UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2007.1

IMPLEMENTAÇÃO DA CAMADA DE REDE *SUPER-PEER* PARA PONTOS DE INTEGRAÇÃO E PONTOS DE DADOS NO SISTEMA SPEED

TRABALHO DE GRADUAÇÃO

Aluno – Domingos Sávio Ribeiro Mendes (dsrm@cin.ufpe.br)

Orientadora – Ana Carolina Salgado (acs@cin.ufpe.br)

Recife, Agosto de 2007

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CENTRO DE INFORMÁTICA

2007.1

DOMINGOS SÁVIO RIBEIRO MENDES

IMPLEMENTAÇÃO DA CAMADA DE REDE *SUPER-PEER* PARA PONTOS DE INTEGRAÇÃO E PONTOS DE DADOS NO SISTEMA SPEED

Trabalho apresentado à graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora – Ana Carolina Salgado (acs@cin.ufpe.br)

Recife, Agosto de 2007

Resumo

Nos últimos anos, a necessidade de acesso a dados distribuídos fez surgir uma grande quantidade de sistemas de gerenciamento de dados. Devido ao crescimento dos sistemas *peer-to-peer* (P2P), surgiram então os Sistemas de Gerenciamento de Dados *Peer-to-peer* (*Peer Data Management Systems* – PDMS) [Sung 2005].

Os PDMS possibilitam o compartilhamento de dados estruturados e semi-estruturados através de redes P2P. São compostos por uma grande quantidade de pontos que representam fontes de dados disponibilizando seus dados através de esquemas de exportação, utilizando diferentes tipos de topologias de rede.

O SPEED [Pires 2007] é uma proposta de um PDMS baseado em uma topologia de rede mista, utilizando uma rede DHT (*Distributed Hash Table*) [Sung 2005] e uma rede *Super-peer* [Fiorano 2003]. O SPEED explora o conceito de comunidade semântica, relacionada ao conteúdo exportado pelos pontos de dados para encontrar outros pontos com interesses em comum. Essas comunidades são formadas pelos pontos semânticos, que fornecem uma ontologia sobre um domínio de conhecimento. Os pontos semânticos se relacionam com os pontos de integração, que são responsáveis pelo gerenciamento de dados dos pontos de dados, formando um cluster semântico. Em cada cluster semântico, é utilizada a rede *super-peer*, conectando os pontos de integração com os pontos de dados em uma comunidade semântica.

O propósito deste trabalho de graduação é propôr e desenvolver a camada de rede *super-peer* para os clusters semânticos do sistema SPEED, com entrada de pontos de dados e de pontos de integração.

Agradecimentos

Primeiramente, agradeço a Deus pela vida e por ter me dado a oportunidade de estudar em uma universidade conceituada, onde a maioria da população não tem esse privilégio.

Agradeço a toda minha família, em especial a minha mãe Maria José, a minha avó Cléa, a meus irmãos Thomaz e Gerardo, ao amigo Haroldo e a minha noiva Bárbara, por terem me apoiado nessa luta e por me aguentar durante os períodos de stress na faculdade.

Agradeço a minha orientadora, a professora Doutora Ana Carolina Salgado, pela oportunidade de trabalhar em um projeto e pelos carões necessários para a realização das tarefas.

Agradeço também a Carlos Eduardo, Damires Souza, Juliano Souza e Guilherme Souza, companheiros do projeto, pela orientação e apoio, nos momentos mais difíceis na realização do trabalho.

Agradeço a todos os meus amigos do CIn, pelo apoio durante o curso, e pelos momentos de descontração proporcionados entre tantos projetos e provas difíceis realizadas.

Agradeço, enfim, a todas as pessoas que de alguma forma contribuíram na realização deste trabalho.

Sumário

Índice de figuras

1 Introdução

Neste capítulo, será feita uma introdução do contexto onde foi desenvolvido o trabalho. Nele, serão abordados aspectos da motivação para a implementação e também os objetivos do trabalho.

1.1 Motivação

O crescimento da quantidade de dados e, conseqüentemente, dos sistemas de informação, mostra que a integração de dados vem se tornando uma área de bastante pesquisa, principalmente na tentativa de resolver problemas de gerenciamento de dados, como o processamento de consultas. Juntamente com esse crescimento, os sistemas *peer-to-peer* vêm se desenvolvendo, impulsionados pelo aumento de banda na internet e do aumento do poder computacional dos pontos na rede, com o surgimento de diversas aplicações [Sung 2005].

Surgem então os Sistemas P2P de Gerenciamento de Dados (ou *Peer* Data Management Systems - PDMS) [Sung 2005], possibilitando o compartilhamento de dados estruturados e semi-estruturados. Os PDMS possuem características como compartilhamento de dados descentralizado, processamento e armazenamento de dados distribuídos em *peers* autônomos, mapeamentos semânticos entre esquemas exportados pelos *peers* e escalabilidade.

Este trabalho tem como base o SPEED [Pires 2007], um PDMS que utiliza semântica e uma topologia de rede mista, com dois níveis, empregando os conceitos de s*uper-peer* e DHT. A rede DHT é usada para a busca de comunidades semânticas e, dentro de uma comunidade, os *peers* são organizados na topologia de *super-peer*, facilitando o processamento de consultas e a geração e gerenciamento dos mapeamentos semânticos.

1.2 Objetivo

Nesse contexto, este trabalho de graduação tem como objetivo fundamentar e implementar uma solução para uma rede *super-peer* [Fiorano 2003] para os pontos de integração e pontos de dados do sistema SPEED. O trabalho colaborará com duas teses de doutorado [Pires 2007, Souza, D. 2007] e um trabalho de graduação [Souza, G. 2007] já concluído no CIn/UFPE. O trabalho foi dividido em três partes: estudo sobre sistemas P2P e PDMS; estudo de simuladores de redes *super-peer* existentes; especificação e do comportamento da rede *super-peer* do SPEED e implementação dessa rede.

1.3 Estrutura do documento

O restante desse trabalho está organizado como segue. O Capítulo 2 aborda os conceitos relativos a *peer-to-peer* e suas topologias de rede, bem como das características e funcionalidades de um PDMS.

O Capítulo 3 descreve o SPEED, seus fundamentos, suas principais características, o seu funcionamento e seus componentes.

O Capítulo 4 descreve o estudo com o simulador de rede, a definição dos parâmetros para a implementação da rede s*uper-peer*, sua arquitetura, e detalhes de implementação.

E, por fim, o Capítulo 5 descreve as conclusões obtida a partir do estudo e a implementação realizada, suas contribuições para o sistema SPEED, as dificuldades encontradas durante o trabalho e a proposta de trabalhos futuros.

2 *Peer-to-peer* e PDMS

Sistemas e aplicações *peer-to-peer* são sistemas distribuídos onde não existe nenhuma forma de controle centralizado ou hierarquia organizacional [Stoica 2001]. Nesse capítulo, serão abordados os conceitos sobre topologias e sistemas *peer-to-peer* e também sobre os PDMS.

Os primeiros sistemas P2P tinham como objetivo o compartilhamento de dados não-estruturados [Kolweyh 2004], sendo o Napster [Napster 2007] o primeiro grande exemplo desse tipo de sistema. O paradigma *peer-to-peer* possibilita a implementação de sistemas com características como volatilidade, auto-organização, tolerância a falhas, descentralização, balanceamento de carga, escalabilidade e roteamento.

2.1 Topologias

De acordo com as diferentes implementações das redes *peer-to-peer*, elas foram classificadas em topologias, que serão descritas a seguir.

2.1.1 Topologia P2P Pura ou Descentralizada

A topologia pura é o modelo de rede onde todos os *peers* exercem as mesmas funções na rede, não existindo nenhum tipo de hierarquia (ver Figura 1). Um *peer* realiza a busca por um arquivo enviando a consulta por *flooding* para os outros peers conectados a ele (passo 1, Figura 1). Caso o *peer* não tenha o arquivo solicitado, ele reenvia a consulta para outros peers, e assim sucessivamente (passo 2, Figura 1), até que algum deles tenha o arquivo. Então ele envia o seu *ip* para o peer que o solicitou (passo 3 e 4, Figura 1), para que possa ser feito o download do arquivo (passo 5, Figura 1). Seu funcionamento é totalmente descentralizado, com os pontos se conectando automaticamente e compartilhando recursos, serviços e/ou conteúdo com qualquer outro ponto conectado. Exemplos de sistemas *peer-to-peer* baseados em topologia pura são: Gnutella [Gnutella 2007] e FreeNet [FreeNet 2007].

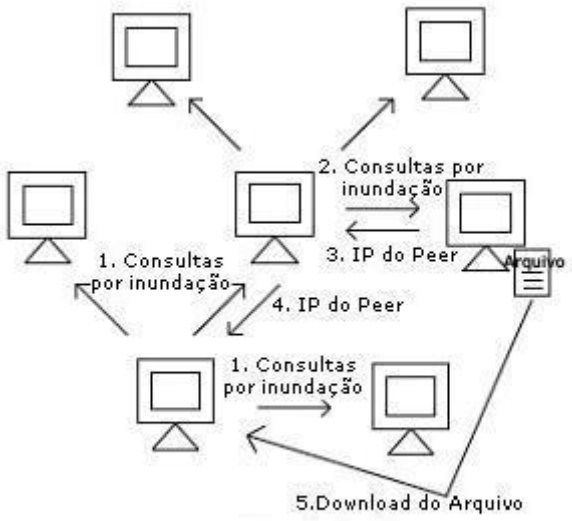

Figura 1 - Topologia Pura. Adaptado de [Pires 2007]

Na topologia pura, existe uma subclassificação: não estruturada e estruturada. Na rede pura não estruturada não existe qualquer restrição quanto à localização, à distribuição e à busca dos recursos na rede, o que se caracteriza pelo uso de *broadcast* (*flooding*) para o envio de mensagens, tornando a rede menos eficiente e escalável. Já uma topologia pura estruturada apresenta um certo tipo de controle em relação à localização dos recursos, com cada *peer* mantendo informações sobre seus recursos e também sobre os recursos compartilhados por outros *peers*. Estes sistemas utilizam um mecanismo para otimizar a busca e recuperação de recursos na rede, denominado DHT (Tabela Hash Distribuída) [Sung 2005], que proporciona uma melhor busca e acesso aos dados distribuídos, tornando o sistema escalável e mais confiável.

