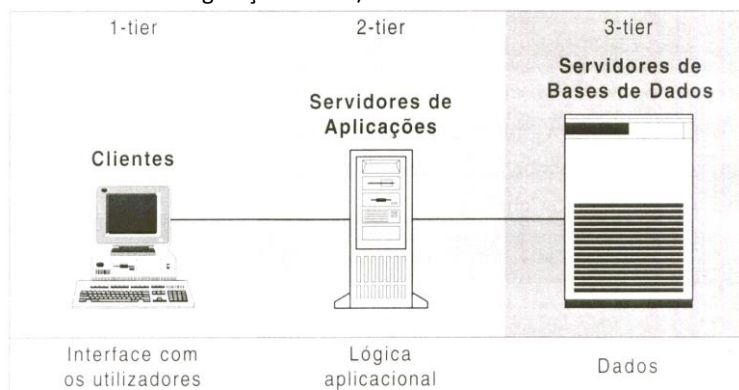
- a linguagem do código escrito é normalmente proprietário ou específico de cada SGBD, limitando a portabilidade
- O abuso na utilização deste tipo de facilidades pode sobrecarregar os servidores, significando, de certo modo, um retrocesso para o modelo centralizado em que tudo corre numa máquina

3

DEIS - AABD - 21/22
Baseado nos slides da prof. Fernanda Correia

Arquitetura Three-Tier

Configuração cliente/servidor em three-tier.



existe um servidor de aplicações entre os clientes (interface com o utilizador) e os SGBD (bases de dados) para suportar a lógica das aplicações

4

DEIS - AABD - 21/22
Baseado nos slides da prof. Fernanda Correia

Arquitetura Three-Tier

A divisão explícita em *three-tier* pode suportar adequadamente as necessidades de processamento de grande parte das organizações:

- Servidores de bases de dados suportam a gestão e armazenamento dos dados
- Servidores de aplicações armazenam e executam o código das diferentes aplicações
- Clientes correm a interface das várias aplicações.

Vantagens desta configuração:

- Maior desempenho e escalabilidade com a divisão da carga total de processamento
- Reutilização do código aplicacional pelos vários clientes em condições mais vantajosas que na solução *fat servers*.

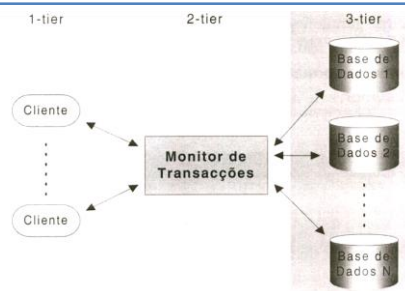
5

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

Arquitectura Three-Tier

- Código aplicacional desenvolvido sob a forma de serviços transacionais
 - Conjuntos de operações a realizar no contexto de uma transação.

No suporte da lógica aplicacional (servidores de aplicações), outra opção a considerar são os monitores de transações (*transaction processing monitors* ou *TP monitors*).



6

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

Arquitectura Three-Tier

- Os serviços transacionais são colocados no monitor de transações e invocados por qualquer cliente sempre que for necessário.
- Os monitores de transações são uma boa opção nos casos para aplicações que durante a mesma transação têm de aceder a várias bases de dados eventualmente em plataformas distintas
- Os monitores de transações permitem assegurar que:
 - ou todas as operações sobre cada uma das bases de dados envolvidas na transação têm sucesso
 - ou então nenhuma tem e por isso devem ser desfeitas

7

Arquitectura Three-Tier

O Monitor de transações permite:

- implementar o conceito de transação distribuída
- reduzir ao mínimo o n.º de conexões aos SGBD
- partilhar um conjunto de conexões a SGBD entre vários clientes
 - otimizando o seu aproveitamento
 - com um menor número de conexões permite atender mais clientes sem perdas de desempenho.

