

JOÃO PAULO AIRES

**UMA ARQUITETURA DISTRIBUÍDA PARA O
MODELO DE METADADOS METAMÍDIA**

**CURITIBA
2005**

JOÃO PAULO AIRES

**UMA ARQUITETURA DISTRIBUÍDA PARA O
MODELO DE METADADOS METAMÍDIA**

**Dissertação apresentada como
requisito parcial à obtenção do
grau de Mestre. Programa de Pós-
Graduação em Informática, Setor
de Ciências Exatas, Universidade
Federal do Paraná.**

**Orientador: Prof^a Dr^a Maria Salete
Marcon Gomes Vaz**

**CURITIBA
2005**

*“À minha esposa e minha filha
que sempre estão juntas comigo e
fizeram vencer mais este obstáculo.”*

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus por permitir a realização desse trabalho.

Agradeço à minha orientadora Prof.^a Dr.^a Maria Salete Marcon Gomes Vaz que aceitou me orientar, bem como, pelas constantes cobranças que fizeram com que me dedicasse e pudesse desenvolver meu trabalho.

Agradeço à minha esposa e minha filha, pelos sorrisos e carinhos que me faziam arrumar forças para continuar neste importante trabalho. Obrigado!

Aos meus pais muito obrigado! Vocês que me criaram e me educaram, hoje agradeço por tudo o que vocês fizeram por mim a vida inteira. É por vocês que também dedico este trabalho.

À Coordenação de Informática do CEFET-PR / Unidade de Ponta Grossa pelo apoio e companheirismo nos momentos que precisei.

Ao programa de Mestrado em Informática da Universidade Federal do Paraná que permitiu que realizasse mais este sonho.

RESUMO

O Modelo de Metadados Metamídia é um modelo para indexação e recuperação de metadados para banco de dados multimídia, sem fazer distinção do tipo da mídia manipulada (imagem, vídeo, som ou texto). Este modelo foi projetado para um ambiente centralizado e, por armazenar objetos multimídia em apenas um servidor, o tempo de resposta às requisições poderá ser comprometido. Dessa forma, para o desenvolvimento dessa dissertação, foi feita a extensão do Modelo de Metadados Metamídia, através da distribuição dos metadados indexados pelo modelo, possibilitando aumento na disponibilidade dos metadados, balanceamento de carga computacional e sincronismo. A ferramenta criada para validar o modelo foi implementada usando uma linguagem de programação orientada a objetos e sistemas gerenciadores de banco de dados *open-source*, para comprovar a flexibilidade do modelo proposto. As contribuições referentes ao estudo são as seguintes: (i) integração de dados de diferentes bancos de dados, com tratamento uniforme, através do uso de uma base de metadados descritivos; (ii) descrição de objetos multimídia, usando um modelo de metadados; (iii) compartilhamento de dados entre objetos, através do mecanismo de herança; (iv) distribuição de metadados, aumentando a disponibilidade; (v) criação de mecanismo de sincronia de metadados, permitindo que os bancos de dados sejam idênticos.

ABSTRACT

The Metamedia Metadata Model is a model for metadata indexation and recovery for database multimedia, without making distinction of type media manipulated (image, video, sound or text). This model was projected for a centered environment and, by storing objects multimedia in only one server, the time of reply solicitations could be compromised. This form, for the development this study, was made the extension of the Metamedia Metadata Model, through the distribution of the metadata indexed by the model, making possible increase in the availability of the metadata, computational load balancing and synchronism. The created tool to validate the model was implemented using a programming language oriented-objects and database management systems open-source, to prove the flexibility of the considered model. The referring contributions to the study are the following: (i) data integration of different databases, with treatment uniform, through the use of a base metadata descriptive; (ii) description object multimedia, using a model of metadata; (iii) sharing of data between objects, through the inheritance mechanism; (iv) distribution of metadata, increasing the availability; (v) creation mechanism of synchronism metadata, allowing that the distributed databases are identical.

SUMÁRIO

1 INTRODUÇÃO	10
1.1 MOTIVAÇÃO	10
1.2 OBJETIVOS ESPECÍFICOS.....	15
1.3 ESTRUTURA	15
2. BANCO DE DADOS DISTRIBUÍDOS	16
2.1 CONCEITOS DE DISTRIBUIÇÃO DE BANCO DE DADOS	16
2.2 VANTAGENS	20
2.3 OBJETIVOS DE BASES DE DADOS DISTRIBUÍDAS	21
2.3.1 Autonomia local	22
2.3.2 Não dependência de um servidor central	23
2.3.4 Independência de localização.....	24
2.3.5. Independência de fragmentação	24
2.3.6 Independência de replicação	26
2.3.7 Gerenciamento de transações distribuído	27
2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO	27
3 MODELO DE METADADOS METAMÍDIA	28
3.1 ARQUITETURA METAMÍDIA	28
3.3 DEFINIÇÃO DE METADADOS	32
3.3.1 Categorias de Metadados.....	34
3.3.2 Metadados Técnicos e de Negócios.....	38
3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO	39
4 METAMÍDIA DISTRIBUÍDO	40
4.1 ARQUITETURA DISTRIBUÍDA DO METAMÍDIA.....	40
4.2 FUNCIONAMENTO DA DISTRIBUIÇÃO DOS METADADOS	44

4.3 REPLICAÇÃO DE METADADOS	45
4.4 BALANCEAMENTO DE CARGA	47
4.5 PROTÓTIPO DO METAMÍDIA DISTRIBUÍDO	49
4.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO	55
5 TRABALHOS RELACIONADOS	56
5.1 SISTEMA MDV	56
5.2 SISTEMA <i>LAZY HYBRID</i>	58
5.3 GOULART	61
5.4 SPRING	62
5.5 SISTEMA OVID	63
5.6 ANÁLISE COMPARATIVA	64
5.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO	67
6 CONCLUSÕES E TRABALHOS FUTUROS	68
6.1 CONTRIBUIÇÕES	69
6.2. TRABALHOS FUTUROS	70
REFERÊNCIAS	72

LISTA DE FIGURAS

Figura 1 - Modelo Metamídia Centralizado	13
Figura 2 - Metadados Distribuídos.....	14
Figura 3 - Servidores de um banco de dados distribuído.....	17
Figura 4 - Sistema de banco de dados local.....	19
Figura 5 - Fragmentação Horizontal	25
Figura 6 - Fragmentação Vertical	26
Figura 7 - Arquitetura do Modelo de Metadados.....	30
Figura 8 - Hierarquia de Metadados/Valores	32
Figura 9 - Modelo Metamídia Centralizado	41
Figura 10 - Metadados Distribuídos.....	42
Figura 11 - Modelo da Base do Metamídia Distribuído.....	50
Figura 12 - Tela de cadastro de Objetos e Associação de Metadados.....	52
Figura 13 - Visualização de Metadados e Objetos	54
Figura 14 - Visão da arquitetura MDV	57
Figura 15 - Arquitetura do sistema de armazenamento.....	59

LISTA DE TABELAS

Tabela 1 - Metadados Descritivos	31
Tabela 2 - Comparativo entre ferramentas estudadas e Metamídia Distribuído	65

1 INTRODUÇÃO

1.1 Motivação

Mesmo com a facilidade de manipulação de dados em forma de texto, os dados não convencionais tornaram-se cada vez mais populares, como fonte de informação nos banco de dados [32]. Eles ganharam espaço, pois é observado um grande interesse com relação ao mercado de informações. Os sistemas estão exigindo maior interatividade e flexibilidade, proporcionando o surgimento de técnicas [32] para uma maior aproximação da percepção humana.

Entre as técnicas existentes, destacamos as que consideram as metodologias usadas em indexação convencional serem estendidas para banco de dados multimídia e as que reconhecem a natureza particular dos dados multimídia, como um objeto complexo [4].

Um banco de dados multimídia [32] contém imagens gráficas, clipes de vídeo, arquivos de som, textos, tornando o processamento extremamente pesado e ocasionando demora na resposta.

Dessa forma, o objeto armazenado no banco de dados multimídia deve ser descrito de modo que o mesmo possa ser facilmente localizado e recuperado. No contexto desta dissertação, a indexação é usada para determinar quais metadados devem ser armazenados para descrever os objetos multimídia e permitir a pesquisa e recuperação dos mesmos.

Os metadados, de uma forma tradicional, correspondem aos dados que descrevem dados. Eles dão suporte a um número de funções como: localização, descoberta, avaliação, documentação e seleção. Um Modelo de metadados fornece o contexto para entender os dados através do tempo, fornecendo ferramentas para transformação dos dados em conhecimento [14].

Os metadados provêm uma descrição concisa a respeito do dado podendo ser um documento, uma coleção de documentos, gráficos, tabelas, imagens, vídeos. Por exemplo, a definição de uma classe, em orientação a objetos [2], corresponde a

metadados. Os modelos são inerentemente metadados, visto que descrevem os tipos sendo modelados, assim como as aplicações do mundo real – tais como catálogos e dicionários – têm metadados. Implementações de linguagens também utilizam metadados. Em banco de dados, informações a respeito dos dados são tão importantes quanto os dados.

Os Sistemas Gerenciadores de Banco de Dados Relacionais [7] também utilizam metadados para descrever o esquema da base de dados, como a estrutura das tabelas, índices, proprietário, atributos, tipos de dados, gatilhos, etc. As tabelas no banco de dados são usadas para armazenar dados. De forma similar, um Sistema de Gerenciamento de Banco de Dados Relacional tem várias meta-tabelas que armazenam definições das tabelas.

Em se tratando de recuperação de dados, as pesquisas realizadas pelos usuários de banco de dados convencionais tendem a ser altamente estruturadas, enquanto usuários de sistemas de banco de dados multimídia podem desejar realizar consultas baseadas na descrição do conteúdo de um objeto multimídia. Esta tarefa pode ser difícil se não existirem modelos que permitam a indexação de objetos multimídia [31] no momento de sua criação.

Deste modo, o Metamídia - um modelo de indexação e recuperação de objetos multimídia [32], apresenta uma estrutura de metadados para auxiliar no processo de indexação e recuperação desses objetos. O Metamídia armazena, recupera e indexa dados multimídia através de uma hierarquia de metadados/valores. A indexação é feita de maneira manual, onde o usuário define o conjunto de metadados e semi-automática, onde o usuário define alguns metadados.

Uma das vantagens obtidas com a definição/utilização do Modelo de metadados Metamídia é o ganho em termos de tempo, visto que não será necessário acessar inúmeros objetos até encontrar o objeto desejado, nem tampouco realizar processamentos utilizando algoritmos complexos para a recuperação. Se algum usuário desejar encontrar todas as mídias que tenham uma “bola”, não será necessário realizar um processamento em cada objeto armazenado, pois existirão metadados/valores associados a cada objeto, indicando o que o usuário provavelmente irá querer recuperar.

O Modelo Metamídia foi projetado para um ambiente centralizado cliente/servidor, onde todas as solicitações são realizadas sobre uma máquina específica, que contém

todos os metadados que descrevem os objetos multimídia. Esses objetos ficam armazenados na mesma máquina.

Dentre as vantagens de sistemas de banco de dados cliente-servidor, conforme [9] [27], destacam-se:

- *Independência de plataforma*: os usuários executam aplicações de banco de dados em qualquer tipo de plataforma ou sistema;
- *Centralização dos dados*: os dados ficam disponíveis em um local específico, facilitando a recuperação e manutenção do sistema;
- *Crescimento do sistema*: estações poderão ser acrescentadas no sistema de maneira a modernizá-lo;
- *Realização de backup e manutenção*: como a base de dados é centralizada, facilita a implementação de mecanismos de cópia de dados, bem como a manutenção do banco de dados;
- *Controle de transações*: as transações são controladas com maior facilidade em sistemas centralizados, uma vez que existe apenas um servidor que sofrerá as modificações no sistema;

As desvantagens de sistemas cliente-servidor são as seguintes segundo as referências [9] [27]:

- *Falta de Tolerância à falhas*: como a base de dados é centralizada, se ocorrer alguma falha no servidor que mantém essa base de dados, o sistema inteiro pára de funcionar;
- *Sobrecarga de máquina*: como existe apenas uma máquina que provê os dados a serem consultados/modificados, essa, no momento que estiver atendendo um número elevado de estações, responderá às requisições de maneira mais lenta;

Verifica-se dentre as vantagens e desvantagens comentadas anteriormente, que uma aplicação utilizando uma base de dados multimídia em um sistema centralizado, terá uma eficiência baixa, uma vez que os objetos multimídia (imaginemos um intervalo de um filme armazenado no banco de dados) [31], serão processados por uma única máquina.

A Figura 1 ilustra como o Metamídia trata os metadados de maneira centralizada. Nota-se que nesse sistema, temos o servidor de metadados que armazena os metadados e os objetos multimídia. Este servidor, é acessado por várias estações (representadas na Figura 1 por Cliente 1, Cliente 2 e Cliente N), que realizam consultas/modificações diretamente no Servidor de metadados, ocasionando demora na resposta, uma vez que o mesmo contém objetos multimídia de diversos tamanhos, que são requisitados mesmo tempo por todos os clientes conectados no servidor.

Dessa forma, o Servidor de metadados sofrerá uma sobrecarga de solicitações, fazendo com que a resposta do sistema seja demorada; por tratar-se de dados complexos do ponto de vista do seu tamanho, bem como a quantidade de informação a ser disponibilizada.

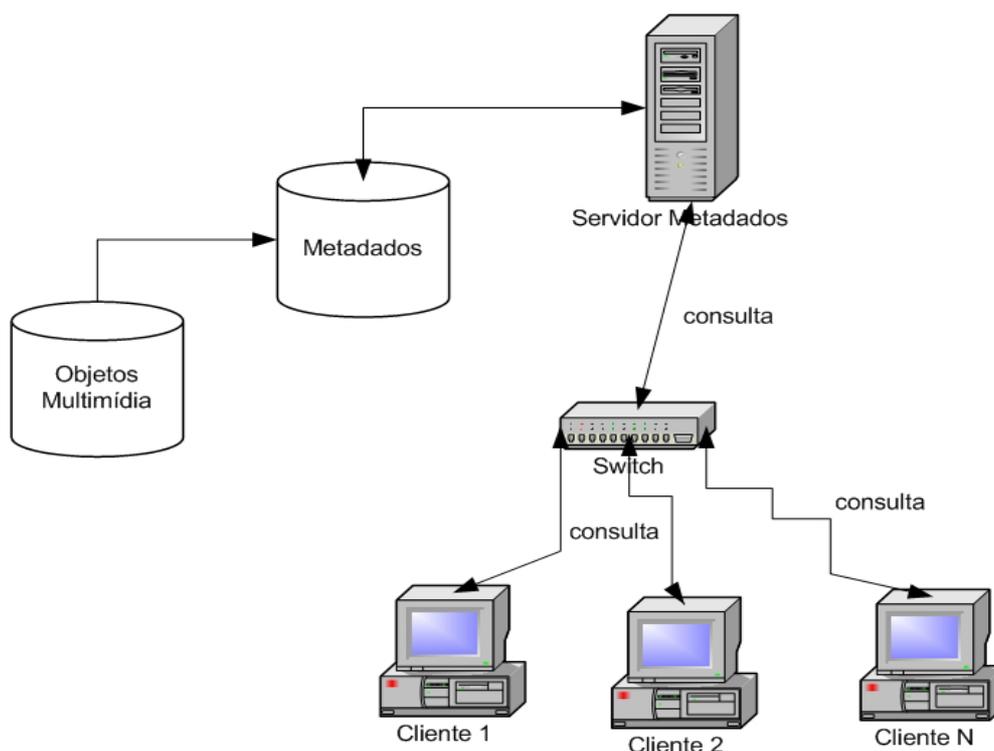


Figura 1 - Modelo Metamídia Centralizado

No ambiente distribuído [23] os vários bancos de dados que compõem o sistema, são logicamente inter-relacionados, distribuídos por uma rede de computadores e definidos como um sistema que permite o gerenciamento do banco de dados, tornando a distribuição transparente para o usuário.

Na Figura 1, cada estação cliente realiza a requisição ao servidor de metadados, que contém as informações dos objetos multimídia, porém somente um servidor atende todas as solicitações do sistema, causando atraso na resposta das solicitações atendidas.

Com a distribuição dos metadados (Figura 2), temos o seguinte cenário: o *Cliente 1* acessará um dos dois servidores disponíveis (conforme figura), por exemplo o Servidor de *Metadados 1*. Quando esse cliente efetuar a inserção de um novo objeto, os metadados associados serão armazenados inicialmente no Servidor de Metadados 1, pois este será o “dono” dos metadados. Após a inserção neste servidor, esses metadados serão copiados através de replicação para o Servidor de Metadados 2 (e para todos os demais que formam o sistema)¹.

Diante do cenário descrito e ilustrado na Figura 2, vamos supor que o Cliente 2 esteja acessando o Servidor de Metadados 2. Todas as consultas que esse cliente (Cliente 2) efetuar serão resolvidas pelo servidor no qual ele estará conectado (Servidor de Metadados 2), pois a base de metadados será igual em todos os servidores. Quando esse cliente consultar um tipo de metadados referente a algum objeto de outro servidor, o objeto será “carregado” automaticamente do servidor que é o “dono” do mesmo.