2.1.2 Topologia P2P Parcialmente Centralizada ou Híbrida

A topologia P2P parcialmente centralizada ou híbrida se caracteriza pela presença de um ou mais servidores centrais (ver Figura 2).

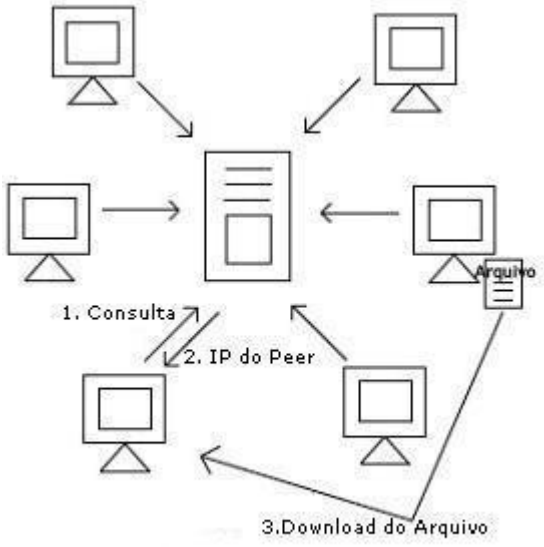

Figura 2 - Topologia Híbrida. Adaptado de [Pires 2007]

As informações sobre os recursos compartilhados pelos *peers* são enviadas para o servidor central, ficando este responsável pelas buscas e indexação. Um *peer* realiza a busca por um arquivo enviando a para o servidor central (passo 1, Figura 2). O servidor então ele envia o *ip* do *peer* que tem o arquivo para o *peer* que o solicitou (passo 2, Figura 2), para que possa ser feito o download do arquivo (passo 3, Figura 2). Essa topologia apresenta a existência de um ponto de falha: no caso de desligamento do servidor central, o sistema ficará parcialmente comprometido. Além do ponto de falha, com o fluxo de informações concentrado, essa topologia não é indicada para sistemas que necessitem de alta escalabilidade. Exemplos de sistemas com topologia híbrida: Napster [Napster 2007] e BitTorrent [BitTorrent 2007]

2.1.3 Topologia P2P *Super-Peer*

Na topologia *super-peer*, alguns *peers* podem assumir papéis diferentes, sendo eleitos os de maior capacidade computacional para coordenar um subconjunto de *peers*. Esse *peer* é chamado de *super-peer* e tem características tanto da topologia pura como da topologia híbrida [Fiorano 2003], combinando a eficiência da busca centralizada com a autonomia e balanceamento da busca descentralizada. As aplicações KaZaA [Kazaa 2007]

e Morpheus [Morpheus 2007] são baseadas na topologia *super-peer*. A topologia s*uper-peer* é ilustrada na Figura 3.

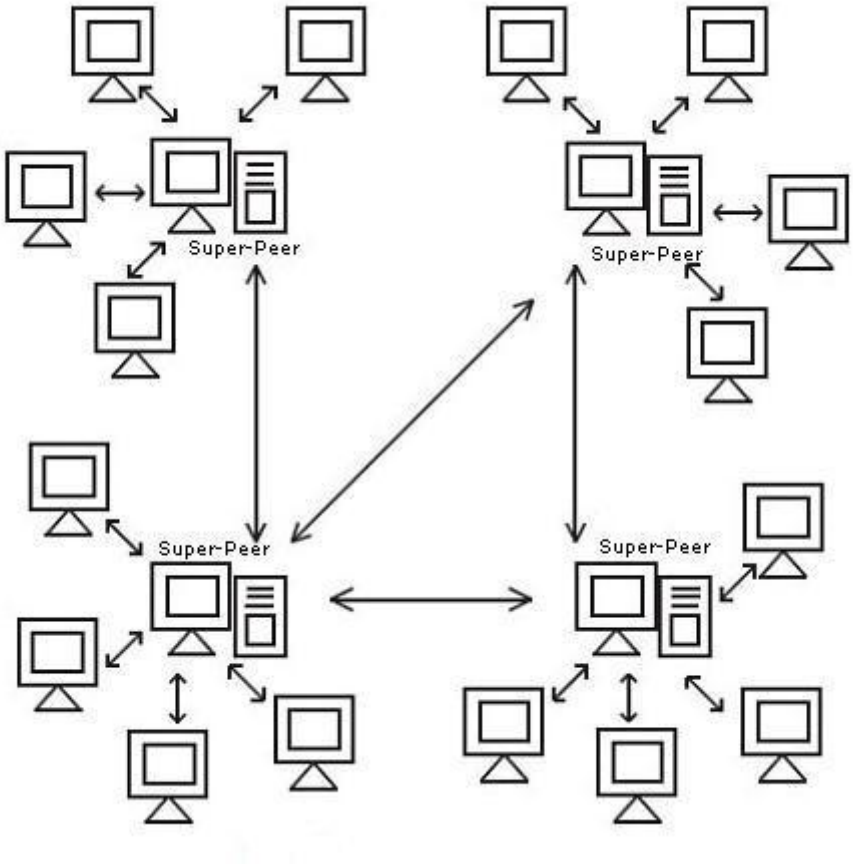

Figura 3 - Topologia s*uper-peer.* Adaptado de [Pires 2007]

Fiorano (2003) lista uma série de vantagens dessa topologia, destacadas a seguir.

Em relação à busca, é considerada mais rápida do que outras topologias porque é separada em um conjunto pequeno de s*uper-peers*, e cada um tem indexado as informações sobre seu conjunto de *peers*. Uma busca que leva o tempo O(n) em uma rede pura ou híbrida com *n peers* irá levar o tempo de O(n/m) numa rede s*uper-peer*, com *m* sendo o número de *peers* por s*uper-peer*.

Em relação à autonomia dos *peers*, na rede *super-peer*, cada cluster (conjunto de um *Super-peer* e seus *peers* relacionados) é uma unidade autônoma, que não depende de um servidor central para trocar informações.

Uma outra vantagem da topologia de rede *super-peer* está no desempenho. Na rede pura, todo *peer* tem igual responsabilidade de

processamento e conexão. Isso diminui o desempenho quando a rede tem *peers* com baixa capacidade. Esse problema é reduzido consideravelmente na topologia *super-peer* porque apenas os *peers* com alta capacidade de processamento e de largura de banda são promovidos para *super-peers*, dividindo a carga de processamento de acordo com a capacidade dos *super-peers* e, consequentemente, melhorando a desempenho da rede. Com isso, surge um problema existente nas redes híbridas: o s*uper-peer* passa a ser um ponto de falha no sistema, caso esse venha a se desconectar. Esse problema pode ser resolvido com uma eleição do *peer* com melhor capacidade conectado ao *super-peer* desconectado. Essa eleição pode ser feita por vários parâmetros, mas o mais recomendado é pela capacidade do *peer*, para não comprometer a desempenho da rede.

Com o aumento dos sistemas P2P, surgiu os *Peer Data Management Systems* (PDMS), cujas características principais serão descritas a seguir.

2.2 PDMS

Os PDMS são Sistemas de Gerenciamento de Dados *Peer-to-peer* (*Peer Data Management Systems*) que provêem um ambiente de integração de dados armazenados em fontes autônomas, heterogêneas e dinâmicas.

Um PDMS é composto por uma grande quantidade de *peers*. Cada *peer* representa uma fonte de dados e disponibiliza seus dados através de esquemas exportados, ou seja, a parte do esquema em que um *peer* deseja compartilhar com os demais *peers* [Pires 2007]. Caso algum *peer* do PDMS venha a ficar indisponível, os outros *peers* irão continuar a funcionar, sem que a rede fique inoperante. Isso porque o mapeamento entre esquemas é descentralizado, evitando a inviabilidade da rede. Os *peers* são considerados autônomos, pois não há controle sobre a entrada ou saída deles, nem mesmo sobre as alterações dos esquemas de suas bases de dados.

Um diferencial entre os PDMS e os outros sistemas de integração de dados é a substituição de um único esquema global por uma coleção de mapeamentos semânticos entre os esquemas de cada *peer*, que representam as fontes de dados.

Estes sistemas possuem as seguintes características: compartilhamento de dados descentralizado, processamento e armazenamento de dados distribuídos em *peers* autônomos e escalabilidade. A escalabilidade é necessária devido à grande quantidade de *peers* que compõem a rede. A rede deverá manter serviços de buscas e indexação sem que haja um grande custo computacional associado a cada operação.

3 SPEED

Nessa seção será abordado o sistema SPEED [Pires 2007], com sua arquitetura e seus principais componentes.

O sistema SPEED é um PDMS que adota uma topologia de rede mista, utilizando uma rede DHT e uma rede *super-peer*. Ele vem sendo pesquisado e desenvolvido com o propósito de prover soluções para problemas críticos de gerenciamento de dados em ambientes *peer-to-peer*, como conectividade, mapeamentos entre esquemas, processamento de consultas e qualidade de serviços. Para isso, utiliza semântica como base para o desenvolvimento e gerenciamento de seus serviços [Souza, D. 2007].

A seguir, serão descritos a arquitetura do SPEED e os conceitos de todos os seus componentes.

3.1 Arquitetura do SPEED

A arquitetura do Sistema SPEED (***S***emantic ***P****eer* ***D***ata ***M***anagement ***S***ystem) foi formulada por [Pires 2007] e é composta por três diferentes categorias de *peers* e relacionamentos entre eles: os *peers* semânticos, os *peers* de integração e os *peers* de dados. A arquitetura do SPEED pode ser observada na Figura 4.