Alguns monitores de transações disponíveis no mercado:

- CICS e o Encina da IBM Corp.
- Tuxedo da BEA Systems Inc.
- Top End da NCR Corp.
- Microsoft Transaction Server (MTS) da Microsoft Corp.

monitores de transações apresentam preços elevados.
O MTS é mais acessível e fácil integração no ambiente Windows.

8

Bases de dados distribuídas

sistema de bases de dados cujos dados se encontram fisicamente dispersos por várias máquinas ligadas por meios de comunicação integrados logicamente

“é um conjunto de centros de computação, ligados por redes de computadores, em que cada centro constitui um sistema de bases de dados e um utilizador de um dos centros pode utilizar dados armazenados nos outros centros como se fossem locais.”

[Date 1995]

9

Bases de dados distribuídas

Classes - em termos de modelos de bases de dados

Homogéneas	Heterogéneas
<p>Todos os nós usam o mesmo SGBD</p> <p>Os dados estão armazenados em vários repositórios ligados por meios de comunicação</p>	<p>SGBDs diferentes nos vários nós.</p> <p>No entanto estes SGBDs possuem:</p> <ul style="list-style-type: none">- o mesmo modelo de dados, consistindo em implementações diferentes do modelo relacional (Ex: Oracle, SQLServer,...)- modelos de dados diferentes, de carácter semântico entre bases de dados cooperantes (o mesmo termo pode ter significados diferentes em bases de dados (locais) diferentes)

10

Bases de dados distribuídas

Homogéneas

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

11

Bases de dados distribuídas homogéneas

- Vamos considerar para já a subclasse das bases de dados distribuídas homogéneas baseadas no modelo relacional.
- Segundo Codd o modelo relacional constitui a melhor tecnologia existente para suportar a gestão de bases de dados distribuídas. Grande parte dos sistemas de bases de dados distribuídas existentes e em desenvolvimento são relacionais.

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

12

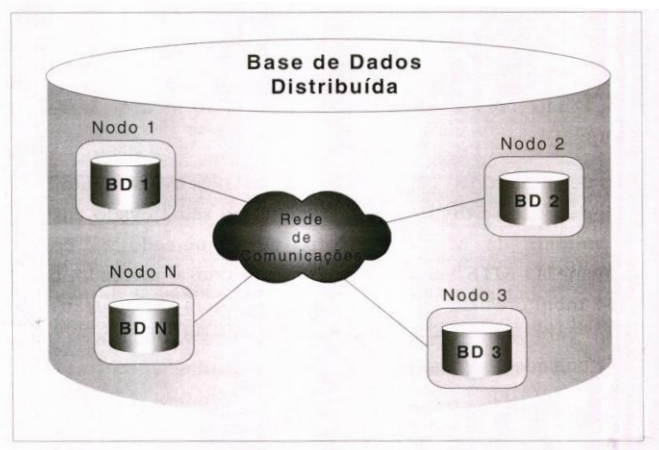
Sistemas Distribuídos

- Um modelo mais ambicioso —→ o que corresponde aos sistemas distribuídos de bases de dados.
- Consiste na distribuição dos dados por vários centros de processamento independentes e a partilha transparente desses dados entre os vários centros (designados por nós) usando meios de comunicação.
- Uma única base de dados lógica, distribuída fisicamente por vários nós que cooperam entre si partilhando dados e capacidade de processamento.

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

13

Sistemas Distribuídos



DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

14

Sistemas Distribuídos

Razões que sugerem a abordagem distribuída de uma base de dados:

- Refletir o estado distribuído de uma organização
 - Organizações em zonas geograficamente distantes, com necessidades de coordenação comum e estruturas organizacionais cada vez mais descentralizadas em termos de funções.
 - mesmo não estando dispersas geograficamente, possuem unidades departamentais autónomas dispondo das suas bases de dados com processamentos específicos
- Por necessidade de crescimento
 - a abordagem distribuída permite um crescimento quase ilimitado da organização, que se dá organizando novos nodos na rede

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

15

Sistemas Distribuídos

De que forma distribuir os dados pelos vários nós?


