

Universidade Federal de Pernambuco
Centro de Informática
Pós-Graduação em Ciências da Computação

Um Sistema P2P de Gerenciamento de Dados com Conectividade Baseada em Semântica

Por
Carlos Eduardo Santos Pires

Exame de Qualificação e Proposta de Tese

Orientadora: Profa. Dra. Ana Carolina Salgado

Recife (PE), Abril de 2007

Resumo

A necessidade de acesso a dados armazenados em múltiplas fontes de dados ocasionou o surgimento de inúmeros tipos de sistemas de gerenciamento de dados distribuídos: sistemas de bancos de dados distribuídos, federações de bancos de dados, bancos de dados múltiplos e sistemas de integração de dados. Recentemente, os *Peer Data Management Systems* (PDMS) aparecem como uma evolução natural destes sistemas, principalmente impulsionados pelo crescimento dos sistemas ponto-a-ponto (P2P).

Um PDMS é composto por uma quantidade excessiva de pontos. Cada ponto representa uma fonte de dados e disponibiliza seus dados através de esquemas exportados, ou seja, a parte do esquema em que um ponto deseja compartilhar com os demais pontos. Os pontos estão interligados entre si através de mapeamentos semânticos entre os esquemas exportados. As consultas podem ser formuladas em qualquer ponto do sistema e respondidas pelo ponto inicial e pelos demais pontos conectados ao sistema, desde que existam mapeamentos semânticos entre os mesmos.

Diferentes tipos de topologias de rede podem ser utilizadas para organização dos pontos em um PDMS: pura não-estruturada, pura estruturada (DHT) e super-ponto. No entanto, poucos dos PDMS propostos na literatura preocupam-se em detalhar a topologia de rede utilizada, a qual é normalmente abstraída. Além disso, não existe um consenso em relação à topologia mais apropriada para um PDMS.

Neste trabalho, propomos um PDMS baseado em uma topologia mista (DHT e super-ponto). Uma estratégia de conectividade que explora a semântica associada ao conteúdo exportado pelos pontos é utilizada para formação de comunidades semânticas de pontos. A rede DHT serve para simplificar a descoberta de comunidades semânticas por pontos que desejam conectar-se ao sistema. Em cada comunidade, os pontos são organizados segundo a topologia super-ponto, formando os clusters semânticos. Neste caso, a topologia super-ponto tem como objetivo facilitar a identificação de mapeamentos semânticos entre pontos semanticamente equivalentes. Além disso, o processamento de consultas é beneficiado pois as consultas são direcionadas apenas para clusters relevantes. Para manutenção das comunidades semânticas, são propostas também estratégias para balanceamento de carga e tolerância a falhas.

Abstract

The need for accessing data stored at multiple data sources motivated the development of diverse types of distributed data management systems: distributed database management systems, federated database systems, multidatabase systems, and data integration systems. Recently, Peer Data Management Systems (PDMS) represent a natural evolution of such systems. PDMS are mainly inspired in the growth of the peer-to-peer (P2P) systems.

A PDMS is composed by an excessive number of peers. Each peer represents a data source, and shares its data through exported schemas, i.e. the part of the schema that a peer wishes to share with other peers. Peers are connected through semantic mappings established between exported schemas. Queries can be formulated at any peer, and answered by the original peer and its connected peers, assuming the existence of semantic mappings among them.

Several types of network topologies can be used to organize peers in a PDMS: pure non-structured, pure structured (DHT), and super-peer. However, only a few number of PDMS proposed in the literature concerns on detailing the adopted network topology, which is frequently abstracted. Besides, there is not a consensus on the most appropriate network topology for a PDMS.

In this work, we propose a PDMS which employs a mixed network topology (DHT and super-peer). A connectivity strategy is proposed to form semantic communities of peers. Such strategy explores the semantic associated with the content exported by peers. The DHT network is used to simplify the discovery of semantic communities by peers wishing to connect to the system. At each community, peers are organized according to the super-peer topology, forming semantic clusters. In this case, the goal of the super-peer topology is to facilitate the identification of semantic mappings among semantically equivalent peers. Besides, query processing is improved since queries are sent only to relevant clusters. Regarding the maintenance of semantic communities, we also propose strategies for load balancing of clusters and fault tolerance.

Sumário

Capítulo 1 – Introdução	11
1.1 Motivação	11
1.2 Definição do Problema	13
1.3 Objetivo	14
1.4 Escopo	14
1.5 Estrutura do Documento.....	14
Capítulo 2 – O Paradigma Ponto-a-Ponto (P2P).....	15
2.1 Visão Geral e Terminologia.....	15
2.2 Topologias de Rede P2P	16
2.2.1 Topologia P2P Pura ou Descentralizada.....	19
2.2.2 Topologia P2P Híbrida ou Parcialmente Centralizada.....	20
2.2.3 Topologia P2P Super-Ponto.....	21
2.2.4 Comparativo entre Topologias de Rede P2P.....	22
2.3 Propriedades dos Sistemas P2P	23
2.3.1 Volatilidade.....	23
2.3.2 Auto-Organização	24
2.3.3 Tolerância a Falhas.....	24
2.3.4 Descentralização	25
2.3.5 Balanceamento de Carga	25
2.3.6 Escalabilidade.....	26
2.3.7 Roteamento	27
2.4 Sistemas P2P e Grades Computacionais	30
2.4.1 Grades Computacionais	31
2.4.2 Comparativo entre Sistemas P2P e Grades Computacionais.....	32
2.5 Classes de Aplicações P2P	33
2.5.1 Compartilhamento de Arquivos.....	33
2.5.2 Troca de Mensagens	34
2.5.3 Processamento Distribuído	35
2.5.4 Trabalho Colaborativo.....	36
2.6 Plataformas para Desenvolvimento de Sistemas P2P	36
2.6.1 A Plataforma JXTA.....	37
2.6.2 Groove Development Kit.....	39
2.7 Considerações Finais	40

Capítulo 3 – Gerenciamento de Dados em Sistemas P2P	41
3.1 Evolução dos Sistemas Gerenciadores de Dados Distribuídos.....	42
3.1.1 Sistemas Gerenciadores de Bancos de Dados Distribuídos.....	43
3.1.2 Sistemas de Bancos de Dados Federados.....	44
3.1.3 Sistemas de Bancos de Dados Múltiplos	45
3.1.4 Sistemas de Integração de Dados	46
3.1.5 Sistemas P2P de Gerenciamento de Dados	50
3.1.6 Comparativo entre Sistemas Gerenciadores de Dados Distribuídos.....	51
3.2 Arquitetura Genérica de um Ponto em um PDMS	53
3.3 Problemas do Gerenciamento de Dados em Sistemas P2P.....	54
3.3.1 Identificação de Mapeamentos Semânticos entre Esquemas	55
3.3.2 Processamento de Consultas	59
3.4 Exemplos de PDMS	61
3.4.1 Piazza.....	61
3.4.2 PeerDB.....	63
3.4.3 XPeer.....	66
3.4.4 Edutella	67
3.4.5 Outros PDMS.....	69
3.4.6 Estudo Comparativo entre PDMS	70
3.5 Considerações Finais	73
Capítulo 4 – Conectividade Baseada em Semântica em PDMS.....	74
4.1 Motivação	75
4.2 Comunidades Semânticas de Pontos.....	76
4.3 Semantic Overlay Networks.....	78
4.4 Semantic Overlay Clusters	82
4.5 PDMS com Conectividade Baseada em Semântica.....	84
4.5.1 Framework H ³	84
4.5.2 PARIS.....	88
4.5.3 Humboldt Discoverer.....	90
4.5.4 Mapster.....	92
4.5.5 Análise Comparativa dos PDMS	94
4.6 Considerações Finais	94
Capítulo 5 – Proposta de Tese	95
5.1 O SPEED.....	95
5.1.1 Visão Geral da Arquitetura.....	96
5.1.2 Arquitetura Interna dos Pontos	98
5.2 Conectividade no SPEED	101

5.2.1 Interesse de um Ponto Solicitante	102
5.2.2 Descoberta da Comunidade Semântica	103
5.2.3 Descoberta do(s) Cluster(s) Semântico(s)	104
5.3 Balanceamento de Carga nos Clusters Semânticos	107
5.4 Tolerância a Falhas.....	109
5.5 Contribuições Esperadas.....	110
5.6 Metodologia	112
5.7 Cronograma.....	114
Referências Bibliográficas	116

Índice de Figuras

Figura 2.1 – Classificação dos Sistemas Computacionais [Lin 2004].....	16
Figura 2.2 – Exemplo de uma Rede Overlay.....	17
Figura 2.3 – Topologia Cliente-Servidor.....	17
Figura 2.4 – Topologias de Rede P2P [Lin 2004].....	18
Figura 2.5 – Topologia P2P Pura.....	19
Figura 2.6 – Topologia P2P Híbrida.....	20
Figura 2.7 – Topologia Super-Ponto.....	21
Figura 2.8 – Roteamento de Consultas no Modelo Híbrido.....	27
Figura 2.9 – Roteamento de Consultas no Modelo de Inundação.....	28
Figura 2.10 – Modelo DHT [Milojicic et al. 2002].....	29
Figura 2.11 – Rede Virtual JXTA [Traversat et al. 2003].....	37
Figura 2.12 – Arquitetura em Camadas da Plataforma JXTA [Gong 2002].....	38
Figura 3.1 – Um SGBDD com dois sites [Valduriez et al. 2004].....	43
Figura 3.2 – Arquitetura em 5 Níveis de um Sistema de Banco de Dados Federado.....	44
Figura 3.3 – Arquitetura Simplificada de um Sistema de Informação baseado em <i>Data Warehouse</i>	47
Figura 3.4 – Arquitetura de Mediação [Wiederhold 1992].....	47
Figura 3.5 – Arquitetura de um Sistema de Informação baseado em Mediadores.....	49
Figura 3.6 – Evolução dos Sistemas de Gerenciamento de Dados Distribuídos.....	51
Figura 3.7 – Arquitetura Genérica de um Ponto em um PDMS [Sung et al. 2005].....	53
Figura 3.8 – Exemplos de Esquemas Exportados, adaptado de [Halevy et al. 2004].....	56
Figura 3.9 – Mapeamentos em um PDMS.....	57
Figura 3.10 – Tipos de Mapeamentos entre Esquemas, adaptada de [Halevy et al. 2004].....	58
Figura 3.11 – Processamento de Consultas Típico em um PDMS.....	59
Figura 3.12 – Uso de Ontologias para definição de “atalhos” entre Pontos não-adjacentes.....	60
Figura 3.13 – Tipos de Mapeamento no Piazza [Tatarinov et al. 2003].....	62
Figura 3.14 – Exemplo de Mapeamentos no Piazza, adaptado de [Tatarinov et al. 2003].....	63
Figura 3.15 – Arquitetura de um Ponto PeerDB [Ooi et al. 2003a].....	64
Figura 3.16 – Clusters no PDMS XPeer [Bellahsène et al. 2004].....	66
Figura 4.1 – Um cenário de rede P2P onde pontos semanticamente similares poderiam ficar isolados [Castano et al. 2003].....	76
Figura 4.2 – Semantic Overlay Networks, adaptada de [Crespo et al. 2002].....	79
Figura 4.3 – Exemplos de Classificações.....	79
Figura 4.4 – Processo de Construção das SON [Crespo et al. 2002].....	81

Figura 4.5 – Arquitetura do Sistema [Li et al. 2005]	81
Figura 4.6 – Abordagem de Clusterização Baseada em um Modelo de Metadados [Löser et al. 2003a]	83
Figura 4.7 – Arquitetura de um Ponto H ³ [Castano et al. 2003].....	84
Figura 4.8 – Exemplo da formação de uma comunidade semântica [Castano et al. 2005]	88
Figura 4.9 – Rede Overlay Semântica do PDMS PARIS.....	89
Figura 4.10 – Arquitetura estendida do PDMS Humboldt Discoverer [Herschel et al. 2005] ...	90
Figura 4.11 – Índice Semântico em 3 Camadas [Herschel et al. 2005]	91
Figura 4.12 – Exemplo de uma Rede no PDMS Mapster [Rouse et al. 2006]	93
Figura 5.1 – Visão Geral da Arquitetura do SPEED.....	96
Figura 5.2 – Componentes Arquiteturais dos Pontos do SPEED	99
Figura 5.3 – Representação do Interesse de um Ponto.....	102
Figura 5.4 – Descoberta de uma Comunidade Semântica através de Comparação Ontológica.	103
Figura 5.5 – Descoberta de Clusters Semânticos através de Comparação Ontológica.....	105
Figura 5.6 – Parte de uma Ontologia do Domínio “Educação”	105
Figura 5.7 – Interesse Semântico dos Pontos Solicitantes P ₁ até P ₆	106
Figura 5.8 – Exemplo de Regra ECA para Manutenção do Tamanho Médio dos Clusters Semânticos.....	109

Índice de Tabelas

Tabela 2.1 – Comparativo entre Topologias de Rede P2P [Fiorano 2003]	22
Tabela 2.2 – Resumo das Diferenças entre Computação P2P e Computação em Grade	32
Tabela 3.1 – Quadro Comparativo entre Sistemas de Gerenciamento de Dados Distribuídos ..	52
Tabela 3.2 – Palavras-Chave para Nomes de Relações e Atributos [Ng et al. 2003].....	65
Tabela 3.3 – Quadro Comparativo entre PDMS.....	72
Tabela 4.1 – Funcionamento de uma SON	80
Tabela 4.2 – Comparativo entre Semantic Overlay Networks e Semantic Overlay Clusters	83
Tabela 4.3 – Graus suportados pelo algoritmo H-Match	86
Tabela 4.4 – Resultado do Processo de <i>Ontology Matching</i>	88
Tabela 5.1 – Formação de Clusters na Comunidade Semântica “Educação”	107
Tabela 5.2 – Exemplos de Regras ECA para Manter o Tamanho Médio dos Clusters.....	109

Capítulo 1

Introdução

1.1 Motivação

Nos últimos anos, o desenvolvimento de soluções para integração de dados têm sido importante em diversos ambientes, incluindo ambientes distribuídos na Web e, mais recentemente, ambientes ponto-a-ponto (P2P). Um sistema P2P caracteriza-se pelo compartilhamento de recursos e serviços computacionais entre os pontos participantes [Bernstein et al. 2002]. Numa perspectiva de alto nível, uma rede P2P pode ser considerada uma rede *overlay*, uma vez que funciona como uma rede virtual, formada pela interconexão de pontos, executando sobre a infra-estrutura de uma rede física já existente [Touch 2001].