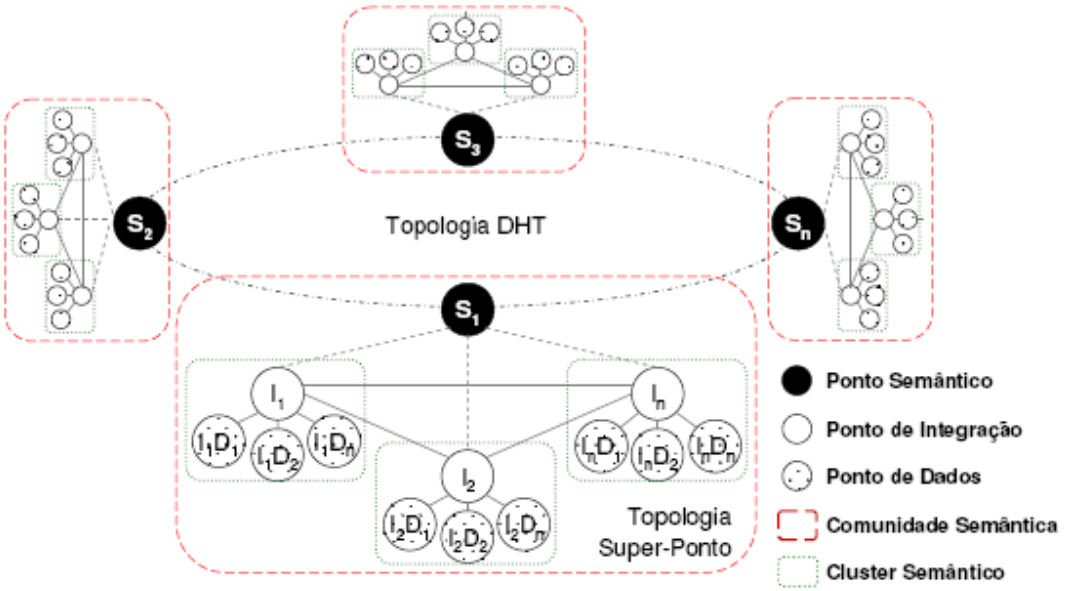

Figura 4 - Arquitetura do SPEED [Pires 2007]

O SPEED foi projetado com duas topologias de redes distintas: a DHT como um anel superior e mais abstrato, que organiza os *peers* semânticos, e a *Super-peer* para o gerenciamento de pontos de integração e pontos de dados. A topologia DHT foi abordada em Souza, G. (2007) e forma a camada de mais alto nível do sistema. Ela tem a funcionalidade de localizar e encaminhar os novos *peers* para a comunidade semântica, de acordo com a função de *hash* [Souza, G. 2007].

Os *peers* semânticos, representados na Figura 4 por S_1, S_2, S_3 e S_n, servem como pontos de entrada para comunidades semânticas e atuam como servidores de ontologias padrões específicas de domínios. Estas ontologias de domínio são usadas para enriquecer semanticamente esquemas exportados de *peers* de dados e distribuir eficientemente *peers* de dados dentro de comunidades semânticas e *clusters* [Souza, D. 2007].

Os *peers* de integração estão ilustrados por I_1, I_2 e I_n na Figura 4. Eles são *peers* de dados diferenciados em um *cluster* semântico por oferecer melhores recursos computacionais. Estes *peers* possuem um conhecimento detalhado sobre os *peers* de dados de um *cluster* [Pires 2007]. Os *peers* de integração se comunicam tanto com os outros *peers* de integração quanto com os *peers* de dados que estão conectados a eles. A necessidade de possuir melhores recursos computacionais é justificada pela maior capacidade

computacional exigida para efetuar controle e processamento de consultas sobre os *peers* de dados.

Os *peers* de dados são quaisquer *peers, e cada um* corresponde a uma fonte de dados cujas informações são compartilhadas com os outros *peers* mediante o estabelecimento de mapeamentos semânticos [Pires 2007]. Os *peers* de dados são os locais onde são executadas as consultas. Estes *peers* podem ser visualizados na Figura 4 sob a legenda de I_1D_1, I_1D_2, I_1D_n, I_1D_1, I_1D_2, I_1D_n, I_nD_1, I_nD_2 e I_nD_n.

Um *cluster* semântico é um conceito lógico associando um conjunto de *peers* de dados a um *peer* de integração com interesses semânticos semelhantes. Cada *cluster* semântico está relacionado a um interesse comum entre *peers* de dados e possui um *peer* de integração que é responsável por tarefas como indexação de metadados, processamento de consultas e integração dos dados [Souza, D. 2007]. Os clusters servem para prover um ambiente estável onde serão aplicadas as técnicas de associação dos esquemas exportados.

Já a comunidade semântica é o mais alto nível de um conjunto semântico. Ela é um conceito lógico para o agrupamento de *clusters* semânticos que possuem interesses semânticos em comum.

3.2 Principais Componentes

Nesta seção será estudada a composição dos principais componentes da arquitetura do sistema SPEED. A Figura 5 mostra os diferentes papéis que os *peers* podem assumir no sistema.

Os *peers* de dados correspondem a fontes de dados, que compartilham esses dados com os outros *peers* da rede. Um *peer* de dados é composto de uma camada P2P, que faz a comunicação com um ponto de integração. A camada de acesso aos dados é composta por um *wrapper*, responsável por traduzir dados da fonte para o modelo de dados comum e pela tradução de consultas para a linguagem específica do ponto, e pela fonte de Dados, onde são armazenados os dados locais a serem compartilhados de acordo com o esquema exportado.

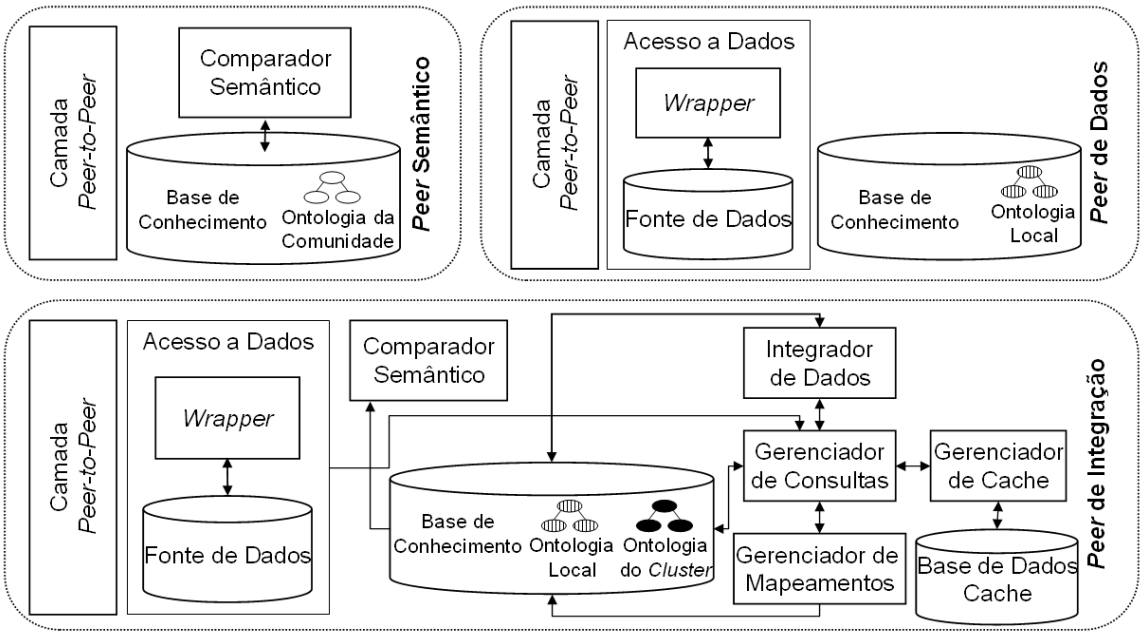

Figura 5 - Principais componentes do SPEED. Adaptado de [Pires 2007]

Um *peer* de integração também é um ponto de dados, mas, por ter mais recursos computacionais, tem responsabilidades diferentes dentro de um cluster semântico. Ele possui um conhecimento detalhado dos *peers* de dados associados a ele, que é mantido em uma Base de Conhecimento (BC). Essa base armazena informações para fazer a associação de esquemas exportados pelos *peers* de dados (mapeamentos entre esquemas). No gerenciador de mapeamentos, o *peer* de integração armazena e gerencia os mapeamentos entre a ontologia dos outros *peers* de integração na comunidade semântica. O *peer* de integração é composto também de um gerenciador de consultas, que processa as consultas enviadas para ele e envia para os *peers* de dados capazes de responder à consulta, além de enviar a consulta para outros *peers* de integração. Possui também um integrador de dados, que integra resultados da consulta retornados dos pontos de dados e dos pontos de integração e envia o resultado final para o ponto que originou a consulta.

O *peer* semântico atua como um servidor de ontologias. Estas ontologias são utilizadas para distribuir pontos de dados dentro de comunidades semânticas e dos clusters. O ponto semântico é o ponto de entrada para uma comunidade, realizando uma comparação entre a ontologia que representa a

ontologia local do *peer* que está se conectando ao sistema com a ontologia do domínio de cada comunidade.

4 A implementação.

A fase de implementação para o projeto foi dividida em três partes: primeiro com o estudo dos simuladores de redes *Super-peers* existentes. Com base nesse estudo e na arquitetura do sistema SPEED, foi elaborada uma arquitetura para a implementação. E, por fim, a implementação da rede propriamente dita.

4.1 Simuladores

Os simuladores são as ferramentas úteis para se simular as redes e as aplicações *peer-to-peer*, tentando deixar os resultados da simulação próximos da realidade. Isso porque existe um alto custo para montar um ambiente *peer-to-peer* real para realização de testes em larga escala, com quantidade de máquinas suficientes e capacidade de banda e de processamento. Esse estudo dos simuladores existentes foi realizado em conjunto com [Souza, G. 2007], para a implementação de camada de rede DHT do SPEED [Pires 2007].

Existem vários simuladores de rede *peer-to-peer*, tendo sido os mais conhecidos [Brown 2006] analisados. Eles foram os seguintes: Narses [Narses 2003], 3LS [Ting 2003], P2PSim [P2PSim 2005], NeuroGrid [Neurogrid 2001], PlanetSim [García et al. 2005], PeerSim [PeerSim 2005], OMNet++ [OMNet++ 2006], NS2 [NS-2 2007] e SSFNet [SFFNet 2004].

Para a avaliação dos simuladores, foram levados em consideração fatores como a disponibilidade de código fonte, linguagem de programação, sistema operacional, escalabilidade, documentação existente, além, é claro, da proposta do simulador, ou seja, o que ele pretende simular.

Com relação à simulação da rede *Super-peer*, o simulador escolhido para um estudo mais aprofundado foi o PeerSim [PeerSim 2005], um projeto *open-source* desenvolvido na Universidade de Paderborn, Alemanha.

O estudo realizado no simulador, com base na documentação existente e no código fonte do projeto, mostrou que o Peersim realiza simulações para fazer o balanceamento da rede s*uper-peer* a partir de uma rede pura, gerada randomicamente com um número *n* de *peers*, e iniciando com um número aleatório de s*uper-peers*. A simulação é dividida em ciclos, realizando a eleição e/ou rebaixamento dos s*uper-peers* com base em um atributo que generaliza a capacidade de processamento e de largura de banda do *peer*. Esse algoritmo de balanceamento desenvolvido em [Monstreso 2004] garante que na medida em que acontece a eleição dos s*uper-peers*, a rede estará balanceada em um número finito de ciclos do algoritmo, inclusive, com possibilidade de retirada aleatória de alguns *peer* durante a simulação, acrescentando assim o fator da dinamicidade e escalabilidade da rede *peer-to-peer*.