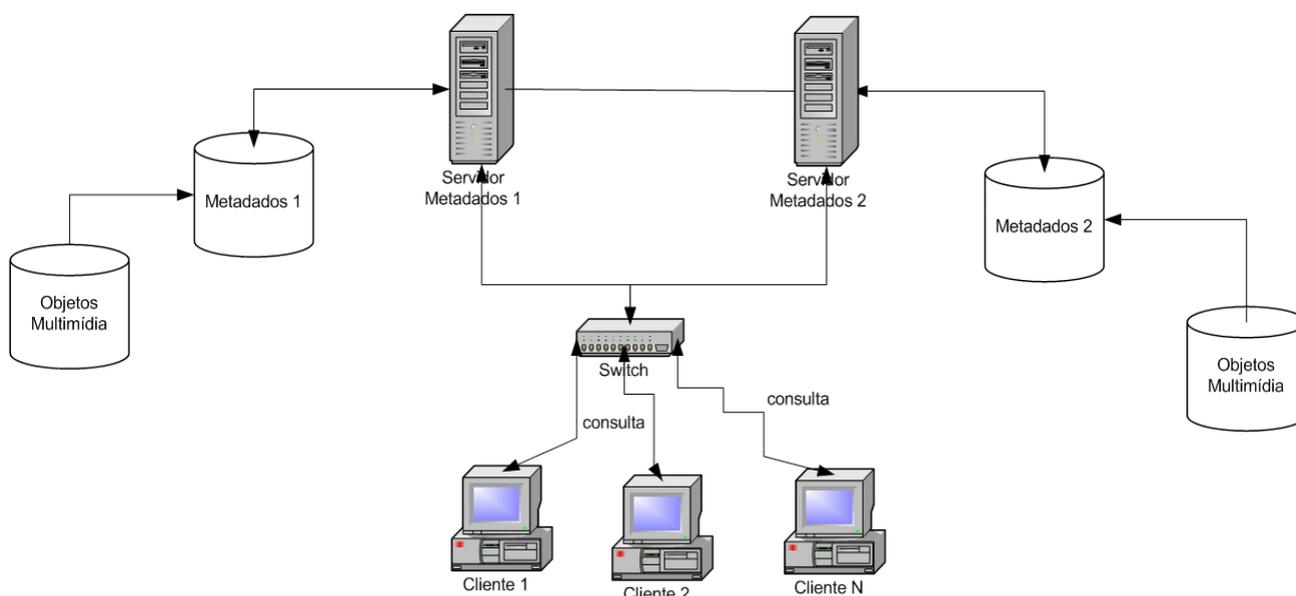


Figura 2 - Metadados Distribuídos

¹ Apenas os metadados serão replicados. Os objetos multimídia ficarão armazenados somente no servidor de origem.

O objetivo principal deste trabalho é apresentar uma arquitetura de distribuição para o Metamídia. Deve-se observar que o Metamídia trata de dados multimídia e seu acesso pode comprometer o desempenho do sistema e, portanto, uma solução é a distribuição dos metadados em vários servidores, para aumentar a tolerância à falhas, a disponibilidade dos metadados e permitir balanceamento de carga computacional.

Na seção seguinte, os objetivos encontram-se mais detalhados.

1.2 Objetivos específicos

Dentre os objetivos deste trabalho, destacam-se:

- Aplicação de metadados para indexação de e recuperação de informações multimídia;
- Apresentação do modelo de distribuição do Metamídia para auxiliar na recuperação de objetos multimídia e levantamento de requisitos para Sistemas de Banco de Dados Distribuídos, contendo:
 - Modelo de distribuição do Metamídia (mecanismo);
 - Criação de ferramenta que implementa o modelo proposto;
 - Estudo de caso para validar a ferramenta;
 - Levantamento de trabalhos relacionados e comparação com o Metamídia Distribuído.

1.3 Estrutura

Tendo em vista o exposto, esta dissertação está estruturada como segue. No Capítulo 2 são descritos os fundamentos de banco de dados distribuídos. No Capítulo 3 é abordado o Metamídia, mostrando suas características e a forma de indexação de dados multimídia proposta por este modelo. No Capítulo 4 é apresentado o modelo de distribuição para o Metamídia bem como o protótipo baseado no modelo de distribuição proposto. No Capítulo 5 são analisados alguns trabalhos correlatos e comparados com o modelo proposto. No Capítulo 6 são feitas as conclusões e perspectivas de trabalhos futuros.

2. BANCO DE DADOS DISTRIBUÍDOS

Um banco de dados é um repositório integrado e compartilhado dos dados operacionais de um empreendimento. Um sistema de gerência de bancos de dados é uma ferramenta destinada a isolar os programas de aplicação dos detalhes de armazenamento do banco, controlando o acesso ao banco e evitando inconsistências ou acessos indevidos, entre outros requisitos funcionais.

Os bancos de dados locais devem ser interligados e coordenados por um Sistema de Gerência de Bancos de Dados Distribuídos (SGBDD) que permita acesso a dados remotos de forma transparente e confiável, mas que mantenha a autonomia dos bancos locais [1] [25].

Esta seção apresenta as principais funcionalidades de um sistema de gerência de bancos de dados distribuídos. Desenvolve ainda argumentos indicando quando bancos de dados distribuídos são uma alternativa, acompanhando uma lista de requisitos funcionais para Sistemas de Gerenciamento de Banco de Dados Distribuídos.

2.1 Conceitos de Distribuição de Banco de Dados

Os Sistemas de Gerência de Bancos de Dados Distribuídos (SGBDD) [9] estendem as facilidades usuais de gerência de dados de tal forma que o armazenamento de um banco de dados possa ser dividido ao longo dos nós de uma rede de computadores.

Para ÖZSU e VALDURIEZ [23], Banco de Dados Distribuídos são uma coleção de múltiplos bancos de dados logicamente relacionados sobre uma rede de computadores e um sistema de gerenciamento de banco de dados, como sendo um sistema que permite o gerenciamento dos sistemas de banco de dados distribuídos, fazendo a distribuição de forma transparente para o usuário.

Em um sistema de banco de dados distribuído, denomina-se *site* o local formado por clientes (estações) e servidores que formam o sistema [7] [10] [23]. Por exemplo, temos a Figura 3 que ilustra a existência dos *sites* numerados de 1 a 5. Nesta figura, os

clientes que acessam dados do *site* 2 (servidor 2) poderão, na existência de uma falha neste servidor, acessar qualquer um dos demais servidores dos outros *sites*, de forma totalmente transparente à aplicação.

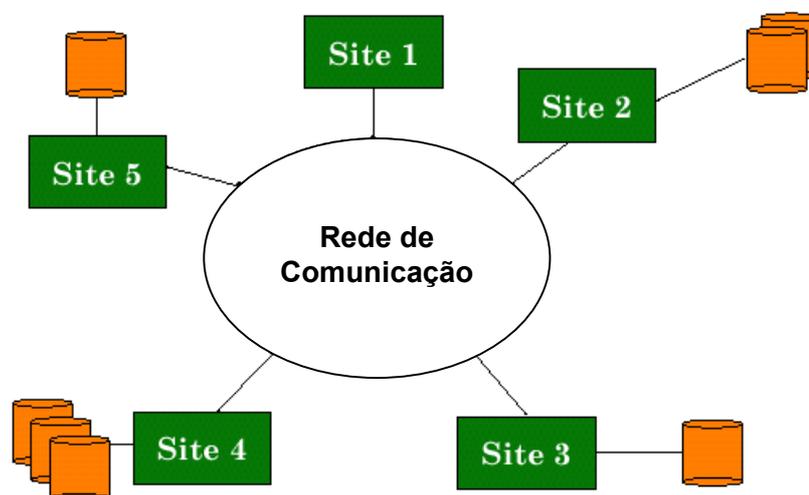


Figura 3 - Servidores de um banco de dados distribuído

A criação de sistemas de banco de dados distribuídos contribui de forma significativa para o aumento da produtividade em desenvolvimento de aplicações. Tais sistemas simplificam a tarefa de definir aplicações que requerem o compartilhamento de informação, programas ou organizações, onde os usuários da informação, ou mesmo as fontes de informação estão geograficamente dispersas. As aplicações com estas características incluem, por exemplo, sistemas de consulta a saldos bancários, outros sistemas voltados para clientes e bancos de dados censitários.

Os Sistemas de banco de dados distribuídos permitem que cada setor de uma organização geograficamente dispersa mantenha controle de seus próprios dados, mesmo oferecendo compartilhamento a nível global no uso destes dados. Os sistemas de banco de dados distribuídos podem diminuir os custos de comunicações, que atualmente tendem a ser maiores do que o próprio custo de equipamento, com o tradicional declínio dos custos de *hardware* [16].

Finalmente, sistemas de banco de dados distribuídos facilitam o crescimento modular do sistema (em contraste principalmente com um sistema centralizado de grande porte), aumentando a confiabilidade através da replicação [11] das partes críticas do

banco em mais de um nó, e podendo aumentar a eficiência através de um critério de particionamento e replicação que coloca os dados próximos do local onde são mais freqüentemente usados (em contraste com acesso remoto a um banco de dados centralizado) [10].

Os Bancos de Dados Distribuídos [27] podem refletir a estrutura organizacional ou geográfica do empreendimento dando maior autonomia e responsabilidade local ao usuário, mas preservando uma visão unificada dos dados. Do lado tecnológico, o desenvolvimento de redes de comunicação de dados permitiu a interligação de um grande número de processadores independentes de forma confiável e com custo previsível.

Do ponto de vista puramente econômico, o preço/desempenho de equipamentos de menor porte tem melhorado substancialmente, argumentando a favor de equipamentos de grande porte. Além disso, os bancos de dados distribuídos problemas de comunicação se a maior parte dos acessos gerados em um nó puderem ser resolvidos localmente, sem acesso a dados armazenados em nós remotos [25].

Os bancos de dados distribuídos podem ser projetados de forma a melhorar a disponibilidade e confiabilidade do sistema através da replicação de dados [11], além de permitirem crescimento da aplicação, simplesmente acrescentando-se novos processadores e novos módulos do banco ao sistema.

Em um sistema de banco de dados distribuído, o conhecimento do estado global do sistema é necessário para processamento de consultas e para o controle de concorrência, enquanto que não só os dados, mas também o controle e informação sobre o estado do sistema estão distribuídos.

Portanto, esses sistemas diferem significativamente de sistemas centralizados do ponto de vista técnico, e um sistema distribuído não pode ser entendido como a simples replicação de sistemas centralizados em vários nós.

Segundo a referência [10], as opções de distribuição de sistemas de banco de dados para dar suporte a armazenamento dos dados de uma aplicação, podem ser classificadas em:

- *Sistemas Distribuídos*: o sistema operacional provê recursos para que as aplicações em execução, em diferentes máquinas, possam trocar dados. O dado está centralizado e os processos distribuídos. Múltiplas aplicações podem acessar o mesmo dado, respeitando o protocolo de acesso definido;

- *Sistemas Cliente-Servidor*: o SGBD provê recursos para que as diversas aplicações possam compartilhar os dados. O dado está centralizado, mas diversos aplicativos podem solicitar a manipulação de dados ao SGBD. A característica básica é a possibilidade de diversos programas poderem acessar os dados tanto para consulta quanto para modificação, simultaneamente;
- *Sistemas de Bases de Dados Distribuídas*: a informação está distribuída em diversos servidores. Cada servidor atua como no sistema cliente-servidor, porém as consultas oriundas dos aplicativos são feitas para qualquer servidor, indistintamente. Caso o dado solicitado seja mantido em outro servidor ou servidores, o sistema encarrega-se de obter a informação necessária, de maneira transparente, que passa a atuar consultando a rede, independentemente de conhecer seus servidores.

De um ponto de vista bem geral, um SGBD distribuído pode ser visto como um conjunto de sistemas de banco de dados centralizados, autônomos, chamados de sistemas *locais* (Figura 4), que são interligados por uma camada de software chamado de *sistema de banco de dados da rede* ou *sistema global* [11].

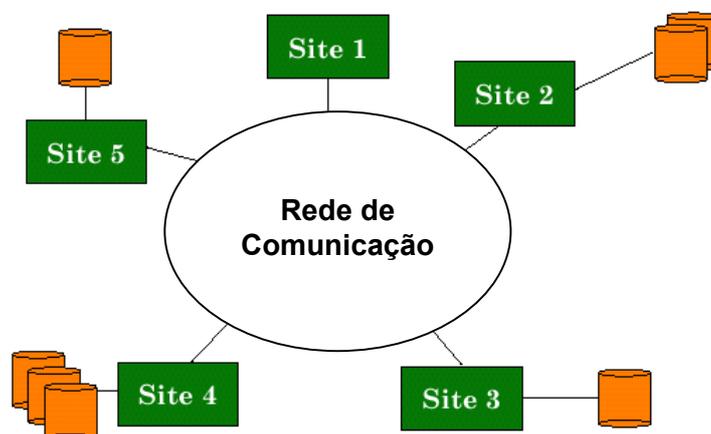


Figura 4 - Sistema de banco de dados local

Um sistema de banco de dados local (Figura 4) é, para todos os efeitos, um sistema centralizado gerenciando de forma autônoma o *banco de dados local* [11], exceto que poderá receber comandos tanto de usuários locais quanto da cópia local do sistema global. O sistema local faz uso do sistema operacional local que provê as seguintes

facilidades básicas: métodos de acesso, gerência de processos e gerência de memória. Cada *site* ilustrado na figura gerencia o banco de dados local, permitindo que todas as consultas sejam realizadas localmente, bem como o tratamento de transações (como, por exemplo, concorrência). Esse tipo de sistema funciona como se não existisse a rede de comunicação.

O sistema de banco de dados global (Figura 4) executa como uma aplicação sob o sistema operacional da rede de comunicação de dados. Isso significa que todos os problemas de comunicação de dados e distribuição de recursos são transparentes ao sistema global. Na figura, o sistema global é representado pela rede, uma vez que é através dela que as cópias locais são idênticas. Cada *site* é autônomo, porém quando alguma modificação ocorre em algum *site*, esta deverá ser comunicada a todos os demais.

2.2 Vantagens

A principais vantagens da distribuição de banco de dados, segundo [1] [7] [9], são:

- **Gerenciamento de dados distribuídos com transparência:** um sistema de banco de dados deve ter transparência na distribuição, ou seja, sem informar detalhes de onde cada dado está armazenado. Esta transparência pode ser de três tipos:
 - **na rede:** detalhes de funcionamento/configuração da rede não são conhecidos pelo usuário;
 - **na replicação:** cópias dos dados são armazenadas em vários locais (*sites*) permitindo maior disponibilidade, desempenho e tolerância a falhas. Através desta transparência, o usuário não tem conhecimento da existência da cópia;
 - **na fragmentação:** a fragmentação horizontal divide a relação em um conjunto de linhas, enquanto que a fragmentação vertical divide-a em um conjunto de colunas, formando sub-relações da relação original. O usuário não tem conhecimento da existência dos fragmentos.
- **Confiabilidade e disponibilidade:** são as vantagens na distribuição dos dados. A *confiabilidade* é a freqüência da existência de falhas em

determinado local, enquanto que *disponibilidade* é a quantidade de vezes em que o dado pode ser acessado, mesmo na presença de alguma falha. Com sistema de banco de dados distribuído, qualquer servidor (*site*) pode falhar, porém os demais existentes continuam operando. Somente os dados existentes no servidor (*site*) falho estarão indisponíveis, diferentemente de sistemas centralizados, nos quais, qualquer falha faz com que todos os sistemas tornem-se indisponíveis.

- **Melhora no desempenho:** fragmentos do banco de dados ficam mais próximos dos locais onde são mais acessados. A localização reduz processamento de entradas/saídas em disco, uso de processador e conseqüentemente diminui o tempo de acesso.
- **Expansão:** facilidade no acréscimo de processadores ou aumento no tamanho do banco de dados.

2.3 Objetivos de Bases de Dados Distribuídas

Segundo as referências [7] [23] [27] existe uma taxonomia que classifica as possíveis alternativas para distribuição de bases de dados em três dimensões:

- **Autonomia:** refere-se à distribuição do controle e indica o grau em que sistemas de gerenciamento de banco de dados individuais podem operar independentemente, de acordo com a consideração de alguns fatores como a necessidade de interoperabilidade dos sistemas componentes e a execução de transações de forma independente por parte desses sistemas;
- **Distribuição:** refere-se ao local de armazenamento físico dos dados, que podem estar armazenados em um único local ou de forma distribuída sobre vários locais;
- **Heterogeneidade:** refere-se à utilização de sistemas de banco de dados distintos ou não para gerenciamento dos dados, ou seja, sistemas de banco de dados heterogêneos ou homogêneos respectivamente.

A taxonomia proposta pelos autores [7] [23] [27] apresenta uma visão diferenciada do conceito de autonomia, considerando a existência de três tipos de autonomia: *forte integração*, *semi-autonomia* e *total autonomia* (ou total isolamento), que correspondem a três tipos de sistemas:

- *Sistemas fortemente integrados*: uma visão única de toda base de dados é disponibilizada aos usuários que querem compartilhar a informação que não pode estar fisicamente armazenada em múltiplas bases de dados;
- *Sistemas semi-autônomos*: consistem em sistemas de banco de dados que podem operar independentemente, mas que participam de um conjunto de sistemas para permitir o compartilhamento de seus dados;
- *Sistemas totalmente isolados*: os componentes individuais são sistemas de banco de dados *standalone* que desconhecem a existência de outros sistemas de banco de dados, bem como a maneira de se comunicar com eles.

Um SGBD distribuído será chamado de homogêneo se os sistemas locais são semelhantes, caso contrário será chamado de heterogêneo. Mais precisamente, um SGBD distribuído é homogêneo se todos os seus sistemas locais: oferecem interfaces idênticas ou, pelo menos, da mesma família e fornecem os mesmos serviços aos usuários em diferentes nós.

Os sistemas de banco de dados distribuídos heterogêneos surgem usualmente quando há necessidade de integrar sistemas já existentes. A escolha entre uma arquitetura ou outra é influenciada pelo aproveitamento de "*hardware*" e "*software*" já existentes e pelo próprio hábito e grau de cooperação esperado dos usuários em caso de uma mudança para um sistema diferente [25].