- Fragmentação
 - Como dividir?
 - Como distribuir?
- Replicação dos dados (ou duplicação)
 - Quando replicar?
 - Como replicar?
 - É necessário manter as cópias sempre sincronizadas?
 - É aceitável estarem temporariamente inconsistentes?
 - Se puderem estar temporariamente inconsistentes quanto tempo isso é tolerável?

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

16

Replicação dos dados

Uma alternativa para reduzir estes custos é:

- Replicar os dados mais frequentemente solicitados, pelos nós que mais os solicitam.  Maior probabilidade dos dados necessários estarem armazenados localmente



isto supondo que as operações de consulta predominam sobre as de atualização

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

17

Replicação dos dados

A prevenção de falhas também aconselha à replicação de dados nalguns nós



- Melhora a robustez do sistema
- Diminui a carga de comunicações
- Aumenta o seu desempenho geral (maior probabilidade do processamento ser local).

Necessário manter a sua coerência global.

As atualizações devem garantidamente ser propagadas para todos os nós onde os dados afetados por essas atualizações estão duplicados.

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

18

Replicação dos dados

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

- Manter as réplicas permanentemente sincronizadas traz dificuldades.
- garantir que cada atualização só entra em vigor quando todas as réplicas forem também atualizadas com sucesso.
- A finalização de uma transação distribuída, que envolve a atualização de dados, tem de atualizar um protocolo específico (o mais divulgado é o **two-phase commit**) que garanta que todas as réplicas ficam sincronizadas, caso contrário a transação deve falhar em todas elas, sendo feito o *rollback*.

19

Replicação dos dados

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

Fatores que num sistema distribuído podem colidir com a necessidade de manter todas as réplicas sincronizadas:

- Falhas de comunicação nos nós individuais.
- Interferência das transações locais com as transações distribuídas, pois concorrem no acesso aos dados locais.
- Maior quantidade de nodos, maior a probabilidade de ocorrerem problemas.

20

Replicação dos dados

Melhor abordagem nessas situações para minimizar o período de dessincronização entre as réplicas:

- Propagar as atualizações feitas num nó para as correspondentes réplicas logo que a transação termine nesse nó.
- Alternativamente, se esse período puder ser mais longo, uma abordagem mais económica é propagar não todas as atualizações individuais, mas sim o resultado final de várias atualizações.

21

Replicação dos dados

- Primary site replication.
 - Um nó (*primary site*) detém a posse dos dados e é o único que os atualiza (mais simples)
 - Propaga as atualizações para os restantes réplicas (*snapshots*)
- Dynamic ownership.
 - A autorização para atualizar dados replicados vai-se movendo entre os nós
 - Em cada momento apenas um dos nós possui autorização.

22

Replicação dos dados

- Replicação simétrica (aka Shared ownership ou Update anywhere)
 - permite que qualquer réplica seja atualizada em qualquer altura, eventualmente em simultâneo
 - o efeito dessas atualizações é propagada de forma sincronizada, ou não, para as restantes réplicas
 - mais complexo

Questão:

- conflito de atualização quando duas ou mais transações tentam atualizar os dados em duas ou mais réplicas diferentes ao mesmo tempo?

Forma de resolução destes conflitos:

- Recurso ao fator tempo.
- Prevalece a atualização mais recente.

23

Fragmentação dos dados

- Dada uma tabela global da base de dados global a fragmentação é a forma como essa tabela pode ser dividida em tabelas menores, estas últimas a distribuir pelos nós.



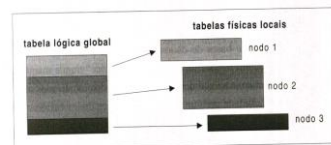
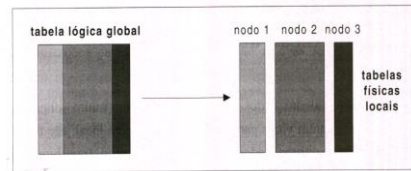
- A tabela global é uma tabela lógica que se obtém integrando os vários fragmentos (tabelas dispersa pelos nós).
- Essas tabelas são tabelas com existência física.