Verificou-se então que a simulação realizada pelo Peersim, não se adequa à proposta do SPEED, pelos seguintes pontos: a rede s*uper-peer* é gerada randomicamente, não tendo um controle sobre quais *peers* estarão sobre responsabilidade de cada s*uper-peer*, ficando difícil a visualização dos clusters; da mesma forma, a retirada de *peers* da rede ocorre aleatoriamente, não tendo a possibilidade de escolha dos *peers* e/ou s*uper-peers* que deixarão a rede; não existe o conceito de clusters por domínio, ou seja, cada cluster é formado sem nenhuma classificação e, no balanceamento, um *peer* que esteja em um cluster pode vir a se tornar um s*uper-peer* de outro cluster apenas pela característica de capacidade.

Concluiu-se que, de acordo com a proposta do SPEED, descrita no Capítulo 3, a utilização do Peersim para simular a rede *Super-peer* do SPEED seria bastante demorada, com a necessidade de várias mudanças no simulador, o que faria com que a proposta inicial do simulador ficasse perdida, não tendo a garantia de que os resultados seriam relevantes para a implementação do SPEED.

Como solução para a implementação da rede s*uper-peer* para o SPEED, surgiu então a idéia, por parte do aluno de Mestrado do CIn Juliano Freitas e que participa do grupo de estudos BD e P2P, a utilização de um trabalho por ele realizado para a disciplina de mestrado Sistemas Distribuídos, que será descrito no próximo seção.

4.2 O Kaleph

O sistema Kaleph [Kaleph 2007] foi desenvolvido por alunos de mestrado do CIn para a disciplina de Sistemas Distribuídos, e tem como objetivo criar um PDMS com base numa topologia pura. Os *peers* do Kaleph são pontos de dados, com suas respectivas bases e que realizam consultas na rede, através dos mapeamentos entre esquemas dos outros *peers*. Esse mapeamento é realizado de forma manual, e armazenado em arquivo XML. A Figura 6 ilustra a visão geral do sistema Kaleph.

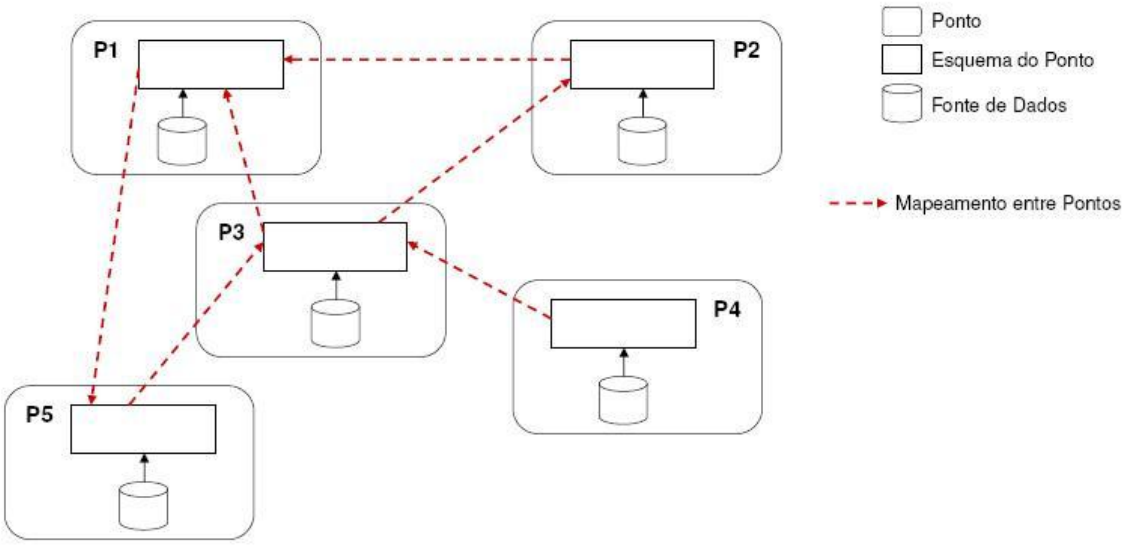

Figura 6 - Sistema Kaleph. Visão geral [Kaleph 2007]

O Kaleph foi desenvolvido na linguagem Java [Java 2007] e para a comunicação entre *peers*, foi utilizado RMI [Java 2007], uma abordagem dessa tecnologia para prover as funcionalidades de objetos distribuídos. Os peers se conectam aos outros exportando seus esquemas e realizando os mapeamentos entre eles.

A estrutura de um *peer* do sistema Kaleph está ilustrada na Figura 7. Em seguida, cada camada do *peer* será explicada.

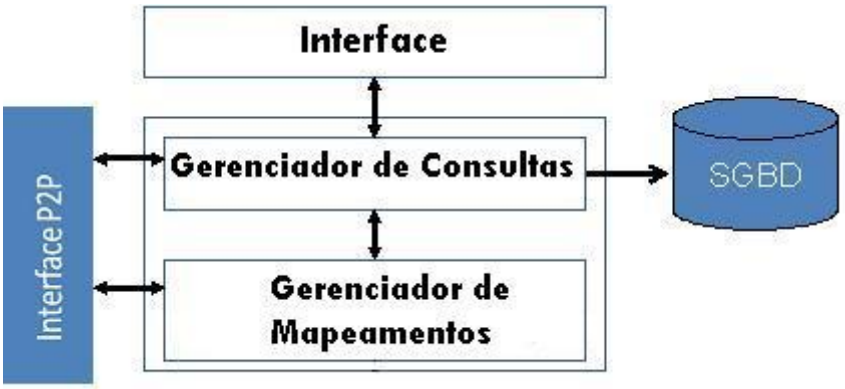

Figura 7 - Estrutura de um *peer* no sistema Kaleph [Kaleph 2007]

A interface com o usuário é responsável pela realização das consultas no *peer*, que serão executadas na base local e enviadas para os outros *peers* via rede. O usuário não tem conhecimento de quantos *peers* estão conectados a ele.

O gerenciador de consultas é responsável pela execução de consultas na base local do *peer* e pela reescrita de consultas enviadas para outros *peers*, além de integrar os resultados para serem exibidos ao usuário.

O gerenciamento de mapeamentos está responsável pelo gerenciamento de *peers* conhecidos, guardando os endereços *ip* dos *peers,* e pela construção dos mapeamentos entre eles, que é feito de forma manual. Esse mapeamento é feito a partir do esquema exportado pelo *peer*, podendo ser apenas um subconjunto do esquema de dados. Tanto o esquema exportado quanto os mapeamentos são armazenados em arquivos XML. Esses mapeamentos são direcionados, ou seja, se o *peer* 1 tem o mapeamento para o *peer* 2, a consulta parte apenas do *peer* 1 para o *peer* 2. Para o *peer* 2 consultar os dados do *peer* 1, ele deve ter o mapeamento do *peer* 2 para o *peer* 1. No SGBD são armazenados os dados para consulta local e as consultas enviadas pelos outros *peers* via interface P2P.

A interface P2P realiza o envio e recebimento de consultas e de resultados dos outros *peers*. O Kaleph foi desenvolvido para a topologia de rede pura não estruturada, que utiliza como técnica de consulta o *flooding*, além das outras características explicadas no tópico 2.1.1.

O Kaleph tem então uma proposta de rede muito próxima dos *peers* de dados do sistema SPEED, e foi utilizado para implementação da rede *Super-peer*, que será descrito no próximo tópico.

4.3 Implementação da rede super-peer no Kaleph

Por questões de implementação, a arquitetura dos *peers* de integração foi simplificada. Foram abstraídas as camadas de base de conhecimento e comparador ontológico, tendo a garantia de que todos os *peers* de dados que se conectam são da mesma ontologia do cluster ao qual o *peer* de integração representa.

A Figura 8 mostra a estrutura implementada do *peer* de integração.

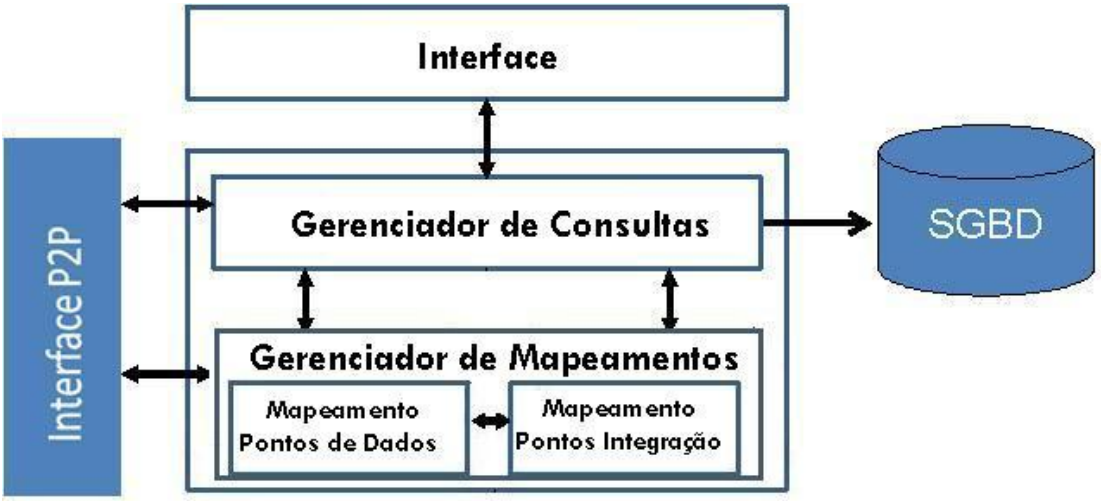

Figura 8 – Arquitetura interna de um *Peer* de integração

O gerenciador de consultas no ponto de integração continua responsável pela execução de consultas na base local e pela reescrita de consultas enviadas para os *peers* de dados e os *peers* de integração que possam estar conectados a ele, além de integrar os resultados para serem devolvidos ao *peer* que solicitou a consulta.

O gerenciamento de mapeamentos foi dividido em duas partes: os mapeamentos dos *peers* de dados, que estejam conectados a esse *peer* de

integração; e os mapeamentos com os esquemas dos outros *peers* de integração que estiverem na mesma comunidade semântica.