Segundo a referência [7], os objetivos de um sistema de banco de dados distribuído são úteis para compreensão da tecnologia distribuída e como uma estrutura caracteriza a funcionalidade de sistemas distribuídos específicos.

Nas seções a seguir, serão abordadas as principais características de Sistemas Gerenciadores de Banco de Dados Distribuídos, dando ênfase na distribuição de metadados, que é o objetivo desta dissertação.

2.3.1 Autonomia local

Os servidores em um sistema distribuído devem ser autônomos [23]. Autonomia local significa que todas as operações em um determinado servidor são controladas por esse

servidor; nenhum servidor X deve depender de algum outro servidor Y para sua operação ser bem-sucedida (caso o servidor Y estivesse inativo, o servidor X não poderia funcionar, mesmo que X estivesse funcionando corretamente).

A autonomia local indica que os dados são gerenciados por algum banco de dados local, mesmo que sejam acessíveis através de servidores remotos. Assim, a segurança, integridade e armazenamento são de responsabilidade do servidor local.

Fazendo analogia com metadados, o servidor responsável em mantê-los deverá ser acessado independentemente dos demais servidores que formam o sistema, ou seja, toda consulta ou atualização dos metadados acontecerá e será gerenciada localmente. Acesso concorrente, problemas com transações e falhas físicas ou lógicas, são de responsabilidade do servidor local.

2.3.2 Não dependência de um servidor central

A autonomia local implica que todos os servidores devem ser tratados como iguais. Portanto, não deve haver nenhuma dependência de um servidor central que forneça algum serviço – por exemplo processamento de consultas e gerenciamento de transações – tais que o sistema inteiro dependa desse servidor central [27].

A dependência de um servidor central seria indesejável pelo gargalo criado na utilização deste; bem como o sistema seria vulnerável, pois se ele parasse de funcionar, todo o sistema pararia.

Imaginemos que exista um servidor que seja responsável em manter os metadados do sistema. Toda vez que ocorrer modificações ou consultas, apenas este único servidor será responsável em resolver todas as solicitações, o que representará em diminuição do tempo de resposta, uma vez que toda requisição sobrecarregará uma máquina central.

2.3.3. Operação contínua

Uma vantagem dos sistemas distribuídos é que eles devem fornecer maior confiabilidade e maior disponibilidade.

A confiabilidade – sistema funcionando sem queda em qualquer momento – melhora porque sistemas distribuídos podem continuar a funcionar mesmo diante da falha de algum componente individual, como um servidor isolado [7] [27].

A disponibilidade – sistema funcionando continuamente sem queda durante um período – também melhora devido à possibilidade de replicação dos dados [7].

Não será preciso desligar o sistema para a execução de alguma tarefa, como adicionar um servidor ou atualizar o sistema de banco de dados.

Esta característica é fundamental, uma vez que os sistemas garantem o funcionamento/ acesso dos dados/metadados, independente do tempo necessário para manipulá-los. Como os metadados estarão distribuídos (replicados) em todos os servidores, mesmo na presença de uma falha, algum servidor disponibilizará os metadados para serem manipulados.

2.3.4 Independência de localização

Também conhecida por transparência de localização [7], é quando os usuários não precisam saber onde estão fisicamente armazenados os dados.

É uma característica desejável, pois simplifica programas e atividades em aplicações e permite que dados migrem de um servidor para outro, sem invalidar qualquer desses programas e atividades. Essa capacidade de migração é desejável porque permite que dados sejam deslocados pela rede em resposta a alterações de exigências de desempenho [7].

Utilizando-se de metadados, a transparência de armazenamento faz com que os usuários das aplicações desconheçam a forma (em que local) como os metadados são mantidos, simplificando a adição de novos servidores no sistema distribuído.

2.3.5. Independência de fragmentação

Um sistema admite fragmentação de dados se uma dada variável de relação armazenada pode ser dividida em pedaços (fragmentos) para fins de armazenamento físico.

A fragmentação é desejável por razões de desempenho: os dados podem ser armazenados no local em que são mais freqüentemente utilizados, de modo que a maior parte das operações seja apenas local e o tráfego na rede seja reduzido [23].

Aplicando as técnicas de fragmentação horizontal e vertical a metadados, temos que o particionamento das linhas (fragmentação horizontal) ou das colunas (fragmentação vertical) sugerem que os metadados mais acessados estejam disponíveis localmente, ou seja, mais próximo do local que precise, evitando tráfego de dados pela rede, causando congestionamento.

Existem dois tipos de fragmentação: horizontal (Figura 5) e vertical (Figura 6).

A Fragmentação Horizontal (Figura 5) divide a tabela através da divisão das linhas em vários fragmentos. A reconstrução da tabela pode ser obtida através da operação de união dos fragmentos. Esse tipo de fragmentação está ilustrado conforme a Figura 5, representando os fragmentos de funcionários trabalhando em sedes diferentes.

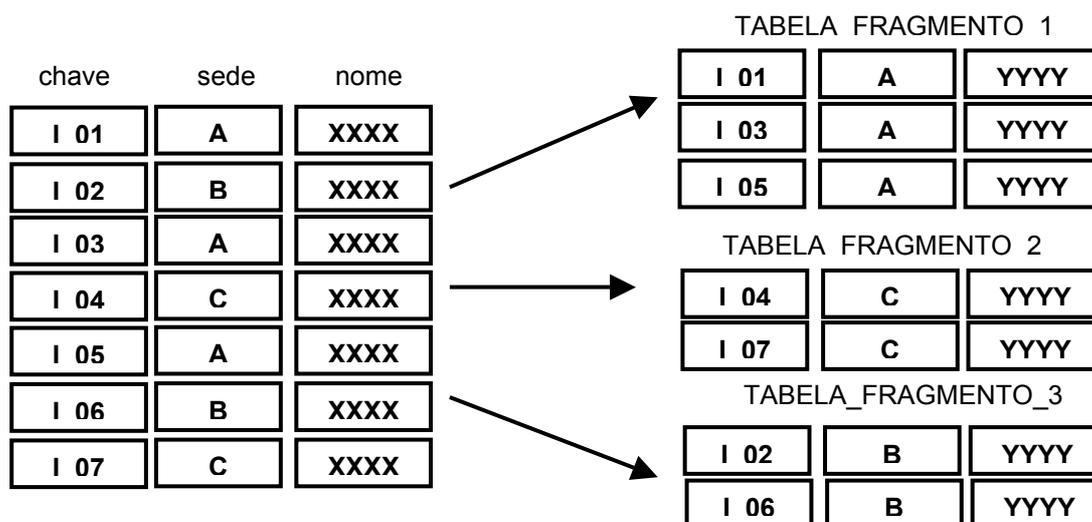


Figura 5 - Fragmentação Horizontal

A Fragmentação Vertical (Figura 6) divide a tabela pelas colunas sob os critérios determinados pelo usuário. A reconstrução da tabela pode ser obtida através da operação de junção dos fragmentos.

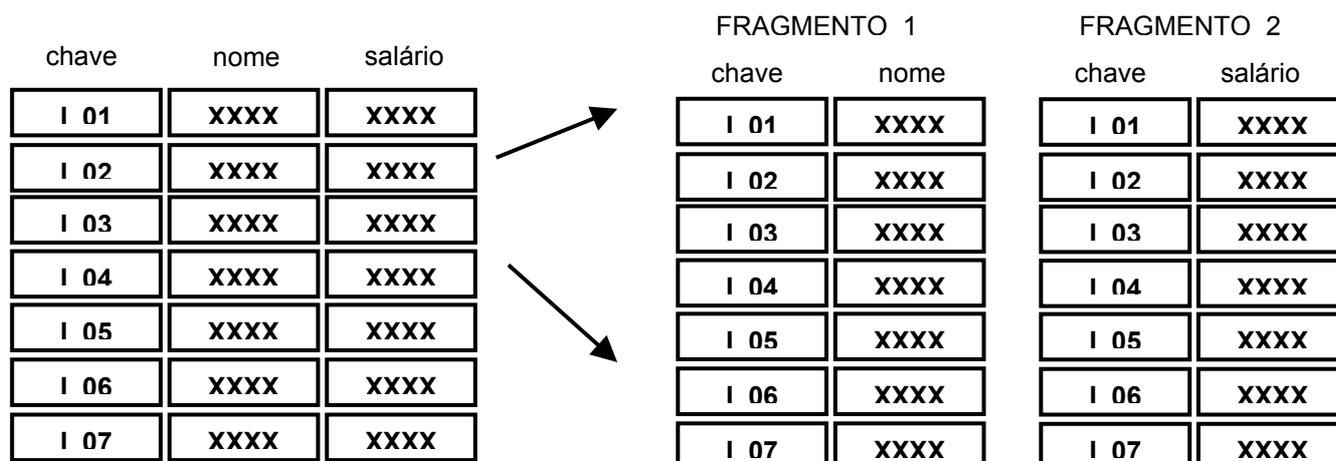


Figura 6 - Fragmentação Vertical

As técnicas de replicação e fragmentação podem ser aplicadas sucessivamente à mesma tabela, ou seja, um fragmento pode ser reproduzido e réplicas de fragmentos podem ser fragmentadas depois, e assim por diante. A próxima seção aborda sobre a replicação de dados.

2.3.6 Independência de replicação

O SGBD mantém várias cópias (réplicas) idênticas dos dados. Cada réplica é armazenada em um servidor diferente, resultando em reprodução de dados. A reprodução, em geral, aumenta o desempenho de operações de leitura e aumenta a disponibilidade de dados para transações de escrita. Entretanto, atualizações de transações ficam sujeitas a baixos desempenhos, pois elas devem ser refletidas a vários servidores [23].

Porém, a replicação torna-se desejável, quando existe a necessidade da criação de sistemas tolerantes à falhas, uma vez que o dado existirá em mais de um local e poderá ser acessado sempre que o servidor local estiver indisponível. Através da replicação, podemos criar sistemas com balanceamento de carga computacional, dividindo o acesso entre os servidores que compõem o sistema.

Ao manipularmos informações multimídia (imagens, vídeo, sons), temos várias informações de diversos tamanhos. Desta forma, torna-se inviável a replicação dos dados

armazenados (uma imagem, por exemplo). Sendo assim, apenas os metadados precisarão ser replicados a todos os servidores, representando um menor fluxo de informações e aumentando a disponibilidade dos metadados existentes no sistema.

2.3.7 Gerenciamento de transações distribuído

Existem duas características no gerenciamento de transações: o controle de recuperação e o de concorrência, que exigem enorme tratamento no ambiente distribuído.

Em um sistema distribuído, uma única transação pode envolver a execução de código de vários servidores, podendo haver atualizações em muitos servidores [6].

Para garantir que uma transação distribuída seja atômica, o sistema deve assegurar que o conjunto de agentes para essa transação seja efetivado ou desfeito. O controle de concorrência na maioria dos sistemas distribuídos se baseia no bloqueio, da mesma forma dos sistemas que não são distribuídos [23].

Para o caso de metadados, poderá acontecer que em uma determinada consulta envolva metadados do servidor local que originou a pesquisa, como provavelmente envolverão os demais servidores existentes, no momento da recuperação dos dados multimídia envolvidos na resposta. Dessa forma, se um determinado tipo de metadados precisar ser modificado ou adicionado, a atomicidade das transações será garantida pelo sistema de banco de dados distribuído, fazendo com que a transação seja efetivada ou cancelada.

2.4 Considerações Finais do Capítulo

Neste capítulo, foram destacados os aspectos de distribuição de bases de dados, tratando como será realizado utilizando-se metadados, que é o principal objetivo deste trabalho. Podemos perceber que alguns pontos focam a utilização de um banco de dados distribuído, destacando: a independência de replicação, independência de *site* central, autonomia local e a operação contínua.

3 MODELO DE METADADOS METAMÍDIA

Um banco de dados multimídia [33] contém imagens gráficas, clipes de vídeo, arquivos de som, textos, entre outros. Assim, o objeto armazenado no banco de dados multimídia deve ser descrito de tal modo que o mesmo possa ser facilmente localizado e recuperado. A indexação é usada para determinar quais dados devem ser armazenados para descrever os objetos multimídia e permitir a pesquisa e recuperação dos mesmos.

Neste capítulo serão destacadas as características do Modelo Metamídia, a arquitetura utilizada para a indexação e recuperação dos metadados, bem como a definição e a categorização de metadados.

3.1 Arquitetura Metamídia

As pesquisas realizadas pelos usuários de banco de dados convencionais tendem a ser altamente estruturadas, enquanto usuários de sistemas de banco de dados multimídia podem desejar realizar consultas baseadas na descrição do conteúdo de um objeto multimídia [33].

Esta tarefa pode ser difícil se não existirem modelos que permitam a indexação de objetos multimídia no momento de sua criação. No caso de um processo demorado para identificar uma característica, pode ser indexado após a criação. Assim, o Metamídia apresenta um modelo de metadados para auxiliar no processo de indexação e recuperação de objetos multimídia.

Uma das vantagens obtidas com a definição do modelo de metadados é o ganho em termos de tempo, visto que não será necessário acessar inúmeros objetos até encontrar o objeto desejado. Outra vantagem é que no uso de descrições de objetos existentes, as informações desses objetos são herdadas pelo novo objeto, permitindo o compartilhamento de metadados/valores entre meta-objetos multimídia [32].

O processo tradicional de indexação envolve determinar os dados que serão armazenados e os relacionamentos que existem entre eles. As tabelas de banco de dados tradicionais podem ser usadas para descrever os dados formatados.

Segundo a referência [32] existem duas abordagens para modelo de dados multimídia. A primeira considera que as técnicas e metodologias usadas em indexação convencional são estendidas para banco de dados multimídia. A segunda reconhece a natureza particular dos dados multimídia, como um objeto complexo. Para tanto, podem existir informações associadas com tais objetos que são classificadas como segue:

- **O próprio objeto**, que é composto de outros objetos. Um usuário pode visualizar todo ou somente parte do objeto. Por exemplo, uma seqüência de vídeo é um agregado de objetos dos tipos vídeo, áudio e texto;
- **As características e propriedades que definem o objeto**, tais como o formato do registro, duração de vídeo, tamanho do frame e técnicas de compressão;
- **Atributos que descrevem o objeto complexo e seus relacionamentos com outros objetos do banco de dados**, muitas vezes descritos como atributos abstratos, tais como título, diretor, data da produção e estúdio de produção.

Adicionalmente, operações/métodos sobre os objetos multimídia são definidas e essas operações diferem daquelas que operam sobre os dados tradicionais mantidos dentro de um banco de dados. Os objetos multimídia são objetos do mundo real, tais como clipes de vídeo ou imagens gráficas [31].

Em bancos de dados convencionais são conceitos abstratos que descrevem objetos do mundo real. Por exemplo, o banco de dados de empregado mantém as informações a respeito de empregados de uma organização e não o próprio empregado. O banco de dados multimídia mantém as informações do mundo real e não atributos dos mesmos.

Tendo em vista o exposto, o Metamídia propõe um modelo de metadados que permite indexar (descrever) objetos multimídia com o objetivo de recuperá-los. A arquitetura do modelo de metadados (Figura 7) é composta por três módulos: *hierarquia de metadados/valores*, *esquema de metadados* e os *objetos multimídia* propriamente ditos.

A *hierarquia de metadados/valores* corresponde à expressão semântica e pragmática de aplicações particulares. Conceitos são definidos como metadados/valores pelos usuários. Neste contexto, o usuário definirá os tipos de metadados e valores que fazem parte da aplicação.

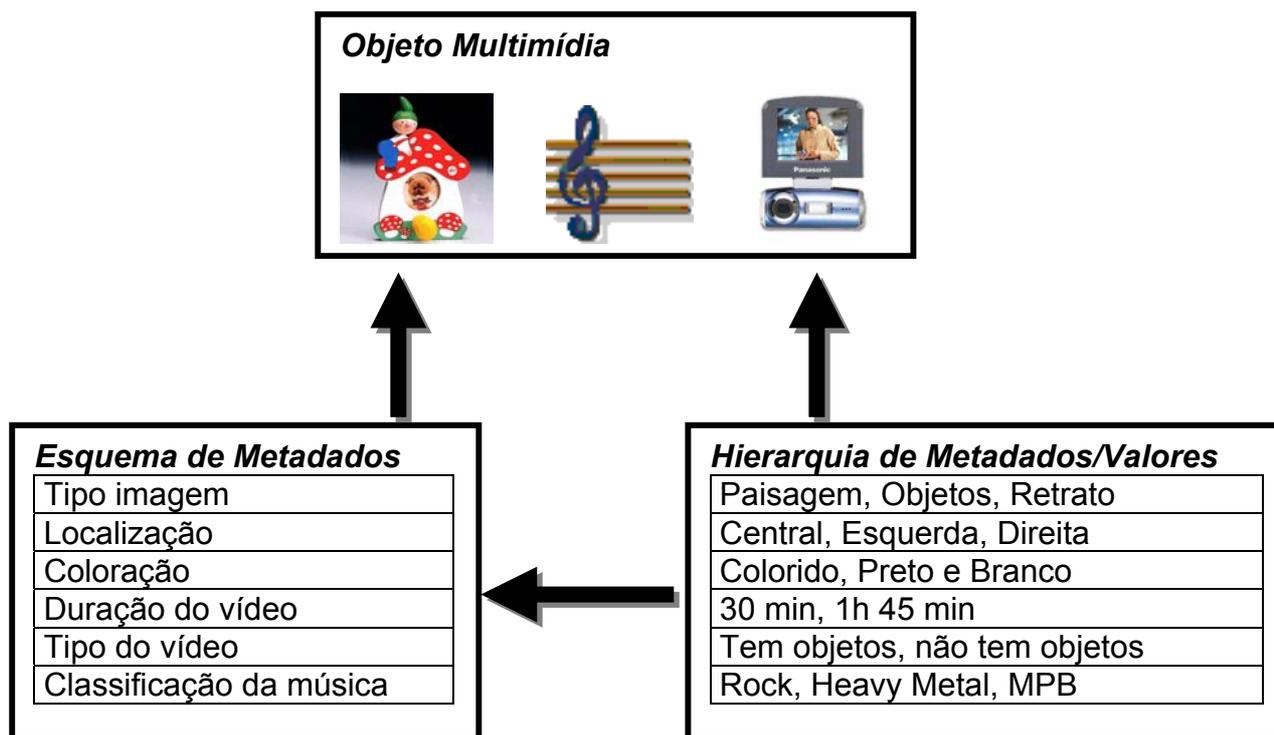


Figura 7 - Arquitetura do Modelo de Metadados

O esquema de metadados é criado para os objetos multimídia indexados. Os metadados/valores são criados para os objetos, gerando os meta-objetos multimídia. Nesse módulo é especificada a forma de representação, a sintaxe dos meta-objetos multimídia. O último componente do meta-modelo corresponde aos objetos multimídia armazenados e descritos para futura recuperação, ou seja, descrito utilizando os metadados existentes, baseado no conhecimento do objeto.

