Sistema de Manipulação Semântica de Dados

Daniel Bento de Paula

Supervisor: Prof. Dr. João Eduardo Ferreira

1 Índice

2 Introdução 4
3 Conceitos e tecnologias estudadas 5
3.1 Semântica de arquivos 5
3.2 Sistemas de arquivos atuais 6
4 Descrição Técnica 8
4.1 O objetivo 8
4.2 Implementação do modelo semântico básico 9
4.3 Buscas Semânticas 9
4.3.1 Busca por intersecção 9
4.3.2 Busca por semelhança 9
4.4 O modelo semântico estendido 10
4.5 Arquitetura do Sistema 10
4.5.1 Server 11
4.5.2 Job 12
4.5.3 Persistence Manager 12
4.5.4 Memory Manager 12
4.5.5 Core 13
4.6 Diagrama de classes 14
4.7 Estruturas e tipos de dados 15
5 Resultados e produtos obtidos 18
5.1 Sistema de Manipulação Semântica de Dados 18
5.2 MP3 Playlist Builder 19
6 Conclusão 19
7 Parte Subjetiva 20
7.1 Desafios e frustrações 20
7.2 Disciplinas importantes do curso 20
7.2.1 Estrutura de Dados 20
7.2.2 Laboratório de programação 21
7.2.3 Sistemas Operacionais 21

7.2.4 Programação Orientada a Objetos 21
7.3 Futuro 21
8 Referências Bibliográficas 22

2 INTRODUÇÃO

A principal e mais popular forma de armazenamento de dados em um sistema operacional é por meio de arquivos. Os diretórios, de forma intuitiva, provêm uma estrutura para distribuir e mantê-los organizados. Por outro lado, pouca ou nenhuma informação a respeito dos dados armazenados nos arquivos é oferecida.

A semântica oferecida pela tríade <nome do diretório> + <nome do arquivo> + <extensão do arquivo> é insuficiente e inflexível para classificar um arquivo em diversas categorias. Desta maneira, a estrutura de armazenamento baseada na tríade torna-se deficiente, pois não é possível armazenar um arquivo em mais de um diretório sem que exista duplicidade do mesmo arquivo. Uma solução paliativa seria criar links simbólicos ou atalhos, mas essa solução torna-se custosa ao gerenciar uma grande quantidade de arquivos que possam se encaixar em diversas categorias.

Para minimizar este problema, seria interessante que o sistema de arquivos fosse capaz de manipular semanticamente seus arquivos. O WinFS (1), sistema de arquivos semântico baseado no modelo relacional, é a solução da Microsoft para esse tipo de problema. No mundo open source, não existe uma solução única e centralizada. Existem diversos projetos com aplicações semelhantes, mas que não convergem para uma padronização.

A manipulação semântica de dados não se restringe apenas ao ambiente desktop. Com o aumento na quantidade e complexidade das informações disponíveis na internet, surgiu-se a necessidade de criar um modelo capaz de organizar e dar significado a toda essa informação. Essa nova etapa da internet, capaz de lidar com dados semânticos, é conhecida como Web 3.0 ou Web semântica (2).

Analogamente a Web semântica, que dirige seus esforços para atribuição e padronização da semântica de dados de sistemas distribuídos e heterogêneos, existe o desenvolvimento de sistemas de arquivos semânticos, como o WinFS, que agrega semântica e integra os dados entre as aplicações locais.

O objetivo desse trabalho de formatura supervisionado consiste em desenvolver um sistema de manipulação semântica de dados. Entenda por manipular semanticamente a capacidade de adicionar, remover, atualizar e realizar buscas levando em consideração a semântica entre os arquivos.

3 CONCEITOS E TECNOLOGIAS ESTUDADAS

3.1 SEMÂNTICA DE ARQUIVOS

De acordo com Vasudevan e Pazandak (3), semântica de arquivo é qualquer informação, de alto nível, relacionada ao arquivo. A semânticas podem ser:

- De definições, que incluem a extensão do arquivo e seu tipo.
- Associativas, como palavras-chaves do arquivo que caracterizam seu conteúdo;
- Estruturais, que contém a organização lógica e física dos dados, incluindo os relacionamentos intra e inter arquivos.
- Comportamentais, que possuem a semântica de visualização e modificação, gerenciamento de mudanças
- De ambiente, que contém informações a respeito do proprietário do arquivo, permissões, histórico de revisões, etc.

O foco deste trabalho é o estudo das semânticas de arquivos do tipo associativas. Assim, quando citada a palavra semântica, interprete-a como sendo a expressão “semântica de arquivos associativa”.

Atribuir semântica a um arquivo tem a finalidade de descrever seu conteúdo. A semântica é um formato que pode ser lido e usado por softwares, permitindo-lhes encontrar, compartilhar e integrar informação mais facilmente.

A semântica de arquivos é separada em duas estruturas: a primeira é um conjunto que descreve o conteúdo de um arquivo; a segunda é o próprio arquivo.

É natural supor que existam dados que são semanticamente parecidos. De fato, os dados em si não possuem semelhança, uma vez que não passam de um conjunto de bits. A semelhança existe nos conjuntos que descrevem o conteúdo dos mesmos. A modelagem semântica deve ser capaz de mensurar o quanto um dado é semelhante ou está relacionado a outro em função dos conjuntos que os descrevem.

Considere os elementos dos os conjuntos que descrevem os dados de um sistema. Como esses elementos são formados por palavras-chaves, temos que eles são definidos por um domínio de aplicação. Assim, é natural de supor que eles se relacionam de alguma forma. Uma modelagem semântica mais poderosa pode ser obtida caso seja definido um relacionamento do tipo “elemento A realiza uma ação sobre elemento B”, que é equivalente a “elemento B sofre uma ação do elemento A”. Diferentemente do relacionamento por semelhança, que mostra o quanto os elementos são próximos, o relacionamento por ação mostra a maneira como os elementos interagem.

