Em Anais SBBD'93 - VIII Simpósio Brasileiro de Banco de Dados, 383–393. Maio 1993.

HEROS: Um Sistema de Banco de Dados Heterogêneos Orientado a Objetos

Carlos H. C. Duarte
Esther Pacitti
Rubens N. Melo
Sidney D. da Silva

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de S. Vicente, 225, Rio de Janeiro, RJ, 22453, Brazil

Resumo

Sistemas de bancos de dados heterogêneos têem sido apresentados como uma possível solução para problemas de manipulação e integração de dados distribuídos entre diferentes bancos de dados. Este trabalho apresenta as características do HEROS, um sistema de gerência de bancos de dados heterogêneos em desenvolvimento na PUC-Rio, com a descrição de sua arquitetura e funcionalidade.

Abstract

Heterogeneous database systems have been presented as a possible solution for problems in handling and integrating distributed data in diferent databases. This paper presents the characteristics of HEROS, a heterogeneus database management syste being developed at PUC-Rio, describing its architecture and functionality.

1 Introdução

Sistemas de bancos de dados têm proliferado no âmbito das organizações para satisfazer necessidades específicas. Uma empresa de grande porte pode ter vários sistemas de gerência de banco de dados (SGBDs), de fabricantes distintos e utilizados em uma diversidade de plataformas de hardware e software.

O avanço da tecnologia de comunicação tem possibilitado que uma mesma aplicação tenha acesso aos dados mantidos por esses vários SGBDs. A manipulação dos dados pela aplicação implica no conhecimento da localização física de cada um dos

bancos de dados nas redes de comunicação e dos caminhos de acesso disponíveis, no conhecimento detalhado dos respectivos esquemas e ainda no conhecimento das linguagens de manipulação de dados empregadas.

Os sistemas de banco de dados existentes foram criados independentemente e administrados por critérios próprios, sem a preocupação com uma possível integração futura. Essa situação é semelhante àquela verificada nos anos que antecederam ao surgimento dos SGBDs, caracterizada pela existência de sistemas de arquivos próprios e com reduzidas possibilidades de compartilhamento dos dados neles contidos.

Esses SGBDs são considerados heterogêneos por serem característicos de diferentes ambientes operacionais e pela possibilidade de cada um utilizar um diferente modelo de dados. Mesmo no caso de SGBDs que se utilizam de um mesmo modelo de dados, é comum que as linguagens de manipulação de dados difiram entre si, assim como os seus esquemas internos de controle de concorrência, de recuperação de falhas e de tratamento de impasses (*deadlocks*).

Sistemas de gerência de bancos de dados heterogêneos (SGBDHs) estão sendo propostos como um forma de se conseguir a cooperação entre múltiplos e pré-existentes SGBDs, viabilizando a integração dos dados contidos nos bancos por eles geridos. Um SGBDH é construído sobre o conjunto de SGBDs, criando para o usuário a ilusão de um banco de dados integrado e escondendo os detalhes relativos aos caminhos de acesso aos dados e das linguagens de manipulação particulares específicas dos SGBDs envolvidos [BGMS92].

Um pressuposto básico na construção de SGBDHs é o respeito à autonomia dos sistemas de banco de dados já existentes. Isso implica em se evitar soluções que violem, dentre outras, a liberdade que cada sistema possui quanto à execução de transações, à representação interna de dados e à associação [SL90].

Neste trabalho apresentamos o SGBDH em desenvolvimento no Departamento de Informática da PUC-Rio, denominado HEROS (Heterogeneous Object System), descrevendo sua arquitetura e esquema geral de funcionamento. O HEROS utiliza um modelo de dados orientado a objetos como modelo integrador, permitindo que, além dos SGBDs que se utilizam dos modelos de dados tradicionais, também novos gerenciadores com orientação para aplicações não convencionais sejam integrados.

O restante deste trabalho está organizado da seguinte maneira:

- A Seção 2 apresenta a descrição detalhada da arquitetura projetada para o HEROS e a dos seus componentes funcionais;
- A Seção 3 contém a descrição do esquema geral de gerência de transações de consulta e atualização;
- A Seção 4 contém observações sobre a implementação da versão inicial do HEROS, em andamento;

- Finalmente, trabalhos relacionados são brevemente referenciados — dando o contexto no qual o HEROS se insere — e são apresentadas nossas conclusões e trabalhos futuros.