Os primeiros sistemas P2P tinham como objetivo o compartilhamento de dados não-estruturados, por exemplo arquivos de música [Napster 2007, Leibowitz et al. 2003]. Os sistemas utilizavam a topologia de rede pura não-estruturada, na qual as consultas são propagadas na rede através da técnica de *flooding* (ou inundação). Este tipo de busca exige grande utilização de largura de banda para proporcionar desempenho razoável, além de apresentar problemas de escalabilidade quando o objetivo é alcançar todos os pontos de uma rede. Não existe garantia de que um arquivo será encontrado mesmo estando disponível na rede.

Inúmeros esforços têm sido despendidos no sentido de aperfeiçoar as topologias de rede e o processamento de consultas nos sistemas P2P. As melhorias referem-se ao armazenamento de índices nos pontos. Os índices representam informações sobre outros pontos conectados e seus respectivos arquivos compartilhados. Como exemplo, podemos citar os sistemas baseados na topologia pura estruturada, ou simplesmente

Distributed Hash Table (DHT), como Chord [Stoica et al. 2001] e CAN [Ratnasamy et al. 2001]. Estes sistemas garantem tempos de busca de ordem logarítmica, porém demandam uma preocupação excessiva em relação à localização do conteúdo e posicionamento dos pontos. Além disso, apresentam pouca flexibilidade em relação à realização de consultas. Apenas consultas simples podem ser submetidas, por exemplo, por palavras-chave. Na topologia super-ponto [Yang et al. 2003], os índices são mantidos em pontos com maior poder computacional (super-pontos). Os super-pontos são interligados entre si e cada um gerencia uma determinada quantidade de pontos.

Recentemente, um novo tipo de sistema P2P surgiu para aumentar a funcionalidade dos atuais sistemas. Estes sistemas possibilitam o compartilhamento de dados estruturados e/ou semi-estruturados, oferecendo representações semânticas mais “ricas” para os dados compartilhados, além de funcionalidades para o processamento de consultas que vão além de simples buscas por palavras-chave [Arenas et al. 2003, Tatarinov et al. 2003, Ooi et al. 2003a]. Tais sistemas são denominados Sistemas P2P de Gerenciamento de Dados (ou *Peer Data Management Systems* - PDMS). Um PDMS pode ser definido como um “*sistema de gerenciamento de dados com arquitetura descentralizada, facilmente extensível, na qual qualquer usuário pode contribuir com novos dados, novos esquemas, ou mapeamentos entre os esquemas dos pontos*” [Halevy et al. 2003a].

O compartilhamento de dados em um PDMS pode ser considerado uma tarefa complexa e desafiadora em virtude das propriedades intrínsecas aos sistemas P2P: volatilidade, autonomia, descentralização, auto-organização e escalabilidade. Dessa forma, muitas das abordagens tradicionais para gerenciamento de dados distribuídos utilizam soluções que muitas vezes não se aplicam (ou se aplicam parcialmente) a um PDMS [Katchaounov 2003].

Em um PDMS, a integração de dados não concentra-se em um único esquema global centralizado [Valduriez et al. 2004]. Ao contrário, cada ponto representa uma fonte de dados autônoma e é responsável por exportar os dados que deseja compartilhar com os demais pontos. Os dados compartilhados são representados através de um esquema exportado que, por sua vez, pode ser o esquema completo da fonte ou apenas uma parte dele [Halevy et al. 2003b].

Os mapeamentos entre os esquemas exportados pelos pontos definem a equivalência semântica entre elementos presentes em dois ou mais esquemas distintos [Kementsietsidis et al. 2003]. Quando um novo ponto conecta-se ao sistema é preciso

estabelecer mapeamentos com outros pontos adjacentes (pontos vizinhos). Normalmente, os mapeamentos são definidos manualmente, entretanto várias técnicas para auxiliar a automatização podem ser aplicadas [Rahm et al. 2001, Belian 2005]. Idealmente, para facilitar a identificação de mapeamentos semânticos, os esquemas dos pontos vizinhos devem ser semanticamente similares ao do novo ponto.

Em um PDMS, as consultas são submetidas a partir de qualquer ponto conectado e respondidas pelos demais pontos do sistema, desde que tenham sido estabelecidos mapeamentos semânticos entre eles [Rouse 2006]. Em geral, a escolha dos pontos capazes de responder uma consulta não é uma tarefa simples e pode ser facilitada pela escolha da topologia de rede. A topologia super-ponto favorece o agrupamento de pontos com interesses similares. O agrupamento de pontos de acordo com o conteúdo compartilhado, por exemplo, pode auxiliar o processamento de consultas no sentido que as consultas são direcionadas para um número menor de pontos, porém mais relevante para a consulta [Löser et al. 2003b].

1.2 Definição do Problema

A conexão arbitrária de pontos em um PDMS é considerada ineficiente [Löser et al. 2003a, Castano et al. 2003] pois pontos com pouca ou talvez nenhuma informação em comum podem tornar-se vizinhos. Conseqüentemente, pode ocorrer um aumento no volume de mensagens transmitidas entre pontos, comprometendo a escalabilidade do sistema. Além disso, diversas outras tarefas importantes podem ser afetadas: identificação de mapeamentos semânticos e processamento de consultas [Heese et al. 2005].

A utilização de uma estratégia para definição dos vizinhos de um ponto é um aspecto essencial em um PDMS. Uma destas estratégias é explorar a semântica do conteúdo exportado pelos pontos para formação de comunidades semânticas [Castano et al. 2005, Brito 2005, Comito et al. 2006, Herschel et al. 2005, Heese et al. 2005, Li et al. 2005, Rouse et al. 2006]. Uma comunidade semântica é um exemplo de rede *overlay* onde os pontos participantes compartilham dados semanticamente equivalentes. Este conceito é puramente lógico, ou seja, pontos pertencentes a domínios de conhecimento comuns tornam-se vizinhos, mesmo localizando-se distante fisicamente.

Neste sentido, para utilização das comunidades semânticas inúmeras questões precisam ser consideradas como, por exemplo, formato de representação do conteúdo exportado pelos pontos, definição de critérios para formação dinâmica de comunidades,

nível de agrupamento semântico (por exemplo, apenas comunidades ou comunidades de comunidades), além do processo de formação e manutenção de uma comunidade.

As tecnologias oriundas da Web Semântica [Berners-Lee et al. 2001] podem ser utilizadas para adicionar um nível maior de semântica aos dados compartilhados. Dentre as vantagens oferecidas, destacamos: representação mais “rica” do conteúdo compartilhado pelos pontos e das consultas submetidas pelos usuários, linguagens de consulta mais poderosas e formação de comunidades semânticas entre pontos.

1.3 Objetivo

O objetivo principal deste trabalho é realizar uma análise detalhada das soluções existentes para o problema da conectividade de pontos em PDMS e propor uma estratégia que explore a semântica associada ao conteúdo exportado pelos pontos.

Neste sentido, propomos uma arquitetura de PDMS baseada nas topologias de rede DHT e super-ponto. A topologia DHT é utilizada para facilitar a descoberta de recursos, auxiliando pontos que desejam se conectar a encontrarem outros pontos com interesses semânticos similares e, por conseguinte, formarem comunidades semânticas. Em cada comunidade, a topologia super-ponto é utilizada para organizar os pontos em clusters semânticos. A rede super-ponto proporciona um ambiente mais adequado para aplicação de técnicas de identificação de mapeamentos semânticos e processamento de consultas.

Em virtude do comportamento dinâmico dos pontos, são necessárias soluções para manutenção dos clusters semânticos. Assim, propomos também estratégias para balanceamento de carga nos clusters semânticos e a tolerância a falhas.

1.4 Escopo

Neste trabalho, nossa preocupação está voltada unicamente para as tarefas relacionadas à conectividade de pontos, balanceamento de carga e tolerâncias a falhas. Embora relacionadas, as tarefas de identificação de mapeamentos semânticos e processamento de consultas não fazem parte do escopo deste trabalho e estão sendo desenvolvidas em paralelo [Souza 2007].

1.5 Estrutura do Documento

O restante deste trabalho está organizado como se segue. No **capítulo 2**, oferecemos uma visão geral sobre os sistemas P2P tradicionais, a terminologia empregada e as principais topologias de rede. Em seguida, são discutidas as propriedades mais importantes dos sistemas P2P. Um comparativo entre sistemas P2P e grades computacionais também é exposto, além das principais classes de aplicação dos sistemas P2P e as plataformas para desenvolvimento destes sistemas. Finalizando o capítulo apresentamos nossas considerações finais.

No **capítulo 3**, apresentamos um breve resumo acerca da evolução dos sistemas gerenciadores de dados distribuídos, enfocando os PDMS. Em seguida, discutimos os principais problemas enfrentados pelos PDMS no que diz respeito à identificação de mapeamentos semânticos e ao processamento de consultas. Apresentamos também alguns dos principais PDMS existentes na literatura. Finalizamos o capítulo expondo um estudo comparativo entre os PDMS discutidos e apresentando as considerações finais.

No **capítulo 4**, apresentamos um resumo das pesquisas mais recentes sobre a formação dinâmica de comunidades semânticas em sistemas P2P tradicionais e PDMS. Enfatizamos a importância do agrupamento semântico de pontos nestes sistemas e mostramos uma definição sobre comunidade semântica. Em seguida, mostramos uma proposta para utilização de comunidades semânticas em sistemas P2P dedicados ao compartilhamento de arquivos, e sua extensão para PDMS. Uma descrição sobre alguns dos principais PDMS que utilizam comunidades semânticas é apresentada. Finalizamos o capítulo com uma análise comparativa destes PDMS e as considerações finais.

No **capítulo 5**, discutimos sobre a proposta de tese. Inicialmente, apresentamos a arquitetura de um PDMS baseada em uma topologia de rede mista (DHT e super-ponto). Em seguida, expomos nossa estratégia para conectividade baseada em semântica. Os pontos que disponibilizam conteúdo semanticamente equivalentes são agrupados em comunidades e clusters semânticos. Em virtude do comportamento dinâmico dos pontos, estratégias para manutenção das comunidades também são discutidas. Concluimos o capítulo apresentando as contribuições esperadas para a tese, a metodologia de trabalho a ser aplicada e o cronograma.

Capítulo 2

O Paradigma Ponto-a-Ponto (P2P)

Devido a sua natureza genérica, existem várias definições para a expressão ponto-a-ponto (ou *peer-to-peer* - P2P). Shirky define o termo ponto-a-ponto como sendo “*uma classe de aplicações que reutiliza recursos já disponíveis na Internet, tais como armazenamento, ciclos e conteúdo*” [Shirky 2000]. Para Bernstein, a expressão significa “*um conjunto de pontos computacionais distribuídos, que podem compartilhar dados e serviços com um conjunto de outros pontos*” [Bernstein et al. 2002]. Em [Milojicic et al. 2002], a expressão é definida como o “*compartilhamento de recursos e serviços computacionais entre sistemas, incluindo informações, ciclos de processamento e espaço em disco*”.

Neste capítulo, oferecemos uma visão geral sobre os sistemas P2P tradicionais, a terminologia empregada e as principais topologias de rede utilizadas pelos sistemas P2P, isto é, a forma como os pontos podem ser organizados na rede. Em seguida, são discutidas algumas das propriedades mais importantes destes sistemas: volatilidade, auto-organização, escalabilidade, entre outras. Um comparativo entre sistemas P2P e grades computacionais também é exposto, assim como as principais classes de aplicação dos sistemas P2P e as plataformas para desenvolvimento destes sistemas.

2.1 Visão Geral e Terminologia

A computação P2P pode ser definida como um paradigma genérico de arquitetura de software, classificado no mesmo nível de abstração da computação cliente-servidor (ver Figura 2.1). Os sistemas baseados no paradigma P2P consistem de componentes de software, chamados de pontos, nós ou *peers*, que compartilham e/ou utilizam os recursos de outro(s) ponto(s) para execução de tarefas de modo descentralizado [Oram 2001]. Cada ponto atua como cliente e servidor simultaneamente,

o que fez surgir o termo *servent* (*server* e *client*) [Schollmeier 2002]. Os recursos compartilhados podem ser os mais variados: poder computacional, espaço de armazenamento, largura de banda e conteúdo. Não existe restrição em relação à quantidade de pontos que podem participar do sistema. Os trabalhos de [Aberer et al. 2002a, Milojevic et al. 2002] apresentam uma visão geral sobre o paradigma P2P e uma descrição atualizada de seus principais sistemas.

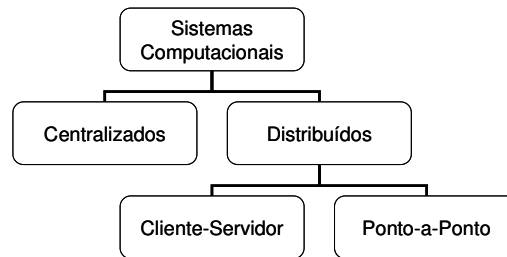


Figura 2.1 – Classificação dos Sistemas Computacionais [Lin 2004]

A computação P2P pode oferecer inúmeras vantagens [Milojevic et al. 2002], entre elas: (i) aumento do poder computacional, aproveitando o tempo ocioso das estações de trabalho conectadas à rede; (ii) capacidade dos pontos poderem assumir funções diferenciadas na rede, ora como clientes, ora como servidores; (iii) redução de custo para o compartilhamento de recursos, possibilitando a realização de tarefas utilizando a infra-estrutura disponível no momento, por exemplo armazenamento de *backup*; (iv) manutenção da autonomia dos pontos participantes, evitando que um ponto seja controlado por outro.

2.2 Topologias de Rede P2P

As redes P2P são comumente confundidas com as redes *overlay* [Touch 2001]. Uma rede *overlay* é uma rede virtual criada sobre uma rede já existente, por exemplo a Internet, com infra-estrutura IP. A rede *overlay* cria uma estrutura com nível mais alto de abstração, de modo a solucionar vários problemas que, em geral, são difíceis de serem tratados ao nível dos roteadores da rede subjacente [Andersen et al. 2001]. Numa perspectiva de alto nível, uma rede P2P pode ser considerada uma rede *overlay*, uma vez que funciona como uma rede virtual, formada pela interconexão de pontos, executando sobre a infra-estrutura de uma rede física (ver Figura 2.2). Portanto, podemos afirmar que uma topologia de rede P2P define o *layout* da rede subjacente e determina o modo como os pontos comunicam-se entre si [Doval et al. 2003, Schlosser et al. 2003].