24

Fragmentação dos dados

A fragmentação pode assumir três formas:

- Fragmentação vertical
- Fragmentação mista
- Fragmentação horizontal



25

Caraterísticas de base de dados distribuída

- Transparência na localização dos dados relativamente ao nível aplicacional
- Um utilizador:
 - Não necessita ter conhecimento da localização dos dados
 - Não deve se aperceber de quais os nós cooperantes.
 - Não se deve aperceber que dados estão armazenados nesses nós
 - deve ter a impressão de que toda a base de dados está disponível localmente

O nível aplicacional deve aceder a dados localizados algures na rede com a mesma facilidade com que acederia, se esses dados residissem numa mesma máquina.

26

Caraterísticas de base de dados distribuída

- A tecnologia de bases de dados distribuída combina dois conceitos divergentes:
 - Distribuição física dos dados.
 - Integração lógica dos mesmos.

27

Caraterísticas de base de dados distribuída

A transparência na localização dos dados:

- Facilita o desenvolvimento e manutenção de sistemas
- Fornece aos utilizadores e a todo o nível aplicacional a facilidade de utilizar “pesquisas genéricas”.
- Deixa ao SGBD a tarefa de resolver os detalhes de navegação na rede
- Assim aplicações e utilizadores devem ter a impressão que toda a base de dados está disponível localmente.

28

Caraterísticas de base de dados distribuída

Princípio fundamental dos sistemas distribuídos

- um sistema de bases de dados distribuído deve comportar-se da mesma forma que um sistema de bases de dados centralizado.

[Date 1995] chama a este princípio **regra zero** dos sistemas distribuídos.

29

Caraterísticas de base de dados distribuída

Características desejáveis :

- **Disponibilidade permanente.**
Alterações na rede (adição, remoção ou falha dos nós) não deve impedir os outros nós de continuarem a operar total ou pelo menos parcialmente
- **Fragmentação transparente.**
uma tabela fragmentada por vários nós, essa fragmentação deve ser transparente aos utilizadores que deverão “ver” apenas uma tabela lógica
- **Duplicação transparente.**
Alguns dados por prevenção e/ou desempenho podem estar duplicados em vários nós.
Essa duplicação deve ser transparente aos utilizadores e o SGBD é único responsável por manter a sua coerência.

30

Caraterísticas de base de dados distribuída

Características desejáveis :

- **Processamento de pesquisas distribuído.**

O processamento de uma pesquisa pode necessitar da cooperação de vários nós e todo esse processo é coordenado pelo sistema sem intervenção dos utilizadores.

- **Independência do hardware, sistema operativo e tecnologias de rede.**

Não deverá ser obrigatório que o hardware, o sistema operativo ou a tecnologia de rede utilizada sejam iguais em todos os nodos.

Os SGBDs utilizados nos vários nós devem poder ser diferentes.

O utilizador deve poder aceder aos dados noutras máquinas usando o mesmo interface que usa nos dados locais.

31

Conceção de Bases de Dados Distribuídas

- Conceção de bases de dados distribuídas



forma como a base de dados global (esquema lógico) vai ser distribuído pelos vários nodos (esquema físico).



- Vai influenciar o comportamento do sistema global.

Localizando os dados nos nós adequados :

- Minimiza os custos de comunicação.
- Aumenta o desempenho e robustez do sistema.

32

Conceção de Bases de Dados Distribuídas

Duas abordagens :

- **Top down.**

Divisão da base de dados pré-existente, ou mais genericamente de um esquema de base de dados pré-definido, em várias partes, a armazenar em vários nodos.

Normalmente o resultado é um sistema distribuído homogéneo de bases de dados.