3.2 Hierarquia de Metadados/Valores

A hierarquia de metadados/valores apresentada no Metamídia é uma especificação dos conceitos, isto é, uma descrição dos conceitos e relacionamentos que podem existir para um conjunto de objetos multimídia [32].

A generalização de uma representação para vídeo, som, imagem e texto, pode ser feita em um domínio de aplicação. Para apresentar o modelo, um conjunto de metadados/valores é mostrado na Tabela 1, como exemplo, para uma agência de Publicidade, que trabalha com todos esses tipos de objetos multimídia.

A representação tabular (Tabela 1) é utilizada para melhor ilustrar os metadados/valores sugeridos para a aplicação, os quais são nomeados como descritivos [32]. Esses metadados são gerados manualmente pelo usuário. As colunas correspondem a: (i) **M** é a designação simbólica dos metadados que são utilizados na construção da seqüência de caracteres (*strings*) para a descrição de cada objeto; (ii) os metadados que podem ser utilizados na aplicação; e (iii) alguns valores que esses metadados podem assumir (entre outros).

Tabela 1 - Metadados Descritivos

M	Metadados	Valores
M ₁	Localização	Central, Esquerda, Direita
M ₂	Tipo de Imagem	Paisagem, Objetos, Retrato, Nu
M ₃	Classe da Imagem	Realismo, Pictorialismo
M ₄	Coloração	Colorida, Preto e Branco
M ₅	Profissional	Experiente, Amador
M ₆	Formato do Filme	35mm, Slide
M ₇	Textura	Randômica, Regular, Orientada
M ₈	Cor Predominante	Azul, Vermelho, Verde
M ₉	Tempo do Frame	A que ponto pertence a história
M ₁₀	Classe de vídeo	Esporte, documentário, romance
M ₁₁	Tipo de vídeo	Tem objetos, não tem objetos
M ₁₂	Luminosidade do ambiente	Azul, vermelha, ...
M ₁₃	Cor do objeto	Verde, azul, amarelo, ...
M ₁₄	Mídia original	Filme, vídeo
M ₁₅	Tipo de Peça	Outdoor, spot, vídeo, anúncio, jingle

Na Tabela 1, o **M1** representa o metadado **Localização** e possui os seguintes valores: **Central**, **Esquerda** e **Direita**. Assim, a lista $M = [M1, M2, \dots, Mn]$, onde o n variará em função do número de metadados definidos para a aplicação, corresponde aos metadados. Os valores dos metadados são listados da seguinte forma: $M1 = [V11, V12, \dots, V1m]$, $M2 = [V21, V22, \dots, V2m]$, e assim por diante, onde o m variará em função do número de valores assumidos para cada tipo de metadados.

Os metadados/valores, que aparecem na Tabela 1, são colocados em uma hierarquia (Figura 8) representada por um grafo acíclico dirigido. Neste tipo de grafo, metadados/valores podem ter mais de um tipo de metadados de mais alto nível. Por exemplo, o valor **Azul** tem dois metadados de mais alto nível **Luminosidade** e **Cor**.

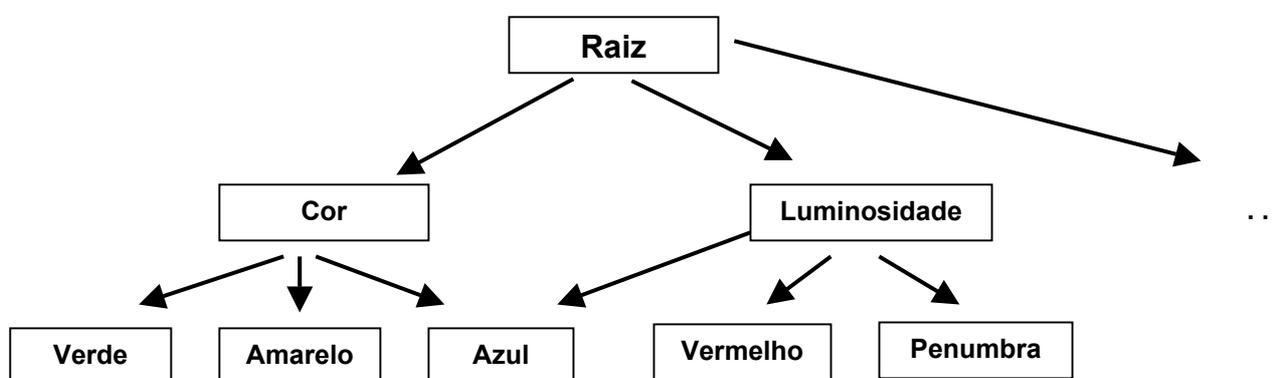


Figura 8 - Hierarquia de Metadados/Valores

Assim, por exemplo, o tipo de metadados **Cor**, pode ser visto como um objeto genérico, dos valores: **Verde**, **Amarelo** e **Azul**. O tipo de metadados **Raiz** representará a aplicação multimídia que pode ser descrita com os metadados/valores definidos.

Dessa maneira, torna-se fácil a associação de valores a um conjunto de metadados existentes, bem como adicionar novos metadados para melhor representar um determinado objeto.

3.3 Definição de Metadados

Em termos simples, os metadados são definidos como sendo **dados sobre o dados**. Em outras palavras, os metadados descrevem ou qualificam outro dado, incorporando, a este,

significado. Sem metadados, a informação se restringe a um conjunto de dados sem significado [13].

Metadados freqüentemente não existem, ou se existem, são incompletos, desatualizados, escritos de maneira pobre e de difícil compreensão. São, muitas vezes, do conhecimento apenas daqueles que criam ou usam os dados. Além disso, costumam estar espalhados em vários documentos, e raramente explicam completamente o conteúdo, significado ou a qualidade dos dados [3] [18].

Nos sistemas comerciais antigos, as aplicações eram responsáveis por manipular diretamente seus próprios arquivos de dados. Sendo assim, os conhecimentos da organização dos dados e método de acesso precisavam ser embutidos no código da aplicação. Além disso, o significado dos dados era também mantido na própria aplicação, o que dificultava o acesso a esses dados por parte de um outro programa.

Com o advento dos Sistemas Gerenciadores de Banco de Dados (SGBD), os metadados passaram a ser armazenados no catálogo do SGBD (dicionário de dados) e não mais de forma individual nos programas, permitindo, dessa forma, que as aplicações se tornassem independentes dos dados [13].

Assim, com a definição dos dados mantida no próprio catálogo, os programas tornaram-se capazes de formular consultas ao banco de dados sem o conhecimento de *como* os dados estão armazenados e da estrutura de acesso a empregar.

A recuperação, inclusão, exclusão e modificação dos dados passaram a ser implementadas nas aplicações através da Linguagem de Manipulação de Dados (DML) suportada pelo SGBD.

O conceito de metadados é muito amplo, devendo ser entendido como um descritor dos dados, responsável por qualificá-lo e incorporá-lo a um contexto. As possibilidades para representá-lo são ilimitadas [20].

São exemplos de metadados, entre vários outros:

- Descrições de visões Lógicas em um catálogo de servidor de dados;
- Comandos SQL de criação de índices ou tabelas;
- Diagramas de Entidade-Relacionamento em um repositório de ferramenta CASE;
- O esquema de um Banco de Dados;

- O padrão convencionado para designação de tabelas e colunas;
- Documento que estabeleça os novos requisitos de um sistema.

A flexibilidade para representação de um esquema de metadados (meta-modelo) motivou muitas organizações ao desenvolvimento de padrões de metadados.

Os padrões determinam como o dado deve ser projetado, capturado, designado, estruturado e documentado. No entanto, esses padrões são muito variáveis e freqüentemente inconsistentes entre as organizações, apresentando formatos variados, com diferentes níveis de detalhes [3] [18].

Na próxima seção, serão descritas as categorias de metadados, que permitem uma divisão dos objetos por elementos específicos, de acordo com sua contextualização.

3.3.1 Categorias de Metadados

A idéia de utilizar metadados para facilitar a descrição e descoberta de informação (ou de recursos de informação) não é nova [3] [18].

Os padrões de metadados possuem uma base sólida na área de Biblioteconomia, Ciência da Informação e Arquivologia, anterior à Internet [14].

Em geral, as abordagens baseadas em metadados para localizar informação, permitem ao usuário realizar uma busca, por um objeto armazenado em um repositório, utilizando campos de consulta típicos, como título, autor, ou ano de publicação.

No entanto, metadados não apenas descrevem o conteúdo de outros dados, mas também indicam muitos outros aspectos, por exemplo, os aspectos relacionados ao gerenciamento, preservação, contexto, e histórico de uso de recursos de informação.

Algumas classificações de metadados podem ser encontradas nas referências [3] [14] [18]. A referência [20] abrange as classificações de BÖHM e RAKOW [3] e [18], e foi estendida com as categorias propostas pela referência [14].

Conforme as referências [3] [14] [18], são quatro categorias que são descritas nas seções seguintes.

Categorias de metadados para descoberta de recursos

Os elementos de metadados das categorias para descoberta de recursos representam o conjunto de termos necessários para descobrir e identificar recursos de informação. Essas categorias são as seguintes:

- **Metadados para descrição de bibliografia.** Esses metadados são elementos descritivos para a identificação e caracterização do recurso.
- **Metadados para identificação única dos recursos.** Esses metadados fornecem uma identificação única para o recurso.
- **Metadados para descrição geral do conteúdo dos recursos.** Esses metadados correspondem a elementos para descrever o conteúdo do recurso.
- **Metadados para descrição do assunto dos recursos.** Esses metadados podem incluir termos controlados ou não controlados, utilizados para caracterizar os conteúdos de informação do recurso.
- **Metadados para descrição da estrutura dos recursos.** Esses metadados são utilizados para descrever a composição de um recurso.
- **Metadados para descrição do relacionamento entre recursos.** Esses metadados incluem elementos para associar um recurso específico com outros recursos relacionados.
- **Metadados para descrição da proveniência dos recursos.** Esses metadados identificam as principais fontes ou fornecedores do recurso.
- **Metadados para descrição do formato e mídia dos recursos.** Esses metadados correspondem a elementos para a representação dos dados e das características de apresentação dos componentes do recurso.

Através dessa categoria os recursos são descritos de maneira a facilitar a recuperação quanto ao assunto, formato, conteúdo, etc., permitindo a descoberta dos recursos usando os metadados associados aos elementos. Dessa forma, torna mais fácil a localização de recursos conforme o que se pretende utilizar.

Categorias de metadados para disponibilidade de recursos

Os elementos de metadados das categorias para disponibilidade de recursos definem os termos e condições necessárias para o acesso e recuperação de recursos, sem considerar se a maneira utilizada é restrita ou irrestrita. Essas categorias são as seguintes:

- **Metadados para distribuição dos recursos.** Esses metadados descrevem como o recurso é disponibilizado.
- **Termos e condições para acesso aos recursos.** Esses metadados especificam as condições prévias de acesso que devem ser preenchidas para garantir a proteção da privacidade e propriedade intelectual associadas ao recurso.
- **Termos e condições para uso dos recursos.** Esses metadados descrevem os termos e condições para utilizar o recurso.
- **Metadados sobre requisitos dos recursos.** Esses metadados especificam os requisitos de software e hardware para o uso do recurso.
- **Metadados para localização dos recursos.** Esses metadados fornecem a informação necessária para transferência do recurso. O principal objetivo é permitir ao sistema selecionar uma cópia ou versão adequada do recurso (por exemplo, para uso local).
- **Metadados sobre autenticidade dos recursos.** Esses metadados descrevem esquemas ou métodos para garantir a autenticidade de um recurso.

Com a associação dos metadados desta categoria, a recuperação dos recursos disponíveis em um determinado momento fica mais simples, possibilitando a descoberta das condições para uso, onde se encontra o recurso e quais os requisitos necessários para a utilização do recurso.

Categorias de metadados para uso de recursos

Os elementos de metadados das categorias para uso do recurso são informações adicionais para permitir o uso adequado de recursos. Essas categorias são as seguintes:

- **Metadados para classificação do conteúdo dos recursos.** Esses metadados representam classificações, baseadas no conteúdo do recurso, de acordo com algum esquema contextual atribuído por uma autoridade.
- **Metadados para descrição da qualidade dos recursos.** Esses metadados especificam a qualidade dos dados do recurso.
- **Metadados para descrição do objetivo dos recursos.** Esses metadados descrevem porque o recurso de informação é fornecido e identifica os programas, projetos, fóruns de discussões, etc., relacionados a esse recurso.
- **Metadados para descrição contextual dos recursos.** Esses metadados fornecem informação relacionada a eventos específicos, situações, configurações, etc., relacionados ao domínio ou finalidade do recurso.

Associando os metadados dessa categoria, os recursos poderão ser utilizados de maneira a adequar-se com o propósito. Efetuando a descrição do objetivo do recurso, permite a identificação da relação do recurso com algum projeto, indicando que somente para aquele tipo de projeto ele poderá ser utilizado. Ao classificarmos recursos nesta categoria, permitimos que esses possam ser melhor utilizados.

Categorias de metadados para administração e controle de recursos

Os elementos de metadados das categorias para administração e controle de recursos fornecem informação para controlar, auditar e rastrear a informação sobre o próprio recurso e também sobre os seus metadados. Essas categorias são as seguintes:

- **Metadados para controle de modificação dos recursos.** Esses metadados controlam as versões do recurso de informação.
- **Metadados para administração dos recursos.** Esses metadados são relacionados a qualquer informação com relação ao gerenciamento e controle do próprio recurso de informação.
- **Metadados para histórico de uso dos recursos.** Esses metadados são reservados para representar informação sobre as operações executadas no recurso de informação, como cópia, edição, remoção, etc.

- **Metadados para administração dos metadados dos recursos.** Esses metadados são relacionados a qualquer informação com relação ao gerenciamento e controle dos próprios metadados do recurso de informação.
- **Metadados para preservação dos recursos.** Esses metadados são relacionados ao gerenciamento da preservação do recurso de informação.

Através da utilização dos metadados dessa categoria, podemos descobrir como um recurso foi utilizado, permitindo gerenciá-lo para um maior controle e preservação do recurso, como versão, cópias, remoção. Com isso, podemos saber a quantidade de vezes que um recurso foi editado ou copiado, garantindo a conformidade dos recursos disponíveis.

3.3.2 Metadados Técnicos e de Negócios

Os metadados *técnicos* são representados pelo dado estruturado, ou seja, é definido através de linguagem ou modelo formal. Sua representação é caracterizada por um conjunto de entidades relacionadas que, por sua vez, são descritas por atributos, e estes por expressões que os quantificam ou qualificam. Esse tipo de metadado é encontrado particularmente em catálogos de bancos de dados, dicionários de dados, modelos de dados, ferramentas CASE e ferramentas de extração [20].

A finalidade dos metadados de *negócio* é prover uma camada semântica entre o analista de negócio e o sistema de suporte à decisão. Os analistas não apresentam um perfil técnico, portanto é necessário que a informação esteja em uma linguagem adequada aos seus entendimentos. São exemplos de metadados apropriados: consultas, relatórios e dados (tabelas e colunas), localização do dado, seu contexto e definição, regras de transformações e sumarizações, sistemas fontes, pessoas responsáveis pelos dados [20].

Esses metadados, diferentemente do que ocorre com metadados técnicos, representam dados não estruturados, assim chamados porque suas definições não são apresentadas através de linguagens formais.

Assim, usuários finais tendem a preferir metadados não estruturados, pois geralmente não são representados por entidades e atributos; mas sim por uma descrição textual, gráficos de um processo de negócios, ou dados de áudio ou vídeo explicando o uso, por exemplo, de uma interface gráfica.

3.4 Considerações Finais do Capítulo

Neste capítulo foi enfatizado o Modelo de Metadados Metamídia, que é um modelo de indexação e recuperação de objetos multimídia utilizando metadados, independente do tipo da mídia utilizado (imagem, som, vídeo, texto). Foi ilustrada a arquitetura deste modelo, bem como foi realizada uma explanação a respeito de metadados, definindo-os e categorizando-os. No contexto deste trabalho, o Modelo de Metadados Metamídia foi utilizado como base para a distribuição dos metadados indexados para a descrição dos objetos armazenados.

4 METAMÍDIA DISTRIBUÍDO

Como mencionado na Seção 3, um banco de dados multimídia contém objetos complexos de serem manipulados, tais como imagens, vídeo, som e textos.

Portanto, devemos descrever o objeto de maneira a facilitar a recuperação, sem a utilização de mecanismos, como, por exemplo, processamento de imagens.