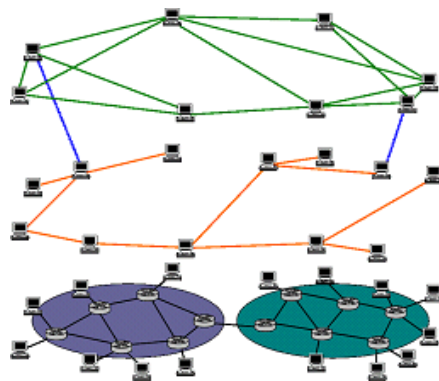


Figura 2.2 – Exemplo de uma Rede Overlay

Embora tenham surgido há algum tempo, os sistemas P2P ainda são confundidos com outros tipos de sistemas distribuídos, especialmente no que se refere ao modelo de rede utilizado. A topologia cliente-servidor, por exemplo, caracteriza-se pela presença de um ponto central (servidor), com alto poder de processamento, ligado a vários outros pontos com poder de processamento inferior (clientes) [Fiorano 2003]. Conforme ilustrado na Figura 2.3, o servidor é o único ponto capaz de fornecer serviços e/ou conteúdo aos clientes. Os clientes, por sua vez, realizam apenas requisições ao servidor, sem compartilhar recursos [Schollmeier 2002].

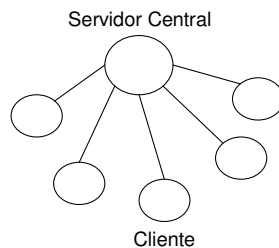


Figura 2.3 – Topologia Cliente-Servidor

A principal vantagem dos sistemas cliente-servidor é sua simplicidade. Como os dados são armazenados em um único local, estes sistemas são mais fáceis de serem administrados e a manutenção da consistência dos dados é minimizada [Dinesh 2004]. O controle de segurança também é facilitado, visto que a preocupação com este quesito resume-se ao servidor. Curiosamente, o fato dos dados estarem armazenados em um único ponto também representa a principal desvantagem dos sistemas centralizados: presença de um ponto único de falha. Se houver uma falha no servidor, todo o funcionamento do sistema é comprometido. Além disso, o servidor pode representar um gargalo no que diz respeito à escalabilidade e desempenho.

Para [Schollmeier 2002], a topologia cliente-servidor não representa uma topologia P2P, uma vez que seu processamento é centralizado e não existe compartilhamento de recursos entre os pontos. Alguns autores inclusive chegam a considerar a topologia cliente-servidor como o oposto da topologia P2P [Singh 2001]. Os autores alegam que, para que uma topologia seja considerada P2P, é necessário que os pontos participantes compartilhem seus recursos, nem que seja parcialmente. Além disso, os pontos devem poder ser acessados diretamente, sem a presença de um ponto intermediário que controle a utilização de recursos.

Na literatura, existem diversas categorizações para as topologias P2P [Lv et al. 2002, Fiorano 2003, Schollmeier 2002]. Estas categorizações levam em consideração diferentes aspectos. Em relação aos sistemas P2P de compartilhamento de arquivos, a classificação baseia-se na localização dos arquivos e dos índices (metadados).

No meio acadêmico, a classificação de [Schollmeier 2002] vem sendo bastante aceita nos últimos anos (ver Figura 2.4). Sua classificação divide as topologias em dois modelos básicos: **pura** ou **descentralizada**, e **híbrida** ou **parcialmente centralizada**. No entanto, um terceiro tipo de topologia, **super-ponto** [Yang et al. 2003], pode ser derivado dos dois primeiros (ver Figura 2.4). Na topologia pura, cada ponto armazena seus arquivos e índices localmente. Veremos mais adiante que a topologia pura pode ser subdividida em não-estruturada e estruturada. Na topologia híbrida, os índices são armazenados em um servidor central (ou um cluster de servidores). Finalmente, na topologia super-ponto, os índices são mantidos em pontos com maior poder computacional (super-pontos). Os super-pontos são interligados entre si e cada um gerencia uma determinada quantidade de pontos. A seguir, apresentamos uma descrição dos modelos de rede P2P, destacando seu funcionamento e principais características. Em seguida, mostramos o resultado de um estudo comparativo entre modelos analisados sob o prisma de diferentes funcionalidades [Fiorano 2003].

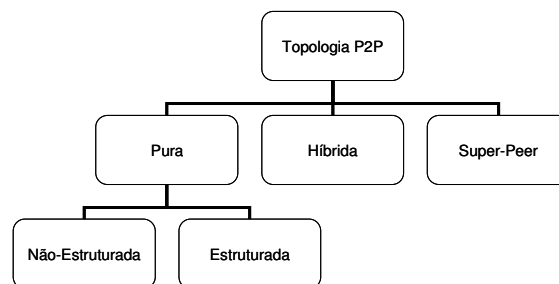


Figura 2.4 – Topologias de Rede P2P [Lin 2004]

2.2.1 Topologia P2P Pura ou Descentralizada

Este modelo de rede é o que segue mais fielmente os princípios do paradigma P2P. Todos os pontos são capazes de se comunicar diretamente entre si e possuem o mesmo papel na rede. Sua principal característica está no funcionamento totalmente descentralizado, ou seja, não existe a idéia de um servidor ou repositório central (ver Figura 2.5). Os pontos podem conectar-se automaticamente e começam a compartilhar recursos, serviços e/ou conteúdo com quaisquer outros pontos já conectados. Os mecanismos de busca e manutenção da infra-estrutura, assim como o compartilhamento de recursos estão distribuídos pela rede. Cada ponto é responsável por manter informações sobre seus próprios dados e, conseqüentemente, ao receber uma solicitação de consulta, pode respondê-la e/ou reescrevê-la, repassando-a aos pontos vizinhos.

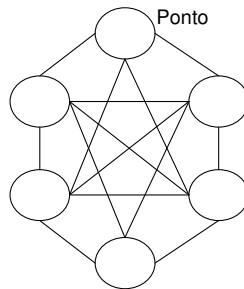


Figura 2.5 – Topologia P2P Pura

A topologia pura pode ainda ser classificada em **não-estruturada** e **estruturada** [Aberer et al. 2002a]. No primeiro caso, não existe qualquer restrição quanto à localização, distribuição e busca dos recursos na rede. Como os pontos não possuem informações sobre os recursos que os outros pontos compartilham, é comum o uso da técnica de *flooding* (ou inundação), que consiste em repassar a consulta recebida a todos os pontos, “inundando a rede”, e retornar os resultados encontrados ao ponto solicitante [Valduriez et al. 2004]. Para evitar que a rede fique congestionada, as consultas normalmente são enviadas a um número limitado de pontos. Entretanto, mesmo que um ponto esteja conectado e apto a responder à consulta, este pode ser descartado em virtude do limite de pontos acessados durante o processamento de uma consulta. Dessa forma, a resposta pode ficar incompleta, diminuindo o nível de confiabilidade do sistema. São exemplos de sistemas desenvolvidos nesse modelo de rede: Gnutella [Gnutella 2007] e Freenet [Freenet 2007].

Diferentemente, a topologia P2P pura estruturada apresenta um certo tipo de controle em relação à localização dos recursos (arquivos). Cada ponto mantém informações sobre seus recursos e os recursos compartilhados por outros pontos. Dessa forma, as buscas são direcionadas e é necessária uma quantidade menor de mensagens, aumentando o desempenho, a eficiência e a confiabilidade do sistema. Estes sistemas utilizam um mecanismo para otimizar a busca e recuperação de recursos na rede, denominado DHT (Tabela Hash Distribuída¹) [Milojicic et al. 2002], a ser descrito na seção 2.3.7. São exemplos de sistemas que utilizam a topologia pura estruturada: Chord [Stoica et al. 2001] e CAN [Ratnasamy et al. 2001].

2.2.2 Topologia P2P Híbrida ou Parcialmente Centralizada

Neste modelo, as informações sobre os recursos compartilhados são armazenadas em um ou mais servidores centrais (ver Figura 2.6). Tais informações são enviadas ao servidor central no momento em que o ponto conecta-se ao sistema. Dessa forma, o servidor central é responsável pelos serviços de busca e manutenção de infraestrutura. Os pontos têm a função de compartilhar entre si recursos, serviços e conteúdo de forma distribuída. Para isso, é necessário primeiro que o ponto envie uma consulta ao servidor central que, por sua vez, devolve o conjunto de pontos que podem atender à requisição. A partir daí, o ponto solicitante pode comunicar-se diretamente com os pontos retornados. São exemplos de sistemas desenvolvidos neste modelo de rede: Napster [Napster 2007] e Groove [Groove 2007].

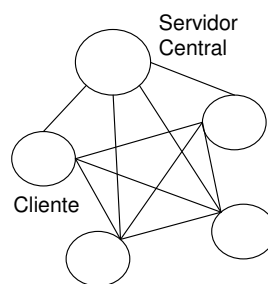


Figura 2.6 – Topologia P2P Híbrida

A topologia híbrida minimiza os problemas de gerenciamento da topologia pura. O servidor central age como um agente de monitoramento dos pontos e garante a consistência dos dados. O aspecto negativo é a existência de um ponto único de falha, ou seja, em caso de desligamento ou falha no servidor central o sistema pode ficar

¹ Em inglês, *Distributed Hash Table*.

parcialmente comprometido. Os pontos que haviam estabelecido comunicação anteriormente continuam sua tarefa, em detrimento à ausência do servidor central, contudo futuras solicitações ao mesmo não podem ser atendidas. Outra questão importante dessa topologia diz respeito à escalabilidade. Embora este problema possa ser amenizado através de atualização de hardware e/ou balanceamento dos pontos (utilização de vários servidores centrais), o fluxo de informações concentrado em um único ponto faz com que a utilização dessa topologia seja indicada para sistemas que necessitem de baixa escalabilidade.

Embora a topologia híbrida possibilite o compartilhamento de recursos, serviços e conteúdo diretamente entre os pontos, a mesma não é considerada verdadeiramente P2P [Aberer et al. 2002b], visto que grande parte do processamento é realizado de forma centralizada, isto é, em um servidor central.

2.2.3 Topologia P2P Super-Ponto

Neste modelo de rede os pontos podem assumir papéis diferenciados. Conforme podemos observar na Figura 2.7, os pontos que apresentam maior poder computacional (super-pontos) são interligados entre si e cada um gerencia uma determinada quantidade de pontos. Cada super-ponto é responsável pelo compartilhamento de recursos entre os pontos a ele conectados. Um super-ponto e o conjunto de pontos associados formam um cluster. Em geral, o agrupamento ocorre em função de interesses similares entre os pontos participantes, por exemplo cluster semântico. Como exemplos de sistemas desenvolvidos nesse modelo de rede, destacamos KaZaA [Leibowitz et al. 2003] e Morpheus [Morpheus 2007].

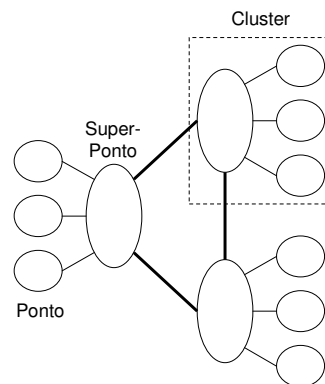


Figura 2.7 – Topologia Super-Ponto

Em geral, a escolha do cluster ao qual o ponto deve interligar-se é feita no instante em que o ponto conecta-se ao sistema. Durante este processo, ponto e super-ponto trocam informações de autenticação, verificando se podem fazer parte do mesmo cluster. Em caso positivo, passam a trocar informações sobre localização, e o ponto informa ao super-ponto os recursos que deseja compartilhar [Bellahsène et al. 2004a]. Em virtude do dinamismo dos pontos, esta topologia é bastante sensível à entrada e saída dos pontos, visto que os metadados armazenados no super-ponto precisam ser constantemente atualizados.

Nesta topologia, as consultas podem ser geradas nos pontos ou super-pontos. Quando realizadas em um ponto, este redireciona-a para seu respectivo super-ponto que, por sua vez, repassa-a aos outros pontos do mesmo cluster e aos super-pontos de outros clusters. Neste caso, os super-pontos comunicam-se com seus respectivos pontos, no sentido de atender à requisição do ponto inicial. Em geral, os super-pontos utilizam o modelo de *flooding* (a ser descrito na subseção 2.3.5) para descoberta de recursos e roteamento de consultas. Porém, outras alternativas são propostas para melhorar a comunicação entre os super-pontos, por exemplo HyperCup [Schlosser et al. 2003].

Uma característica importante dessa topologia é a redução do tempo de consulta e do uso da largura de banda. Uma consulta que, na topologia pura, é respondida em um tempo $O(N)$, na topologia de super-pontos leva $O(N/M)$, onde M é o número médio de pontos conectados a um super-ponto [Fiorano 2003].

2.2.4 Comparativo entre Topologias de Rede P2P

Na Tabela 1, apresentamos o resultado de um estudo comparativo entre as topologias P2P pura, híbrida e super-ponto, realizado em [Fiorano 2003] e estendido em nosso trabalho. O trabalho compara as três topologias segundo os critérios de gerenciamento, tolerância a falhas, escalabilidade, confiabilidade e custo de manutenção da rede. Comparações semelhantes podem ser encontradas em [Backx et al. 2002, Valduriez et al. 2004].

Tabela 2.1 – Comparativo entre Topologias de Rede P2P [Fiorano 2003]

Topologia	Funcionalidade	Gerenciamento	Tolerância a Falhas	Escalabilidade	Confiabilidade	Custo de Manutenção
Pura		Não	Sim	Sim	Baixa	Baixo
Híbrida		Sim	Não	Sim	Alta	Alto
Super-Ponto		Sim	Sim	Sim	Alta	Alto

Os trabalhos de [Fiorano 2003, Brito 2005] apontam a topologia de super-ponto como a mais adequada para o desenvolvimento e manutenção de sistemas P2P, uma vez que possui dentre outras características, redução do tempo de busca e uso de largura de banda, tolerância a falhas, gerenciamento através de super-pontos, escalabilidade e índice aceitável de confiabilidade. Entretanto, destacamos que não existe um consenso quanto a um modelo bem definido para sistemas P2P. Cada topologia possui suas vantagens e desvantagens. A escolha depende do objetivo, funcionamento, desempenho e características do sistema P2P a ser projetado.

2.3 Propriedades dos Sistemas P2P

O trabalho de [Walkerdine et al. 2002] discute uma variedade de propriedades intrínsecas aos sistemas P2P. Algumas propriedades são fortemente inter-relacionadas e precisam ser tratadas conjuntamente (por exemplo, conectividade e auto-organização). Dependendo do tipo de topologia adotado, algumas propriedades podem ser mais ou menos difíceis de serem atendidas [Valduriez et al. 2004]. Por esta razão, é comum encontrar sistemas P2P que não atendam fielmente a todas as propriedades do paradigma P2P. Nesta seção, apresentamos apenas as propriedades destes sistemas mais relacionadas com nosso trabalho: volatilidade, auto-organização, tolerância a falhas, descentralização, balanceamento de carga, escalabilidade e roteamento.