Para a implementação do *peer* de dados não foram necessárias simplificações. A Figura 9 mostra a arquitetura interna do *peer* de dados para a implementação

Figura 9 – Arquitetura interna de um *Peer* de dados

No *peer* de dados não existe um gerenciador de consultas, existindo apenas o executor das consultas. Todo o gerenciamento ocorre na camada de gerenciamento de consultas no *peer* de integração, ou seja, o envio, o recebimento e a integração dos resultados. O *peer* de dados também não tem um gerenciador de mapeamentos, já que estes são gerados apenas nos *peers* de integração. A Figura 10 mostra a interface inicial do sistema para um *peer* de dados.

Um *peer* de dados se conecta ao sistema por meio de um *peer* semântico, como explicado no tópico 3.1. A entrada de um *peer* de dados foi simulada em [Souza, G. 2007], e o resultado é o ip de um *peer* de integração do cluster. Para essa implementação, levou-se em consideração que o *peer* de dados já conhece o ip do *peer* de integração, se conectando e enviando seu esquema.

Para a realização dos testes do sistema, foram utilizados duas fontes de dados, utilizando o SGBD MySQL [MySQL 2007] para criação das bases relacionais.

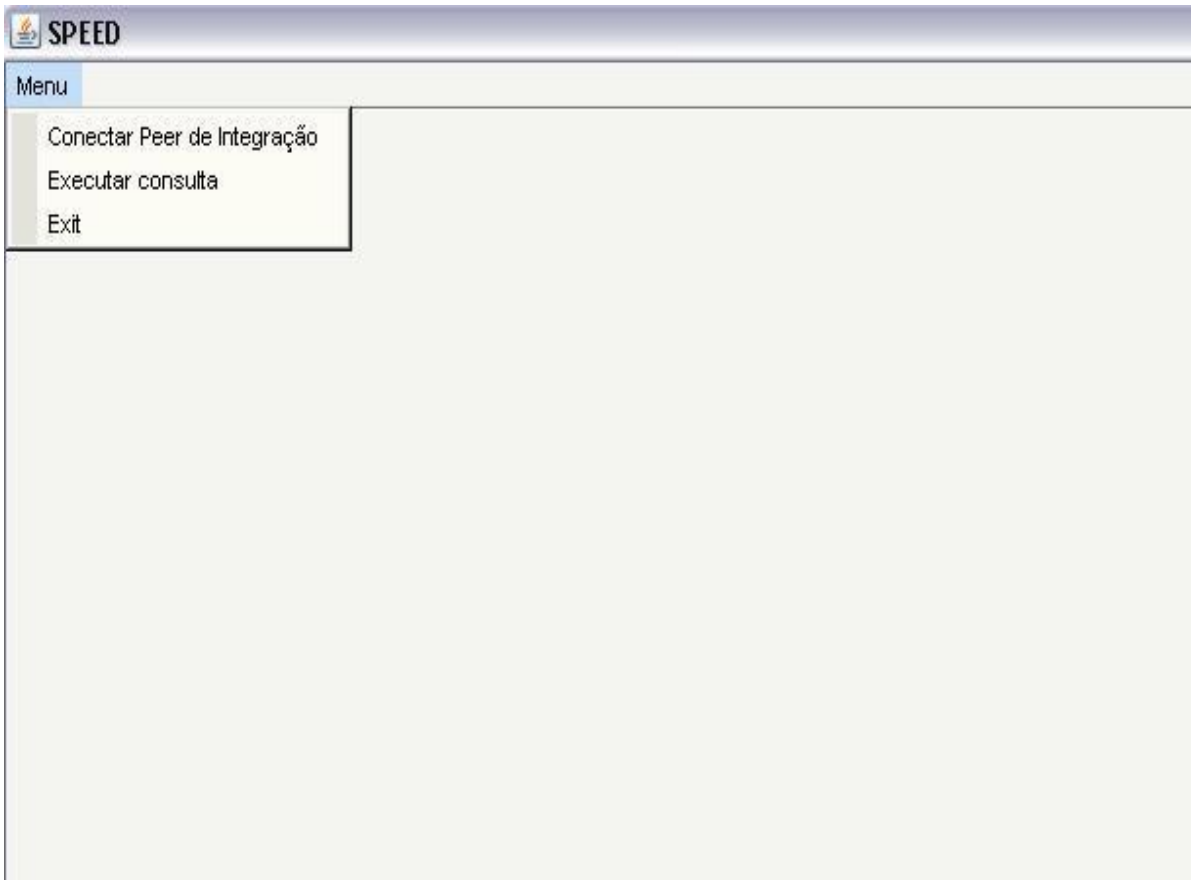

Figura 10 - Tela inicial do Sistema SPEED em um *peer* de dados

O *peer* de integração então faz o mapeamento do seu esquema com o esquema exportado do *peer* de dados. As Figura 11 e Figura 12 mostram as telas de conexão do *peer* de dados com o *peer* de integração. Quando ocorre a conexão, o *peer* de integração recebe o esquema do *peer* de dados, um arquivo XML, e armazena em uma lista de *peers* não mapeados, para que possa ocorrer o mapeamento manualmente.

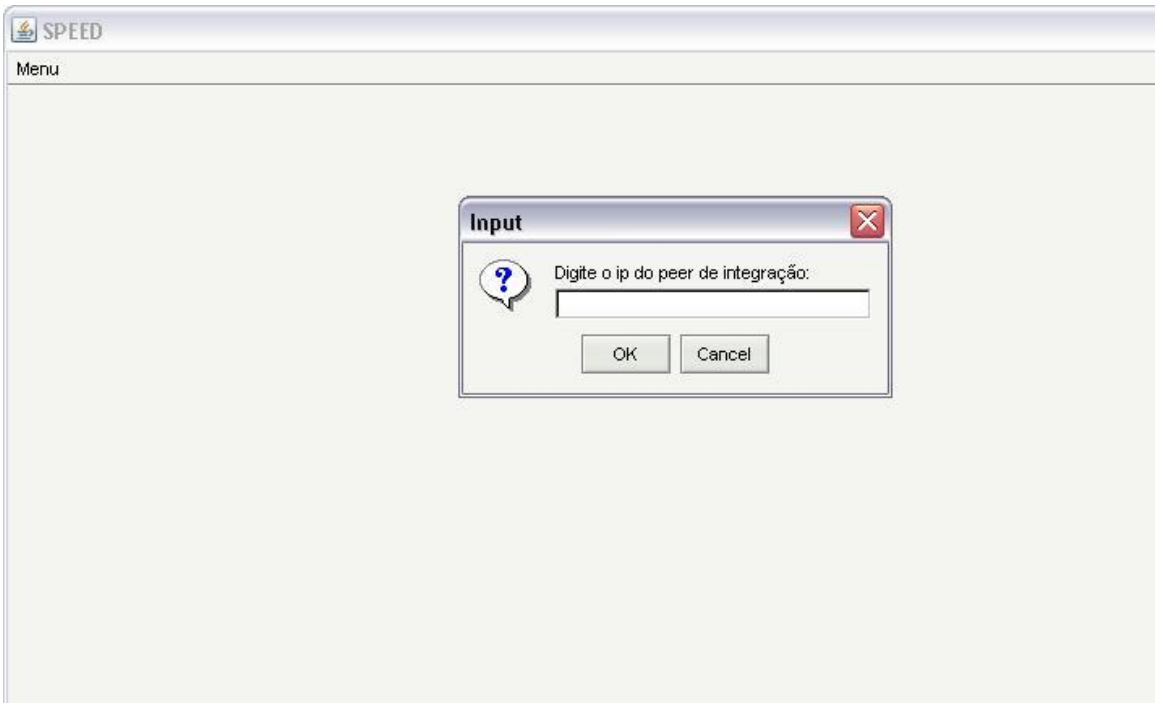

Figura 11 - Tela de conexão com *peer* de integração

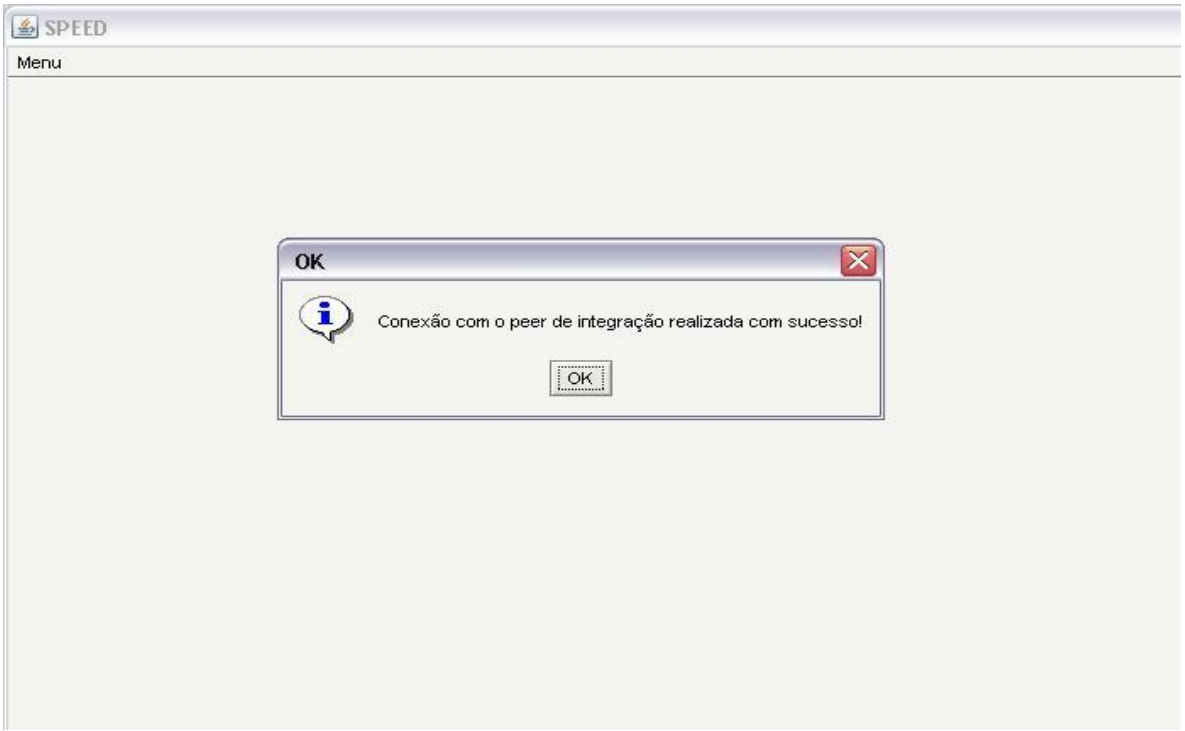

Figura 12 - Tela de confirmação

Caso seja executada uma consulta pelo *peer* de dados, como o mapeamento com o *peer* de integração ainda não foi efetuado, o resultado da

consulta se restringe apenas à fonte do *peer* de dados. A Figura 13 mostra o resultado de uma consulta sem o mapeamento realizado pelo *peer* de integração. Na base do *peer* de dados estão as informações de *instituição*, *nome* e *sexo* da tabela *professor*. Esse peer possui apenas informações sobre os professores da UPFE.