Apenas faz sentido em relação ao desenvolvimento de sistemas de bases de dados distribuídas, de raiz.

- **Bottom up.**

Integração (não desagregação) de várias bases de dados pré-existentes numa base de dados global, distribuída por várias máquinas.

É muito provável a heterogeneidade das bases de dados presentes e por isso esta abordagem oferece mais dificuldades.

Frequente aquando da integração de esquemas de bases de dados pré-existentes.

33

Conceção de Bases de Dados Distribuídas

- Projeto de uma base de dados distribuída
consiste em decidir como distribuir as tabelas que suportam o modelo conceptual de dados global pelos vários nodos da rede.

- A ideia geral é maximizar o processamento local

a regra a seguir na distribuição é que os dados devem ser colocados junto de quem mais precisa deles, de modo que o processamento seja feito localmente sempre que possível.

34

Conceção de Bases de Dados Distribuídas

- Nesta fase decide-se:
 - Quais as tabelas locais (as que não têm de residir por inteiro num só nodo) e a que nodos correspondem.
 - Quais as tabelas a fragmentar.
 - Como fragmentar as tabelas.
 - Onde serão armazenados os fragmentos.
 - Quais as tabelas ou fragmentos a duplicar e em que nodos se duplicarão.

35

Conceção de Bases de Dados Distribuídas

O problema de decidir a configuração ótima da localização dos dados é complexo



uma solução possível é a utilização de heurísticas que não garantem a solução ótima, mas resolvem o problema em tempo útil

Questão:

Em que fase do processo de desenvolvimento de um sistema de bases de dados distribuída surge o problema da distribuição?

36

Conceção de Bases de Dados Distribuídas

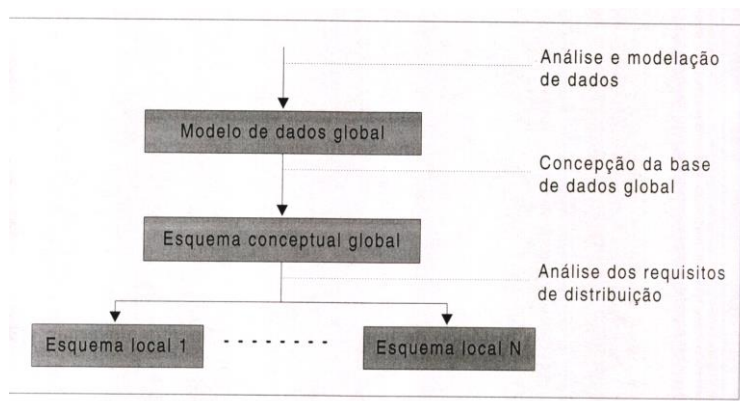
A conceção *top-down* de bases de dados distribuídas não é muito diferente da conceção de bases de dados centralizada:

1. executa-se a modelação conceptual e desenho de bases de dados de forma global independentemente das pesquisas de distribuição
2. consideram-se aspetos relacionados com a distribuição dos dados pelos vários nodos dando origem aos esquemas conceptuais locais

DEIS - AABD - 21/22
Baseado nos slides da prof. Fernanda Correia

37

Conceção de Bases de Dados Distribuídas



DEIS - AABD - 21/22
Baseado nos slides da prof. Fernanda Correia

Concepção de uma base de dados distribuída homogênea.

38

Concepção de Bases de Dados Distribuídas

A separação das duas fases traz algumas consequências:

- Os critérios de distribuição que resultam de condicionalismos organizacionais:
 - Que processamentos são organizados em que nodos?
 - Quais os dados que necessitam.

aparecem desligados da fase de modelação conceptual.

- A fase de modelação conceptual deveria corresponder ao momento em que os condicionalismos e restrições existentes são expressas no modelo.

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

39

Concepção de Bases de Dados Distribuídas

A melhor configuração para a localização dos dados encontra-se com base na análise de:

- Quais são as transações mais frequentes
- Quais os dados a que acedem
- Em que nodos são originados

de modo a maximizar o processamento local.