3.2 SISTEMAS DE ARQUIVOS ATUAIS

Os sistemas de arquivos mais populares (como o Ext3, ReiserFS, NTFS) não possuem estruturas para agregar semântica associativa aos arquivos. Um dos tipos de semântica associada a esses sistemas de arquivos são as semânticas de ambientes. Como os sistemas operacionais que utilizam estes sistemas de arquivos são multiusuários, é necessário que exista um sistema para o controle de permissões e usuários dos arquivos.

O Mac File System, criado em 1984, foi um dos primeiros sistemas de arquivos comerciais a utilizar semântica de arquivos associativa. O MFS foi notável por introduzir ramificação de recursos para permitir o armazenamento de dados estruturados, e também por armazenar metadados necessários ao suporte da interface gráfica de usuário do Mac OS (4).

O armazenamento de arquivos em diretórios tem a finalidade de agrupar arquivos com algum grau de semelhança. Este agrupamento não ocorre ao acaso: procura-se dispor os arquivos de tal forma que a estrutura de diretórios agregue informações aos arquivos ali existentes, seja pelo nome dado ao diretório, seja pelas ramificações criadas de subdiretórios. Toda essa estrutura tem a finalidade de facilitar buscas futuras por arquivos.

As árvores de diretório possuem semântica embutida, porém esta é inflexível. Não há como armazenar o mesmo arquivo, sem duplicidades, em diferentes estruturas de diretórios. Desta maneira, organizar grandes variedades e quantidades de arquivos torna-se uma atividade maçante e custosa de se gerenciar.

O aumento da capacidade dos dispositivos de armazenamento propiciou o aumento da variedade e quantidade de arquivos. Pegue a capacidade máxima de armazenamento de 130 MB, em 1991, e compare-a com a capacidade atual de 750 GB por disco rígido. Temos um aumento na capacidade de 5700 vezes nos últimos 15 anos (5).

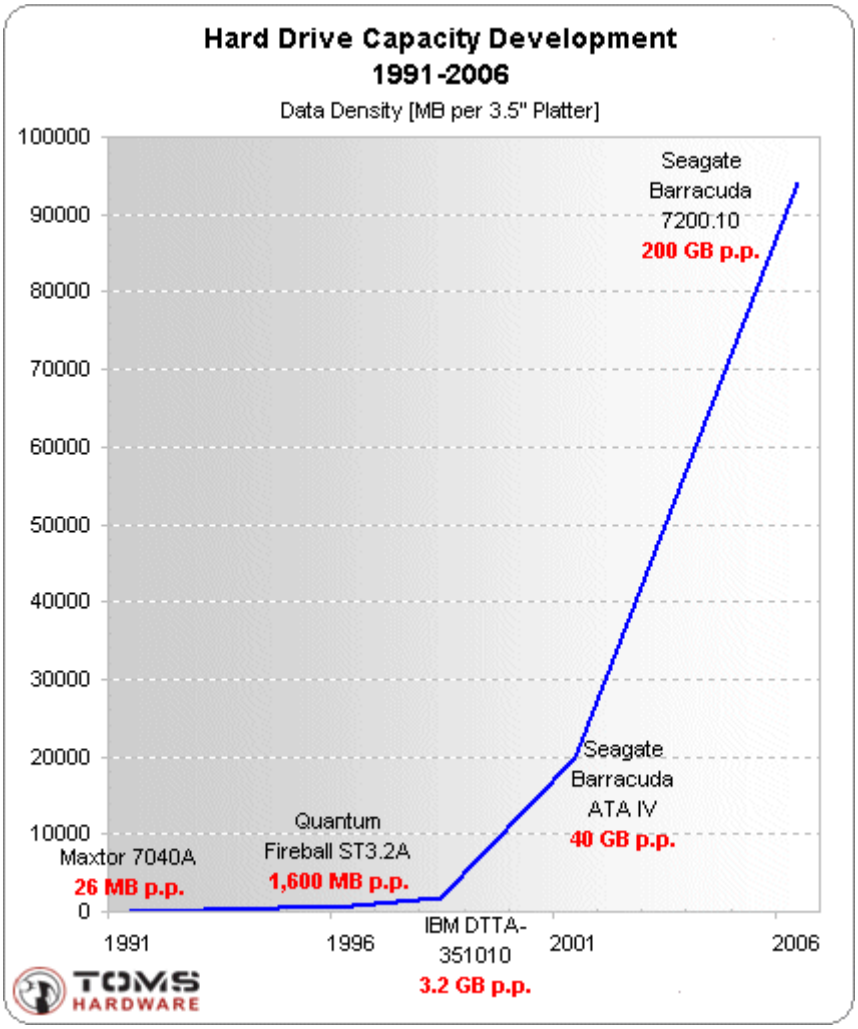


FIGURA 1: WWW.TOMSHARDWARE.COM

O aumento na capacidade de armazenamento, aliado ao aumento na capacidade de processamento e disseminação dos softwares de compartilhamento de arquivos via internet, modificaram as hábitos dos usuários. Armazenar grandes variedades e quantidade de arquivos tornou-se uma necessidade.

Uma solução para o problema do aumento da quantidade e variedade de arquivos, agravada pela inflexibilidade da estrutura de diretórios, é construir uma estrutura que seja capaz de agregar semântica associativa aos arquivos. Seria interessante que essa estrutura fosse incorporada internamente ao sistema de arquivos, provendo todos os recursos necessários para manipular semanticamente arquivos.