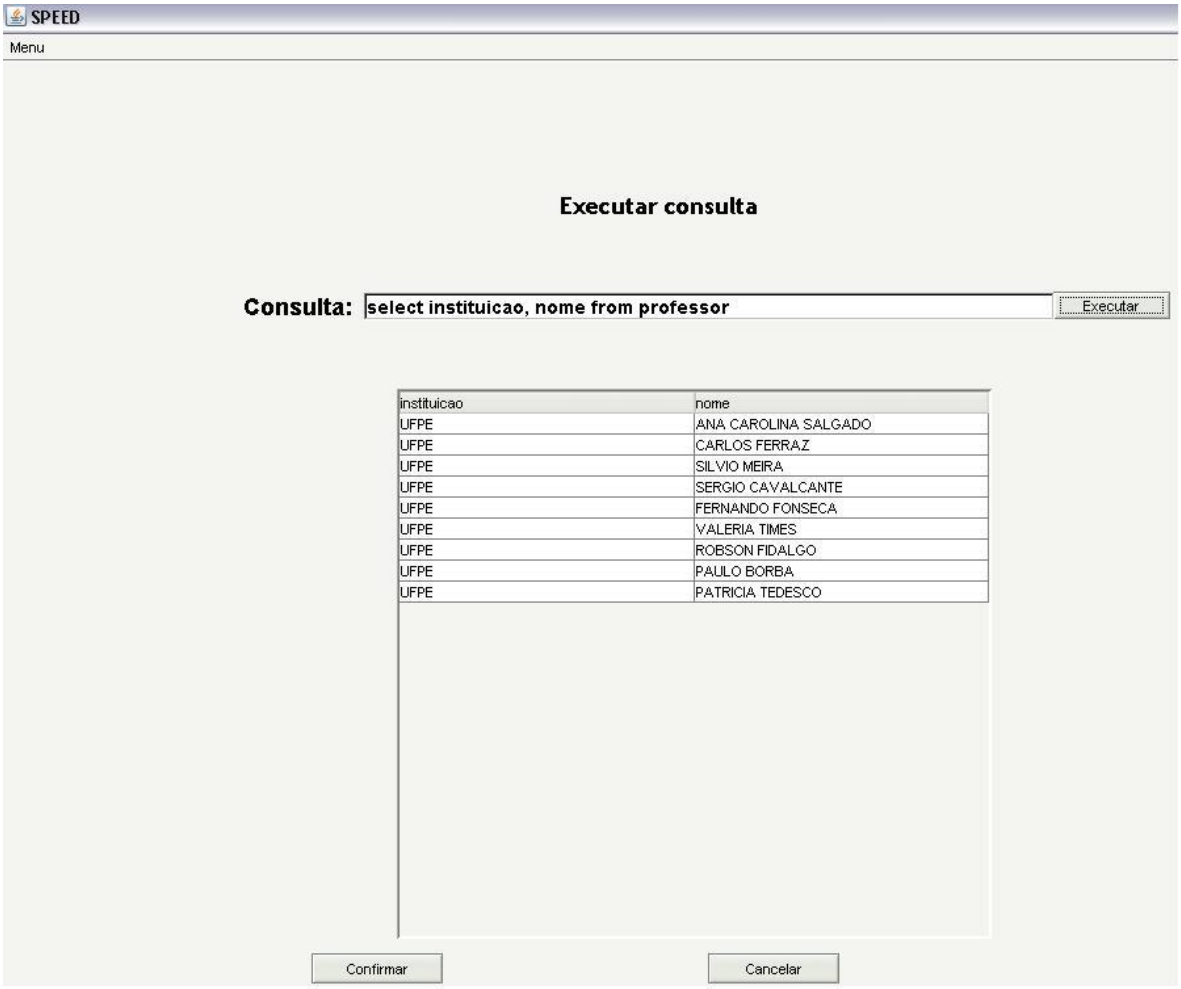

Figura 13 - Consulta na base local

As únicas tarefas que podem ser realizadas por um *peer* de dados são a conexão com um *peer* de integração, a execução de uma consulta enviada pelo *peer* de integração e o recebimento do resultado da consulta feito por ele ao *peer* de integração. A diferença na interface entre um *peer* de dados e um *peer* de integração é a possibilidade de fazer o mapeamento entre o *peer* de dados e o *peer* de integração, como também entre *peers* de integração. A Figura 14 mostra a tela inicial para o *peer* de integração.

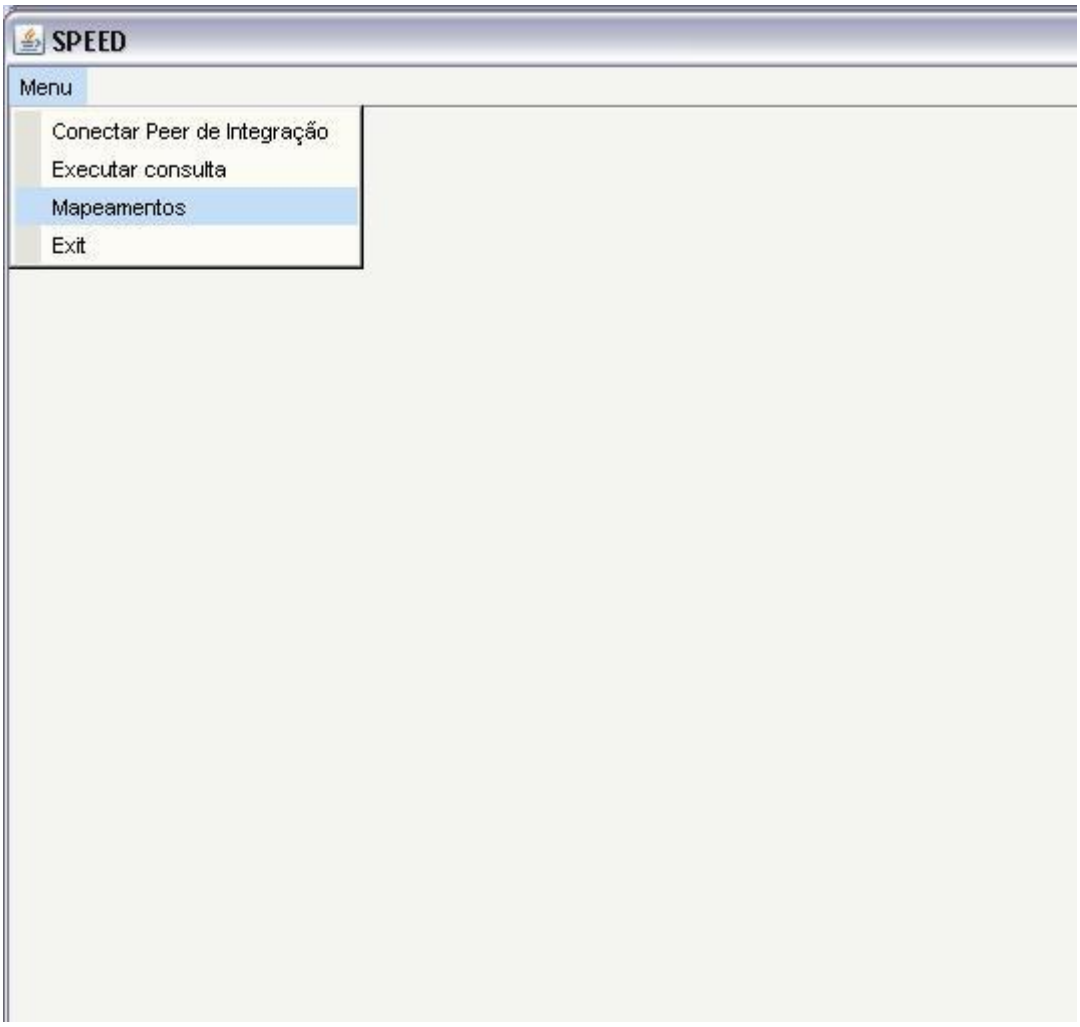

Figura 14 - Tela inicial do *peer* de integração

Como o mapeamento está sendo realizado manualmente, o *peer* de integração tem que escolher um *peer* por vez, dentre os que solicitaram conexão com ele, para fazer o mapeamento. Escolhe-se então o ip do *peer* de integração, e posteriormente quais as tabelas e colunas que são relacionadas com os dois esquemas existentes: o esquema do *peer* de integração, que representa a ontologia do cluster, e o esquema do *peer* de dados ou de outro *peer* de integração, que esteja na mesma comunidade semântica. A Figura 15 mostra a realização do mapeamento entre o *peer* de dados que solicitou conexão e o *peer* de integração. No peer de integração estão as informações de *nome*, *sexo* e *instituição* da tabela *docente*. Esse peer possui apenas informações sobre os docentes da UNICAMP.

Os mapeamentos entre os *peers* de dados e de integração são direcionados apenas do *peer* de integração para o *peer* de dados. No caso dos mapeamentos entre *peers* de integração, esse deve ser feito nas duas direções.