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

40

Concepção de Bases de Dados Distribuídas

- Outra abordagem à distribuição dos dados pelos vários nós é aquela que preconiza a sua contínua redistribuição →
 - é muito provável e natural que as condições de utilização do sistema se alterem ao longo do tempo e poderá ser necessário alterar a distribuição para uma mais adequada.
- a localização dos dados não deve ser estática
 - Por exemplo: com base num histórico de transações os fragmentos deveriam ser transferidos pelo sistema dinamicamente, entre os nodos, de forma a manter a distribuição mais adequada.

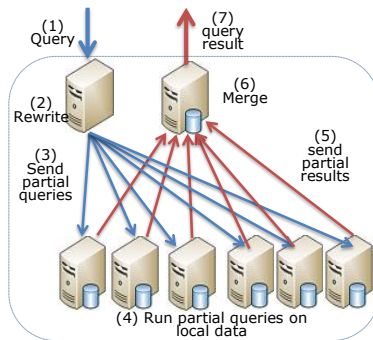
41

Processamento e Otimização de Pesquisas

- Num sistema de bases distribuído a otimização de pesquisas é mais complexo que um sistema centralizado.
- o processo de otimização de pesquisas é distinto pois :
 - utiliza meios de comunicação
 - Utiliza o paralelismo inerente aos sistemas distribuídos

42

Processamento e Otimização de Pesquisas



Execution time influenced by

- Query selectivity
- Data volume in each node
- Execution plan
- Join algorithm (local vs in-network)
- Join order
- Nodes' characteristics

Constrained by the slowest node

Execution time

$$t_{execTime} = t_{rewrite}(Q) + t_{send}(Q) + \max(t_{Nexec}(Q) + t_{Nsend}(Q_{pR}) + t_{receive}(Q_{pR})) + t_{merge}(\sum Q_{pR}) + t_{send}(Q_{fR})$$

BigData - João Costa

43

Processamento e Otimização de Pesquisas

o objetivo da otimização de pesquisas

- nos sistemas centralizados
 - é aumentar o desempenho.
- nos sistemas distribuídos
 - tentar minimizar em termos de recursos o tempo de processamento de pesquisas
 - minimizar o seu tempo de resposta (aqui explorar o paralelismo inerente à distribuição do processamento do sistema).

44

Processamento e Otimização de Pesquisas

- Sempre que possível o processamento deve ser feito localmente ao nodo onde os dados estão armazenados
 - “as pesquisas devem ser enviadas ao encontro dos dados”
 - e não “trazer os dados ao encontro das pesquisas”
- Se uma pesquisa não pode ser processada localmente é preciso dividir, da melhor forma, o seu processamento pelos nodos envolvidos.
- Otimização - seleccionar a estratégia mais eficiente de processar as pesquisas dada a localização dos dados envolvidos

45

Processamento e Otimização de Pesquisas

Processamento distribuído de uma pesquisa

- decomposição dessa pesquisa em subpesquisas
 - de acordo com a distribuição dos dados pretendidos pelos vários nodos
 - em cada nodo, o processamento de cada uma das subpesquisas é otimizada localmente de maneira idêntica aos sistemas centralizados
- Execução de uma pesquisa global
 - normalmente é um conjunto de execuções de subpesquisas locais
 - provavelmente executados em paralelo
 - cujos resultados intermédios são movimentados para os nós participantes
 - finalmente reunidos para constituir a resposta final.

46

Bases de Dados Distribuídas Heterogéneas

- Características de algumas organizações:
 - Sua distribuição geográfica
 - Heterogeneidade dos seus recursos informáticos.