2 A Arquitetura do HEROS

Nesta seção apresentamos a arquitetura do HEROS, descrevendo seus componentes e a funcionalidade que implementam. O HEROS é um SGBD que segue uma arquitetura de esquemas em quatro níveis (v. Figura 1). Seus componentes funcionais estão organizados em três camadas básicas, cada uma compreendendo um conjunto de módulos que implementam funções relacionadas entre si (v. Figura 2). Os níveis de esquema e os componentes dinâmicos são abordados a seguir.

2.1 Níveis de Esquema

A arquitetura de esquemas em três níveis ad ANSI/SPARC [Dat93] não é adequada para descrever a arquitetura de um SGBDH [SL90], necessitando ser estendida para dar suporte à distribuição, heteregoneidade e autonomia. O HEROS se utiliza de uma arquitetura em quatro níveis, descritos a seguir:

- Esquema local: é o esquema local do SGDB componente, expresso no seu próprio modelo de dados. O esquema considerado como local pode ser o próprio esquema conceitual ou um esquema externo que representa uma visão de uma aplicação ou classe de usuários específica;
- Esquema de exportação: é o esquema local do SGBD expresso no modelo de dados do HEROS. O processo de tradução gera mapeamentos entre as representações de objetos em ambos os modelos, utilizados posteriormente para tradução de consultas expressas no model global para o modelo local. O esquema de exportação e os mapeamentos são armazenados em um dicionário de dados local;
- Esquema global: é o esquema obtido pela integração de todos os esquemas de exportação. Este esquema inclui a informação sobre a distribuição dos dados pelos sistemas de bancos de dados componentes e viabiliza a decomposição das consultas globais formuladas, sendo armazenado no dicionário de dados global;
- Esquema externo: é o esquema que representa uma visão do usuário global do esquema integrado do HEROS.

O HEROS se baseia em um esquema de dados global, definido a partir dos esquemas dos bancos de dados integrados e com base em um modelo de dados orientado a objetos. A derivação de esquemas de exportação a partir de esquemas locais é feita pelos administradores de dados locais, com o uso de ferramenta específica, o Tradutor

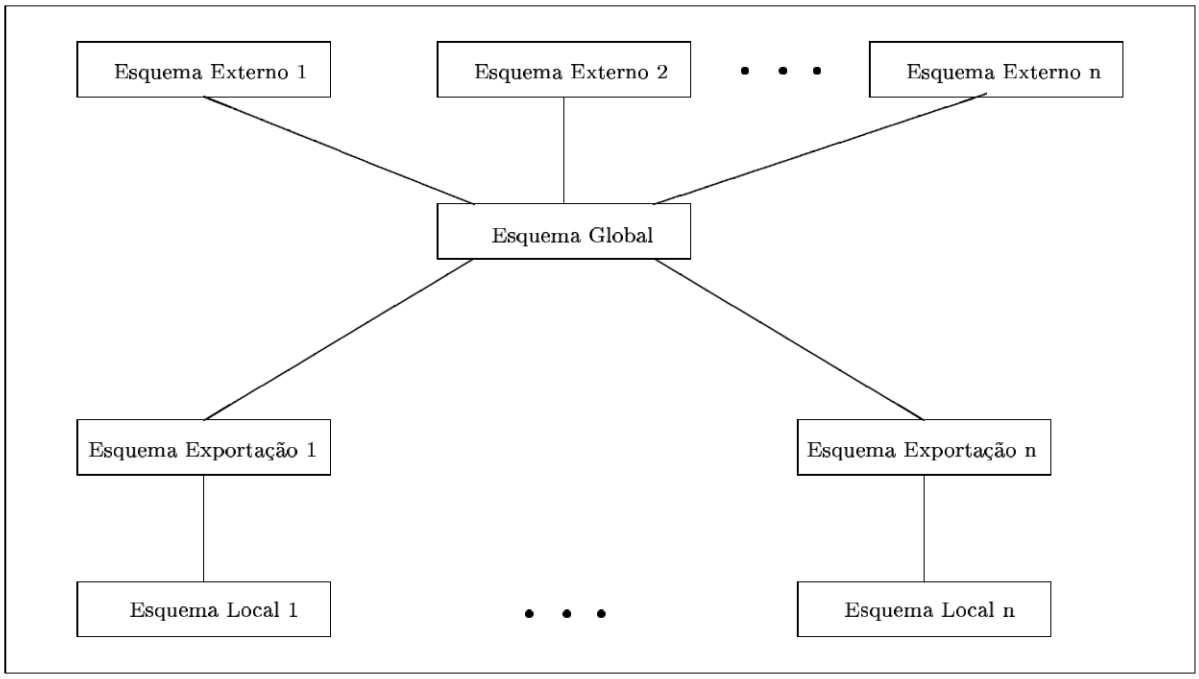


Figura 1: Arquitetura de Esquemas

de Esquemas. A integração dos diversos esquemas de exportação para a ontenção do esquema global é feita também com o suporte de ferramenta específica (o Integrador de Esquemas) pelos administradores de dados locais e pelo administrador de dados global.