O Metamídia utiliza uma estrutura de metadados para indexar e recuperar objetos multimídia, independente do tipo da mídia (imagem, som, vídeo ou texto), através de uma hierarquia de metadados/valores. Nesse contexto, todo objeto inserido na base, é associado a um conjunto de metadados para a descrição do mesmo.

Dessa maneira, esta dissertação procura estender o Modelo Metamídia, propondo um mecanismo de distribuição dos metadados descritos aos objetos, com o intuito de aumentar o desempenho e a disponibilidade dos dados, bem como reduzir tempo de resposta das consultas.

Neste capítulo, serão abordadas a arquitetura de distribuição do Metamídia, as características implementadas no modelo proposto e a justificativa da escolha das ferramentas utilizadas, enfocando a motivação para efetuar a distribuição.

4.1 Arquitetura Distribuída do Metamídia

O Modelo Metamídia, como mencionado anteriormente, foi projetado para um ambiente centralizado (cliente-servidor), e todo trabalho (recuperação, modificação) é realizado em uma máquina – que contém todos os metadados que descrevem os objetos multimídia.

Deve-se observar que o Metamídia trata dados multimídia e seu acesso pode comprometer o desempenho do sistema, uma vez que apenas uma máquina será acessada e deverá responder a toda requisição do sistema.

Apesar de utilizar um modelo de metadados (para facilitar a indexação e recuperação de objetos multimídia), a arquitetura proposta pelo Metamídia sobrecarregará

um servidor (como mencionado acima), pois este deverá estar disponível a toda solicitação realizada, independente da quantidade de objetos a serem indexados ou recuperados.

Outro detalhe da arquitetura utilizada da forma original (cliente-servidor), é a falta de disponibilidade dos metadados, uma vez que, se o servidor parar por algum motivo (queda de energia, problemas de disco, erro lógico ou do sistema), o sistema não funcionará no tempo em que estiver falho.

No ambiente distribuído [27] os vários bancos de dados que compõem o sistema, são logicamente inter-relacionados, distribuídos por uma rede de computadores e definidos como um sistema que permite o gerenciamento do banco de dados, tornando a distribuição transparente para o usuário.

A Figura 9 ilustra como o Metamídia trata os metadados, conforme a requisição de um usuário, enquanto que a Figura 10, mostra como será realizada a distribuição dos metadados pelos servidores que formarão o sistema distribuído.

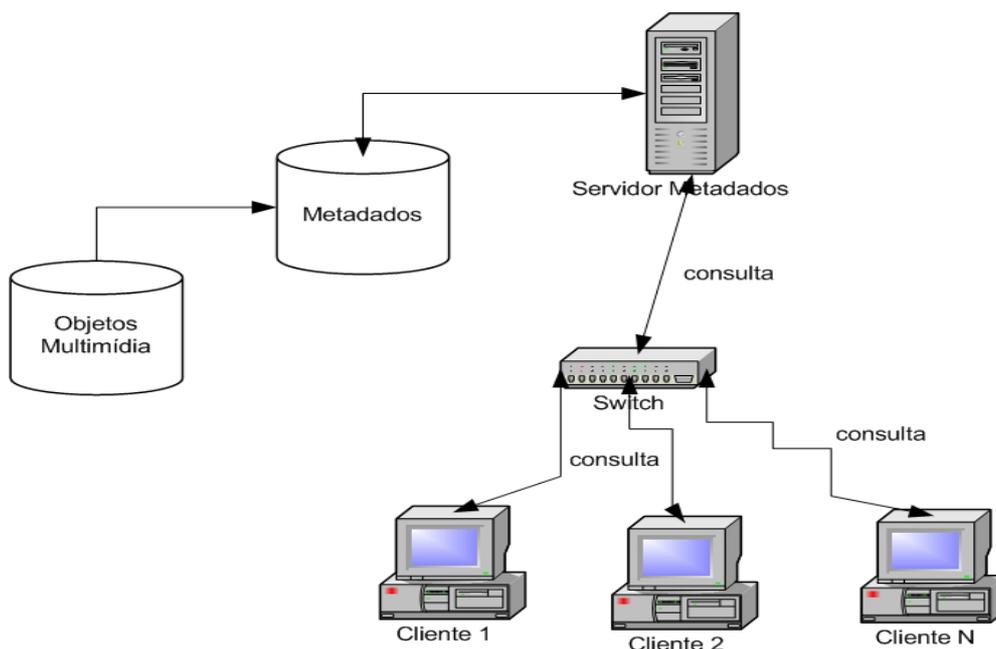


Figura 9 - Modelo Metamídia Centralizado

Na Figura 9, cada estação cliente realiza a requisição ao servidor de metadados (que contém as informações dos objetos multimídia), porém somente um servidor atende todas

as solicitações do sistema e, portanto, pode causar atraso na resposta das solicitações atendidas.

O Modelo Metamídia trata apenas os metadados de maneira centralizada (como ilustrado na Figura 9) e, portanto, não faz distinção do tipo de sistema gerenciador de banco de dados. A única necessidade é a existência de um servidor (de metadados e de objetos) centralizado.

Ao distribuir os metadados, cada servidor conterá os metadados associados a todos os objetos multimídia armazenados, independente onde o objeto estiver armazenado. Como ilustrado na Figura 10, quando um objeto é inserido, ele é armazenado apenas em um servidor, mas os metadados associados são distribuídos a todos os demais servidores. Dessa forma, quando um cliente efetuar uma consulta, essa poderá ser realizada em qualquer servidor (pois os metadados de um objeto estão em todos os servidores), ou seja, devolverá como resultado os metadados/valores e os objetos associados a eles. Se o cliente desejar visualizar algum objeto, a recuperação será realizada no servidor que o armazena.

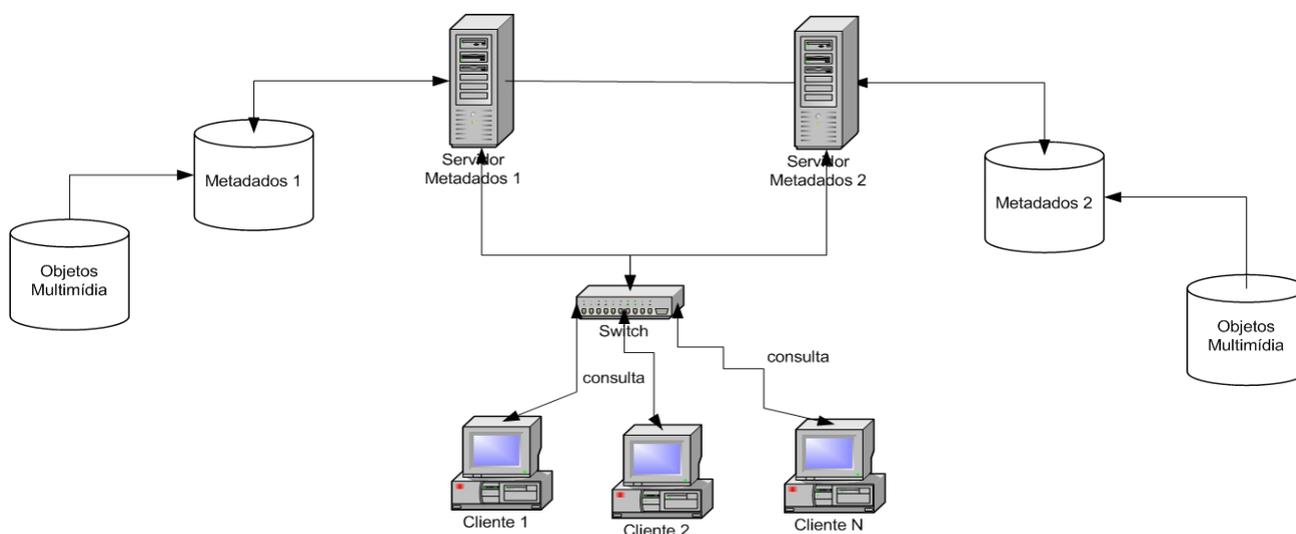


Figura 10 - Metadados Distribuídos

Para a distribuição, neste trabalho foi adotada a *replicação dos metadados*, devido os seguintes fatores:

- Tolerância à falhas: a arquitetura proposta permite que o Metamídia Distribuído possa ser tolerante à falhas, ou seja, mesmo na presença de

uma falha (travamento de um servidor de metadados, por exemplo), o sistema continua operando;

- Balanceamento computacional: como a base de metadados será igual em todos os servidores que compõe o sistema, será possível a ferramenta criada, efetuar a distribuição da carga computacional, evitando que apenas um servidor seja utilizado para responder às requisições;
- Disponibilidade: uma vez replicados os metadados em mais de um local, as consultas estarão disponíveis, independente do local que está sendo consultado.

Em outras palavras, para a montagem da arquitetura de distribuição proposta foi necessário:

- Criar uma base de metadados distribuída;
- Definir quais as tabelas terão seus metadados/valores replicados entre os servidores;
- Definir como será feita a replicação dos metadados/valores;
- Definir a quantidade de servidores para a validação do modelo;
- Definir qual sistema gerenciador de banco de dados a ser adotado;

De acordo com a arquitetura distribuição dos metadados proposta, o Modelo Metamídia Distribuído permitirá que os metadados/valores sejam acessados localmente pelos clientes, uma vez que os metadados encontram-se em mais de um servidor, de forma totalmente transparente ao usuário. Uma observação importante, como já mencionado e ilustrado na Figura 10, é que apenas os metadados são distribuídos; os objetos multimídia permanecerão nos servidores em que foram adicionados inicialmente.

A grande inovação da arquitetura proposta, além da distribuição dos metadados indexados pelo Modelo Metamídia, é a *sincronia dos metadados/valores*, feita através da ferramenta criada para o tratamento da replicação, que, além de permitir a manipulação dos metadados e objetos, é independente de sistema gerenciador de banco de dados utilizado.

Basicamente, a arquitetura proposta funciona conforme ilustrado na Figura 10. Quando algum cliente (estação) executar a ferramenta do Metamídia Distribuído, esta fará o balanceamento, permitindo que os servidores dividam o número de conexões/requisições atendidas. Por exemplo, quando o Cliente 1 executar a ferramenta, esta determina que o servidor que ele deverá acessar é o Servidor 1, pois a rotina verifica que este é o servidor com menos acessos. Toda requisição de recuperação de metadados será direcionada para o servidor selecionado pela ferramenta. Como os objetos multimídia não são distribuídos, quando um cliente selecionar um tipo de metadados que está associado a algum objeto de um servidor diferente daquele determinado para ele acessar, a ferramenta, de forma transparente, faz a requisição para o servidor que contém o objeto, para que o mesmo possa ser carregado.

A Seção 4.5 aborda sobre o protótipo criado, comentando a maneira como os metadados são distribuídos, como um novo servidor de metadados é adicionado no sistema, como os dados são replicados para o novo servidor e como é feito o balanceamento de carga computacional entre os servidores.

4.2 Funcionamento da Distribuição dos Metadados

Com a distribuição dos metadados, cada servidor componente do sistema distribuído, conterà todos os metadados/valores existentes no sistema, tanto àqueles referentes aos objetos multimídia que ele armazena, bem como os metadados/valores dos objetos multimídia armazenados nos outros servidores.

Dessa forma, quando a aplicação efetuar uma requisição, essa será efetuada no servidor que estiver “menos carregado” (selecionado pela ferramenta, através do balanceamento – Seção 4.4), devolvendo a consulta conforme a busca dos metadados envolvidos na solicitação.

Os metadados/valores serão distribuídos a todos os servidores através da replicação; porém, os objetos multimídia ficarão apenas no servidor no qual foi armazenado inicialmente (selecionado pela rotina que realiza o balanceamento). Para exemplificar a arquitetura utilizada para a distribuição dos metadados, temos o seguinte caso: o Cliente 1 executa a ferramenta de manipulação dos metadados e deseja inserir

um objeto qualquer, bem como associar metadados a ele. Esse cliente faz a inserção do objeto, associa metadados/valores a ele e este objeto (bem como seus metadados) é armazenado no servidor no qual o Cliente 1 está acessando (por exemplo Servidor 1). Após inserir os metadados/valores, estes são automaticamente copiados para todos os demais servidores, executando a rotina de replicação existente na ferramenta. Se o Cliente 2 efetuar uma consulta no servidor que ele está acessando (no Servidor 2, por exemplo), que envolva metadados inseridos pelo Cliente 1 (inseridos no Servidor 1 e replicados para o Servidor 2), esses serão mostrados para o Cliente 2. Se desejar visualizar o objeto armazenado, a ferramenta carregará o objeto do Servidor 1, pois esse servidor é o que contém o objeto armazenado. Ou seja, quando um tipo de metadados, associado a um objeto, for recuperado, o objeto será carregado independente da sua localização.

Dessa maneira, os metadados/valores serão inseridos e recuperados localmente, uma vez que, através do mecanismo de replicação, um Cliente fará uma requisição sempre no servidor que ele está conectado e, quando desejar ler o objeto armazenado, esse será lido à partir do servidor que o mantém, podendo ser o servidor no qual o Cliente está conectado, ou qualquer outro existente no sistema distribuído.

4.3 Replicação de Metadados

A replicação dos metadados/valores é utilizada para aumentar a disponibilidade dos dados no Modelo Metamídia Distribuído.

Através da replicação a disponibilidade é aumentada, pois o sistema pode continuar funcionando enquanto existir ao menos um servidor disponível [4]. Melhora também o desempenho nas consultas, pois o resultado será resolvido localmente, uma vez que as bases de dados são idênticas.

A desvantagem de replicar todos os dados é que a velocidade nas atualizações será baixa, pois qualquer modificação ocorrida em algum dado, deverá ser refletida a todos os servidores, para manter os banco de dados consistentes.

O projeto de replicação envolve algumas características como controle de transações, topologia da rede, modelos de replicação de dados e o programa que faz o

gerenciamento da replicação [4]. Esses fatores auxiliam na montagem de um bom projeto de replicação, pois é através dessas características que se faz o levantamento da necessidade dos dados (se para leitura ou escrita), se é necessário o sincronismo das réplicas, se pode haver latência de atualizações e quais servidores dos metadados devem ser armazenados.

Para a replicação dos metadados do Metamídia Distribuído, foi realizado o levantamento de quais as tabelas seriam copiadas para outros servidores, compartilhando os metadados/valores, bem como, indicando em qual servidor um objeto multimídia está armazenado, para fins de permissão de modificações dos metadados/valores.

O tipo de replicação implementado neste trabalho é o *master-slave*, ou seja, quando um tipo de metadados é inserido (e associado a algum objeto), a ferramenta comanda a replicação dos metadados/valores para os demais servidores. Na arquitetura proposta nesta dissertação os metadados sempre serão replicados a todos os servidores, da seguinte maneira:

- Todos os servidores que estão disponíveis receberão os metadados. Caso algum não esteja disponível, é mantido um histórico (*log*) de quais metadados/valores não foram replicados, bem como, qual servidor não recebeu os metadados/valores. Quando o servidor não disponível voltar a funcionar, os metadados/valores serão automaticamente replicados.
- Todos os metadados adicionados/modificados em um servidor são automaticamente replicados para os demais servidores, porém os servidores que recebem os metadados/valores somente poderão ler. Dessa maneira, se um tipo de metadados precisar ser modificado, este deverá ser feito no servidor que mantém o objeto multimídia;
- A ferramenta efetua a replicação para todos os demais servidores componentes do sistema de maneira totalmente transparente ao usuário;
- No momento da replicação, o dado que é transferido, do servidor no qual foi adicionado, para os outros servidores, contém informações referentes aos

metadados, bem como, o identificador do servidor, para indicar em qual servidor o objeto relacionado a um tipo de metadados está armazenado²;

Conforme a explicação anterior, pode-se perceber que a base de dados de cada servidor ativo é consistente, uma vez que se estabelece a sincronia dos metadados no momento da inserção/modificação.

Para o processo de sincronismo, as seguintes etapas são realizadas:

- Quando um tipo de metadados/valores precisa ser replicado, a ferramenta verifica quais os servidores que estão disponíveis para receber o tipo de metadados para efetuar replicação;
- Se todos estiverem *on-line*, os metadados são então enviados a cada servidor e é solicitado o COMMIT para a confirmação da gravação;
- Caso algum servidor não esteja disponível, os metadados não são replicados para este servidor e são reservados para replicação posterior. Neste caso, a replicação será feita quando a ferramenta detectar que o servidor está disponível, fazendo uma varredura dos metadados que não foram replicados e efetuando a distribuição.

Dessa maneira, é garantido que, se os servidores sempre estiverem disponíveis, as bases de metadados estarão atualizadas. Porém, quando algum servidor falhar, ocorrerá atrasos na sincronia para este servidor, uma vez que o problema deverá ser resolvido para que os metadados possam ser replicados, garantindo a consistência de todas as bases.

4.4 Balanceamento de Carga

O balanceamento de carga computacional é uma propriedade desejável quando existem diversas máquinas acessando um único servidor, sobrecarregando-o principalmente na resposta às consultas.

² Como mencionado na Seção 4.1 apenas o metadado é replicado. O objeto permanece no local em que foi armazenado.

Dessa forma, toda requisição será direcionada a uma máquina servidora e esta, deverá ser capaz de atender a qualquer solicitação, independente do grau de complexidade envolvido na questão. Outro fator importante está no fato de que, como existe apenas uma máquina servindo a todos os clientes, se esta parar de funcionar, automaticamente todo o sistema pára.

Como mencionado, na Seção 4.3, os metadados replicados em mais de um servidor, permitem que uma pesquisa possa ser direcionada a qualquer servidor do sistema, uma vez que as bases de metadados são iguais, bem como, permite que as respostas e transações sejam gerenciadas localmente.