4 DESCRIÇÃO TÉCNICA

4.1 O OBJETIVO

O objetivo desse trabalho de formatura supervisionado consiste em construir um sistema de manipulação semântica de dados. Entenda por manipular semanticamente a capacidade de adicionar, remover e atualizar e realizar buscas levando em consideração a semântica entre os arquivos.

O sistema aqui descrito não se trata de um sistema de arquivos nativo, mas uma camada entre o sistema de arquivos e os aplicativos. Ele pode ser visto como um servidor de serviços para manipulação semântica de arquivos. Suas funcionalidades são:

- Inserção / deleção de arquivos.
- Inserção / deleção de tags.
- Associação / remoção de tags dos arquivos.
- Buscas semânticas.

O objetivo é a construção de um sistema simples, que necessite pouca configuração por parte dos usuários e que obter resultados satisfatórios com a manipulação semântica de dados.

O SMSD foi desenvolvido utilizando a linguagem de programação JAVA[1]. Sua escolha deve-se ao fato do JAVA:

- Ser uma linguagem de programação orientada a objetos.
- Possuir grande variedade de bibliotecas.
- Dar suporte de maneira robusta para a criação e manipulação de threads.
- A Java Virtual Machine torna o programa portável para qualquer plataforma.

O uso de uma linguagem orientada a objetos trás os benefícios de utilizarmos padrões de design, que tornam o sistema mais modular, acoplável e simples de realizar futuras manutenções.

[1] http://java.sun.com/

4.2 IMPLEMENTAÇÃO DO MODELO SEMÂNTICO BÁSICO

O SMSD lida com semântica do tipo associativa. Esta tem a finalidade de associar palavras-chave aos arquivos do sistema. As palavras-chaves aqui citadas serão chamadas de Tag. Os arquivos do sistema serão chamados de DataFiles.

Cada Tag pode estar associada a um ou mais DataFiles. Cada DataFile pode estar associado a uma ou mais Tags.

4.3 BUSCAS SEMÂNTICAS

Vamos definir cada Tag t como sendo um conjunto de DataFiles. E cada DataFile df como sendo um conjunto de Tags.

4.3.1 BUSCA POR INTERSECÇÃO

Tem o objetivo de encontrar os DataFiles comuns as tags de entrada. Seja T = {t1, t2, ..., tn} as tags de entrada. A busca por interseção retorna a os DataFiles presentes na intersecção de T.

4.3.2 BUSCA POR SEMELHANÇA

Tem o objetivo de encontrar DataFiles que sejam semelhantes as tags de entrada. Sejam T = {t1, t2, ..., tn} as tags de entrada e Ttodos = {v1, v2, ..., vm} = Todas as tags existentes no sistema. A busca por semelhança realiza os seguintes passos:

- Calcule Df = a intersecção de DataFiles do conjunto T.
- Para cada elemento vi em Ttodos, calcule a quantos DataFiles vi está associado em Df. Chame esse valor de semelhança(vi).
- Ordene os elementos de Ttodos em ordem decrescente de semelhança(vi).
- Calcule a união dos elementos de Ttodos para os quais semelhança(vi) é diferente de zero.
- Retorne esse resultado.

4.4 O MODELO SEMÂNTICO ESTENDIDO

O modelo semântico estendido visa aumentar o poder das buscas semânticas. Ele é o modelo semântico básico acrescido das seguintes estensões:

- Para cada Tag, associa-se um tipo.
- Tags relacionam-se através de um objeto ação. Dizemos que a tag1 faz uma ação na tag2 e a tag2 recebe uma ação da tag1.

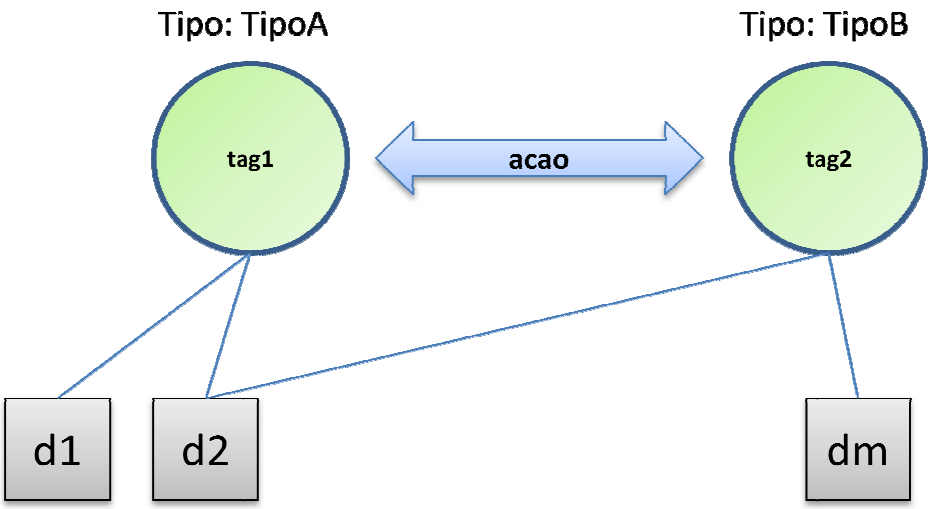


FIGURA 2: MODELO SEMÂNTICO ESTENDIDO

Esse tipo de abordagem semântica é semelhante ao utilizado pela Web Semântica, onde existem metadados e uma estrutura intermediária que descreve como esses metadados interagem entre si.