2.3.1 Volatilidade

Esta propriedade está relacionada com a entrada e saída de pontos no sistema. Em um sistema P2P, estas ações são consideradas dinâmicas e *ad-hoc*, isto é, os pontos passam pouco tempo conectados e podem entrar e sair a qualquer momento, sem nenhum conhecimento prévio dos demais pontos [Sen et al. 2004].

A conexão de um ponto solicitante ocorre a partir de outro que já esteja participando do sistema. Em alguns sistemas, pontos podem atuar simplesmente como ponto de entrada (ou *entry point*) para os que desejam fazer parte do sistema. Em [Li et al. 2004] o processo de conexão inicia-se através da comunicação entre um ponto solicitante e um servidor de registro. Um servidor de registro é um tipo especial de ponto cuja função é oferecer uma lista dos super-pontos conectados através de serviços de “páginas amarelas”.

Durante a conexão, é importante que pontos similares situem-se “próximos” uns dos outros. O conceito de proximidade pode variar de acordo com o sistema. Em um sistema P2P de compartilhamento de arquivos, por exemplo, proximidade pode estar relacionada com a semântica associada ao conteúdo compartilhado pelos pontos.

A saída de um ponto pode ocorrer por desconexão ou por falha. No último caso, um mecanismo de *time-out* pode ser utilizado para determinar a desconexão, caso o ponto não tenha respondido durante um determinado período de tempo. A desconexão de um ponto pode ocasionar ainda uma auto-organização dos demais pontos da rede. A propriedade de auto-organização é discutida na próxima subseção.

Embora a maioria dos pontos fique disponível por um intervalo de tempo reduzido, alguns pontos oferecem uma maior disponibilidade. Os sistemas P2P devem explorar o comportamento diferenciado destes pontos, atribuindo-lhes papéis distintos na rede, como é o caso dos super-pontos.

2.3.2 Auto-Organização

Por serem autônomos, os pontos precisam se coordenar para realização de tarefas globais [Anceaume et al. 2003, Ledlie et al. 2002]. A auto-organização diz respeito à capacidade dos pontos se realocarem sozinhos na rede após a ocorrência de um evento, como conexão, desconexão ou falha. A inexistência de uma administração centralizada nos sistemas P2P faz com que esta propriedade fique ao encargo dos próprios pontos. Em geral, esta propriedade é uma das mais defíceis de ser atendida pelos sistemas P2P [Vazirgiannis et al. 2006]

2.3.3 Tolerância a Falhas

Em geral, as topologias cujos pontos podem assumir papéis diferenciados são as mais “sensíveis” a eventos de falha. Por exemplo, na topologia de super-pontos, a falha de um super-ponto não deve comprometer os demais pontos do cluster. Novamente, as estratégias de tolerância a falha devem estar embutidas nos próprios pontos.

Na literatura, diferentes estratégias têm sido apontadas para garantir a tolerância a falhas. Em [Brito 2005, Johnstone et al. 2005] é proposta a eleição de um novo super-ponto entre os demais pontos do cluster. Em geral, um super-ponto é selecionado segundo critérios como: poder computacional, largura de banda, capacidade de armazenamento e disponibilidade. Como solução pró-ativa, [Li et al. 2004] sugerem a

utilização de servidores de *backup*. Os metadados armazenados no super-ponto são periodicamente replicados em um super-ponto de *backup*. Em caso de falha do super-ponto, o super-ponto de *backup* assume o papel do super-ponto. [Rouse et al. 2006] propõem a presença de vários super-pontos em um mesmo cluster visando melhorias no balanceamento de carga e na tolerância a falhas.

Para ser considerada eficiente, uma solução não deve introduzir efeitos colaterais. Por exemplo, na solução adotada pelo framework JXTA [Gong 2002], voltado para o desenvolvimento de aplicações P2P, um ponto que mantém-se desconectado de um super-ponto por um determinado período de tempo, torna-se automaticamente um super-ponto. Esta estratégia pode ocasionar problemas de desempenho, visto que pontos com capacidade limitada ou pouca largura de banda podem tornar-se super-pontos [Johnstone et al. 2005]. O framework JXTA é descrito detalhadamente na subseção 2.6.1.

2.3.4 Descentralização

Esta propriedade refere-se ao fato dos dados e metadados estarem distribuídos entre os pontos participantes do sistema. Não existe um servidor central responsável por tarefas como descoberta de pontos, auto-organização da rede e armazenamento de metadados, com exceção da topologia híbrida. Para saber se seus pontos vizinhos continuam conectados, um ponto precisa comunicar-se periodicamente com os mesmos. Para descobrir novos pontos na rede, um ponto necessita enviar mensagens de *broadcast* na rede.

Basicamente, o objetivo principal da propriedade de descentralização é evitar a introdução de um ponto concentrador que represente um ponto único de falha e, por conseguinte, comprometa o restante do sistema em caso de desconexão ou falha.

2.3.5 Balanceamento de Carga

Em nosso estudo, o balanceamento de carga refere-se à habilidade em distribuir solicitações entre servidores diferentes no sentido de melhorar o desempenho da rede. Estratégias de balanceamento de carga são mais comuns para as topologias híbrida e super-ponto. Na topologia híbrida, múltiplos servidores centrais podem ser utilizados para melhor distribuir o processamento de consultas [Fiorano 2003].

Em relação à topologia super-ponto, 2 (dois) aspectos são importantes para manter um bom balanceamento de carga:

1. *Quantidade de super-pontos*: a quantidade de pontos por super-ponto necessita ser ajustada de maneira adequada [Zhuang et al. 2004]. A presença de muitos super-pontos pode provocar um aumento na quantidade de mensagens trocadas na rede durante o roteamento de consultas, fazendo com que a topologia tenha um comportamento similar ao de uma rede pura não-estruturada. Por outro lado, a presença de poucos super-pontos faz com que cada um super-ponto fique responsável por lidar com uma quantidade excessiva de pontos, sobrecarregando-os;
2. *Métricas para escolha do super-ponto*: as métricas para escolha de um super-ponto devem ser bem definidas. Em geral, incluem: disponibilidade, capacidade de armazenamento e largura de banda [Montresor 2004, Johnstone et al. 2005]. Funções para cálculo da capacidade de um ponto devem ser disponibilizadas. Estas funções devem atribuir pesos adequados para cada métrica.

2.3.6 Escalabilidade

Para os sistemas P2P, escalabilidade significa a capacidade da rede crescer sem ficar sobrecarregada. Diferentemente dos sistemas cliente-servidor, onde os administradores podem estender ou rebalancear os recursos computacionais para compensar o crescimento da rede, em um sistema P2P as soluções para tratar escalabilidade devem estar embutidas em cada ponto, visto que não existem administradores ou equipamentos centrais. Além disso, a velocidade na qual as consultas necessitam ser processadas torna a questão da escalabilidade ainda mais importante [Rouse et al. 2006].

Esta propriedade varia de acordo com o modelo de rede adotado. Na topologia híbrida, o tratamento é praticamente o mesmo dos sistemas cliente-servidor, ou seja, os pontos centrais podem necessitar de balanceamento e/ou expansão física do hardware para compensar o crescimento da rede. Existe uma preocupação com os custos de manutenção dos pontos centrais. Todavia, a experiência do Napster [Naspter 2007] mostrou que, exceto por questões legais, o sistema era robusto e eficiente. Na topologia pura, o aumento da quantidade de pontos pode provocar uma sobrecarga na troca de mensagens para descoberta de novos pontos e realização de buscas na rede. Na topologia de super-ponto, a escalabilidade pode ser tratada através da realocação de pontos nos clusters, isto é, à medida que a quantidade de pontos aumenta, os pontos podem migrar de um cluster para outro.

2.3.7 Roteamento

Encontrar uma informação numa rede volátil, de larga escala e altamente distribuída não é considerada uma tarefa simples [Rocha et al. 2004]. O roteamento refere-se ao modo como os pontos participantes trocam mensagens durante tarefas como descoberta de recursos e processamento de consultas, por exemplo. Nesta seção, abordamos os principais mecanismos de roteamento existentes, destacando suas vantagens e desvantagens. Os principais modelos de roteamento para redes P2P são: híbrido, *flooding* (ou inundação) e tabela hash distribuída (DHT).

Modelo híbrido – os pontos conectam-se a um servidor central (ou servidor de diretórios), onde publicam informações sobre o conteúdo que desejam compartilhar [Milojicic et al. 2002] (Figura 2.8a). Ao receber uma solicitação de um ponto (Figura 2.8b), o servidor central escolhe, em seu diretório, os pontos mais adequados (Figura 2.8c) para responder à consulta. A escolha geralmente baseia-se na disponibilidade e rapidez dos pontos conectados. A partir daí, o compartilhamento é realizado diretamente entre o ponto solicitante e o ponto selecionado (Figura 2.8d). Esse modelo requer uma infra-estrutura de gerenciamento que facilite a administração do servidor de diretórios, visto que o mesmo armazena informações sobre todos os pontos da comunidade. Conforme mencionado anteriormente, isso pode ocasionar uma limitação de escalabilidade do modelo, uma vez que servidores maiores são necessários à medida que o número de requisições aumenta, e uma quantidade de espaço maior é necessária à medida que a quantidade de usuários cresce.

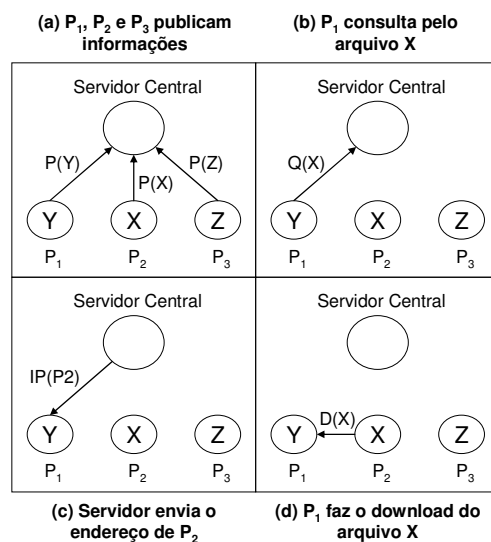


Figura 2.8 – Roteamento de Consultas no Modelo Híbrido

Modelo de inundação – diferentemente do modelo híbrido, o modelo de inundação não se baseia na publicação dos recursos compartilhados [Milojicic et al. 2002]. A requisição de um ponto é enviada a todos os outros pontos conectados a ele diretamente (Figura 2.9a), os quais, por sua vez, reenviam para os pontos diretamente conectados a eles (Figura 2.9b), e assim por diante até a solicitação ser respondida ou o número máximo de encaminhamentos² ser alcançado (Figura 2.9c). Uma vez localizado o recurso, o compartilhamento é realizado diretamente entre o ponto solicitante e o ponto selecionado (Figura 2.9d). Este modelo requer uma grande utilização da largura de banda para proporcionar desempenho razoável. Ainda, o modelo apresenta problemas de escalabilidade quando o objetivo é alcançar todos os pontos de uma rede. Todavia, o modelo é considerado eficiente em comunidades limitadas e corporativas. Diversas pesquisas foram realizadas na tentativa de propor melhorias para o modelo de inundação. O algoritmo *efa* [Li et al. 2004] propõe a eliminação de mensagens duplicadas. [Chawathe et al. 2003] expõem uma solução baseada em adaptação, replicação e controle de fluxo. A solução apresentada em [Yang et al 2002] propõe um aprofundamento iterativo das buscas.

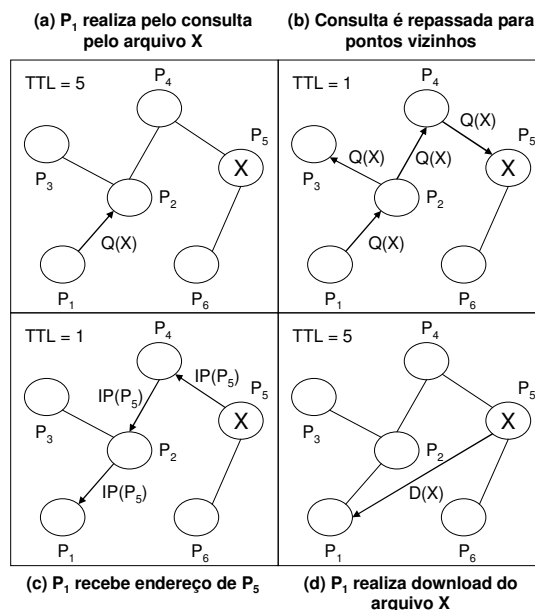


Figura 2.9 – Roteamento de Consultas no Modelo de Inundação

A configuração apropriada do número de conexões de cada ponto e do número de encaminhamentos das mensagens de requisição permite que o modelo de inundação possa ser utilizado por centenas de milhares de pontos [Rocha et al. 2004]. O

² Em geral, um mecanismo de *time-to-live* (TTL) é utilizado para limitar a quantidade de encaminhamentos.

armazenamento de resultados de consultas recentes em cache também pode ser utilizado para melhorar a escalabilidade. A técnica de inundação é bastante questionada em termos de desempenho [Valduriez et al. 2004].

Modelo DHT – a tentativa de melhorias no desempenho dos algoritmos de roteamento de consultas dos sistemas P2P baseados na topologia pura não-estruturada, levou à criação dos sistemas estruturados que utilizam tabelas de hash distribuídas (DHT) [Ratnasamy et al. 2001, Stoica et al. 2001] e sistemas baseados em super-pontos que indexam seus respectivos pontos [Yang et al. 2003]. Os objetivos principais destes sistemas são: reduzir o número de acesso a pontos na busca por documentos e diminuir a quantidade de informações de roteamento que precisa ser mantida em cada ponto.

No modelo DHT, cada ponto da rede possui um identificador randômico e conhece um conjunto de outros pontos. Conforme podemos observar na Figura 2.10, quando um documento é compartilhado por um ponto, um identificador é atribuído ao mesmo como resultado da aplicação de uma função *hash* sobre o nome do documento. Em seguida, o ponto encaminha o documento ao ponto cujo identificador mais aproxima-se do identificador do documento. O processo é repetido até que o identificador do ponto atual seja o mais próximo do identificador do documento. Cada operação de roteamento garante que uma cópia local do documento seja mantida. Quando uma requisição pelo documento é feita, a mesma é enviada ao ponto com identificador mais próximo ao identificador do documento. O processo continua até que uma cópia do documento seja encontrada. Em geral, um ponto responsável por um identificador é encontrado em $O(\log n)$, onde n é o número de pontos no sistema. Uma vez encontrado, o documento é transferido ao ponto que originou a requisição, e cada ponto que participou do roteamento permanecerá com uma cópia local do documento.