A arquitetura adotada pelo HEROS possibilita ao usuário global a independência lógica e física dos dados. Dessa forma, mudanças nos bancos de dados locais integrados pelo HEROS, como alteração da estrutura de armazenamento, localização na rede e migração entre gerenciadores distintos, por exemplo, não impactam as aplicações definidas sobre o esquema global e requerem apenas alteração das traduções e mapeamentos correspondentes pelos administradores de bancos de dados locais e global.

2.2 Arquitetura Funcional

A Camada de Tratamento de Informação realiza o controle sobre todas as informações que entram e saem do sistema. Suas principais funções são o gerenciamento da interação com o usuário e a preparação das informações que serão utilizadas posteriormente nas demais chamadas. Os módulos que compõe esta camada são:

- Gerente da Interface com o Usuário: este módulo é o responsável com o usuário final, derivando do processo de integração consultas globais na linguagem de consulta do HEROS (SQL-like) e retornando-lhe dados;

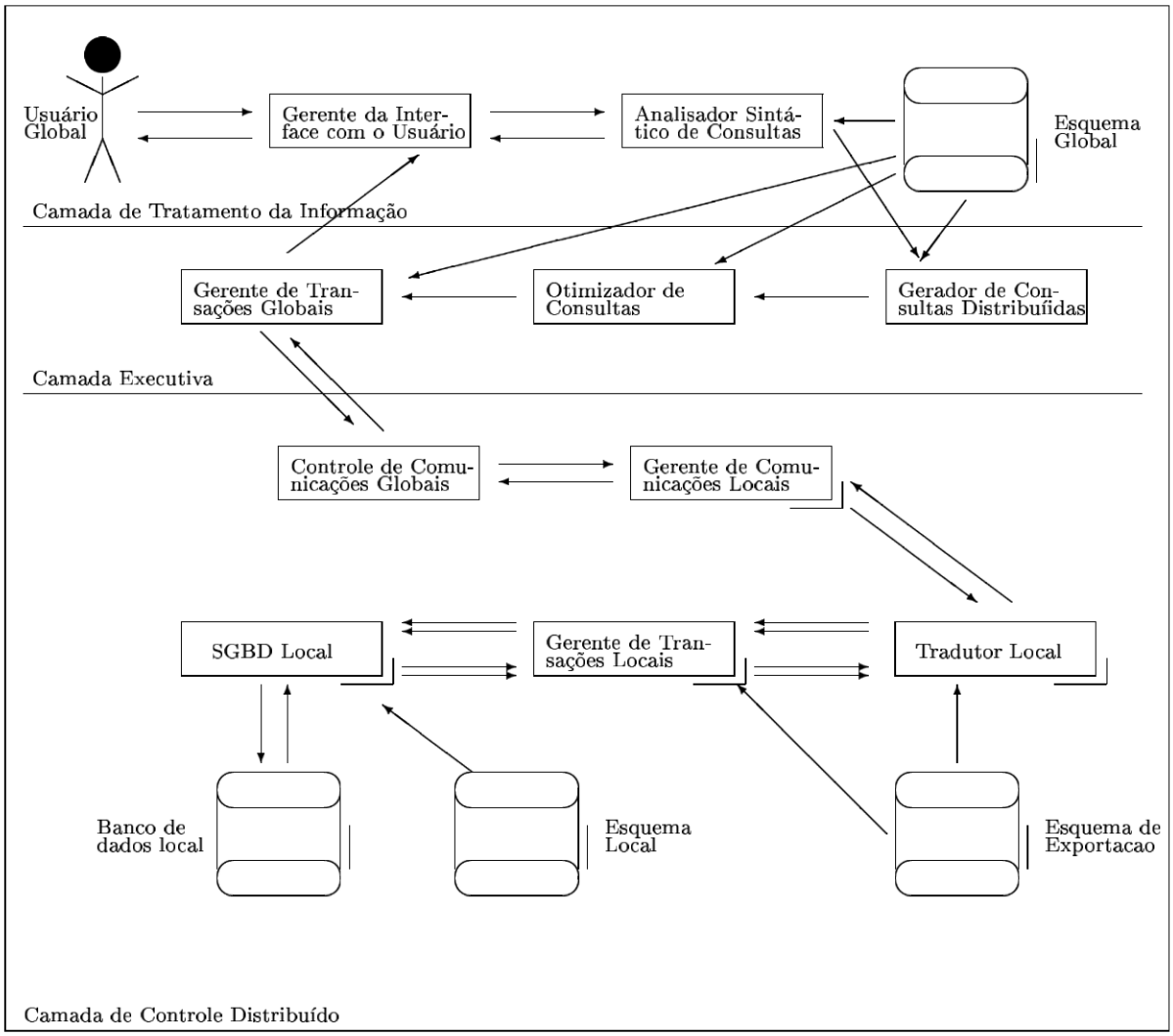


Figura 2: Esquema Funcional do HEROS

- Analisador Sintático de Consultas: tem como função efetuar o *parsing* das consultas, detectando erros de sintaxe ou referências inválidas a objetos do esquema (externo ou global).

A camada intermediária é chamada Executiva e engloba as funções de processamento efetivo das transações globais. Seus módulos são:

- Gerador de Consultas Distribuídas: a partir da consulta global recebida e das descrições contidas no dicionário de dados global deriva as subconsultas que devem ser submetidas aos diversos SGBDs componentes. Cada subconsulta gerada é expressa em termos dos conceitos contidos em um único esquema de exportação. As subconsultas são parte de um plano de execução da consulta global, submetido ao Otimizador de Consultas;

- Otimizador de Consultas: este módulo utiliza como entrada o plano de execução e produz um plano de execução alternativo otimizado. A produção é baseada nos volumes de dados existentes nos sistemas de banco de dados componentes e freqüências de acesso respectivas, coletados periodicamente pelo Gerente de Comunicação Global e pelos agentes locais do HEROS;