Conforme exposto, quando um cliente efetuar uma consulta, esta será direcionada ao servidor que estiver menos “carregado”, ou seja, com o menor número de máquinas. Na arquitetura proposta, foi criada uma rotina que mantém o histórico de quais máquinas estão conectadas a quais servidores, ou seja, a quantidade de máquinas por servidor, independente da localização.

Na ferramenta que gerencia os metadados distribuídos, quando um cliente faz a conexão, a rotina de balanceamento verifica qual servidor tem o menor número de clientes conectados. Dessa forma, toda vez que um cliente se conectar, a ferramenta Metamídia Distribuído avalia qual servidor ele poderá utilizar para gravar/consultar metadados/valores e objetos multimídia. Assim, haverá uma equalização na utilização dos servidores, pois a ferramenta é capaz de computar a quantidade de máquinas que acessam cada servidor, no momento da conexão.

Vamos ao seguinte cenário: supondo que não existam clientes conectados a nenhum servidor. No momento que um cliente efetuar uma conexão, a rotina de balanceamento é executada para verificar qual o servidor possui o menor número de clientes conectados. Sempre que ocorre empate no número de máquinas por servidor, a rotina escolhe de maneira aleatória o servidor para esse cliente conectar-se. Quando um segundo cliente conectar-se, a ferramenta novamente buscará o servidor com menor número de conexões e direciona esse cliente a esse servidor. E dessa maneira é realizada toda vez que uma máquina conectar-se.

Quando um servidor parar de funcionar, a rotina de balanceamento é novamente executada para selecionar um novo servidor para os clientes que estavam acessando o servidor que falhou.

Outro detalhe, uma vez que um cliente conecta-se em um servidor, de acordo com a arquitetura de distribuição proposta para o Modelo Metamídia, os metadados e os objetos serão adicionados no servidor no qual ele está conectado e todas as consultas são resolvidas localmente, pois a base de metadados é igual em todos os servidores (replicação síncrona). Quando uma consulta envolver metadados nos quais o dono não seja o servidor que o cliente conectou-se, o objeto será carregado a partir do servidor no qual ele se está armazenado.

Sendo assim, a ferramenta permite que, no momento em que é adicionado um novo servidor, e após ser efetuada a replicação de metadados, esse servidor começa a formar o sistema de base de dados distribuída e, portanto, poderá receber conexões de qualquer cliente, uma vez que a base de metadados/valores é igual aos demais servidores componentes.

O balanceamento implementado simplesmente faz a divisão de conexões, não preocupando-se com a complexidade de uma requisição, ou seja, basicamente garante que o número de máquinas que estão acessando um determinado servidor será igual a qualquer outro disponível, independente se as máquinas que acessam um determinado servidor, sobrecarregue-o mais que outro. Em outras palavras, significa dizer que esta rotina melhora o acesso aos servidores, pelo fato de que não haverá sobrecarga de conexões em apenas um, ou seja, ocorrerá equalização do número de acessos nos servidores existentes no sistema.

4.5 Protótipo do Metamídia Distribuído

O protótipo foi implementado, para validar a arquitetura de distribuição proposta, utilizando a linguagem de programação orientada a objetos JAVA [30], pois além de ser orientada a objetos, é multiplataforma e possui integração facilitada com ferramentas de modelagem, IDE (*Interface Development Environment*) de programação Eclipse versão 3.0.2 [8] – escolhido pela quantidade de plug-ins que a ferramenta possui, permitindo

maior facilidade de desenvolvimento do protótipo - e sistemas gerenciadores de banco de dados *open-source*, tais como, PostgreSQL [24], Firebird [12] e MySQL [21] para comprovar a flexibilidade do Modelo de Metadados Metamídia Distribuído, possibilitando o uso com qualquer um dos gerenciadores de banco de dados mencionados ou a utilização dos três em conjunto.

A Figura 11 ilustra o modelo da base de dados criado para a distribuição dos metadados entre os servidores. Este esquema criado foi implementado em cada servidor, para garantir a consistência da base de metadados, uma vez que esta deverá ser igual em todos os servidores.

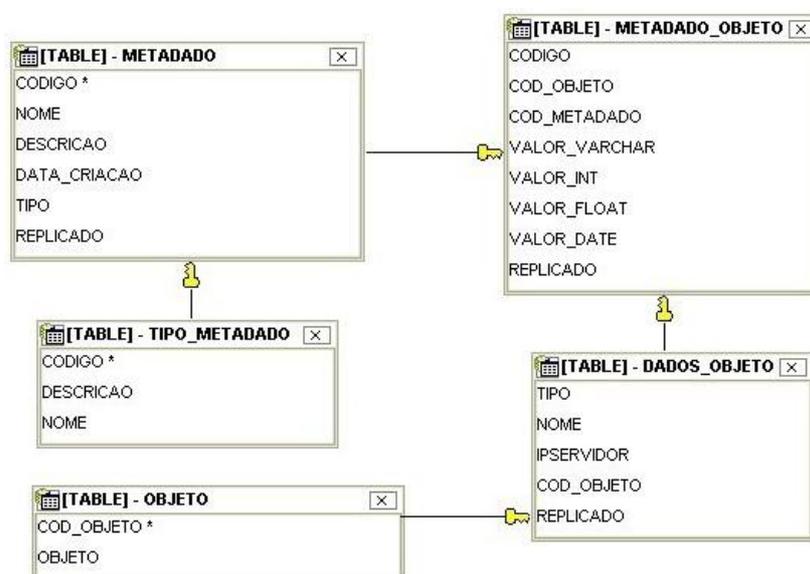


Figura 11 - Modelo da Base do Metamídia Distribuído

Conforme ilustrado no modelo da Figura 11, temos a definição das seguintes tabelas que foram criadas nas bases de dados:

- **Metadado**: tabela que conterà os metadados criados, indicando o tipo de dado a qual pertence, podendo pertencer ao tipo de dados inteiro, real, texto ou data;
- **Tipo_Metadado**: criada para indicar qual a “tipagem” (imagem, documento, vídeo, som) do metadado a ser descrito, segundo a hierarquia de metadados, para permitir a herança dos metadados;

- **Dados_Objeto:** tabela que manterá as informações básicas do objeto, como por exemplo o atributo IPSEVIDOR, que é utilizado para indicar em qual servidor o objeto está armazenado, uma vez que este não é replicado;
- **Objeto:** tabela que armazena o objeto propriamente dito e têm relação direta com a tabela Dados_Objeto;
- **Metadado_Objeto:** tabela que associa os metadados aos objetos armazenados, permitindo indicar todos os metadados que descrevem o objeto gravado.

Nas tabelas *Metadado*, *Metadado_Objeto* e *Dados_Objeto* nota-se a existência do atributo REPLICADO, que indica se o tipo de metadados foi replicado para todas as bases de dados. Supondo que existam dois servidores; se um deles falhar esse atributo indicará quais dados não foram replicados a todos os servidores. Quando o servidor falho voltar a funcionar, a ferramenta encarrega-se de efetuar a replicação dos dados/metadados ainda não replicados.

Conforme mencionado anteriormente, o protótipo criado realiza replicação dos metadados com sincronismo e balanceamento de carga, através da divisão de conexões entre os servidores disponíveis no sistema de banco de dados distribuído, permitindo que as consultas sejam resolvidas localmente.

Basicamente a ferramenta funciona da seguinte forma:

- Quando é executada, uma rotina realiza o balanceamento entre os servidores disponíveis (conforme explicado na Seção 4.4), ou seja, verifica qual dos servidores está com menos conexões e efetua o direcionamento das requisições daquele cliente para o servidor selecionado;
- Depois de realizado o balanceamento, toda solicitação será realizada no servidor que foi escolhido, independente se é a inserção de objetos ou associação de metadados/valores;
- Ao efetuar a inserção de um novo objeto multimídia, apenas os dados e os metadados associados serão replicados; o objeto permanecerá disponível apenas no servidor que foi selecionado para aquele cliente. Por exemplo, vamos supor que existam três servidores disponíveis. Quando a máquina 1

executar a ferramenta, a rotina de balanceamento verificará qual dos três servidores estão menos “carregados”. Dos servidores verificados, foi selecionado o servidor 2. Ao armazenar um objeto e associar metadados a ele, o objeto multimídia será armazenado no servidor e seus dados/metadados serão replicados para os servidores 1 e 3.

- Quando alguma máquina realizar a consulta de algum tipo de metadados, a ferramenta avalia em qual servidor o objeto selecionado na pesquisa está armazenado, para então efetuar a leitura.

A Figura 12 ilustra a tela de inclusão dos objetos multimídia e a associação de metadados ao objeto inserido. Nesta tela, é mostrado o IP do servidor no qual o cliente está conectado, de maneira que quando o usuário efetuar o cadastramento de algum objeto, este será armazenado no servidor cujo IP está mostrado na tela. Após o cadastramento de um objeto, selecionado a partir dos arquivos existentes na máquina local, os metadados poderão ser associados. Uma vez associado um metadado, automaticamente ele é replicado a todos os demais servidores, de maneira transparente ao usuário.

MANIPULAÇÃO DE OBJETOS

IP DO SERVIDOR: 10.1.1.25

TIPO DO OBJETO: IMAGEM

NOME DO OBJETO: C:\Documents and Settings\joao\Meus documentos\Esp...

Novo Editar Excluir Salvar Cancela... Consultar

TIPOS DE METADADOS: COR

VALOR DO METADADO:

Adicio... Remove

METADADO	VALOR
COR	BRANCO

Informações do objeto (Tipo: Documento, vídeo, som, imagem)

Metadados associados ao objeto

Figura 12 - Tela de cadastro de Objetos e Associação de Metadados

Nota-se que, como as bases de dados são idênticas (realizado através da replicação), as consultas serão sempre executadas localmente. Somente quando for carregar o objeto, faz-se a verificação do local onde está armazenado. Por exemplo, a máquina 1 (que está conectada no servidor 2) ao realizar uma consulta, a resposta obtida contém objetos armazenados no servidor 3. Se a máquina 1 desejar visualizar algum objeto (um arquivo de som, por exemplo), esse será carregado do servidor que o mantém (no caso, o servidor 3).

Com a distribuição dos metadados indexados pelo Modelo Metamídia, podemos ter um número maior de servidores que respondem às requisições. Isso se torna uma característica desejável, pois a disponibilidade dos dados é maior, e, além da velocidade que as respostas terão ser maior (tempo de resposta menor), podemos indexar qualquer objeto multimídia, através de um sistema tolerante a falhas.

Imaginemos o seguinte cenário: executar a ferramenta, efetuar o balanceamento. Quando for armazenar algum objeto, o servidor não está disponível. A ferramenta executa novamente a rotina de balanceamento de forma totalmente transparente para o usuário, uma vez que as bases de dados dos servidores são idênticas, permitindo que o acesso ao metadados não seja comprometido.

No cenário descrito acima, a replicação será realizada para todos os servidores que estiverem disponíveis, ou seja, se algum estiver falho, os dados/metadados não serão replicados somente para este servidor. Quando o servidor falho voltar a funcionar, a ferramenta seleciona algum servidor (pois as bases de dados são idênticas) que replicará os dados/metadados para o servidor que entrou em atividade, garantindo o Protocolo de Transações em Duas Fases, empregado na rotina de replicação criada (Seção 4.3). Essa rotina de replicação de servidores é executada sempre que ocorre alguma falha em algum dos servidores, para garantir a consistência das bases de metadados distribuídos.

A ferramenta possui algumas restrições, para evitar a inconsistência dos dados/metadados armazenados. Quando alguma máquina efetua uma consulta de algum tipo de metadados e obtém algum objeto como resposta, o usuário somente poderá associar algum novo tipo de metadados e os valores, se o servidor no qual ele encontra-se conectado for o servidor que mantém o objeto. Caso o “dono” do objeto seja outro servidor, a ferramenta não permite a associação de novos metadados.

A Figura 13 ilustra a tela de visualização de metadados e objetos multimídia cadastrados. Os metadados são colocados em uma árvore, para facilitar a organização e a manipulação dos metadados/valores. São ilustrados os elementos que comprovam a distribuição dos metadados em mais de um servidor. São eles: o servidor no qual a ferramenta está conectada e o servidor no qual o objeto está armazenado. Dessa forma, fica comprovado que os metadados estão em mais de um servidor.

A árvore ilustrada na figura possui 4 níveis: o primeiro nível indica o nome do tipo de metadados, o segundo nível indica o valor associado aos metadados, o terceiro nível indica o tipo do objeto armazenado e o último mostra o nome do objeto armazenado.

Na tela ilustrada nessa figura, a árvore de metadados e os valores mostrados, são carregados do servidor no qual a ferramenta está conectada. Se desejar visualizar o objeto, basta clicar no botão “Visualizar Objeto” que o objeto será carregado de duas formas: se for imagem, será mostrada na tela, se for outro tipo de mídia, será gravado em disco para facilitar a abertura do objeto.

Nota-se na Figura 13 que os servidores são diferentes e, portanto, o botão “Adicionar Metadados” encontra-se desabilitado, impedindo que sejam associados metadados se o servidor conectado não é o servidor que armazena o objeto manipulado.

De 15/07/2005 a 05/09/2005 - 2ª a 6ª feira
 2) LOCAL E HORÁRIO DAS INSCRIÇÕES
 Departamento de Pós-Graduação do CEFET – Unidade de Ponta Grossa
 3) NÚMERO DE VAGAS
 Máximo: 33 (trinta e três)
 4) DOCUMENTAÇÃO NECESSÁRIA
 Preenchimento do formulário de inscrição (disponível em:
<http://www.pg.cefetpr.br/cont/espcco/>)
 Fotocópia dos seguintes documentos:
 Diploma ou certificado de conclusão do curso de graduação;
 Histórico escolar da graduação;
 Documento de identidade;
 Currículo-Vitae;
 Comprovante de pagamento da taxa de inscrição no valor de R\$ 30,00
 5) CONDIÇÕES PARA INSCRIÇÃO
 Ser formado em Curso de Graduação na área de Informática ou outro curso desde
 que possua experiência profissional comprovada na área de Informática.
 6) DATAS IMPORTANTES

Evento	Data
Inscrição de candidatos	15/07/2005 a 05/08/2005
Resultado da classificação dos alunos	12/08/2005
Registro dos alunos	15 a 18/08/2005
Início das atividades letivas	29/08/2005
Término das atividades letivas	27/02/2006
Data limite para entrega do trabalho conclusivo	18/08/2006
Data de Apresentação dos Trabalhos	30/08/2006

8) INVESTIMENTO
 Matrícula: R\$ 300,00
 Taxa de inscrição: R\$ 300,00

Figura 13 - Visualização de Metadados e Objetos

Outra característica implementada na ferramenta é a possibilidade da indicação e novos servidores de metadados. Quando um novo servidor entra em atividade, a ferramenta seleciona um servidor para efetuar a replicação de seus metadados para o novo, uma vez que a base de dados do novo servidor estará vazia. Dessa maneira após replicados todos os metadados, esse servidor formará parte do sistema de banco de dados distribuídos, podendo receber conexões vindas de qualquer máquina, pois a base de dados estará igual a de todos os servidores.

4.6 Considerações Finais do Capítulo

Neste capítulo, foi abordado o Metamídia Distribuído, destacando as características implementadas como a arquitetura do modelo proposto; a distribuição dos metadados indexados pelo Metamídia; o funcionamento da replicação dos metadados, justificando a escolha da replicação; o balanceamento de carga, que permite a divisão de conexões entre os servidores disponíveis e o protótipo do Metamídia Distribuído propriamente dito, que é o responsável pelas características implementadas para a distribuição dos metadados indexados.

5 TRABALHOS RELACIONADOS

Nesta seção são apresentados dois trabalhos relacionados, dando ênfase ao processo de distribuição de metadados, procurando mostrar os principais pontos abordados por cada trabalho estudado, dentro do contexto de distribuição de metadados. As técnicas e as formas de distribuição são comentadas conforme abordagens dos autores.

5.1 Sistema MDV

Conforme a referência [17], a arquitetura do MDV (*Distributed Metadata Management System*), um sistema de gerência distribuído de metadados, é formada por três camadas e dá suporte a uma área temporária de dados e a replicação na camada mediana, de modo que as perguntas possam ser avaliadas localmente. Os usuários e as aplicações especificam a informação necessária usando uma linguagem especializada. Isso reduz a quantidade de dados que têm que ser perguntada localmente tendo por resultado um desempenho melhor da execução da consulta.

O MDV executa um algoritmo para iniciar a replicação dos dados, o qual pode ser executado usando uma linguagem padrão da base de dados relacional para efetuar a pesquisa. Segundo o autor, como a arquitetura é em três camadas, facilita o ajuste às variações de chamadas de trabalho e aos pontos de atividade mais utilizados.

O acesso eficiente dos metadados usando uma área temporária de armazenamento e um mecanismo para publicar/subscrever resultam na preservação da consistência dos dados na área temporária. MDV usa RDF (*Resource Description Framework*) como seu modelo dos dados, que usa a sintaxe de XML para documentos e o esquema de RDF para definição do esquema de metadados.

A Figura 14 [17] ilustra a arquitetura do sistema MDV, consistindo de fornecedores de metadados, os repositórios locais de metadados e os clientes do MDV. Os fornecedores de metadados são distribuídos sobre a Internet para fornecer um acesso

uniforme a respeito da latência da rede e do índice dos metadados. Os fornecedores de metadados realizam compartilhamento do mesmo esquema e replicação consistente dos metadados entre si. Todos os metadados armazenados em fornecedores de metadados são considerados como globais e publicamente disponíveis.