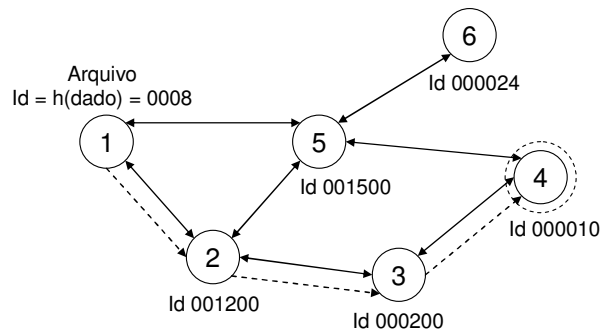


Figura 2.10 – Modelo DHT [Milojicic et al. 2002]

CAN [Ratnasamy et al. 2001], Chord [Stoica et al. 2001], Tapestry [Zhao et al. 2001] e Pastry [Rowstron et al. 2001] são exemplos de sistemas P2P que implementam DHT. Estes algoritmos garantem tempos de busca de ordem logarítmica ou garantem que isso pode ser alcançado com grande probabilidade. As diferenças entre os mesmos são mínimas, entretanto cada qual é apropriado para um determinado ambiente. No Chord, cada ponto mantém informações sobre outros pontos, em uma tabela de roteamento denominada *finger table*. Quando um ponto entra ou sai do sistema, vários pontos precisam ser notificados do evento, atualizando sua tabela de roteamento. No CAN, um ponto mantém informações de roteamento sobre uma quantidade menor de outros pontos. Um conjunto mínimo de pontos é afetado durante a entrada ou saída de um ponto, fazendo com que este algoritmo seja adequado para comunidades mais dinâmicas. Contudo, quanto menor a tabela de roteamento maior é o caminho para as buscas. Tapestry e Pastry são bastante similares. A maior vantagem destes algoritmos sobre os anteriores é que eles tentam efetivamente reduzir a latência e a quantidade de encaminhamentos realizados durante uma busca.

Dentre as limitações do modelo DHT, podemos citar [Li et al. 2004]:

- *Controle sobre a localização de dados e posicionamento de pontos:* estes sistemas exigem uma preocupação excessiva em relação à localização do conteúdo e posicionamento dos pontos, características estas que não se aplicam bem a ambientes como a Internet, onde os usuários estão amplamente distribuídos e normalmente pertencem à organizações sem cooperação;
- *Consultas:* estes sistemas apresentam pouca flexibilidade em relação à realização de consultas. Apenas consultas simples podem ser submetidas, por exemplo por palavras-chave. Em face à rápida evolução dos sistemas P2P, faz-se necessário o suporte a consultas mais “ricas” e complexas. As tendências atuais indicam que novos tipos de consulta devem ser integrados às aplicações P2P: consultas por faixa, envolvendo vários atributos, junções e agregações;
- *Conteúdo compartilhado:* em geral, estes sistemas possibilitam apenas o compartilhamento de arquivos.

2.4 Sistemas P2P e Grades Computacionais

Por tentarem resolver o mesmo tipo de problema – organizar o compartilhamento de recursos entre comunidades virtuais – a computação P2P e a

computação em grade apresentam grande similaridade. Ambos os tipos de computação utilizam uma abordagem única para tentar resolver o problema, ou seja, a criação de estruturas *overlay* que coexistam com as estruturas organizacionais subjacentes. No entanto, existem algumas diferenças entre os mesmos. Nesta seção, expomos uma breve introdução sobre a computação em grade e, em seguida, apresentamos um comparativo entre a computação em grade e a computação P2P.

2.4.1 Grades Computacionais

Segundo [Foster 2002], a computação em grade caracteriza-se pelo “*compartilhamento de recursos coordenados e resolução de problemas em organizações virtuais multi-institucionais dinâmicas*”. Sistemas em grade são sistemas de suporte à execução de aplicações paralelas que acoplam recursos heterogêneos distribuídos, oferecendo acesso consistente e barato aos recursos, independente de sua posição física. A tecnologia de grade possibilita agregar recursos computacionais variados e dispersos em um único “supercomputador virtual”, acelerando a execução de várias aplicações paralelas. Sistemas em grade se tornaram possíveis nos últimos anos, devido a grande melhoria em desempenho e redução de custo, tanto de redes de computadores quanto de microprocessadores [Foster 2002]. São exemplos de sistemas em grade: Globus [Globus 2007] e OurGrid [OurGrid 2007].

A computação em grade é um conceito que explora as potencialidades das redes de computadores, com o objetivo específico de disponibilizar camadas virtuais que permitem ao usuário ter acesso às aplicações de alta demanda computacional, assim como aderir às comunidades virtuais de larga escala, com uma grande diversidade de recursos de computação e de repositórios de informações.

A principal diferença entre a computação distribuída e a computação em grade é que a computação distribuída consiste na possibilidade de resolver um determinado problema computacional através da utilização de diferentes recursos distribuídos geograficamente. Podemos afirmar que a computação distribuída torna-se uma “computação em grade” a partir do momento em que existe uma infra-estrutura física e uma infra-estrutura lógica (software) que permita coordenar os trabalhos a serem processados e garantir a qualidade de serviço.

2.4.2 Comparativo entre Computação P2P e Grades Computacionais

Apesar das inúmeras semelhanças entre a computação em grade e a computação P2P, existem algumas diferenças importantes entre ambas [Ooi et al. 2003b, Foster et al. 2003, Talia et al. 2003]. Um resumo destas diferenças é apresentado na Tabela 2.2 e descrito a seguir.

Tabela 2.2 – Resumo das Diferenças entre Computação P2P e Computação em Grade

Critério	Computação P2P	Computação em Grade
Nós Participantes	- muitos (usuários comuns) - conexões variadas - computadores pessoais	- poucos (organizações) - redes de alta velocidade - servidores (clusters)
Segurança	- comunidades abertas - tópico pouco explorado - anonimato - difícil de gerenciar	- comunidades fechadas - tópico muito explorado - autenticação/autorização/confiabilidade - mais simples de gerenciar
Conectividade	instável	estável
Consultas	resultados incompletos	resultados completos
Aplicações	simples	alto desempenho
Descoberta de Recursos	informações disponíveis em um ponto	pontos enviam mensagens periodicamente
Tolerância à Falhas	tópico bastante explorado	tópico pouco explorado

Em um sistema P2P, os pontos normalmente são computadores pessoais, enquanto que em uma grade computacional é comum a participação de computadores mais robustos. Dessa forma, o custo total para construção de uma grade é consideravelmente superior ao de um sistema P2P. Além disso, como os computadores pessoais representam a maioria dos terminais ligados a Internet, a quantidade de pontos em um sistema P2P tende a crescer rápida e progressivamente. Em contrapartida, em uma grade computacional, a quantidade de pontos é normalmente pequena, aumentando em menor taxa.

A dificuldade em controlar segurança é maior nos sistemas P2P. Dependendo do modelo de rede, pontos maliciosos podem facilmente conectar-se ao sistema. Diferentemente, em um ambiente em grade, ao iniciar a participação no sistema os pontos autorizam e definem os recursos que desejam compartilhar. É utilizado um princípio de justiça, ou seja, um ponto que freqüentemente atende solicitações tem sempre preferência quando necessita utilizar algum recurso disponível em um outro ponto.

Em relação ao comportamento dos pontos, um sistema P2P é considerado instável em virtude da alta taxa de conexão/desconexão dos pontos participantes, o que consequentemente dificulta a descoberta de recursos na rede. Por sua vez, um ambiente em grade é mais estável, facilitando a descoberta de recursos. A instabilidade do ambiente P2P faz com que as consultas executadas passem a ter respostas incompletas, o que normalmente não acontece no ambiente em grade.

As aplicações desenvolvidas para ambientes em grade geralmente exigem alto desempenho dos pontos, visto que seus objetivos são a realização de cálculos complexos e simulações. Diferentemente, em um ambiente P2P as aplicações são mais simples, por exemplo, compartilhamento de arquivos de música.

Os sistemas P2P apresentam soluções que não oferecem padrões para protocolos e infra-estrutura para interoperabilidade. Por outro lado, os sistemas em grade baseiam-se em uma infra-estrutura de serviços padronizados. À medida que os sistemas em grade crescem, passam a necessitar de técnicas aplicadas nos sistemas P2P, e vice-versa. A utilização da tecnologia P2P para desenvolvimento de aplicações cada vez mais complexas poderá provocar convergência entre a computação P2P e em grade. O resultado pode ser o surgimento de uma nova classe de tecnologias combinando elementos de ambas: escalabilidade, auto-organização, recuperação de falhas, além de uma infra-estrutura persistente e padronizada para interoperabilidade.

2.5 Classes de Aplicações P2P

Nesta seção descrevemos as principais classes de aplicação P2P existentes [Leuf 2002]: compartilhamento de arquivos, troca de mensagens, computação distribuída e trabalhos colaborativos. Para cada classe de aplicação, apresentamos exemplos de sistemas.

2.5.1 Compartilhamento de Arquivos

O compartilhamento de arquivos é uma das classes de aplicação P2P que tem alcançado maior sucesso. Basicamente, estes sistemas são utilizados para troca de arquivos (por exemplo, músicas, vídeos e softwares), onde um usuário disponibiliza seus arquivos e solicita arquivos de outros usuários. Além de manter seus arquivos, os pontos também armazenam índices, estruturas utilizadas durante o processamento de consultas para mapear arquivos em pontos. Os índices são mantidos sob a forma de metadados

definidos pelo sistema. Cada sistema é responsável por definir e atualizar seus metadados. Os metadados incluem nome e tamanho de arquivos, número de *uploads/downloads*, largura de banda, entre outros. As pesquisas por arquivos são normalmente realizadas através de buscas por palavras-chave. Um dos primeiros sistemas P2P construídos com esse propósito foi o Napster [Napster 2007], cujo objetivo principal era a troca de arquivos de áudio. Depois surgiram outros: Gnutella [Gnutella 2007], KaZaA [Leibowitz et al. 2003] e e-Mule [Emule 2007].

Um dos fatores que levaram ao desenvolvimento dessas aplicações é a visível limitação dos servidores no que diz respeito à largura de banda que, em geral, impede a transferência de arquivos de tamanho grande. Os sistemas de armazenamento distribuído baseados no paradigma P2P utilizam as vantagens da infra-estrutura existente para oferecer as seguintes características [Rocha et al. 2004]: áreas potencialmente ilimitadas para armazenamento de arquivos, alta disponibilidade do conteúdo garantida através de replicação, mecanismos eficientes de localização e captura de conteúdos disponíveis na rede.

Dentre as principais questões técnicas referentes aos sistemas de compartilhamento de arquivos P2P, destacam-se: segurança, consumo de largura de banda e capacidade de busca [Milojicic et al. 2002].

2.5.2 Troca de Mensagens

A possibilidade de comunicar-se diretamente e em tempo real com outras pessoas tem tornado as aplicações de mensagem instantânea (ou *instant messaging*) uma das mais populares entre os usuários da Internet.

Ao contrário dos programas de correio eletrônico, onde as mensagens enviadas são armazenadas temporariamente em um servidor para posteriormente serem acessadas pelos destinatários, as aplicações de mensagem instantânea fornecem a entrega de mensagens em tempo real. Normalmente, os usuários criam listas de contato onde é possível verificar previamente o estado de cada um: ativo, inativo, ocupado ou ausente. Além disso, os usuários podem enviar e receber arquivos, desde que estejam ativos.

Em virtude da popularidade das aplicações de mensagem instantânea, uma ampla variedade delas pode ser encontrada na Internet gratuitamente:

- **ICQ:** é o mensageiro instantâneo mais antigo, criado em 1996. Possibilita o envio de mensagens e arquivos, estabelecimento de *chats* e ativação de outros programas de (*netmeeting*, jogos) e aviso sobre a chegada de mensagem de correio eletrônico;
- **MSN Messenger:** é uma das aplicações de mensagens instantâneas mais populares. Possui a funcionalidade de sincronização entre vídeo e voz para oferecer comunicação em tempo real. Durante a comunicação, o usuário pode utilizar uma câmera Web que permite exibir imagens ao vivo para outros usuários. A funcionalidade de voz permite aos usuários conversarem entre si usando o microfone do computador;
- **Google Talk (GTalk ou Gtalk):** é um serviço de mensagens instantâneas e de VoIP desenvolvido pelo Google. Com uma interface simples, o Google Talk oferece a troca de mensagens instantâneas de texto e bate-papo por voz. O programa também funciona como notificador de novos e-mails em contas do GMail. Para utilizar o Google Talk é preciso ter uma Conta Google;
- **Yahoo! Messenger:** sua principal característica é a integração com outros serviços e conteúdos do Yahoo!. O usuário pode enviar e receber mensagens através de aparelho celular, ser alertado instantaneamente quando chegar mensagem no Yahoo! Mail ou quando chegar a hora de um de seus compromissos no Yahoo! Agenda.

2.5.3 Processamento Distribuído

A computação distribuída vislumbra a possibilidade de agregar e utilizar a capacidade de processamento e armazenamento que fica subutilizada em máquinas ociosas [Rocha et al. 2004]. Embora não seja uma idéia recente, a computação distribuída vem sendo utilizada por inúmeros projetos [Sterling et al. 1995, Seti 2007].

O projeto SETI@home [Seti 2007] tem como objetivo utilizar a capacidade de processamento ociosa de milhões de computadores, para analisar os sinais obtidos a partir do rádio-telescópio do observatório de Arecibo, Porto Rico, à procura de algum sinal de vida inteligente. Seu poder computacional é de aproximadamente 25 Tflops/s (trilhões de operações de ponto flutuante por segundo). O sistema opera dividindo o problema computacional em tarefas menores e independentes. Em seguida, as sub-tarefas são distribuídas entre os pontos disponíveis. O processamento de cada uma é realizado em um computador individual na rede e os resultados são coletados por um servidor central. O servidor central é responsável por distribuir as sub-tarefas entre os

computadores na Internet. Em cada computador registrado é instalado um software cliente que executa alguma computação requisitada pelo servidor. O processamento é realizado toda vez que o computador entra em período de inatividade, normalmente caracterizado pela ativação da proteção de tela. Após o término da computação, o resultado é retornado ao servidor, e uma nova sub-tarefa é alocada para o cliente. O principal problema do sistema diz respeito à escalabilidade, em virtude da alta taxa de transferência de dados.

2.5.4 Trabalho Colaborativo

Os programas de trabalho colaborativo ou *groupware* são projetados para melhorar a produtividade de indivíduos que possuem metas e interesses comuns. O termo *groupware* é definido como um software que suporta colaboração, comunicação e coordenação de um grupo de usuários em uma rede. Para realização das tarefas em comum, o grupo utiliza ferramentas de calendário, listas de discussão, correio eletrônico, vídeo-conferência, espaços de trabalho, entre outras. Nesse caso, os sistemas P2P oferecem a possibilidade dos usuários colaborarem entre si para realização de suas tarefas, normalmente em tempo real, sem a utilização de uma infra-estrutura centralizada. Como exemplos de aplicações que exploram essa área, destacamos o NetMeeting e o Groove [Groove 2007].