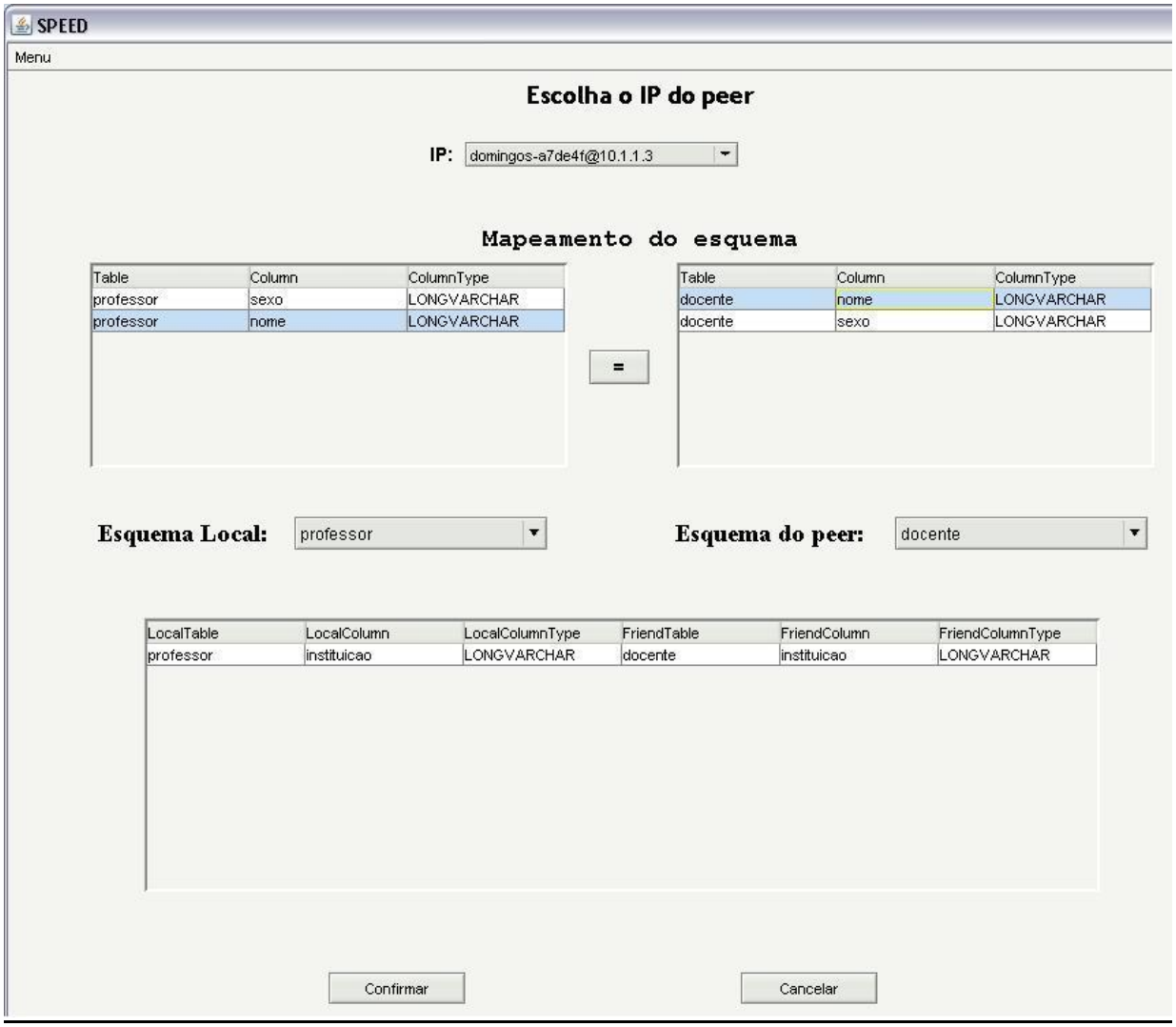

Figura 15 - Tela de Mapeamento

Para realizar o mapeamento, são escolhidas as tabelas dos dois esquemas: do *peer* de dados que solicitou a conexão e do *peer* de integração; no caso, as tabelas escolhidas foram *professor* e *docente*, respectivamente. As colunas das duas tabelas são exibidas para que o usuário faça o mapeamento. Depois de realizado, o *peer* de dados agora pode realizar uma consulta recebendo também os dados do *peer* de integração. A Figura 16 mostra a conclusão do mapeamento.

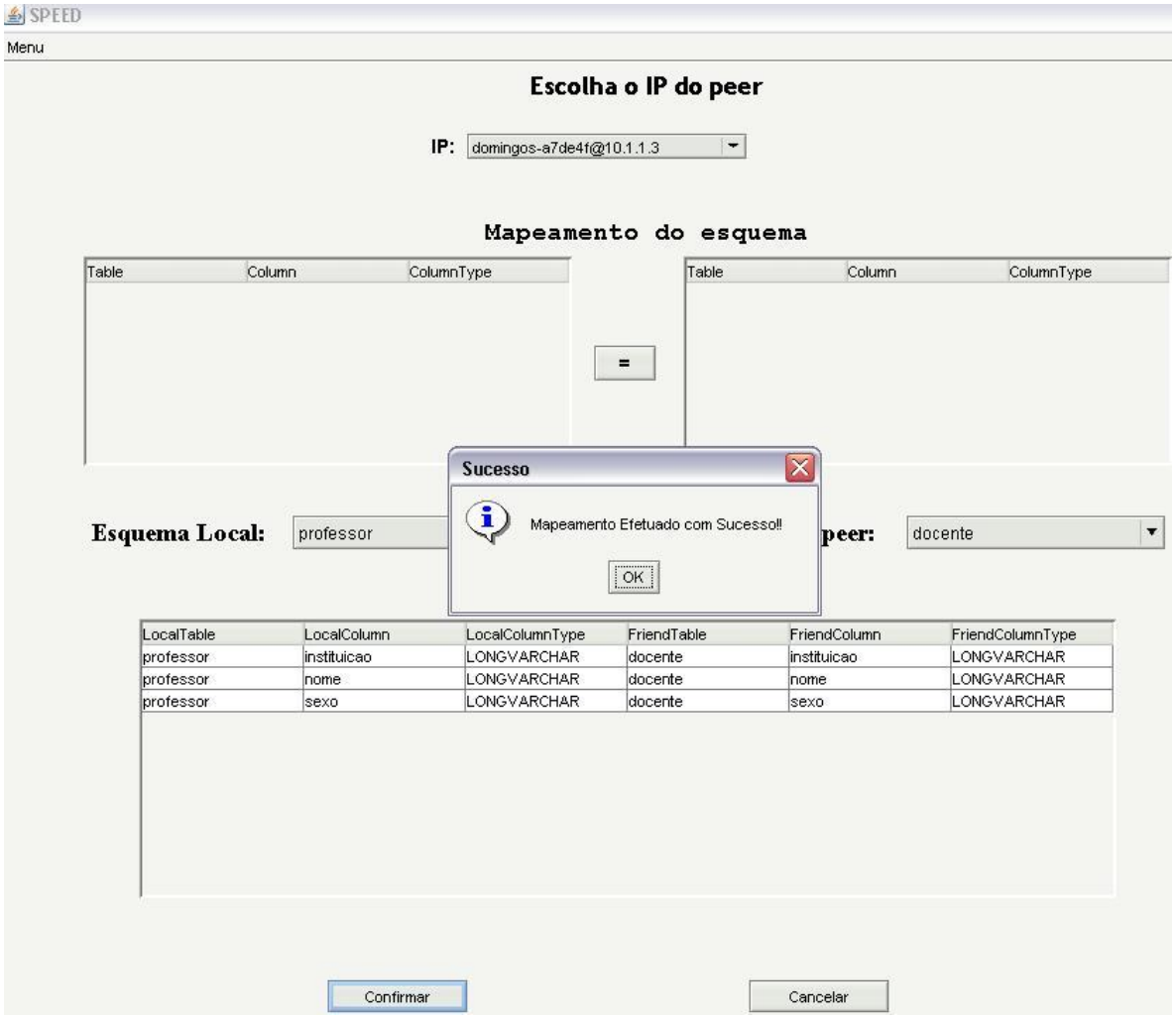

Figura 16 - Conclusão do Mapeamento

Quando o mapeamento é realizado, é gerado então um arquivo XML (Figura 17) que fica armazenado no *peer* de integração. Esse arquivo tem as tabelas mapeadas com suas respectivas colunas relacionadas.

```
<?xml version="1.0" encoding="UTF-8"?>
<node>
  <ip>domingos-a7de4f/10.1.1.3</ip>
  <tabela local="professor" remota="docente">
    <coluna local="instituicao" remota="instituicao"/>
    <coluna local="nome" remota="nome"/>
    <coluna local="sexo" remota="sexo"/>
  </tabela>
</node>
```

Figura 17 - Arquivo XML de mapeamento

Com o arquivo de mapeamento, o gerenciador de consulta do *peer* de integração consegue fazer a reescrita da consulta para enviá-la aos *peers* conectados a ele.

Se a mesma consulta for realizada novamente pelo *peer* de dados, agora mapeado pelo *peer* de integração, a consulta será reescrita baseada no arquivo XML de mapeamento e executada nos *peers* de dados que possam respondê-la. O *peer* de integração, então, faz a reescrita da consulta, recebe os dados e faz a integração das respostas, que para a implementação, foi apenas concatenar as duas respostas, como exibido na Figura 18.