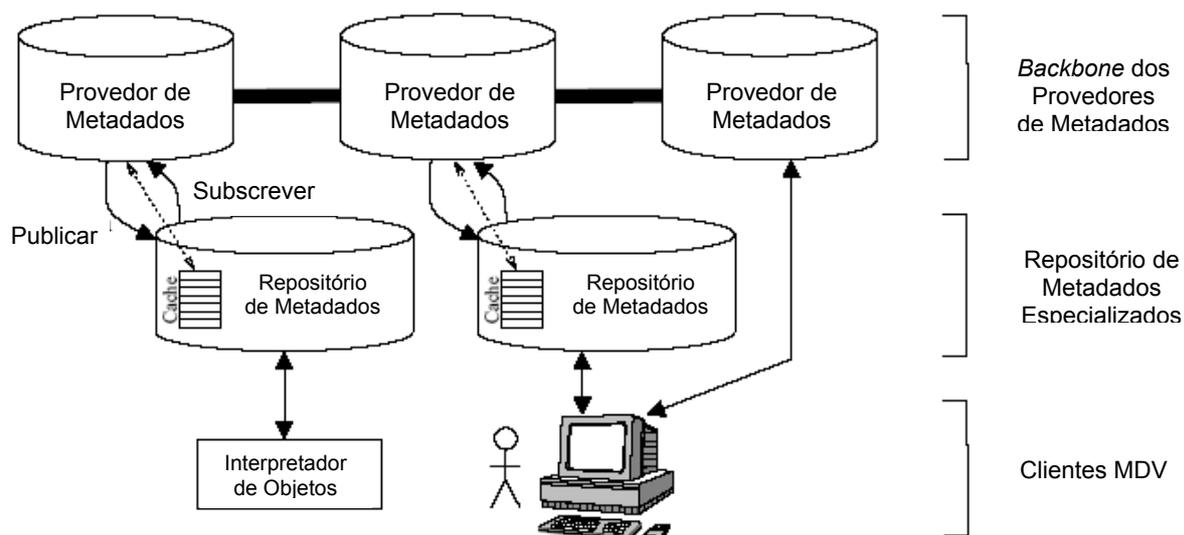


Figura 14 - Visão da arquitetura MDV

Os repositórios locais de metadados são os componentes do sistema MDV que processam a consulta de metadados. Conseqüentemente, repositórios funcionam perto das aplicações que perguntam os metadados, por exemplo, na mesma rede. A área de armazenamento temporária de metadados deve conter os relevantes, apropriados aos usuários ou às aplicações usadas. Os repositórios locais de metadados usam o mecanismo de publicação e subscrição para buscar metadados relevantes de um provedor e para receber atualizações de seus dados, isto é, para manter suas áreas de armazenamento consistentes. O provedor de metadados usa regras de subscrição do uso de provedores de metadados para publicar atualizações, inserções ou exclusões de metadados dos repositórios locais de metadados. Além dos metadados globais, o armazenamento dos metadados locais não deve ser acessível ao público e conseqüentemente não é enviado à rede [17].

As aplicações e os usuários que acessam o sistema MDV são referenciados como clientes. Os clientes podem pesquisar um tipo de metadados em um LMR usando a

linguagem de pergunta (declarativa) de MDV's que não foi apresentada no artigo. A administração dos metadados, ou seja, registrar, atualizar ou excluir metadados; é feita em provedores de metadados. Os novos metadados devem ser registrados dentro de um documento válido de RDF. A exclusão pode ser realizada removendo as partes de um documento e atualizando ou removendo o documento completo com todo seu índice. Esta é a única maneira para adicionar, atualizar ou excluir um tipo de metadados. A linguagem de pergunta de MDV's não fornece nenhuma funcionalidade de atualização nem exclusão.

MDV é implementado em Java, de modo que seja portátil, permitindo a instalação com esforço muito pequeno, e usa um sistema de gerência da base de dados relacional para armazenamento de dados básicos.

5.2 Sistema *Lazy Hybrid*

O Sistema *Lazy Hybrid* [26] corresponde a uma nova arquitetura da gerência de metadados, projetada para fornecer alto desempenho e escalabilidade.

Tradicionalmente, os metadados e os dados são controlados pelo mesmo sistema de arquivo, na mesma máquina, e armazenados no mesmo dispositivo. Para a eficiência, o tipo de metadados é armazenado freqüentemente fisicamente perto dos dados que descreve. Em alguns sistemas distribuídos modernos, os dados são armazenados nos dispositivos que podem diretamente ser alcançados através da rede.

Os sistemas de armazenamento baseado em objetos separam a gerência dos dados e dos metadados como ilustrado na Figura 15. Os dispositivos de armazenamento semi-independentes baseados em objetos controlam tarefas de baixo nível do armazenamento de dados, tais como, programar o pedido e disposição de dados e apresentar uma relação baseada em objeto simples do acesso dos dados aos arquivos do sistema. Um conjunto separado dos metadados controla a hierarquia do diretório, permissões de arquivos e do diretório, reservando para transferências simultâneas e eficiente de dados entre um grande número clientes e dispositivos de armazenamento semi-independentes baseados em objetos [26].

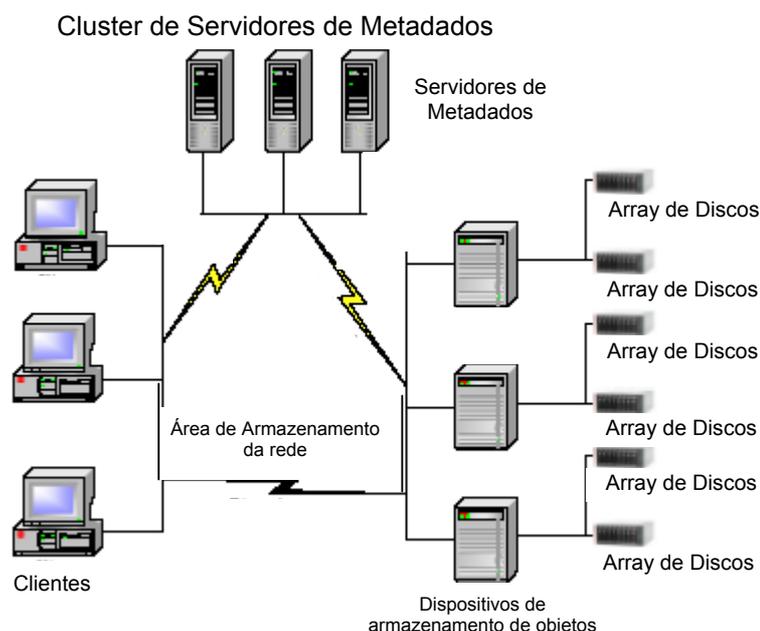


Figura 15 - Arquitetura do sistema de armazenamento

O *Lazy Hybrid* dirige-se aos problemas de usar uma combinação de *hashing* dos diretórios hierárquicos, da re-locação dos metadados e das listas atualizadas de controle de acesso. A posição dos metadados em arquivos individuais é determinada por uma associação do nome do caminho, nos metadados distribuídos e em evitar "pontos de gargalo" onde uma porcentagem desproporcional de pedidos ativos dos metadados é de um único servidor [26].

No *hashing*, os nomes dos caminhos permitem o acesso direto aos metadados do arquivo sem envolver todos os servidores de metadados, os quais armazenam diretórios ao longo do caminho. Os diretórios hierárquicos são mantidos a fim fornecer a semântica e operações padrão do diretório. As políticas de atualização permitem atualizações eficientes dos metadados quando o arquivo ou os nomes ou as permissões de diretório são mudados ou quando os servidores dos metadados são adicionados ou removidos do sistema.

O *Lazy Hybrid* foi projetado com os seguintes objetivos [26]:

- **Alto desempenho:** o conjunto do servidor de metadados deve fornecer acesso muito rápido aos metadados. Um pedido típico deve envolver uma única mensagem a um único servidor de metadados. O conjunto do servidor de metadados deve:

- Dar suporte a um grande número de acessos paralelos a um único arquivo, diretório ou subdiretório;
- Ter controle eficiente dos diretórios muito grandes que contêm dez dos milhares de arquivos, dividindo os múltiplos servidores de metadados dos servidores para evitar sobrecarga.
- **Escalabilidade:** o desempenho dos metadados deve escalar com o número de servidores do metadados no conjunto;
- **Flexibilidade:** o conjunto de servidores dos metadados deve suportar eficientemente a alteração do nome do diretório, as mudanças da permissão dos arquivos e do diretório e a adição, a remoção, e a alteração de servidores de metadados;
- **Conformidade dos padrões:** fornece uma relação padrão que faça o sistema do armazenamento útil em sistemas existentes.

O *Lazy Hybrid* usa *hashing* para distribuir os metadados através do conjunto de servidores. Entretanto, mantém a hierarquia de diretórios para dar suporte à semântica para permissão de acesso.

Para acessar os dados, os clientes buscam o nome do caminho do arquivo para produzir o valor que o servidor de metadados está procurando. O cliente contata então o servidor de metadados apropriado para abrir o arquivo e obter o objeto do arquivo. O resultado do acesso é extremamente eficiente, envolvendo tipicamente uma única mensagem a um único servidor de metadados.

Quando um servidor de metadados é contatado, uma de três situações pode ocorrer [26]. Primeira, o tipo de metadados existe no servidor e o cliente tem a permissão fazer a operação pedida. Nesse caso, a operação é terminada e é dada ao cliente a informação necessária para obter os dados do arquivo dos dispositivos de armazenamento semi-independentes baseados em objetos. A segunda possibilidade é que o tipo de metadados existe no servidor e o cliente não tem a permissão de fazer a operação solicitada. Nesse caso, uma resposta é emitida ao cliente que indica que a operação solicitada não é permitida. A terceira possibilidade é que o tipo de metadados não existe. Nesse caso, uma resposta apropriada é emitida ao cliente.

5.3 Goulart

O servidor de metadados para vídeo [15], corresponde a um sistema que tem por característica, proporcionar o gerenciamento de dados multimídia (vídeos) armazenados em um servidor, procurando prover acesso eficiente através da utilização de um repositório de metadados.

Na implementação, utiliza a Linguagem de Programação JAVA, executada sobre uma rede TCP/IP de 10 Mbits e a base de metadados armazenada no sistema gerenciador de banco de dados PostgreSQL versão 6.2.1 [15] [24].

Utilizou-se a linguagem JAVA, por ser orientada a objetos e ser multiplataforma, facilitando a expansão e utilização em qualquer que seja o sistema operacional, não causando dependência nem de *hardware* quanto de *software* [15]. O sistema gerenciador de banco de dados escolhido foi o PostgreSQL [24], por ser de domínio público e incorporar diversas características que o tornam flexível, como por exemplo a existência de *driver* JDBC (*Java Database Connectivity*) para permitir acesso à base através a linguagem.

Os metadados estão centralizados em uma única máquina, permitindo que as requisições sejam direcionadas de forma objetiva.

O sistema é composto de um servidor de dados multimídia, por um servidor de metadados centralizado e por clientes. O servidor de metadados e o servidor de dados multimídia são dois módulos distintos, implementados no software, podendo estar fisicamente na mesma máquina ou não e o acesso é feito através da *Web* [15].

O Módulo Cliente é responsável por facilitar o acesso do usuário aos metadados, através de uma *applet* criada em JAVA, inserida em uma página HTML e acessada através de navegador [15]. É através desse módulo que uma consulta (simples – como pesquisar um nome – ou apresentação de um vídeo) é enviada para o servidor de metadados e são exibidos os resultados no navegador.

Para a apresentação dos vídeos, foi implementado um *player* utilizando a ferramenta *Java Media Framework* (JMF), provendo facilidades na apresentação de vídeos na *Web*. A carga do vídeo é realizada por *streaming vídeo*, provendo rapidez na resposta e amostragem [15].

O servidor de metadados (implementado como uma aplicação) é responsável em receber as solicitações dos clientes, fazer o tratamento, realizar a consulta e enviar a resposta. Além do gerenciamento dos metadados, o servidor de metadados faz o gerenciamento da base de dados. O servidor fica aguardando que um cliente estabeleça uma conexão (realizada por intermédio do navegador que interpreta a *applet*) e quando é realizada, o servidor cria uma *thread* para poder gerenciar as diversas requisições realizadas por vários clientes acessando ao mesmo tempo [15].

5.4 SPRING

O SPRING é um Sistema de Informações Geográficas, contendo funções de *processamento de imagens, análise espacial, modelagem numérica de terreno e consulta a bancos de dados espaciais* [28].

O ambiente SPRING foi proposto visando unificar o tratamento de imagens de Sensoriamento Remoto (ópticas e microondas), mapas temáticos, mapas cadastrais, redes e modelos numéricos de terreno [29]. Ele é baseado em um modelo de dados orientado a objetos, o SPRING combina as idéias de "campos" e "objetos geográficos", atingindo grandes resultados em integração de Dados [28].

O SPRING conta com duas características importantes:

- ✓ Ambiente multiplataforma: permitindo a compatibilidade com as diversas versões do *UNIX* e *Windows*;
- ✓ Interface amigável: desenvolvendo uma interface simples, permitindo uma fácil utilização pelos usuários.

A primeira versão do SPRING, liberada para a utilização de entidades externas ao INPE, foi concluída em dezembro de 1994, já permitindo a realização de estudos ambientais importantes [28] e, atualmente, encontra-se na Versão 4.1.

O banco de dados geográfico [5] construído pelo SPRING implementa uma arquitetura *Dual* onde as representações dos dados espaciais e as informações descritivas (dados não espaciais) são armazenadas em ambientes diferentes [29] e permite que o usuário faça a opção por um formato de arquivo a ser utilizado para

armazenar as informações. Na última versão estão disponibilizados os formatos *DBase*, *Access*, *Oracle* e *MySQL*.

5.5 Sistema OVID

Oomoto e Tanaka projetaram o sistema OVID (*Object-oriented Video Information Database*) [22] [32] que trata informação do tipo vídeo. A informação pode ser recuperada através de um identificador e por sentenças condicionais.

Essas recuperações são efetuadas a partir de textos, que são descrições (pares de atributos/valores) associadas a cenas significativas de vídeo. As descrições de cenas de vídeo ou objetos do tipo vídeo são subdivididas em objetos menores e esses herdam as descrições do objeto original. Um objeto do tipo vídeo também pode ser valor de um atributo em um outro objeto. Assim, cenas de vídeo podem ser descritas mais naturalmente através de outra cena de vídeo. As descrições são definidas durante o processo de codificação e estruturação. Essas descrições são armazenadas em listas.

A consulta e recuperação dos objetos do tipo vídeo são feitas através de uma linguagem de consulta chamada VideoSQL. O resultado da consulta é uma coleção de objetos que satisfazem uma condição especificada. Antes das consultas passarem pelo processo de avaliação; os objetos são avaliados com base no mecanismo de herança de intervalos de vídeo [32].

O processo de herança, deste sistema, é diferente dos sistemas orientados a objetos. A herança de inclusão de intervalos ocorre entre instâncias e não entre classes. Os objetos do tipo vídeo pertencem a uma mesma classe.

A herança de inclusão de intervalos permite que informações de um objeto A sejam herdadas por um objeto B, desde que os intervalos de B sejam definidos sobre alguma porção dos intervalos de A e os atributos estejam incluídos em um conjunto de atributos herdáveis. Esse conjunto de atributos é definido pelo usuário.

O modelo OVID [22] foi especificado formalmente [32] na linguagem de especificação MooZ (*Modular object-oriented Z*) [19], onde foram expostos de maneira mais clara os conceitos apresentados neste modelo. Na especificação original de OVID

[32], foi considerado que conjuntos de intervalos também podem ser valores de atributos. No entanto, ao especificar o modelo de dados OVID, chegou-se a conclusão que a especificação original está inconsistente ou então incompleta, pois: (i) nos exemplos apresentados não há menção de intervalo como valor de atributo; (ii) intervalos já são componentes de objetos e assim não há sentido em colocá-los como valores de atributo [32].

A indexação efetuada de forma manual, como ocorre em OVID, pode ter diferentes imagens armazenadas, que deveriam possuir a mesma descrição, porém, as descrições podem ser totalmente diferentes. O problema é que não existem definições claras de como cada imagem deve ser descrita. No sistema OVID não existe um modelo de indexação a ser seguido, tornando o trabalho tedioso e podendo resultar em descrições errôneas [32].

No sistema OVID, a herança é bastante adequada, pois ajuda o usuário na navegação em objetos, indo de níveis abstratos para detalhes de cenas, de forma gradual.

A consulta em texto, apresentada por OVID, é adequada apenas se as informações a respeito da imagem de vídeo forem necessárias, porém inadequada se for necessário pesquisar um conjunto de imagens diferentes, onde as mesmas, ou similares a estas, devem ter a mesma descrição. Pode ocorrer que essas imagens tenham descrições totalmente diferentes [32], o que dificultaria na recuperação.

5.6 Análise Comparativa

Os critérios utilizados para comparar cada uma das ferramentas estão numerados conforme segue:

- 1) **Tipos de mídia:** Saber quais as mídias que a ferramenta manipula;
- 2) **Distribuição de metadados:** Saber se a ferramenta trabalha com metadados distribuídos;
- 3) **Balanceamento de carga computacional:** Verificar se existe equalização no uso dos servidores;

- 4) **Ferramenta baseada na Web:** Verificar se a ferramenta pode ser acessada via Internet;
- 5) **Multiplataforma:** Validar a utilidade da ferramenta em diversos ambientes;
- 6) **Sistemas Gerenciadores de Banco de Dados utilizados:** Levantar quais os sistemas dão suporte ao gerenciamento dos metadados manipulados pelas ferramentas;
- 7) **Alteração de metadados distribuídos:** Verificar se a ferramenta permite que os metadados possam ser modificados;
- 8) **Foco de utilização da ferramenta:** Verificar as áreas do conhecimento que podem fazer uso da ferramenta;
- 9) **Sincronismo dos metadados:** Indicar se a ferramenta utilizada dá suporte à sincronia dos metadados.