2.6 Plataformas para Desenvolvimento de Sistemas P2P

Inicialmente, para o desenvolvimento de sistemas P2P, era necessário implementar um protocolo próprio de comunicação para permitir a interoperabilidade entre os pontos na rede. Além de implicar num esforço adicional, as várias formas de implementação impediam que as aplicações pudessem comunicar-se entre si, mesmo que o serviço oferecido pelas mesmas fosse equivalente. Por exemplo, arquivos compartilhados por sistemas como o KaZaA, e-Mule e Gnutella são acessíveis exclusivamente dentro de suas próprias redes, forçando os usuários a manterem instalados, em suas máquinas, clientes para cada um dos sistemas de compartilhamento de arquivos que pretenda usar.

Com a popularização das aplicações P2P, surgiu um esforço maior em construir plataformas e frameworks para desenvolvimento deste tipo de aplicação, de modo que as

mesmas possam comunicar-se entre si naturalmente. Dentre elas destacamos JXTA [Gong 2002] e Groove [Groove 2007].

2.6.1 A Plataforma JXTA

JXTA [Gong 2002] é uma plataforma aberta, desenvolvida inicialmente pela *Sun Microsystems*, para construção de sistemas P2P. A plataforma disponibiliza um conjunto de protocolos para comunicação, colaboração e compartilhamento de recursos, entre diferentes tipos de dispositivo: celulares, estações de trabalho, PDA e servidores. Seus principais objetivos são: interoperabilidade (entre diferentes sistemas e comunidades P2P), independência de plataforma (diversas linguagens de programação, sistemas operacionais e redes), generalidade (qualquer tipo de dispositivo digital) e segurança.

Conforme ilustrado na Figura 2.11, os dispositivos (pontos) JXTA criam uma rede virtual *ad-hoc* no topo de redes existentes, ocultando a complexidade das camadas de rede inferiores. Em uma rede virtual JXTA, qualquer ponto pode interagir com os demais, independente de sua localização, tipo de serviço ou ambiente operacional – mesmo quando alguns pontos e recursos estejam localizados atrás de *firewalls* ou utilizem diferentes tecnologias de transporte de rede. Assim, o acesso aos recursos da rede não é limitado por incompatibilidades de plataforma ou restrições da topologia cliente-servidor.

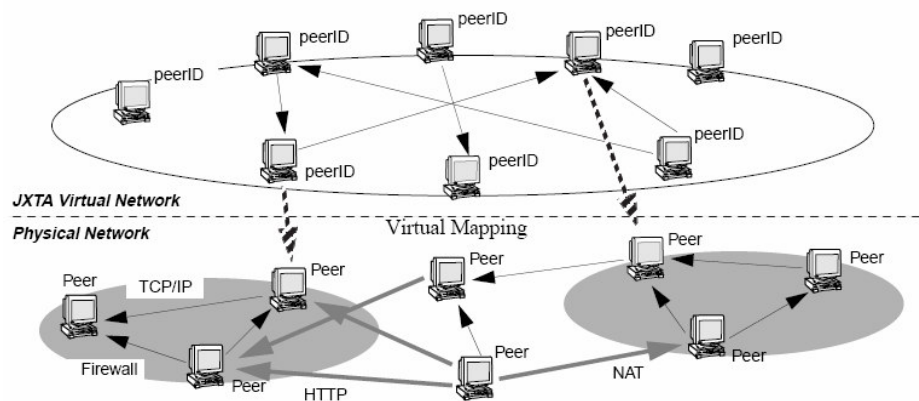


Figura 2.11 – Rede Virtual JXTA [Traversat et al. 2003]

A plataforma JXTA está posicionada como uma pilha P2P, uma camada localizada acima do sistema operacional ou máquina virtual, e abaixo das aplicações e dos serviços P2P. Ela pode ser descrita simplesmente como uma tecnologia que permite a comunicação entre pontos. Cada ponto é associado a um identificador único, um “*peer ID*”, e pertence a um ou mais grupos, denominados *peer groups*. Dentro dos *peer groups* os

pontos cooperam e têm funções similares sob um conjunto unificado de capacidades e restrições. A plataforma disponibiliza protocolos para funções básicas: criar e encontrar grupos, entrar e sair de grupos, monitorar grupos, conversar com outros grupos e pontos, compartilhar conteúdo e serviços – tudo isso é realizado através da publicação e troca de anúncios XML e mensagens entre os pontos.

Conceitualmente, cada ponto na plataforma JXTA abstrai três camadas: o núcleo, a camada de serviços e a camada de aplicação [Gong 2002]. A Figura 2.12 ilustra uma visão da arquitetura. O núcleo é responsável por gerenciar o protocolo JXTA. Sua função é encapsular o conhecimento das operações P2P básicas, ou seja, conter as funcionalidades e a infra-estrutura necessárias para o desenvolvimento de qualquer aplicação P2P.



Figura 2.12 – Arquitetura em Camadas da Plataforma JXTA [Gong 2002]

A camada de serviço armazena as funcionalidades comuns, oferecendo funcionalidades similares às de uma biblioteca, que pode ser controlada pelas aplicações JXTA através de lógica na camada de aplicação.

A camada de aplicação é onde a aplicação P2P realmente reside. Ela pode permitir que o usuário controle diferentes serviços, ou pode ser onde a lógica de uma aplicação autônoma opera. Por exemplo, um simples programa de bate-papo pode ser desenvolvido nessa camada, fazendo uso tanto do serviço quanto do núcleo para permitir que os pontos troquem mensagens.

Uma rede virtual JXTA é formada por vários tipos de pontos, e cada ponto conectado à rede pode assumir os seguintes papéis:

- *Edge Peers*: são pontos simples, como computadores pessoais, conectados por uma LAN ou modem à Internet, ou outros dispositivos computacionais;
- *Minimal Peers*: são pontos com restrições de recursos, como celulares e *palms*. Em geral, não podem aproveitar toda a funcionalidade disponível pela plataforma JXTA;
- *Proxy Peers*: são pontos instalados em computadores que realizam funções de *proxy* (para mensagens da plataforma JXTA) para os seguintes pontos: *minimal peers* que não possuem endereço IP; pontos que, mesmo possuindo IP, não podem realizar operações intensivas em termos de recursos computacionais; ou pontos localizados atrás de um *firewall*;
- *Rendezvous Peers*: são pontos com maior poder computacional, com endereço IP fixo, que servem como cache de informação para os pontos conectados, facilitando a descoberta de recursos e fornecendo operações de resolução, tal como resolução de nome de ponto para endereço IP;
- *Relay Peers*: são pontos que adquirem informação de roteamento, bem como realizam passagem de mensagens para outros pontos atrás de um *firewall*, um NAT ou, simplesmente, através de um roteador.

2.6.2 Groove Development Kit

O Groove é uma plataforma para desenvolvimento de aplicações P2P baseada na reutilização de componentes. Para a construção de aplicativos é necessária a utilização do *Groove Development Kit* (GDK) e ter conhecimento em linguagens baseadas na tecnologia *Microsoft Component Object Model* (COM), como Visual C++, Visual Basic, Delphi, entre outras. Apesar de gratuita, é necessária a obtenção de uma licença para qualquer produto construído através do kit.

O Groove reúne uma série de ferramentas de comunicação e pode ser utilizada para criar novos aplicativos ou integrar aplicativos já existentes, aproveitando-se de componentes reutilizáveis. O sistema utiliza espaços de trabalho que são criados e compartilhados entre os pontos participantes. O Groove usa um mecanismo cliente-servidor para encontrar os pontos e iniciar os serviços. Em seguida, toda a comunicação acontece diretamente entre os pontos participantes.

2.7 Considerações Finais

Iniciamos este capítulo com uma visão geral sobre os sistemas P2P, a terminologia empregada nestes sistemas e as principais topologias de rede. Discutimos também sobre as propriedades dos sistemas P2P mais relacionadas com nosso trabalho, como volatilidade, auto-organização, tolerância a falhas, descentralização, balanceamento de carga, escalabilidade e roteamento. Um breve comparativo entre sistemas P2P e grades computacionais também foi apresentado. Finalizamos o capítulo discutindo sobre as principais classes de aplicação dos sistemas P2P e as plataformas para desenvolvimento destes sistemas.

Os conceitos apresentados neste capítulo são de fundamental importância para entendimento do assunto tratado no Capítulo 3: Sistemas Gerenciadores de Dados P2P (ou PDMS), um tipo de aplicação P2P destinada ao compartilhamento de dados semi-estruturados e estruturados. Nestes sistemas, cada ponto representa uma fonte de dados que deseja compartilhar todos os seus dados ou apenas uma parcela dos mesmos através de esquemas exportados. Mapeamentos semânticos são estabelecidos entre os esquemas exportados. Assim como nos sistemas P2P tradicionais, os pontos em um PDMS podem entrar e sair do sistema a qualquer momento (embora em uma taxa menor).

Capítulo 3

Gerenciamento de Dados em Sistemas P2P

Os benefícios proporcionados pelos sistemas P2P têm despertado a atenção de pesquisadores de diversas áreas: redes de computadores, sistemas distribuídos, inteligência artificial e banco de dados. A comunidade de banco de dados está particularmente interessada no paradigma P2P, com o objetivo de oferecer uma infraestrutura mais adequada para o gerenciamento de dados armazenados em fontes distribuídas, heterogêneas e autônomas.

Conforme discutimos anteriormente, os primeiros sistemas P2P tinham como objetivo compartilhar dados não-estruturados (por exemplo, arquivos de música). Recentemente, inúmeros sistemas P2P têm sido propostos para compartilhamento de dados estruturados e/ou semi-estruturados [Arenas et al. 2003, Tatarinov et al. 2003, Ooi et al. 2003a]. Tais sistemas são denominados Sistemas P2P de Gerenciamento de Dados ou *Peer Data Management Systems* (PDMS).

Neste capítulo, apresentamos um breve resumo acerca da evolução dos sistemas capazes de gerenciar dados distribuídos: Sistemas Gerenciadores de Bancos de Dados Distribuídos, Sistemas de Bancos de Dados Federados, Sistemas de Bancos de Dados Múltiplos, Sistemas de Integração de Dados e PDMS. Em seguida, apresentamos os principais problemas enfrentados pelos PDMS no que diz respeito à identificação de mapeamentos semânticos e ao processamento de consultas. Apresentamos também alguns dos principais PDMS existentes: Piazza [Tatarinov et al. 2003], PeerDB [Ooi et al. 2003a], XPeer [Bellahsène et al. 2004a], Edutella [Nejdl et al. 2002], APPA [Valduriez et

al. 2004], Hyperion [Arenas et al. 2003] e Rosa-P2P [Brito 2005]. No final do capítulo, um estudo comparativo entre os PDMS discutidos é exposto e as considerações finais são apresentadas.

3.1 Evolução dos Sistemas Gerenciadores de Dados Distribuídos

O gerenciamento de dados em sistemas distribuídos foi inicialmente tratado pelos sistemas gerenciadores de bancos de dados distribuídos [Özsu et al. 2001], os quais permitem que os usuários, através de uma linguagem de consulta de alto nível, acessem e atualizem de forma transparente os dados armazenados em inúmeros bancos de dados espalhados em uma rede de computadores. Sua utilização é voltada para aplicações que requerem um controle centralizado de dados distribuídos e necessitam utilizar todas as funcionalidades dos bancos de dados. Entretanto, quando comparada com outras arquiteturas, a arquitetura dos SGBDD mostra-se pouco escalonável.

Quando existe a necessidade de acesso a fontes de dados pré-existentes, heterogêneas e distribuídas, têm-se o problema da integração de dados [Katchaounov 2003]. Os sistemas de integração de dados apresentam uma visão lógica unificada dos dados de modo que as aplicações não necessitam lidar com inúmeras fontes, interfaces e representações dos dados. É necessário a definição de um esquema global sobre os dados existentes e de mapeamentos entre o esquema global e os esquemas das fontes de dados. Os sistemas de integração de dados receberam diversos nomes ao longo dos anos, tais como sistemas de bancos de dados federados, sistemas de bancos de dados múltiplos (ou *multidatabases*) e, mais recentemente, sistemas de integração. No contexto da Web, os sistemas de integração permitem o acesso a fontes de dados autônomas apenas para leitura dos dados. Portanto, tais sistemas não oferecem todas as funcionalidades de um banco de dados, por exemplo controle de transações e replicação. Os PDMS representam uma evolução natural dos tradicionais sistemas de integração, no qual o esquema global único é substituído por uma coleção de mapeamentos semânticos entre os esquemas exportados pelos pontos que compõem a rede [Herschel et al. 2005].

Nesta seção, apresentamos uma visão geral sobre o gerenciamento de dados em ambientes distribuídos, enfatizando os sistemas e abordagens mais tradicionalmente utilizados. Discutimos também os principais problemas destes sistemas e sua evolução para ambientes distribuídos como o ambiente P2P.

3.1.1 Sistemas Gerenciadores de Bancos de Dados Distribuídos

Um banco de dados distribuído é uma coleção de múltiplos bancos de dados logicamente interrelacionados e distribuídos ao longo de uma rede de computadores. Um sistema gerenciador de bancos de dados distribuídos (SGBDD) é um software que controla os bancos de dados distribuídos, tornando transparente a distribuição dos dados para os usuários [Özsu et al. 2001]. Esta transparência estende o conceito de abstração dos dados de forma que os usuários não percebem a distribuição dos dados [Valduriez et al. 2004].

Na arquitetura de bancos de dados distribuídos, os componentes são chamados de *sites* ou nós e consistem de um computador único e independente. Os *sites* executam suas aplicações de forma independente e são interligados através de uma rede de computadores. As aplicações submetem consultas e transações ao sistema e este, por sua vez, as transformam em consultas e transações locais, enviando-as aos demais *sites* e integrando os resultados. Um sistema de bancos de dados distribuídos pode ser executado a partir de qualquer *site*, inclusive em um *site* que não armazena dados, como mostra a Figura 3.1.