- Gerente de Transações Globais: os controles operacionais do ambiente global estão a cargo deste módulo, que recebe um plano de consulta e submete as consultas locais ao módulo Gerente de Comunicação Global. Esxte módulo implementa operações de banco de dados necessárias para composição dos resultados parciais e armazenamento temporário de resultados. O controle da atomicidade da transação e a detecção e o tratamento de *deadlocks* globais são também funções deste módulo.

A última camada é a de Controle Distribuído, responsável pela comunicação entre o Gerente de Transações Globais e os diversos SGBDs subjacentes ao HEROS. Esta camada é constituida pelos seguintes módulos:

- Gerente de Comunicações Global, responsável pela realização física da comunicação. Sua função é prover a conexão com os servidores distribuídos HEROS-L, detectando e reportando ao Gerente de Transações Global qualquer problema relacionado com o acesso a dados remotos;

- Gerente de Comunições Local, que possui localmente função análoga ao Gerente de Comunicações Global;

- Tradutor Local, que traduz uma subconsulta recebida na linguagem de consulta global e com referências aos conceitos do esquema de exportação em uma consulta expressa na linguagem de consulta e conceitos locais. A tradução é feita através de consulta ao esquema de exportação local e também através da aplicação de regras de tradução aplicáveis a cada SGBD específico;

- Gerente de Transações Locais, com a função de submeter e controlar a execução das subtransações locais. Este módulo é o responsável pela criação do estado de "preparado", necessário ao protocolo de sincronismo que implementa a atomicidade das transações globais. A coleta de dados estatísticos sobre o sistema de banco de dados local é efetuada pelo módulo e seus resultados remetidos ao Gerente de Transações Globais para atualização do dicionário de dados global.

Os módulos Gerente de Comunicações Local, Tradutor Local e Gerente de Transações Locais são vistos funcionalmente como um único componente, atuando como agentes locais do HEROS e chamados HEROS-L.

3 Gerência de Transações

Os usuários do HEROS são chamados usuários globais. Esses usuários formulam suas consultas consultas globais sobre um esquema único, obtido pela integração dos diversos esquemas locais dos sistemas de bancos de dados subjacentes ao HEROS. Nesta seção são apresentados os esquemas básicos de processamento adotatos.