Tabela 2 - Comparativo entre ferramentas estudadas e Metamídia Distribuído

TRABALHOS	1	2	3	4	5	6	7	8	9
METAMÍDIA DISTRIBUÍDO	Vídeo Som Imagem Texto	SIM	SIM	NÃO	SIM	MySQL PostgreSQL Firebird	SIM	Qualquer, basta adicionar os metadados	SIM
METAMÍDIA CENTRALIZADO	Vídeo Som Imagem Texto	NÃO	NÃO	NÃO	SIM	Oracle	SIM	Qualquer, basta adicionar os metadados	NÃO
LAZY HYBRID	*	SIM	NÃO	NÃO	SIM	*	SIM	*	SIM
MDV	Texto	SIM	NÃO	SIM	SIM	*	SIM	Documentos	NÃO
Goulart (**)	Vídeo	NÃO	NÃO	SIM	SIM	PostgreSQL	SIM	Vídeo	NÃO
SPRING	Imagem	NÃO	NÃO	SIM	SIM	XBASE ACCESS MySQL Oracle	SIM	Informações Geográficas	NÃO
OVID	Vídeo	NÃO	NÃO	NÃO	NÃO	*	SIM	Vídeo	NÃO

As linhas da Tabela 2 correspondem aos trabalhos analisados, enquanto que as colunas, os atributos que foram considerados relevantes no estudo. O (*) que aparece na Tabela 2, corresponde ao fato de não ter sido mencionado na literatura analisada sobre o atributo pesquisado. O (**) ilustrado, o trabalho nomeado utilizando o criador, pois não existe um nome definido para a ferramenta criada. A seguir, são feitas algumas considerações a respeito de cada item citado.

Em geral, os sistemas estudados tratam da recuperação de informação do tipo imagem. Os sistemas Goulart e OVID tratam informações do tipo vídeo e o Sistema

SPRING trata imagem através de reconhecimento de fotos provenientes de satélite. Já o *Metamídia Centralizado* e o *Metamídia Distribuído* não fazem distinção entre os vários tipos de mídia. A hierarquia de metadados/valores pode ser usada e o tratamento é dado a qualquer tipo de mídia.

Dos trabalhos analisados, apenas o *Lazy Hybrid* e MDV fazem a distribuição dos metadados utilizados. Os demais não efetuam distribuição e a ferramenta SPRING, como faz reconhecimento de imagens, não utiliza metadados para descrever as imagens armazenadas. No *Metamídia Distribuído*, a distribuição é utilizada para permitir o balanceamento computacional, característica que determina uma melhora de desempenho na utilização do modelo, uma vez que existe uma equalização na quantidade de máquinas que usam um servidor e disponibilidade dos metadados existentes.

Outra característica que merece destaque é com relação à *alteração dos metadados criados por outro* servidor. No *Metamídia Distribuído*, os metadados poderão ser alterados/adicionados, porém, somente se a conexão for realizada no servidor que armazena o objeto a ser alterado o tipo de metadados/valores. Dessa forma, somente poderá alterar/adicionar metadados a algum objeto se a máquina estiver conectada no servidor que é o “dono” do objeto. Caso contrário, não poderá realizar a modificação.

Basicamente, o foco de utilização das ferramentas é ou somente informações geográficas (SPRING), vídeos (Goulart e OVID) ou documentos (MDV). Tanto o Modelo Metamídia Centralizado como o Modelo Metamídia Distribuído não fazem distinção do tipo de aplicação que poderá ser utilizada a ferramenta que implementa o modelo. Outro detalhe, não se faz necessária a utilização de um padrão específico de metadados (como W3C ou *Dublin Core*), ou seja, poderá ser utilizado qualquer padrão existente, seja para bibliotecas digitais, informações geográficas, etc; dependendo apenas do que pretende-se indexar.

Outro ponto interessante é com relação à facilidade de utilização da ferramenta, como por exemplo, a disponibilidade do uso de navegadores para acessar através da *Web*. Os trabalhos que permitem essa utilização são MDV, Goulart e SPRING. O *Metamídia Distribuído*, por ter sido implementado em JAVA, bastam alguns ajustes para

que permita a manipulação dos metadados na Internet através da criação de páginas JSP (*Java Server Pages*), uma vez que as classes principais já estão criadas.

O Metamídia Distribuído caracteriza-se por realizar o sincronismo dos metadados, permitindo que, quando uma estação efetuar uma pesquisa na base de metadados – que estará idêntica em todos os servidores – a resposta contará inclusive com metadados que acabaram de ser adicionados. O Metamídia Distribuído faz o sincronismo dos metadados baseado em um controle *master/slave*, ou seja, quando algum servidor recebe um novo objeto ou um novo tipo de metadados foi associado, a ferramenta envia os metadados a todos os demais servidores, para garantir a igualdade das bases em todos os servidores. Outro trabalho que possui sincronia de metadados é o *Lazy Hybrid*, que a implementa através de um mecanismo de *hashing*, permitindo que as atualizações sejam eficientes dos metadados quando os arquivos ou os nomes sofrem mudanças ou quando os servidores dos metadados são adicionados ou removidos do sistema.

5.7 Considerações Finais do Capítulo

Neste capítulo foi feito um levantamento das características de trabalhos relacionados a distribuição de metadados, verificando as propriedades implementadas em cada trabalho estudado. Em seguida, foi realizada uma análise comparativa dos trabalhos com o Metamídia Distribuído proposto, tomando por base algumas características desejáveis em sistemas de distribuição de metadados, destacando: os tipos de mídia suportados pelas ferramentas, o balanceamento de carga para divisão de conexões entre servidores, os sistemas gerenciadores de bases de dados utilizados para indexação dos metadados e o sincronismo dos metadados nos servidores componentes do sistema.

6 CONCLUSÕES E TRABALHOS FUTUROS

Os dados no formato texto são simples de armazenar e recuperar. Porém, os dados não convencionais (imagem, som, vídeo), estão ganhando espaço nos sistemas, exigindo maior flexibilidade, proporcionando o surgimento de técnicas que se aproximem das pesquisas dos usuários de sistemas multimídia (pesquisa baseada em conteúdo, por exemplo).

Algumas técnicas destacam-se por usarem a indexação convencional de maneira estendida para banco de dados multimídia, reconhecendo o dado como um objeto complexo. Sendo assim, todo objeto armazenado no banco de dados multimídia deve ser descrito de modo que o mesmo possa ser facilmente localizado e recuperado. No contexto deste trabalho, a indexação foi usada para determinar quais metadados devem ser armazenados para descrever os objetos multimídia e permitir a pesquisa e recuperação dos mesmos.

Quando tratamos da recuperação de dados, as pesquisas que os usuários de banco de dados convencionais realizam, são totalmente focadas no contexto, enquanto que os usuários que manipulam bases de dados de imagens, vídeos e som, procuram relacionar o conteúdo do objeto na pesquisa, tornando a tarefa difícil de ser realizada se não existirem modelos que façam a indexação dos objetos multimídia quando são armazenados.

O Modelo Metamídia, utilizado neste estudo, foi projetado para um ambiente centralizado cliente-servidor, e, como os metadados utilizados na indexação ficarão armazenados em uma máquina, a recuperação poderá ser comprometida, pois apenas uma máquina será utilizada para atender todas as requisições.

O objetivo principal deste trabalho foi apresentar um modelo de distribuição para o Modelo Metamídia, que por tratar de dados multimídia, seu acesso pode comprometer o desempenho do sistema. Portanto, a solução aplicada, foi através da distribuição dos metadados em vários servidores, aumentando a tolerância à falhas, disponibilidade e permitindo o balanceamento de carga computacional.

Foi criada uma ferramenta para validar o modelo proposto, permitindo uma extensão do Modelo de Metadados Metamídia Centralizado, contando com um alto grau de disponibilidade dos metadados através da replicação síncrona entre os servidores.

Na ferramenta criada, foi implementado um mecanismo de balanceamento de carga, que permite a distribuição de conexões entre servidores, garantindo que o sistema não sobrecarregue apenas um servidor.

Através do sincronismo implementado, os metadados tornam-se disponíveis instantaneamente, ou seja, no momento que uma máquina efetua o armazenamento de um objeto (imagem, som, texto ou vídeo) e associa metadados a ele, esses são automaticamente replicados para os demais servidores que compõe o sistema de banco de dados distribuído.

Dessa maneira, tem-se um Modelo de indexação e recuperação de metadados distribuídos flexível, independente do tipo de mídia, independente de padrão de metadados, independente de sistema gerenciador de banco de dados, uma vez que a ferramenta criada faz a replicação, o balanceamento e a sincronia dos metadados.

Enquanto o Modelo Metamídia Centralizado somente efetua a indexação dos metadados e permite a associação com objetos multimídia, armazenando-os em uma máquina, o Modelo Metamídia Distribuído permite a distribuição dos metadados indexados, tornando o sistema mais rápido na resposta e permitindo incremento de máquinas e independência de plataforma.

6.1 Contribuições

Das contribuições do trabalho realizado, podemos citar:

- Aplicação de meta-modelo para indexação e recuperação de informações multimídia;
- Criação do Modelo de Distribuição do Metamídia;
- Criação de ferramenta que implementa e valida o modelo proposto;
- Implementação de mecanismo de sincronia de metadados distribuídos;

- Implementação de rotina para balanceamento de carga computacional.

Pode-se observar quando foram realizados os testes da ferramenta que, a rotina criada para replicar os metadados, bem como, efetuar a sincronia entre os servidores garantiu que o sistema pudesse contar com metadados distribuídos em diversos locais, facilitando o trabalho local dos servidores e permitindo disponibilidade dos dados/metadados.

Outro ponto importante no estudo realizado é com relação a rotina que faz o balanceamento. Esta sempre é executada quando um dos servidores parar de funcionar (simulação realizada para validar a rotina). Neste caso a ferramenta realiza um novo balanceamento, migrando as máquinas que estavam conectadas no servidor falho, para outro servidor que estivesse disponível. Desta maneira, se qualquer servidor falhar (independente do tipo da falha), a ferramenta faz com que as estações acessem um novo servidor.

Como o trabalho realizado foi a distribuição do Modelo de Metadados Metamídia, pode-se perceber uma validação importante para a área de informática, que é a utilização de um modelo de indexação/recuperação de metadados que não faz distinção do objeto multimídia a ser manipulado, nem tampouco se faz necessária a utilização de um padrão de metadados, muito menos utilizar algum sistema gerenciador de banco de dados específico, tornando a ferramenta distribuída flexível o suficiente para ser utilizada em qualquer situação/ambiente, seja para indexação de imagens médicas, imagens de satélite, bibliotecas digitais, servidores de vídeos, etc.

6.2. Trabalhos Futuros

Como perspectivas de trabalhos futuros, pretende-se estender o modelo proposto para utilização de *clustering*, permitindo a divisão do trabalho em várias máquinas, diminuindo o tempo de resposta do sistema, uma vez que existirão mais máquinas processando.

Pretende-se também, criar uma ferramenta para *Web*, possibilitando que qualquer máquina possa utilizar o mecanismo de indexação através de navegadores, permitindo maior flexibilidade no uso da ferramenta.

Outro ponto importante para extensão da ferramenta é a associação de novos tipos de metadados e valores a algum objeto, independente do servidor no qual a máquina está conectada. A ferramenta atual somente permite a associação de novos valores, se o servidor que a máquina encontra-se conectada for o “dono” do objeto. Uma possível melhoria da ferramenta seria permitir que os usuários associem novos metadados/valores a algum objeto e, se o servidor no qual a ferramenta está conectada não for o mesmo que armazena o objeto, criar um módulo de avaliação dos metadados propostos, ou seja, algum usuário será responsável em validar a real necessidade da associação realizada por algum usuário. Dessa forma, um tipo de metadados somente será associado se o avaliador verificar que a associação é necessária.

REFERÊNCIAS

- [1] S. Abiteboul; R. Hull; V. Vianu. **Foundations of databases**. Reading: Addison-Wesley, 1995.
- [2] G. Booch; J. Rumbaugh; I. Jacobson. **UML: Guia do Usuário**. Rio de Janeiro: Campus, 2000.
- [3] K. Bohm; T. C. Rakow. **Metadata for Multimedia Documents**. *SIGMOD Record*, 23(4), 1994, p. 21-26
- [4] C. H. M. Braga. **Metamodelo para Controle de Estratégias Assíncronas de Replicação de Dados**. Dissertação de Mestrado. São Paulo, 2001. Setor de Ciências Exatas. Universidade de São Paulo.
- [5] G. Camara. **Desenvolvimento de Sistemas de Informações Geográficas no Brasil: Desafios e Oportunidades**. Semana de Geoprocessamento do Rio de Janeiro. Rio de Janeiro - RJ, 1996.
- [6] G. F. Coulouris; J. Dollimore; T. Kindberg. **Distributed systems: concepts and design**. 3rd ed. Harlow: Addison-Wesley, 2001
- [7] C. J. Date. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2000.
- [8] ECLIPSE. Disponível em www.eclipse.org, acesso em 06 de julho de 2005.
- [9] R. Elmasri; S. B. Navathe. **Fundamentals of Database Systems**. The Benjaming Cummings Publishing Company, Inc, 1994.

- [10] J. E. Ferreira; M. Finger. **Controle de Concorrência e Distribuição de Dados: a teoria clássica, suas limitações e extensões modernas**. 12^a Escola de Computação, IME-USP, 2000.
- [11] A. L. C. Freitas. **Proposta de um Sistema de Banco de Dados Distribuído e Replicado Utilizando Serviço RMI-Java**. III Congresso Brasileiro de Computação (CBCComp), Univali, 2003.
- [12] FIREBIRD. Disponível em <http://firebird.sourceforge.net/>, acesso em 10 de julho de 2005.
- [13] P. A. B. Garcia. **Provedores de Dados de Baixo Custo: Publicação Digital ao Alcance de Todos**. Dissertação de Mestrado. Curitiba, 2003. Setor de Ciências Exatas. Universidade Federal do Paraná.
- [14] A. J. Gilliland-Swetland. **Setting the Stage: In Introduction to Metadata: Pathways to Digital Information**. In Baca, *Getty Information Institute*, Los Angeles, CA, 2002 <http://www.getty.edu/research/institute/standards/intrometadata/>, Acesso em: 15 de maio de 2005.
- [15] R. Goulart. **Utilização de Metadados no Gerenciamento de Acesso a Servidores de Vídeo**. Dissertação de Mestrado. São Carlos, 1998. Setor de Ciências Exatas. Universidade de São Paulo.
- [16] C.A.Heuser. **Projeto de Banco de Dados**. 4^a ed. São Paulo: Sagra Luzzatto, 2001.
- [17] M. Keidl; S. Seltzsam; A. Kemper. **Reliable Web Service Execution and Deployment in Dynamic Environments**. Apresentado no *International Workshop on Technologies for E-Services (TES)*, Berlin, Germany, September 2003.

- [18] V. Kashyap, K. Shah, A. Sheth. **Metadata for Building the Multimedia Path Quilt**. In *Multimedia Database System: Issues and Research Directions*, Springer-Verlag, 1995.
- [19] S. R. L. Meira; A. L. C. Cavalcanti; C. S. Santos. **The Unix File System: A MooZ Specification**. In K. Lano and H. Haughton, editors, *Object-Oriented Specification Case Studies*, Object-Oriented Series. Prentice-Hall, 1993.
- [20] A. M. C. Moura, M. L. M. Campos, C. M. Barreto. **A Survey on Metadata for Describing and Retrieving Internet Resources**. *World Wide Web Journal*, 1(4), 1998, p. 221-240.
- [21] MySQL. Disponível em www.mysql.com, acesso em 10 de julho de 2005.
- [22] E. Oomoto; K. Tanaka. **Video database systems – recent trends in research and development activities**. *The Handbook of Multimedia Information Management*, Prentice Hall, 1997.
- [23] M. T. Özsu; P. Valduriez. **Princípios de sistemas de bancos de dados distribuídos**. Rio de Janeiro: Campus, 2001.
- [24] POSTGRESQL. Disponível em www.postgresql.org, acesso em 10 de julho de 2005.
- [25] R. Ramakrishnan; J. Gehrke. **Database management systems**. 2nd ed. Boston: McGraw-Hill, 2000.
- [26] S. A. Weil, K. T. Pollack, S. A. Brandt, E. L. Miller. **Dynamic Metadata Management for Petabyte-Scale File Systems**. ACM/IEEE SC 2004 Conference (SC'04).

- [27] A. Silberschatz; H.F. Korth; S. Sudarshan. **Sistema de banco de dados**. 3. ed. São Paulo: Makron Books, 1999.
- [28] SPRINGa. **Sistema de Processamento de Informações Georeferenciadas**. Disponível em: <http://www.dpi.inpe.br/spring/>, acesso em 15 de Junho de 2005.
- [29] SPRINGb. **Manual de Ajuda do SPRING**, disponível no Software SPRING versão 4.1, 2005.
- [30] SUN MICROSYSTEMS. Disponível em www.sun.com/java, acesso em 05 de Julho de 2005.
- [31] R. Steinmetz; B. Lindsay. **Multimedia: Computing, Communications, and Applications**. Prentice Hall, 1995.
- [32] M. S. M. G. Vaz. **MetaMida: Modelo de Metadados na Indexação e Recuperação de Objetos Multimídia**. Tese de Doutorado, Centro de Informática, Universidade Federal de Pernambuco, 2000.
- [33] M. S. M. G. Vaz; A. M. L. Vasconcelos; F. F. Souza; A. C. Salgado. **Descrevendo um modelo de informação de vídeo através de uma linguagem de especificação orientada a objetos**. IDEAS'98 – Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software, 1998.