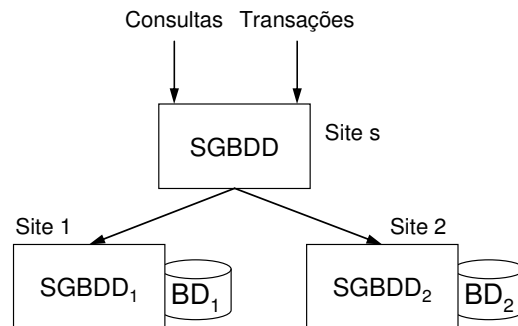


Figura 3.1 – Um SGBDD com dois sites [Valduriez et al. 2004]

A distribuição física do banco de dados entre os sites é feita através da fragmentação e replicação dos dados. Em um esquema de banco de dados relacional, por exemplo, a fragmentação subdivide cada tabela em várias partições através da aplicação de uma função sobre os valores dos campos. Os fragmentos podem ser replicados para melhorar o desempenho das consultas e a disponibilidade dos dados.

As funcionalidades oferecidas pelos SGBDD são basicamente as mesmas dos sistemas de bancos de dados tradicionais (gerenciamento de esquema, controle de acesso, processamento de consultas, suporte à transação, entre outras). Entretanto, em virtude da distribuição dos dados estes sistemas são mais difíceis de serem implementados, e é

justamente por isso que muitos dos sistemas de bancos de dados distribuídos suportam apenas uma parcela dessas funcionalidades. Conforme veremos a seguir, a arquitetura dos SGBDD, quando comparada com outras arquiteturas, é considerada pouco escalonável, suportando uma pequena quantidade de bancos de dados homogêneos.

3.1.2 Sistemas de Bancos de Dados Federados

O termo banco de dados federado é usado para se referir a uma coleção de banco de dados cooperantes e autônomos onde o compartilhamento é feito através da exportação de esquemas, o qual define a parte compartilhável de cada banco de dados local [Kim 1995]. Neste contexto, cada aplicação pode definir seu próprio esquema global através da integração destes esquemas. A característica chave de uma federação é a cooperação entre sistemas independentes.

A arquitetura de cinco níveis de um sistema de banco de dados federado [Sheth et al. 1990] é apresentada na Figura 3.2. Seus componentes principais são: (i) *esquema local*: é o esquema conceitual do sistema de banco de dados componente; (ii) *esquema componente*: é derivado do esquema local pela sua tradução para um modelo de dados comum; (iii) *esquema exportado*: representa o subconjunto de dados do esquema componente que estão disponíveis para a federação; (iv) *esquema federado*: é a integração de múltiplos esquemas exportados; (v) *esquema externo*: define um esquema para um usuário, aplicação ou classes de usuários e aplicações.

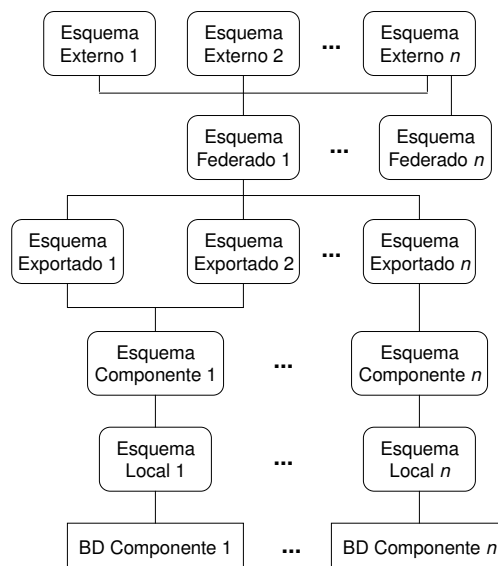


Figura 3.2 – Arquitetura em 5 Níveis de um Sistema de Banco de Dados Federado

Os esquemas exportados são definidos a partir de entendimentos entre o administrador de banco de dados da federação e o do sistema de banco de dados componente que tem controle sobre o que está incluído no esquema exportado. Os esquemas federados são responsabilidade do administrador da federação que deve criá-los e controlá-los. Os esquemas externos são criados a partir de entendimentos entre o usuário da federação e o administrador da federação que tem autoridade sobre o que está incluído em cada esquema externo.

Diferentes arquiteturas de sistemas federados são criadas de acordo com o nível de integração dos bancos de dados componentes e com o responsável pelo gerenciamento da federação. Um sistema federado possui duas categorias clássicas:

- *Fracamente acoplado*: quando é responsabilidade do usuário criar e manter a federação e não existe controle por parte da federação ou de seus administradores. Cada usuário da federação é responsável por criar e administrar o seu esquema federado;
- *Fortemente acoplado*: quando existe uma autoridade central responsável por criar e manter a federação e ativamente controlar o acesso aos bancos de dados componentes.

3.1.3 Sistemas de Bancos de Dados Múltiplos

Existem aplicações que necessitam lidar com diversos bancos de dados pré-existentes localizados em ambientes heterogêneos de hardware e software. Neste caso, é praticamente impossível converter cada banco de dados para um formato comum sem comprometer as aplicações existentes. Assim, é necessário uma camada de software adicional, em um nível superior ao dos bancos de dados existentes, para tratar as informações armazenadas nestes bancos de dados heterogêneos. Essa camada de software é denominada sistema de banco de dados múltiplo ou *multidatabase* [Korth et al. 2006, Elmagarmid et al. 1999].

Nos sistemas de banco de dados múltiplos, os bancos de dados locais podem empregar diferentes modelos lógicos, além de linguagens de definição e manipulação de dados distintas. No entanto, um modelo de banco de dados comum precisa ser aplicado já que estes sistemas criam a ilusão de uma integração lógica dos bancos de dados sem necessariamente exigir uma integração física correspondente. Uma vez definido o modelo de dados comum, o sistema deve integrar os esquemas dos bancos de dados formando um esquema global visível para as aplicações. Esta integração é uma tarefa complexa,

principalmente devido à heterogeneidade semântica, sintática e estrutural dos bancos de dados.

Os bancos de dados de um sistema de banco de dados múltiplo podem ainda diferir nos mecanismos de controle de concorrência e gerenciamento de transações. Sob essas circunstâncias, o processamento destas tarefas torna-se bastante complexo.

3.1.4 Sistemas de Integração de Dados

Com o advento da Web, os sistemas de gerenciamento de dados distribuídos passaram a sentir a necessidade de integrar dados armazenados em fontes de dados heterogêneas, incluindo não somente bancos de dados (relacionais ou orientados a objeto, por exemplo), mas também outros tipos de fontes de dados, como fontes de dados XML e sistemas de arquivos. Além disso, como a maioria das fontes na Web são autônomas, em geral os dados disponibilizados pelas mesmas são apenas para leitura. Assim, para lidar com estes novos requisitos, um novo tipo de sistema surgiu, os sistemas de integração de dados.

Duas grandes abordagens são utilizadas para projetar sistemas de integração de dados: a **abordagem materializada**, baseada na tecnologia de *data warehouse* [Kimball 2002], e a **abordagem virtual**, fundamentada no conceito de mediadores [Wiederhold 1992]. *Data warehouses* são sistemas centralizados onde dados de fontes distribuídas são previamente coletados, unificados e armazenados em um repositório comum, para em seguida serem acessados pelo usuário, sem que o mesmo necessite interagir diretamente com as fontes de dados. Os sistemas mediadores fornecem uma visão lógica e unificada de fontes distribuídas, autônomas e heterogêneas. Por intermédio desta visão lógica, o usuário pode acessar e consultar os dados relevantes diretamente das fontes. A seguir, apresentamos um resumo de cada abordagem.

Na abordagem materializada, a visão integrada é fisicamente materializada através de uma cópia associada a possíveis transformações nos dados das fontes. O grau avançado de maturidade da tecnologia de bancos de dados relacionais, além de sua ampla utilização, fez com que a mesma fosse rapidamente escolhida como solução padrão para implementação dos sistemas de *data warehouse*. Tais sistemas são construídos como bancos de dados orientados a assunto, especializados em responder consultas específicas de apoio à tomada de decisão. A arquitetura de um sistema de informação baseado em *data warehouse* é ilustrada na Figura 3.3.

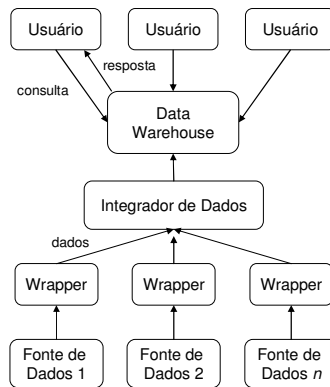


Figura 3.3 – Arquitetura Simplificada de um Sistema de Informação baseado em *Data Warehouse*

Uma alternativa à abordagem materializada é manter os dados em suas fontes originais e acessá-los apenas sob demanda, recuperando e combinando somente os dados relevantes para responder às consultas. Para isto, é necessária a existência de uma camada intermediária que apresente aos usuários uma visão lógica e integrada das fontes de dados. Como a visão integrada não é materializada explicitamente, este enfoque para integração de dados é também conhecido como abordagem virtual. A visão integrada não armazena dados, e sim metadados.

Em [Wiederhold 1992], são analisados os principais requisitos e interfaces para a camada de integração virtual. O trabalho apresenta uma arquitetura para a camada de mediação, adaptada e ilustrada na Figura 3.4. Nela, o mediador é uma camada intermediária virtual que separa as funções relacionadas com a integração de dados, das funções referentes ao gerenciamento de dados das fontes. A camada consiste de módulos denominados mediadores, definidos em [Wiederhold 1992] como “*um módulo de software que explora o conhecimento codificado sobre determinados conjuntos ou sub-conjuntos de dados, para gerar informações destinadas a uma camada de mais alto nível composta de aplicações?*”.

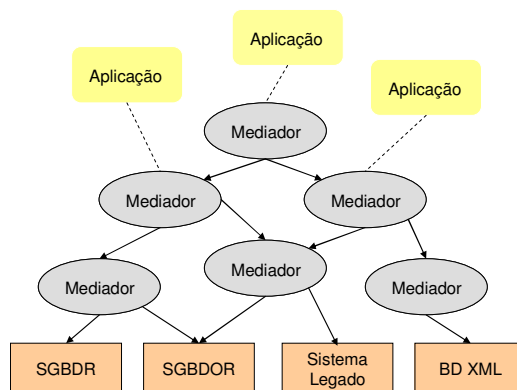


Figura 3.4 – Arquitetura de Mediação [Wiederhold 1992]

Para que sua gerência possa ser facilitada, a camada de mediação deve ser projetada de forma modular, consistindo de um conjunto de mediadores, onde cada um deve ser especializado em um determinado domínio de conhecimento, integrando parcialmente um número menor de fontes. A manutenção dos mediadores pode ser realizada sem a necessidade de um número elevado de especialistas de domínio. Os mediadores podem compartilhar suas abstrações com outros mediadores e aplicações, os quais podem fazer uso do conhecimento de domínio acumulado em mediadores de mais baixo nível. Aplicações e mediadores que necessitem de informações pertencentes a mais de um domínio podem utilizar um ou mais mediadores especializados. Cada mediador compartilha a visão integrada originada a partir das fontes que ele integra ou até mesmo de outros mediadores, disponibilizando novos conhecimentos na rede de mediadores. Não existe uma única visão unificada representando todas as fontes, mas uma gama de mediadores. Para facilitar a descoberta e a reutilização de conhecimento, os mediadores devem disponibilizar metadados sobre os dados que disponibilizam.

Assim como os sistemas de bancos de dados múltiplos, é fundamental que o esquema das fontes seja representado através de um modelo de dados comum, já que as fontes de dados podem apresentar modelos de dados diferentes. Este modelo é o mesmo adotado para o mediador. Assim, para que uma fonte possa ser integrada é necessário que seu esquema seja convertido para o modelo de dados comum.

Uma vez representados através de um modelo padrão, os esquemas das fontes podem ser integrados para geração da visão unificada (esquema de mediação). Neste processo, são identificados diferentes tipos de conflito entre os esquemas das fontes: sintáticos, estruturais e semânticos. Ainda, são geradas as assertivas de correspondência [Lóscio 2003], ou seja, mapeamentos entre os elementos da visão unificada e das fontes.

Considerando os dois processos anteriores, conversão de esquemas e integração de esquemas, os sistemas mediadores são normalmente organizados em duas camadas distintas (Figura 3.5). A primeira delas é responsável pelo processo de mapeamento entre modelos de dados. Em geral, essa camada é implementada através de *wrappers*, cujo papel principal é ocultar os detalhes de acesso a fonte. Para cada fonte de dados deve existir um *wrapper* associado. Suas funções são recuperar os dados armazenados na fonte e traduzi-los para o modelo de dados comum, acessar os metadados e as estatísticas da fonte. A segunda camada (camada de mediador) é responsável por fornecer mecanismos para resolução de conflitos entre os esquemas das fontes.

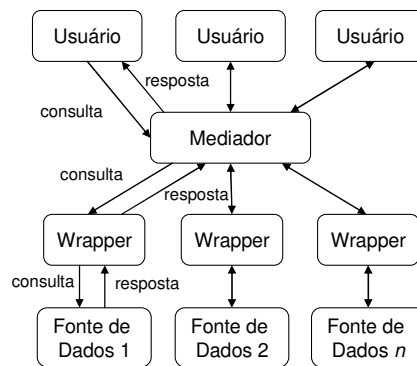


Figura 3.5 – Arquitetura de um Sistema de Informação baseado em Mediadores

Em um sistema de mediadores, as consultas são formuladas de acordo com o esquema de mediação. O sistema traduz a consulta do usuário em subconsultas a serem destinadas às fontes de dados, utilizando mapeamentos semânticos. Existem duas estratégias tradicionais para especificação de mapeamentos entre o esquema de mediação e os esquemas das fontes: *global-as-view* (GAV) e *local-as-view* (LAV). Os trabalhos de [Levy 2000] e [Ullman 1997] apresentam uma visão mais detalhada dos enfoques, os quais baseiam-se na utilização de visões.

Na estratégia GAV [Levy 2000], o esquema de mediação é definido como um conjunto de visões sobre os esquemas das fontes. Sua principal vantagem é que o processo de reescrita de consultas ao esquema de mediação é mais simplificado. Entretanto, a integração de novas fontes de dados torna-se complexa, visto que o esquema de mediação necessita ser atualizado sempre que uma nova fonte for integrada.

Diferentemente, na estratégia LAV [Levy 2000], o esquema de mediação é primeiramente definido e, em seguida, os esquemas das fontes são descritos como visões sobre o esquema de mediação. Sua principal desvantagem é a reescrita de consultas mediante a utilização das visões, um problema considerado complexo e “caro” de ser resolvido [Halevy 2000], visto que as visões podem ser utilizadas na ordem inversa. Entretanto, a integração de novas fontes de dados pode ser gerenciada localmente atualizando-se as visões do esquema das fontes adicionadas. A especificação de mapeamentos LAV é indicada para sistemas de integração que incluem um grande número de fontes de dados, possivelmente heterogêneas, visto que fontes de dados adicionadas ao sistema podem ser descritas através de seu próprio modelo de dados sem a necessidade de serem traduzidas para um modelo de dados comum. Portanto, LAV representa uma boa solução em termos de escalabilidade.