3.1 Processamento de Consultas Distribuídas

Inicialmente, o usuário global compõe sua consulta com a ajuda do Gerente de Interface com o Usuário e produz uma consulta na DML do HEROS, com referências aos objetos constantes do esquema global e aos quais tem acesso permitido. Este módulo implementa para cada usuário uma interface adequada às suas necessidades particulares, empregando os conceitos de manipulação direta e orientação a objetos [HH89].

A sequir, a consulta global é analisada sob o ponto de vista da sintaxe empregada e da propriedade das referências efetuadas pelo Analisador Sintático de Consultas. Caso algum erro, incompatibilidade ou restrição de acesso seja detectado, é dado o retorno ao usuário, através do Gerente de Interface com o Usuário.

As consultas validadas são submetidas ao Gerador de Consultas Distribuídas, que produz um conjunto de subconsultas locais equivalentes. Isto é feito pela transformação de consultas Isto é feito transformando consultas aninhadas em *joins* e pelo agrupamento das sub-consultas que se referem a um mesmo banco de dados local, preservando-se a semântica das construções originais. Um plano de execução é gerado sob a forma de árvore, contendo, aém das subconsultas, também operações de controle e de composição de resultados de subconsultas.

O plano de execução é utilizado pelo Otimizador de Consultas que, utilizando-se de dados adicionais sobre os sistemas de banco de dados subjacentes ao HEROS, produz um grafo de consulta [eJPC91] e calcula um plano alternativo de execução otimizado.

O conjunto de sub-consultas considerado pelo Gerente de Transações Global de acordo com o plano de execução ótimo gerado. As subconsultas que podem ser executadas em paralelo são despachadas para o Gerente de Comunicações Global, at'e que não reste nenhuma subconsulta a ser executada. A cada subconsulta que o Gerente de Comunicações Global envia para um Gerente de Comunicações Local, é inicializada a contagem de tempo, permitindo a detecção de impasses e de falhas da rede. A decisão sobre a conclusão de uma transação com sucesso ou não é tomada pelo Gerente de Transações Globais a partir das informações fornecidas pelo Gerente de Comunicação Local e pelos agentes locais do HEROS, os HEROS-L.

As subconsultas locais são recebidas pelo Gerente de Transações Locais, que controla os acessos ao banco de dados local, assegurando a execução serial das transações. Este componente mantém um mecanismo de registro de todas as operações (*log*) do tipo *write-ahead*, necessário para implementação do protocolo de sincronismo global.

3.2 Processamento de Atualizações Distribuídas

O processamento de transações que modificam o conteúdo dos bancos de dados componentes segue os mesmos passos descritos para o processamento de transações de consultas. Os problemas específicos com a atualização de dados podem ser resumidos em controle distribuído de concorrência, tratamento de *deadlock* distribuído e recuperação distribuída de dados, e são abordados a seguir.

Para viabilizar a gerência de transações efetuamos um particionamento restritivo do conjunto de dados, classificando-os em dois subconjuntos distintos:

- atualizáveis globalmente, construído pelos itens de dados que só podem ser modificados por transações globais, e
- atualizáveis localmente, constituído pelos itens de dados que só podem ser modificados por transações locais.

Esta restrição, embora impacte as aplicações já existentes, é um recurso válido [BST90, M+92] na solução dos problemas relacionados com a gerência de transações em ambientes heterogêneos quando se integram SGBDs que não exportam um estado "preparado".

O controle de concorrência é efetuado para assegurar a serializabilidade das execuções [BHG87]. No caso do HEROS o controle é realizado pelo protocolo *two-phase locking* para controle de concorrência [BHG87]. A nível local, o controle será realizado pelos servidores locais do HEROS (HEROS-L), e a nível global é complementado pela manutenção do *locking* até que todas as (sub)transações locais componentes de uma transação global possam receber o *commit* ou o *abort*. A atomicidade das transações é necessária para garantir a produção de estados consistentes nos bancos de dados no caso de falhas dos componentes envolvidos ou de cancelamento de transações [CP85]. No caso do HEROS, a atomicidade das transações globais na presença de falhas estará assegurada pelo uso do protocolo *two-phase commit* (2PC). Um detalhamento do protocolo 2PC aplicado a bancos de dados distribuídos homogêneos pode ser encontrado em [CP85].