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

47

Bases de Dados Distribuídas Heterogéneas

Motivos da heterogeneidade dos seus recursos informáticos:

- Departamentos diferentes utilizarem soluções diferentes
 - a nível de hardware, sistemas operativos, tecnologias de rede, etc..
- fusões e aquisições entre organizações
 - diferentes organizações tomaram opções tecnológicas diferentes
- Aplicações desenvolvidas para SGBDs específicos
 - adquiridas a empresas de software especializadas

DEIS – AABD – 21/22
Baseado nos slides da prof. Fernanda Correia

48

Bases de Dados Distribuídas Heterogéneas

Solução mais comum:

- Integração de bases de dados locais pré-existentes
 - concebidas sem quaisquer preocupações de integração entre si numa base de dados distribuída.
- Muitas vezes o desenvolvimento de sistemas de bases de dados distribuídas só é viável se conseguir incorporar bases de dados construídas segundo as tecnologias mais recentes com bases de dados vitais para o funcionamento das organizações.

49

Bases de Dados Distribuídas Heterogéneas

- Um dos princípios fundamentais dos sistemas de bases de dados distribuídas é a sua transparência relativamente ao nível aplicacional.
- Nos sistemas homogéneos o problema da transparência envolve fundamentalmente a transparência na localização dos dados, dada a semelhança dos SGBDs presentes.

Nos sistemas heterogéneos temos dois problemas:

- Transparência na localização dos dados.
- Transparência nos acessos. Um utilizador poder aceder aos dados globais com a mesma interface com que acede aos dados locais.

50

Bases de Dados Distribuídas Heterogéneas

Várias abordagens para a integração de bases de dados distintas numa base de dados distribuída heterogénea:

- Utilização de *database gateways/middleware*
 - Normalmente utilizada quando se encontram presentes SGBDs diferentes mas baseados no mesmo modelo de dados.
 - Ex: utilização de ODBC.
- Integração num modelo global
 - abordagem adequada quando os SGBDs presentes são baseados em modelos de dados diferentes.
 - é necessário utilizar um modelo global onde os modelos locais são mapeados.

51

Gestão de Bases de Dados Distribuídas

- Características a que deve satisfazer uma transação:
 - Atomicidade
 - Integridade
 - Isolamento
 - durabilidade

As transações num sistema de bases de dados distribuída são mais complexas e podem envolver a cooperação de vários nós. Nestes casos uma transação dá origem a um conjunto de subtransações (uma para cada nó implicado na transação global).

52

Gestão de Bases de Dados Distribuídas

- Para uma transação distribuída satisfazer as características anteriores é necessário haver a coordenação das várias subtransações locais que constituem a transação global distribuída.
- integridade e isolamento da transação global é assegurada por cada uma das transações locais. Isto acontece se cada uma das transações satisfazer o protocolo *two-phase locking* e as regras de integridade definidas para os nós locais.
- As subtransações da transação global necessitam de:
 - Ser sincronizadas com as transações locais a cada nó
 - Ser sincronizadas com as outras transações globais ativas no sistema.

53

Gestão de Bases de Dados Distribuídas

- O controlo de concorrência é mais complexo e os SGBDs cooperantes no caso de bases de dados distribuídas heterogéneas podem utilizar várias técnicas de controlo de concorrência.

54

Gestão de Bases de Dados Distribuídas

Relativamente à atomicidade e durabilidade das transações distribuídas

- Cada subtransação local deve ser atômica, isto é ou todas as suas operações sucedem ou nenhuma é executada.
- A transação global deve ser atômica, isto é ou todas as suas subtransações sucedem ou nenhuma delas é executada.
- Uma subtransação local não pode individualmente fazer o *commit* sem atender ao estado final de todas as outras subtransações que constituem a transação global distribuída.
- Cada subtransação local deve aguardar uma instrução de *commit* por parte de alguma entidade coordenadora da transação global antes de finalizar.

55

Gestão de Bases de Dados Distribuídas

protocolo two-phase commit (2PC).