4.5 ARQUITETURA DO SISTEMA

A modelagem do sistema levou a criação das seguintes unidades funcionais:

- Server
- Job
- Core
- PersistenceManager
- MemoryManager

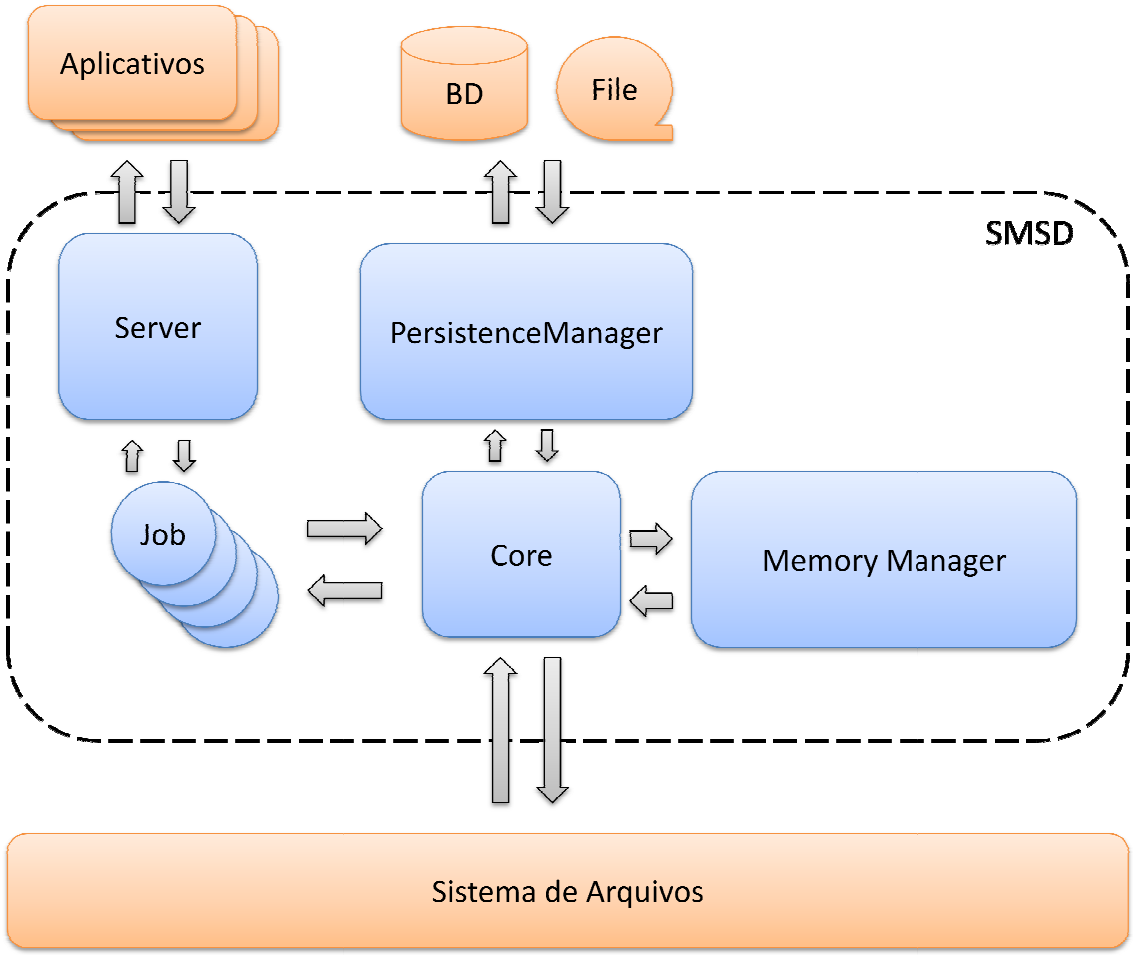


FIGURA 3: UNIDADES FUNCIONAIS

4.5.1 SERVER

O SMSD pode ser visto como um servidor. Seus serviços são responsáveis por gerenciar a manipulação semântica dos arquivos. Existem os seguintes serviços:

- Inserção de tags
- Deleção de tags
- Inserção de arquivos
- Deleção de arquivos
- Associação entre tags e arquivos
- Remoção de tags dos arquivos
- Busca por intersecção
- Busca por semelhança

Os aplicativos interagem com o SMSD por meio de comandos. Cada comando é uma requisição a um serviço do SMSD. O Server analisa o comando e em função deste, cria um Job, que é responsável por executar o comando.

4.5.2 JOB

Job são threads criadas pelo Server responsáveis por enviar comandos ao Core do SMSD.

4.5.3 PERSISTENCE MANAGER

O persistence manager possui todas as estruturas responsáveis por realizar a persistência de dados do SMDS. Para garantir a independência entre os mecanismos de persistência e o SMSD, foi adotado o padrão Data Access Object (DAO). "O DAO é um objeto que provê uma interface abstrata a alguns tipos de banco de dados ou mecanismos de persistência, provendo algumas operações específicas sem expor os detalhes do banco de dados" (6).

Como o DAO é apenas uma interface para o acesso de dados, foi criada uma implementação para esta interface, utilizando a API Java Database Connectivity (JDBC)[2]. A plataforma Java SE inclui uma implementação da API JDBC que utiliza ODBC. Isso possibilita conexões com qualquer banco de dados baseado no modelo relacional com suporte a ODBC. O banco de dados relacional utilizado foi o MySQL[3]. Sua escolha deve-se ao fato de ser um banco de dados open source de fácil configuração e utilização.

É possível implementar tipos alternativos de persistência. Basta seguiro padrão DAO utilizado.