O papel de coordenador central caberá ao Gerente de Transações Global (GTG). Para que uma transação global T_i seja completada, o coordenador central requisita o voto de todos os SGBDs locais onde estiverem sendo executadas as subtransações de T_i, cabendo a cada participante manifestar-se pelo *commit* ou pelo *abort* da transação. Como existem SGBDs que não exportam um estado de "preparado", o voto de cada componente será dado pelos servidores locais HEROS-L, não se violando assim a autonomia de execução e a autonomia de projeto de cada SGBD componente.

A ocorrência de *deadlocks* é possível com a utilização do protocolo de controle de concorrência 2PL, uma vez que a gerência de subtransações é distribuída e cada HEROS-L tem apenas visão local. A detecção de eventuais *deadlocks* será feita pelo Gerente de Transações através do controle dos tempos de resposta (*timeouts*).

4 Considerações sobre a Implementação

No desenvolvimento da versão corrente do HEROS, optamos pela utilização de estações Sun. Esta plataforma apresenta vantagens para a implementação, como a existência de uma interface gráfica de usuário (X-Windows), capacidade de multiprogramação e serviços de rede disponíveis. Como SGBDs componentes, foram escolhidos o POSTGRES [RK85] e o O2 [Tec92], já disponíveis no laboratório do Departamento de Informática da PUC-Rio.

O POSTGRES [S+90] é um SGBD baseado no modelo relacional que oferece várias extensões, buscando assim atender aos requisitos das novas classes de aplicação, chamadas não convencionais. Dentre outras funcionalidades, o POSTGRES suporta definição de tipos do usuário, mecanismos de disparo, de alerta e descrição de dados através do emprego de regras.

O O2 [V+89] é um SGBD orientado a objetos que procura atender aplicações comerciais, transacionais, de automação de escritórios e de multi-mídia. Sendo orientado a objetos, o SGBD possui funcionalidades que implementam várias características da abordagem, como a definição de tipos pelo usuário, a criação de objetos complexos, os mecanismos de herança e o suporte á identidade de objetos.

A arquiterura escolhida é a de cliente-servidor [Sin92]. Os clientes são compostos pelos módulos Gerente de Interface com o Usuário e Analisador Sintático, interagindo com o servidor global do HEROS. O servidor global HEROS, por sua vez é cliente dos servidores locais HEROS-L. A comunicação entre os clientes e servidores é feita com o uso dos protocolos TCP/IP e RPC. TCP/IP é um protocolo que permite o compartilhamento de recursos em redes de computadores heterogêneas. O RPC é um mecanismo que implementa a camada a procedimentos remotos em uma rede, e que caracteriza dois agentes básicos, o chamador (cliente) e o chamado (servidor).

5 Trabalhos Relacionados

O HEROS é uma proposta de SGBD para bancos de dados distribuídos e heterogêneos. Bancos de dados distribuídos têm sido pesquisados intensamente nos últimos quinze anos e resultaram em uma grande diversidade de algorítmos, técnicas e soluções para resolver ou contornar os problemas encontrados. Em [Gup89, Tec92, SL90] são abordados os principais aspectos da interoperabilidade de múltiplos e diferentes sistemas de bancos de dados. Em [SL90], ainda, é proposta uma arquitetura de esquemas em cinco níveis para sistemas de bancos de dados heterogêneos, e são revistas diversas implementações de SGBDHs.

Os sistemas de bancos de dados distribuídos e heterogêneos apresentam diversos desafios que demandam esforço de pesquisa. Em [SL90], são sumarisados os seguintes:

- a definição de um modelo de dados integrador;
- o processo de integração de esquemas;
- a definição de um método para mapeamento de esquemas;
- o processamento concorrente de transações;
- os mecanismos de recuperação de falhas;
- a criação de ferramentas de software para ajudar no processo de tradução e integração de esquemas.

Na literatura são encontradas várias propostas de SGBDHs, algumas com referências a implementações de protótipos como CARNOT [CHS91], DATAPLEX [Chu90], HYDRO [PRR91], PEGASUS [A+91] e INTERBASE [E+90]. O modelo de dados utilizado nível global é o relacional ou o funcional, com exceção do PEGASUS, que menciona a utilização do modelo de dados orientado a objetos do IRIS.

6 Conclusões e Trabalhos Futuros

Apresentamos o HEROS, um sistema de gerência de bancos de dados distribuídos heterogêneos que prove autonomia para os SGBDs componentes. A versão em desenvolvimento do HEROS é dirigida para um ambiente local UNIX-like, baseado em estações Sun, com o uso de TCP/IP e RPC. Estão sendo desenvolvidos servidores específicos (HEROS-L) para dois SGBDs com orientação a objetos, o POSTGRES [RK85] e o O2 [Tec92].