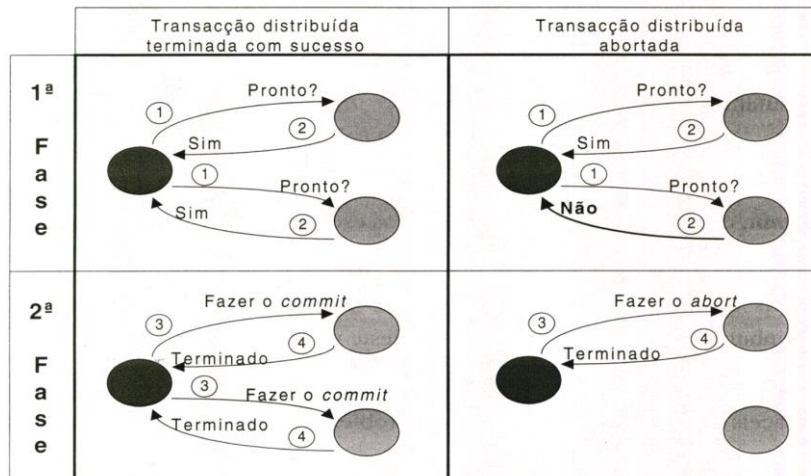
- a abordagem mais comum para resolver estes problemas
- a coerência da base de dados está sempre assegurada pois qualquer atualização num nodo só fica definitiva se a mesma for feita com sucesso em todos os nodos afetados por essa atualização (indivisibilidade da transação global).

No protocolo *two-phase commit* a tarefa da coordenação da finalização da transação distribuída é desempenhada por um dos nodos (normalmente o que originou a atualização)

56

Gestão de Bases de Dados Distribuídas

DEIS - AABD - 21/22
Baseado nos slides da prof. Fernanda Correia



O protocolo *two-phase commit*.

57

Gestão de Bases de Dados Distribuídas

1ª fase:

- O nó coordenador envia uma mensagem de preparação para o *commit*, a cada nó participante na transacção.
- A esta mensagem cada nó responde afirmativamente (se estiver pronto para fazer o *commit* da sua subtransacção local) ou negativamente (se a transacção local abortou e foi feito o *rollback*).
- A fase termina se todos os nós envolvidos responderem ao nó coordenador.

2ª fase:

- Se as respostas à primeira fase forem positivas, o nó coordenador envia uma instrução de *commit* a cada nó participante, e aguarda uma confirmação.
- Basta que um dos nós tenha respondido negativamente à primeira fase (ou não tenha respondido - *timeout*), o nó coordenador envia uma mensagem de *abort* a todos os nós participantes para estes fazerem o *rollback* das suas subtransacções locais.

DEIS - AABD - 21/22
Baseado nos slides da prof. Fernanda Correia

58

Gestão de Bases de Dados Distribuídas

- No protocolo **three-phase commit** (3PC)- existe um passo intermédio, chamado *pré-commit* que confirma que todos os nós estão em condições de finalizar
- O inconveniente deste protocolo é que aumenta o tráfego de mensagens de coordenação entre os nodos.

59

Gestão de Bases de Dados Distribuídas

O problema dos *deadlocks* agrava-se quanto à sua deteção e resolução, principalmente quando o *deadlock* envolve recursos a vários nós.

É designado por **deadlock distribuído**.

Possível solução :

- construir um grafo de espera global a todos os nós

Issues:

- não pode estar centralizado num dos nós, pois uma falha nesse nó compromete completamente a deteção de *deadlocks* no sistema
- origina elevado tráfego para manter o grafo

Solução de compromisso:

- Utilização de grafos de espera para a deteção de *deadlocks* locais.
- Mecanismos de *timeout* para a deteção de *deadlocks* distribuídos.

60

Vantagens e Limitações

Vantagens dos sistemas de bases de dados distribuídos:

- Pode refletir de forma transparente a estrutura da própria organização.
- Permite acompanhar a evolução de uma organização.
- Transporta uma maior resistência a falhas.
- O seu desempenho global pode ser superior ao da mesma versão centralizada.