4.5.4 MEMORY MANAGER

O memory manager é a entidade responsável por armazenar, gerenciar e garantir a integridade dos em memória. Ele possui um conjunto de métodos e estruturas de dados responsáveis pela manipulação dos dados ali presentes.

Mecanismos de persistência, em geral, acessam e manipulam dados diretamente do disco rígido. Os tempos de acesso ao disco rígido são de ordem 100000 maiores do que os tempos de acesso à memória. A necessidade da existência do

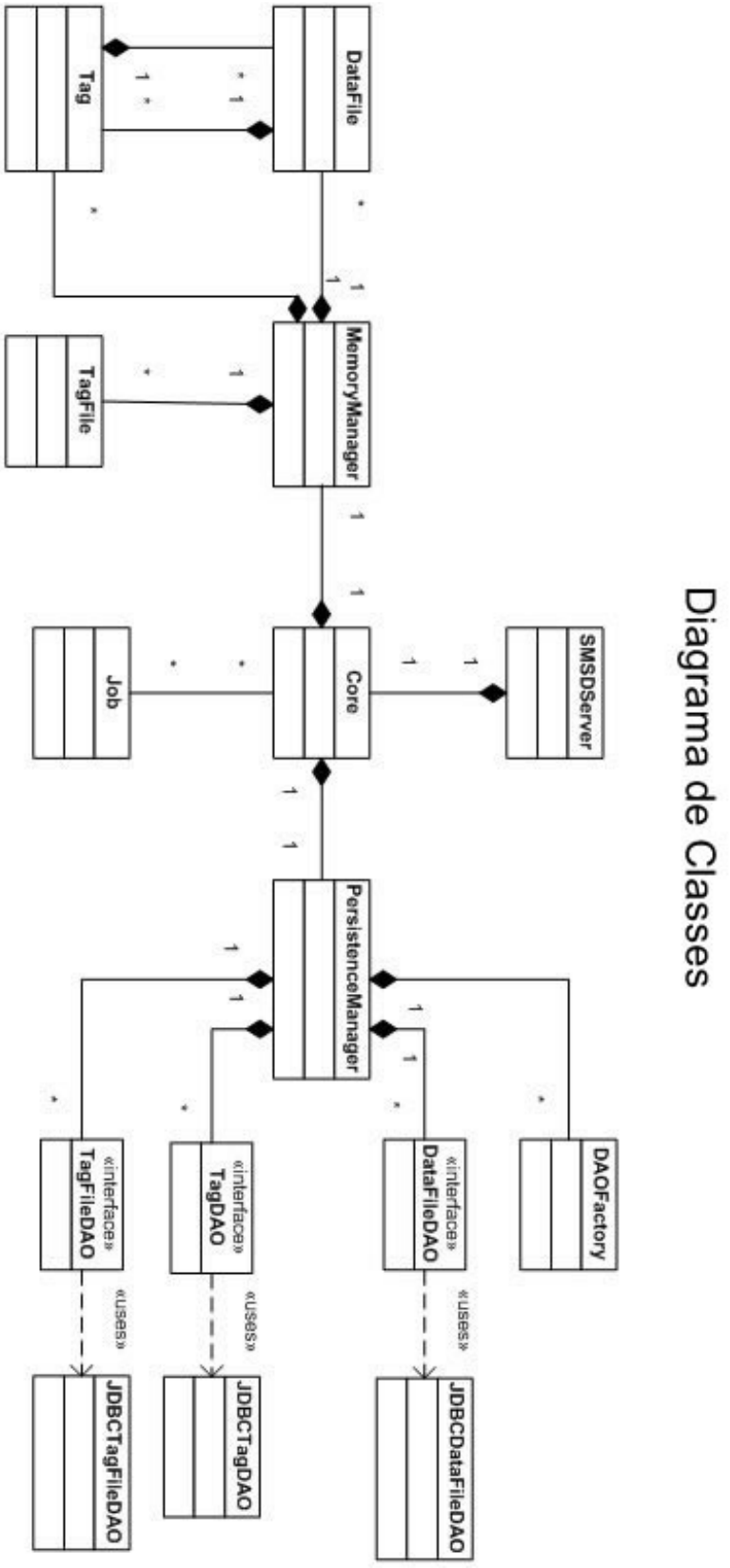
[2] http://java.sun.com/javase/technologies/database/
[3] http://www.mysql.org

memory manager está ligada a este fato. Com o intuito de minimizar os tempos de acesso, é interessante manter em memória todos os dados ativos do sistema, principalmente no momento em que são realizadas buscas complexas.

4.5.5 CORE

O Core, como o próprio nome sugere, é a estrutura central do sistema. Ele é o responsável por executar os comandos que chegam através dos Jobs e intermediar a transferência de dados entre o memory manager e o persistence manager.

4.6 DIAGRAMA DE CLASSES



4.7 ESTRUTURAS E TIPOS DE DADOS

Apresentaremos nesta seção as estruturas de dados utilizadas no sistema, assim como a justificativa para a utilização das mesmas.

Os principais tipos de dados do SMSD são Tag, DataFile e TagFile. Cada Tag é uma palavra-chave que representa o conteúdo de um DataFile. Cada DataFile é uma referência a um arquivo armazenado em disco. Tags podem se associar a um ou mais DataFiles. DataFiles podem se associar a uma ou mais Tags. Desta maneira surge um relacionamento N para M entre Tags e DataFiles. Cada TagFile representa uma instância desse relacionamento. Cada Tag e DataFile armazenam as referências aos objetos associados em uma tabela de Hash.