O modelo de dados utilizado na integração dos esquemas locais e produção do esquema global é orientado a objetos. Um particionamento restritivo dos itens de dados é efetuado, criando um subconjunto de itens de dados que integram o esquema global e que só podem ser atualizados através de transações globais.

A serializabilidade é obtida com o uso de um protocolo baseado no 2PL, e implementada globalmente pelo módulo coordenador local — o Gerente de Transações Globais — e localmente pelos servidores locais do HEROS. Associado a cada SGBD componente existe um servidor (denominado de HEROS-L), com o papel de garantir a execução serial das subtransações globais e de participar das decisões para conclusão (*commit* ou *abort*) das transações globais.

A coordenação da execução das subtransações exexutadas em cada SGBD componente é realizada por um coordenador central e pelos agentes locais HEROS-L através do uso do protocolo 2PC. A detecção de deadlocks é feita com o uso de mecanismos de controle de tempo.

A construção de servidores locais HEROS-L segue idéias propostas em relatorios técnicos de projetos semelhantes [BST90, PRR91] e encontra-se em seu início.

A criação de agentes locais do HEROS — HEROS-L — que filtrem todas as chamadas aos SGBDs locais está sendo avaliada como alternativa ao atual mecanismo de controle baseado no particionamento restritivo dos conjuntos de dados locais.

Referências

[A+91] R. Ahmed et al. The pegasus heterogeneous multidatabase system. *IEEE Computer*, 24(12), December 1991.

[BGMS92] Y. Breitbart, H. Garcia-Molina, and A. Silberchartz. Overview of multidatabase transaction management. Technical Report TR-92-21, Department of Computer Sciences, University of Texas at Austin, 1992.

[BHG87] P. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison Wesley, 1987.

[BST90] Y. Breitbart, A. Silberchartz, and G. R. Thompson. Reliable transaction management in a multidatabase system. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 1990.

[CHS91] C. Collet, M. N. Huhns, and W. Shen. Resource integration using a large knowleadge base in carnot. *IEEE Computer*, 24(12), December 1991.

[Chu90] Chin-Wan Chung. Dataplex: An access to heterogeneous distributed databases. *Communications of the ACM*, 33(1):70 80, January 1990.

[CP85] S. Ceri and G. Pelagatti. *Distributed Databases, Principles and Systems.* McGraw-Hill Book Co., 1985.

[Dat93] C. J. Date. *An Introduction to Database Systems*, volume 2. Addison Wesley, 1993.

[E+90] A. K. Elmagarmind et al. A multidatabase model for interbase. In *Proc. 16 VLDB Conference*, 1990.

[eJPC91] Rosana S. G. Lanzelotte e Jean-Pierre Cheiney. A requirement analysis for optimising declarative database languages: an eternal problem revisited. In *Anais do 6o Simpósio Brasileiro de Banco de Dados*, pages 59–84, May 1991.

[Gup89] A. Gupta, editor. *Integration of Information Systems: Bridging Heterogeneous Databases.* IEEE Press, 1989.

[HH89] Rex Hartson and Debora Hix. Human-computer interface development: Concepts and systems for its management. *ACM Computing Surveys*, 21:5–92, 1989.

[M+92] S. Mehrotra et al. Efficient global transaction management in multidatabase systems. Technical Report TR-92-11, Department of Computer Sciences, University of Texas at Austin, 1992.

[PRR91] W. Perrizo, J. Rajkumar, and P. Ram. Hydro: A heterogeneous distributed database system. In *Proc. ACM SIGMOD International Conference on Management of Data*, 1991.

[RK85] J. Rhein and G. Kemnitz. The postgres user manual. Technical report, University of California at Berkeley, 1985.

[S+90] R. Stonebreaker et al. The implementation of postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1), March 1990.

[Sin92] A. Sinha. Client-server computing. *Communications of the ACM*, 35(7), July 1992.

[SL90] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed heterogeneous and autonomous databases. *ACM Computing Surveys*, 22(3), September 1990.

[Tec92] O2 Technology. The o2 user manual. Technical Report Version 4.0, 92.

[V+89] F. Velez et al. The o2 object manager: an overview. In *Proc. 15th VLDB Conference*, 1989.