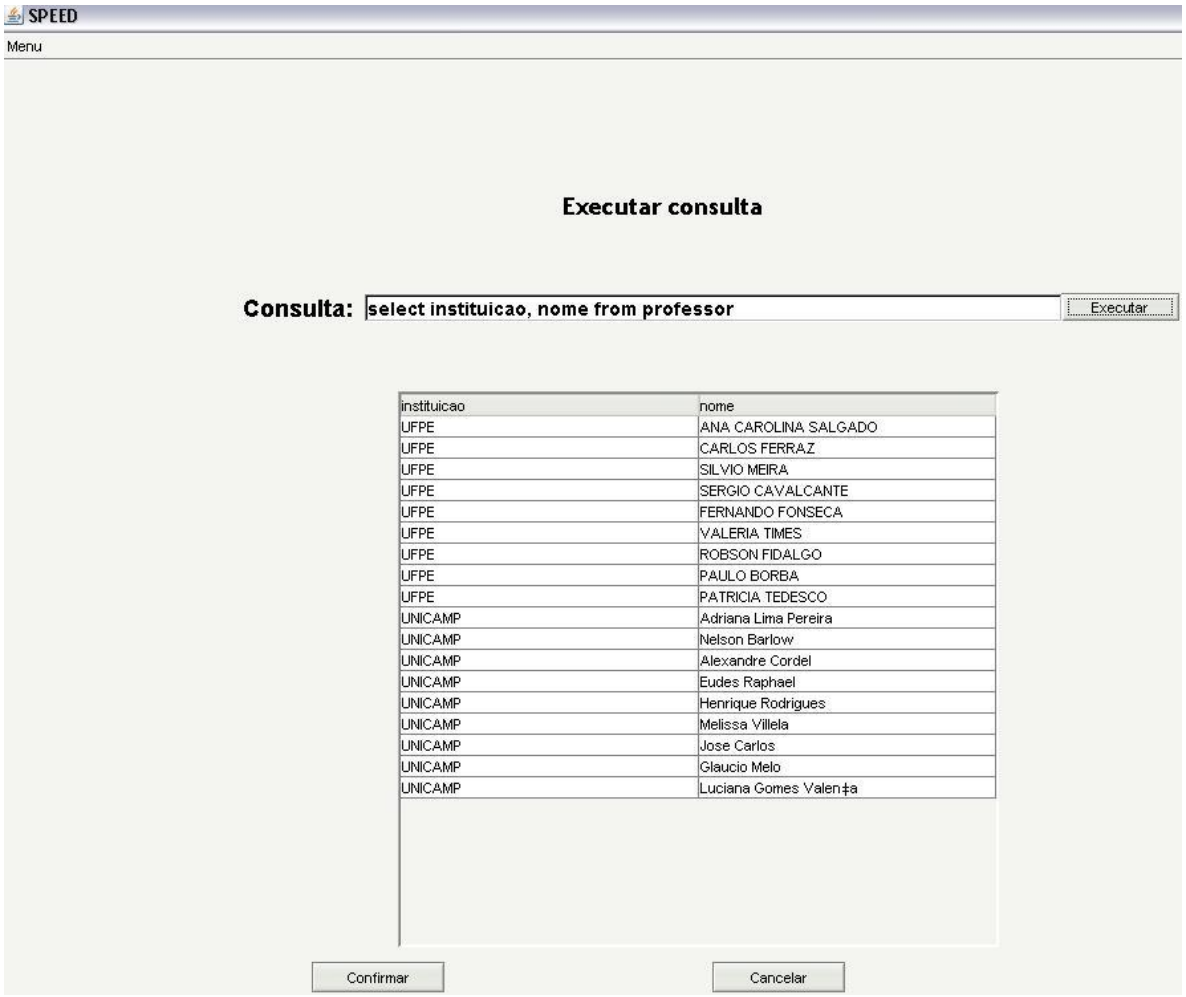

Figura 18 - Consulta realizada na rede

No *peer* de integração foi definido um tempo de espera para o recebimento das respostas. Isso porque, como os *peers* são autônomos, a qualquer momento um *peer* de dados pode deixar a rede, e se o *peer* de

integração ficasse aguardando uma resposta, o desempenho do sistema seria comprometido. Para a realização dos testes, foi utilizado o tempo de 10 segundos para aguardar uma resposta de um *peer* de dados, sendo esse valor ajustável, dependendo do tamanho da rede.

Caso um *peer* de dados venha a se desconectar da rede, cabe ao *peer* de integração refazer e/ou desfazer o mapeamento, agora sem as tabelas do *peer* ausente. Cada *peer* de integração armazena o número de *peers* de dados a ele conectados. Com esse valor, é possível calcular o percentual de carga de processamento utilizado pelo *peer* de integração. Esse percentual será utilizado para fazer o balanceamento da rede, ocorrendo a distribuição dos *peers* de dados entre os *peers* de integração, ou com a eleição de *peer* de dados para um *peer* de integração.

Devido à restrição de tempo para desenvolver este trabalho, o remapeamento dos *peers* com a saída de um deles e o balanceamento da rede quando atingido o percentual de carga do peer de integração não foram implementados.

5 Conclusões e Trabalhos Futuros

Neste trabalho, observamos as principais características de *peer-to-peer* e PDMS que fundamentaram a implementação da rede *super-peer*, com os *peers* de integração e de dados que serão utilizados no sistema SPEED.

5.1 Contribuições

O estudo dos simuladores de redes *peer-to-peer* foi importante para ter uma visão mais clara da implementação da rede *super-peer*, mesmo esses não tendo sido utilizados propriamente para a implementação do trabalho.

A utilização do Kaleph como base dessa implementação facilitou o andamento do trabalho, por ser um sistema em português, modularizado, de fácil entendimento e com características próximas ao SPEED. A implementação da rede *super-peer* mostrou que é possível o desenvolvimento de um PDMS com a utilização de integração de esquemas exportados, da

forma que o SPEED propõe [Pires 2007]. Este trabalho foi implementado de uma forma extensível, pois deve integrar-se com o restante da topologia do SPEED, que será implementada no decorrer desse projeto.

5.2 Dificuldades Encontradas

A primeira dificuldade encontrada foi encontrar um simulador de rede que utilizasse a topologia *super-peer*, para que fosse realizada uma simulação, ao invés de implementação da rede. Como apenas um simulador entre os pesquisados utilizava a rede *super-peer*, uma segunda dificuldade foi a de entender o funcionamento dele, devido a pouca documentação da implementação do simulador. Depois de constatado que uma modificação no simulador para se adequar aos objetivos e arquitetura do SPEED levaria um tempo superior ao proposto por esse trabalho e que não seria uma grande contribuição ao SPEED, partiu-se então para a implementação da rede com base no Kaleph.

5.3 Trabalhos Futuros

O desenvolvimento da rede *super-peer* não está completo, pois algumas simplificações foram necessárias para a realização desse trabalho. Sendo assim, o desenvolvimento da rede deve continuar, com a implementação do balanceamento de rede. Paralelamente a esta implementação, a camada que utiliza uma rede DHT foi desenvolvida [Souza, G. 2007] no simulador PlanetSim [PlanetSim 2005]. Na próxima fase de implementação do SPEED, deve ocorrer uma integração das duas topologias desenvolvidas, que ficará facilitada com a utilização da documentação provida por estes trabalhos.

A interface gráfica do sistema é um ponto a ser aprimorado, sendo a atual estendida do Kaleph e sem nenhuma melhoria. Com uma interface gráfica melhor, o funcionamento do sistema passa a ser mais transparente para o usuário.

6 Referências

[BitTorrent 2007] BitTorrent. http://www.bittorrent.com/, último acesso em: 30/05/2007.

[Brown 2006] Brown, A., Kolberg, M., 2006. Tools for *Peer-to-Peer* Network Simulation. University of Stirling, UK.

[Eclipse 2005] Eclipse 3.1.2, 2005. http://www.eclipse.org/, último acesso em 30/06/2007.

[Fiorano 2003] Fiorano Software (2003) "*Super-peer* Architectures for Distributed Computing". White Paper, Fiorano Software, Inc.

[FreeNet 2007] FreeNet. http://freenetproject.org/, último acesso em 30/05/2007.

[García et al. 2005] García, P., Pairot, C., Mondéjar, R., Pujol, J., Tejedor, H., Rallo, R., 2005. PlanetSim: A New Overlay Network Simulation Framework. T. Gschwind and C. Mascolo (Eds.): SEM 2004, LNCS 3437, pp. 123–136.

[Gnutella 2007] Gnutella, 2007. http://www.gnutella.com/, último acesso em 30/05/2007.

[Java 2007] Java, 2007. http://www.java.com/, último acesso em 30/07/2007.

[Kaleph 2007] Kaleph, 2007. http://www.cin.ufpe.br/~dsrm/kaleph.ppt, último acesso em 30/07/2007

[Kazaa 2007] KaZaA. http://www.kazaa.com/, último acesso em 30/05/2007.

[Kolweyh 2004] Kolweyh, M. (2004). Towards Next-Generation Peer-to-Peer Systems. University of Bremen

[Monstreso 2004] Monstreso, A. 2004 "A Robust Protocol for Building Super*peer* Overlay Topologies". In Technical Report UBLCS-2004-8.

[Morpheus 2007] Morpheus. http://www.morpheus.com/, último acesso em 30/05/2007.

[MySQL 2007] MySQL 2007. http://www.mysql.org, versão 5.0, último acesso em 30/05/2007

[Napster 2007] Napster, 2007. http://www.napster.com/, último acesso em 30/05/2007.

[Narses 2003] Narses Network Simulator, 2003. http://sourceforge.net/projects/narses/, último acesso em 30/05/2007.

[Neurogrid 2001] NeuroGrid Network Simulator, 2001. http://www.neurogrid.net/, último acesso em 30/05/2007.

[NS-2 2007] The Network Simulator (NS-2), 2007. http://sourceforge.net/projects/nsnam/, último acesso em 30/05/2007.

[OMNet++ 2006] OMNeT++ is a discrete event simulation environment, 2006. http://www.omnetpp.org/, último acesso em 30/05/2007.

[P2PSim 2005] P2Psim: A Simulator for *Peer-to-Peer* (P2P) Protocols, 2005. http://pdos.csail.mit.edu/p2psim/, último acesso em 30/05/2007.

[Peersim 2005] Peersim P2P Simulator, 2005. http://Peersim.sourceforge.net/, último acesso em 15/07/2007.

[Pires 2007] Pires, C. E. S., 2007. Um Sistema P2P de Gerenciamento de Dados com Conectividade Baseada em Semântica. Monografia de Qualificação e Proposta de Doutorado. CIn, UFPE, Brasil.

[Risson 2004] Risson, J., Moors, T., 2004. Survey of research towards robust *peer-to-peer* networks: search methods. Technical Report UNSW-EE-P2P-1-1, University of New South Wales, Sydney, Austrália.

[SFFNet 2004] SSF (Scalable Simulation Framework), 2004. http://www.ssfnet.org/, último acesso em 30/05/2007.

[Souza, D. 2007] Souza, D. Y., 2007. Reformulação de Consulta Baseada em Semântica para PDMS. Monografia de Qualificação e Proposta de Doutorado. CIn, UFPE.

[Souza, G. 2007] Souza, G. B. 2007 “Implementação de uma Rede DHT para *Peers* Semânticos no Sistema SPEED”. Monografia e Proposta de Trabalho de Graduação. CIn, UFPE

[Stoica et al. 2001] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H., 200X. Chord: A Scalable *Peer-to-peer* Lookup Service for Internet Applications. MIT Laboratory for Computer Science.

[Sung et al. 2005] Sung, L. G. A., Ahmed, N., Blanco, R., Li, H., Soliman, M. A., Hadaller,D., 2005. A Survey of Data Management in *Peer-to-peer* Systems. School of Computer Science, University of Waterloo.

[Ting 2003] Ting, N. S., Deters, R., 2003. 3LS - A *Peer-to-Peer* Network Simulator. Department of Computer Science, University of Saskatchewan, Canada.

[Yang et al. 2003] Yang, B. Gacrcia-Molina, H. 2003 “Designing a *Super-peer* Network” In Proc of International Conference on Data Engineering (ICDE 03), Bangalore, India