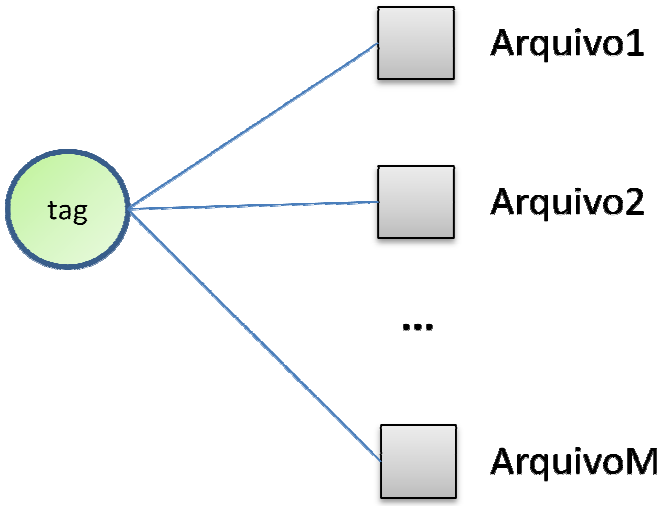


FIGURA 4: TAG ASSOCIADA A M DATAFILES

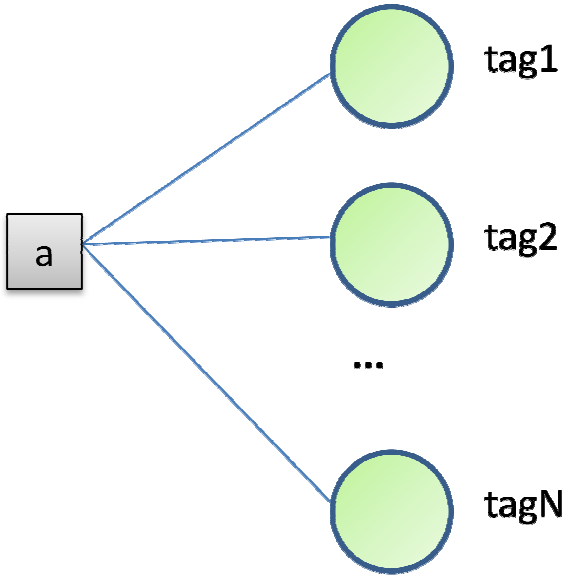
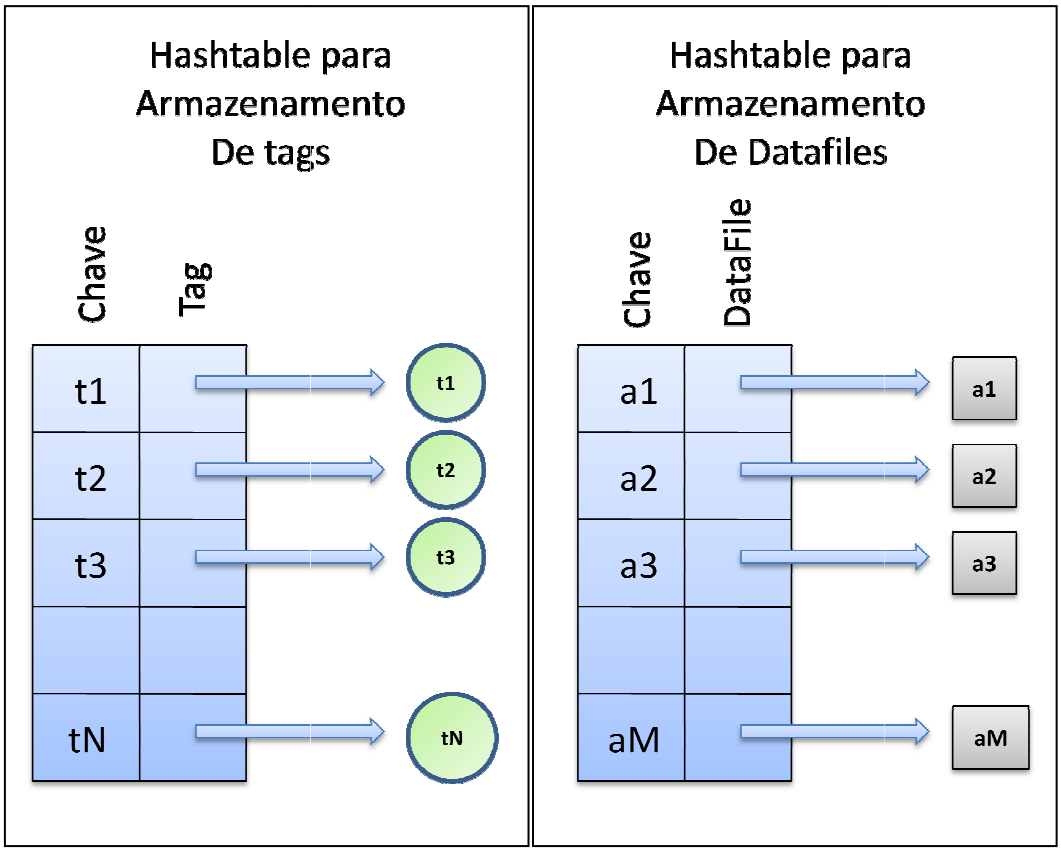


FIGURA 5: DATAFILE D ASSOCIADO A N TAGS

As estruturas de dados foram escolhidas de modo a aumentar o desempenho das buscas semânticas. Pode-se abstrair a Tag como sendo um conjunto de DataFiles e o DataFile como sendo um conjunto de Tags. Vários passos dos algoritmos de busca semântica necessitam encontrar a intersecção entre esses conjuntos. Encontrar a intersecção pode ser uma tarefa computacionalmente cara caso não sejam utilizadas as estruturas de dados corretas.

O memory manager possui duas estruturas de Hash, uma para armazenar referências às Tags e outra para as referências aos DataFiles. A tabela de Hash é uma estrutura de dados que armazena objetos e permite o acesso aos mesmos, através de uma chave, em tempo constante O(1). Elas são utilizadas para que não houvesse a necessidade de percorrer todas as Tags ou DataFiles do sistema para realizar o acesso ou para verificar pertinência. As chaves de acesso às Tags, em sua tabela de Hash, é o próprio nome da Tag. Como os parâmetros das buscas semânticas são os nomes das Tags, conseguimos acessá-las em tempo constante, melhorando o desempenho das buscas.



O memory manager também possui um ArrayList de TagFile. A existência dessa estrutura deve-se unicamente para facilitar a transferência de dados entre o persistence manager e o memory manager, seguindo o padrão DAO.

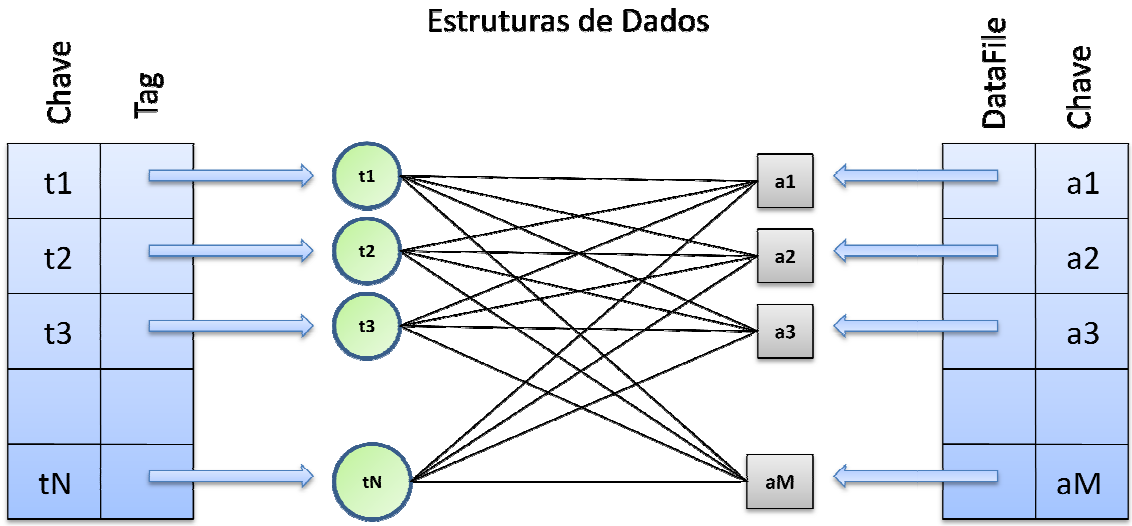


FIGURA 6: UNIÃO DAS ESTRUTURAS DE DADOS

5 RESULTADOS E PRODUTOS OBTIDOS

5.1 SISTEMA DE MANIPULAÇÃO SEMÂNTICA DE DADOS

O resultado principal do trabalho foi o desenvolvimento do sistema citado anteriormente.

Os serviços de adição, deleção e associação entre tags e arquivos, além da busca por semelhança, funcionaram satisfatoriamente. Esses serviços abrangem totalmente os requisitos para a manipulação semântica de dados.

A busca por semelhança foi uma tentativa de mostrar, a partir de um modelo simples, se seria possível mensurar o quanto um arquivo é semanticamente próximo de outro. A busca apresentou resultados abaixo do esperado. O modelo simplificado utilizado não contém todos os elementos necessários para fornecer respostas razoáveis em todas as buscas. O modelo estendido seria uma solução mais eficaz para o problema da semelhança, por conter informações a respeito de como as tags interagem entre si.

O SMSD foi construído sobre a filosofia do software livre e está disponível para futuros desenvolvimentos.

5.2 MP3 PLAYLIST BUILDER

Como o SMSD é um servidor de serviços para manipulação semântica de arquivos, foi construído um software cliente para testar seus serviços.

O MP3 Playlist Builder é um software para geração de playlists. Ele gera listas no formato m3u, que é utilizado pelos principais softwares reprodutores de arquivos mp3, como Winamp e XMMS.

Suas funcionalidades são:

- Geração de listas m3u através de buscas semânticas.
- Extração de tags dos arquivos mp3 e adição das mesmas, juntamente com os arquivos, no SMSD.

6 CONCLUSÃO

A manipulação semântica de dados é um assunto que está em bastante evidencia. A Web Semântica, por contar com o respaldo do W3C, entidade que define padrões para a web, ganhará força em alguns anos, pois o atual modelo não comporta a grande quantidade de informações disponíveis na internet.

A principal promessa para um sistema de arquivos com manipulação semântica de dados é o WinFS. A Microsoft vem investindo em pesquisas para o desenvolvimento desse sistema de arquivos. Mas o desenvolvimento nunca está completo: sua entrega sempre é adiada para cada nova versão do Windows.

O desenvolvimento de um sistema de arquivos semântico no mundo open source parece estagnado. Não há uma convergência de esforços ou mesmo uma padronização, o que cria o entrave para o desenvolvimento unificado de um sistema. Uma tecnologia deste porte necessita de um coordenador que oriente os desenvolvedores em suas atividades. Mas caso o WinFS vingue, o mundo open source tratará de se organizar para criar seu similar.

Assim como o SMSD, outras ferramentas para manipulação semântica de dados são desenvolvidas enquanto não surge um sistema de arquivos semânticos nativo. O NEPOMUK [4], um ambiente baseado em ontologias e conhecimento unificado, estreará com o KDE4 para os desktops Linux. O surgimento contínuo desse tipo de

[4] http://nepomuk.semanticdesktop.org/

tecnologia sugere que a mudança de paradigma na organização e busca de dados parece irreversível.

7 PARTE SUBJETIVA

7.1 DESAFIOS E FRUSTRAÇÕES

A idéia inicial do trabalho era construir um sistema de arquivos para Linux com manipulação semântica de dados. O trabalho seria feito em dupla e teria duas responsabilidades principais: construir a estrutura para agregar semântica associativa aos arquivos e integrar essa estrutura ao sistema de arquivos. Como meu parceiro de trabalho não cursou a disciplina de conclusão de curso, não seria possível tocar a idéia inicial do trabalho sozinho. A solução foi concentrar os trabalhos na criação de um sistema que fosse independente do sistema de arquivos. E esse, de fato, foi o produto final obtido com o trabalho, mas com a ressalva que não é um sistema de arquivos nativo.

O maior desafio nesse trabalho foi o tempo. Conciliar os estudos de outras disciplinas com o desenvolvimento do projeto e o estágio não foi uma tarefa simples. As tecnologias utilizadas no desenvolvimento do projeto possuem grande documentação na internet, o que facilitou o trabalho. Por outro lado, não foram encontrados muitos artigos científicos que ajudassem a modelar o sistema conceitualmente.

O sistema funciona satisfatoriamente em seus serviços básicos para manipulação semântica. Contudo, a busca por semelhança foi minha frustração. Eu apostava que seria possível construir uma busca de semelhança semântica de arquivos, a partir do modelo simples, com resultados satisfatórios. Ao final dos testes percebi que o modelo utilizado não era suficiente para resolver esse tipo de problema. Acredito que resultados mais satisfatórios teriam sido obtidos caso fosse utilizado o modelo estendido para realizar a busca por semelhança.

7.2 DISCIPLINAS IMPORTANTES DO CURSO

Aqui serão relacionadas as disciplinas mais importantes do curso para o desenvolvimento do SMSD.

7.2.1 ESTRUTURA DE DADOS

A meu ver, estrutura de dados é a disciplina mais importante do curso de ciência da computação. Ela fornece a visão de como os programas realmente

funcionam por dentro. Nela, acabamos descobrindo que sistemas mais complexos nada mais são do que associações de diversas estruturas de dados diferentes.

Como as estruturas de dados influenciam diretamente o desempenho de um software, a disciplina forneceu o conhecimento necessário para realizar a escolha das estruturas de dados mais eficientes para cada necessidade.

7.2.2 Laboratório de programação

Laboratório de programação é a primeira disciplina onde encaramos o desenvolvimento de um software de maior porte. Nela aprendemos como seguir uma metodologia de desenvolvimento e adquirir bons hábitos como comentar códigos nos programas e criar documentação detalhada para os sistemas.

7.2.3 Sistemas Operacionais

É a área onde o trabalho de conclusão de curso se insere. Foram apresentadas as áreas de estudo, entre elas sistemas de arquivos.

7.2.4 Programação Orientada a Objetos

O desenvolvimento de um sistema, que utiliza uma linguagem de programação orientada a objetos, deve seguir os padrões de design sugeridos pela “Gang of Four”. A disciplina de programação orientada a objetos mostrou os padrões de design mais relevantes, aos quais alguns são utilizados no SMSD.

7.3 Futuro

Como o SMSD é uma camada entre os aplicativos e o sistema de arquivos, o próximo passo do desenvolvimento seria implementar suas funcionalidades dentro um sistema de arquivos. Para isso é necessário adquirir conhecimentos avançados de sistemas operacionais e como os sistemas de arquivos funcionam.

Para aperfeiçoar a busca por semelhança, o modelo semântico estendido deve ser implementado. Como ele lida com tipagem de objetos, estudos de linguagens de programação com tipagem seriam necessários.

8 REFERÊNCIAS BIBLIOGRÁFICAS

1. **Wikipedia.** WinFS. *Wikipedia, the free encyclopedia.* [Online] 21 de Outubro de 2007. [Citado em: 15 de Novembro de 2007.] http://en.wikipedia.org/wiki/WinFS.

2. **W3C.** W3C Semantic Web Activity. *World Wide Web Consortium.* [Online] [Citado em: 15 de Novembro de 2007.] http://www.w3.org/2001/sw/.

3. **Vasudevan, Venu e Pazandak, Paul.** Semantic File Systems. *OBJS.* [Online] Object Services and Consulting, Janeiro de 1997. [Citado em: 15 de Novembro de 2007.] http://www.objs.com/survey/OFSExt.htm.

4. **Wikipedia.** Macintosh File System. *Wikipedia, the free encyclopedia.* [Online] como vc traduziria o "resource fork" em. http://en.wikipedia.org/wiki/Macintosh_File_System.

5. **Schmid, Patrick e Roos, Achim.** 15 Years Of Hard Drive History: Capacities Outran Performance. *Tom's Hardware.* [Online] 27 de Novembro de 2006. [Citado em: 15 de Novembro de 2007.] http://www.tomshardware.com/2006/11/27/15-years-of-hard-drive-history/.

6. **Wikipedia.** Data Access Object. *Wikipedia, the free encyclopedia.* [Online] 9 de Novembro de 2007. [Citado em: 15 de Novembro de 2007.] http://en.wikipedia.org/wiki/Data_Access_Object.

7. **Gifford, David, et al.** Semantic File Systems. *Proceedings of 13th ACM Symposium on Operating Systems Principles.* 1991.