3.1.5 Sistemas P2P de Gerenciamento de Dados

Os sistemas P2P ganharam destaque com o compartilhamento de arquivos. Nos últimos anos, inúmeros esforços têm sido despendidos no sentido de aperfeiçoar as topologias de rede e o processamento de consultas nestes sistemas. Dentre as melhorias alcançadas podemos citar o sistema Gnutella, baseado na topologia de rede pura não-estruturada, e os sistemas baseados em DHT, como Chord e CAN. No entanto, estes sistemas apresentam limitações em relação ao conteúdo compartilhado (apenas arquivos) e às linguagens de processamento de consulta, normalmente baseadas no nome dos arquivos [Comito et al. 2006].

Recentemente, um novo tipo de sistema P2P surgiu para aumentar a funcionalidade dos sistemas P2P de compartilhamento de arquivos. Estes sistemas possibilitam o compartilhamento de dados estruturados, oferecendo representações semânticas mais “ricas” para os dados compartilhados, além de funcionalidades para o processamento de consultas que vão além de simples buscas por palavras-chave [Halevy et al. 2003a, Arenas et al. 2003, Tatarinov et al. 2003, Ooi et al. 2003a]. Tais sistemas são denominados Sistemas Gerenciadores de Dados P2P (ou *Peer Data Management Systems - PDMS*). Um PDMS pode ser definido como um “*sistema de gerenciamento de dados com arquitetura descentralizada, facilmente extensível, na qual qualquer usuário pode contribuir com novos dados, novos esquemas, ou mapeamentos entre os esquemas dos pontos*” [Halevy et al. 2003a]. Basicamente, a idéia dos PDMS é explorar as principais vantagens oferecidas pelos sistemas P2P e pelos SGBD, formando um sistema único [Furtado 2005].

Em um PDMS, a integração de dados não concentra-se em um único esquema global centralizado [Valduriez et al. 2004]. Ao contrário, cada ponto representa uma fonte de dados autônoma e é responsável por exportar, completa ou parcialmente, o esquema da fonte que deseja compartilhar com os demais pontos. O esquema exportado representa os dados que podem ser acessados pelos outros pontos. Note que, o esquema exportado por um ponto só consegue ser compartilhado caso existam outros esquemas exportados cujos termos sejam semanticamente relacionados. Neste caso, podem ser especificados mapeamentos semânticos entre os termos dos esquemas exportados pelos pontos.

As consultas são formuladas de acordo com o esquema específico de um ponto e enviadas para os demais pontos. Assim, durante o processamento de uma consulta devem ser considerados tanto os dados armazenados no ponto onde a consulta foi

submetida, quanto os dados contidos em outros pontos com os quais o esquema do ponto em questão mantém mapeamentos. Os mapeamentos entre os esquemas podem mudar ao longo do tempo, ou podem ser diferentes dependendo da consulta submetida à fonte de dados ou das fontes de dados sendo consultadas.

Como exemplo de aplicação que pode usufruir das vantagens oferecidas por um PDMS, considere um grupo de cientistas que deseja compartilhar dados sobre pesquisas em comum. Em geral, o conteúdo é compartilhado por um intervalo de tempo suficiente para realização de um experimento e refere-se a todos os resultados produzidos por um cientista ou apenas a uma parte destes resultados.

3.1.6 Comparativo entre Sistemas Gerenciadores de Dados Distribuídos

Os PDMS podem ser considerados uma evolução natural dos sistemas de integração de dados no qual o esquema de mediação é substituído por uma coleção de mapeamentos semânticos entre os esquemas individuais exportados por cada ponto (ver Figura 3.6).

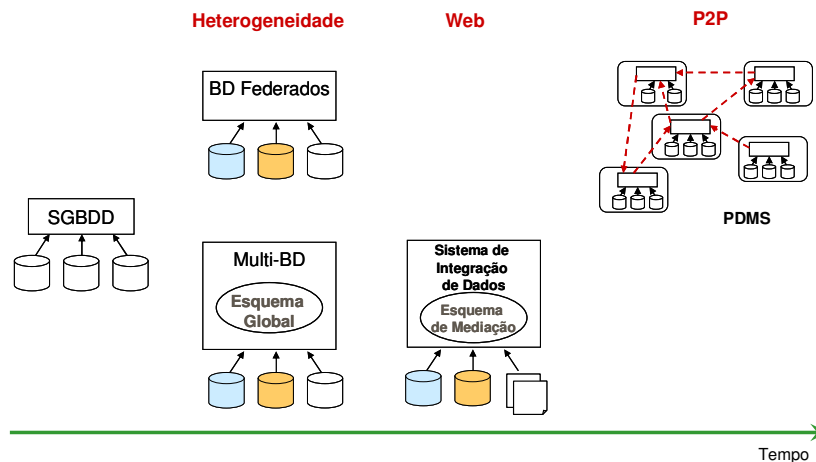


Figura 3.6 – Evolução dos Sistemas de Gerenciamento de Dados Distribuídos

Os SGBDD foram pioneiros ao permitirem o gerenciamento de bancos de dados distribuídos deixando a distribuição dos dados transparente ao usuário. Um banco de dados distribuído armazena dados homogêneos (unicamente dados relacionais, por exemplo) e possibilita consulta e atualização dos dados.

Nos sistemas de bancos de dados federados os bancos de dados podem armazenar dados heterogêneos (bancos de dados relacionais e orientados a objeto, por

exemplo). A manipulação de dados ocorre através da importação e exportação de esquemas (não há esquema global). Os sistemas de bancos de dados múltiplos também lidam com dados heterogêneos. Contudo, as consultas e atualizações ocorrem através de um esquema global.

Os sistemas de integração de dados também utilizam um esquema global. Além disso, consideram a participação de fontes de dados heterogêneas na Web. A maioria das funcionalidades de integração é encapsulada em um único componente central, o mediador. Nele, está concentrado todo o conhecimento necessário para integração das fontes de dados. Dessa forma, o mediador introduz um ponto único de falha no sistema, ou seja, a interrupção do seu funcionamento pode provocar uma parada completa do sistema. Além disso, sua localização centralizada torna o acesso e a manutenção tarefas bastante complexas, dependendo da quantidade de fontes integradas.

Os PDMS caracterizam-se por uma taxa de conexão e desconexão de pontos freqüente e indeterminada, não existe a definição de um esquema global, o resultado das consultas pode ser incompleto e não é possível prever os pontos a serem consultados para atender uma determinada consulta [Ng et al. 2003]. Os mapeamentos semânticos são estabelecidos diretamente entre os pontos participantes do sistema. A quantidade de fontes de dados integradas é bastante alta e as fontes de dados podem pertencer a domínios de conhecimento distintos, o que inviabiliza a presença de um esquema global único. Além disso, a possibilidade de existirem especialistas de domínio com conhecimento suficiente para integrar todas as fontes é mínima. Um resumo das diferenças e similaridades entre os sistemas de gerenciamento de dados distribuídos é apresentado na Tabela 3.1.

Tabela 3.1 – Quadro Comparativo entre Sistemas de Gerenciamento de Dados Distribuídos

Sistema	Fontes de Dados			Esquema Global	Domínio de Conhecimento	Acesso aos Dados
	# Fontes	Tipo da Fonte	Conectividade			
SGBDD	Dezenas	Banco de Dados (Homogêneos)	Estável	Não	Único	Escrita e Leitura
Sistemas de BD Federados	Dezenas	Banco de Dados (Heterogêneos)	Estável	Não	Único	Escrita e Leitura
Sistemas de BD Múltiplo	Dezenas	Banco de Dados (Heterogêneos)	Estável	Sim	Único	Escrita e Leitura
Sistemas de Integração de Dados	Dezenas ou Centenas	Qualquer	Estável	Sim	Único	Leitura
PDMS	Milhares	Qualquer	Instável	Não	Variado	Leitura

3.2 Arquitetura Genérica de um Ponto em um PDMS

Por ser uma área de estudo recente, ainda não existe uma arquitetura definitiva para os pontos participantes de um PDMS. A Figura 3.7, traduzida de [Sung et al. 2005], apresenta uma arquitetura genérica para os pontos de um PDMS. Dependendo do papel exercido pelo ponto no PDMS, um ou mais componentes podem ser acrescentados/retirados ou combinados.

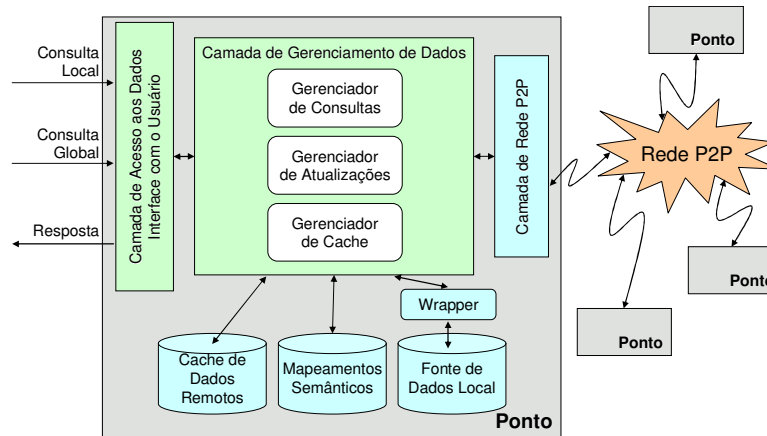


Figura 3.7 – Arquitetura Genérica de um Ponto em um PDMS [Sung et al. 2005]

Um ponto é formado basicamente por três camadas com funções distintas: (i) *camada de interface*, utilizada para submissão de consultas pelos usuários; (ii) *camada de gerenciamento de dados*, responsável pelo processamento de consultas e manipulação de metadados; e (iii) *camada de rede*, que garante a conectividade do ponto na rede.

As consultas submetidas a um ponto podem ser globais ou locais. Uma consulta local caracteriza-se por ter sido formulada no próprio ponto. Uma consulta global é normalmente reformulada em consultas locais e enviadas para outros pontos. Assim, uma consulta local é, na verdade, parte de uma consulta global que foi formulada originalmente em um outro ponto.

Quando uma consulta global é submetida, o módulo Gerenciador de Consultas (GC) recupera informações sobre mapeamentos semânticos armazenadas no Repositório de Mapeamentos Semânticos. Tais informações são utilizadas para identificar outros pontos capazes de responder a mesma consulta. Estes pontos são denominados **pontos relevantes** para uma consulta. Uma vez identificados, as mesmas informações são utilizadas para reformular a consulta global em consultas locais de modo que os pontos relevantes possam compreendê-la. Alguns PDMS armazenam as informações sobre mapeamentos semânticos em pontos especializados. Assim, o ponto original necessita

comunicar-se com os mesmos ou transmitir a consulta global para que possam processá-la.

Durante o processamento de consultas, o GC precisa solicitar os serviços oferecidos pela camada de rede para comunicação com outros pontos. O modo como uma consulta global é executada varia de acordo com a implementação do sistema. Em alguns PDMS, os dados dos pontos relevantes são retornados e integrados no ponto que iniciou a consulta. Outros PDMS possuem pontos especializados que auxiliam o processamento de consultas. Em ambos os casos, para melhorar o processamento de consultas futuras e similares, é comum o uso de cache para armazenar dados retornados por pontos relevantes. O módulo Gerenciador de Cache é responsável por gerenciar o Repositório de Cache de Dados Remotos. O armazenamento de cache pode ser feito em qualquer ponto ou apenas em pontos especializados.

O GC pode também executar consultas locais submetidas a partir de pontos remotos. Como um PDMS admite a participação de fontes de dados heterogêneas, um *wrapper* é utilizado para ocultar a heterogeneidade de linguagens de consulta, formato de armazenamento de dados, ou quaisquer outros tipos de incompatibilidade entre os pontos. O *wrapper* localiza-se entre a fonte dados local e a camada de gerenciamento de dados. Alguns PDMS utilizam ainda replicação para melhorar o desempenho de consultas. Neste caso, quando os dados são atualizados, o Gerenciador de Atualizações é responsável por sincronizar as réplicas mantidas em pontos remotos.

A camada de rede disponibiliza serviços de comunicação para a camada de gerenciamento de dados. Sua implementação depende do tipo de topologia de rede empregada no PDMS. A seguir, apresentamos alguns dos principais problemas enfrentados no desenvolvimento de PDMS.

3.3 Problemas do Gerenciamento de Dados em Sistemas P2P

O compartilhamento de dados estruturados em ambientes P2P pode ser considerado uma tarefa complexa e desafiadora em virtude de vários problemas: autonomia dos pontos, heterogeneidade dos esquemas compartilhados e relutância em compartilhar o esquema armazenado nos diversos pontos. Conforme discutido na seção 3.1, as abordagens tradicionais para gerenciamento de dados distribuídos utilizam soluções que muitas vezes não se aplicam (ou se aplicam parcialmente) a um PDMS.

Neste ambiente, as técnicas precisam considerar principalmente aspectos como escalabilidade, auto-organização e volatilidade dos pontos.

Nesta seção, são discutidos alguns dos problemas mais comuns encontrados no desenvolvimento de PDMS, dentre os quais destacamos: a identificação de mapeamentos semânticos entre esquemas e o processamento de consultas. O problema da conectividade de pontos em PDMS, por ser de especial interesse para este trabalho, será discutido separadamente no Capítulo 4.

3.3.1 Identificação de Mapeamentos Semânticos entre Esquemas

Os mapeamentos entre esquemas definem a equivalência semântica entre termos presentes em dois ou mais esquemas distintos [Kementsietsidis et al. 2003]. Como acontece nos tradicionais sistemas de bancos de dados distribuídos, o objetivo destes mapeamentos é fornecer um ambiente de consultas único que oculte a autonomia, distribuição e heterogeneidade das fontes de dados. Em sistemas de integração de dados os mapeamentos são estabelecidos entre o esquema de mediação e as fontes de dados. Nos PDMS os mapeamentos são estabelecidos entre os elementos dos esquemas exportados pelos pontos. Normalmente, os mapeamentos são definidos manualmente, entretanto várias técnicas para automatização deste processo têm sido propostas [Rahm et al. 2001, Belian 2005].

Quando um novo ponto entra no sistema é preciso fornecer mapeamentos para um pequeno número de pontos adjacentes, denominados pontos vizinhos. Idealmente, os esquemas dos pontos vizinhos devem ser similares ao do novo ponto.

O conjunto de mapeamentos define a rede semântica de um PDMS. A otimização da rede semântica considera principalmente: a eliminação de mapeamentos redundantes, a redução da distância entre os pontos (para reduzir a perda de informação durante a reformulação de consultas) e a identificação de pontos inacessíveis.

Para demonstrar um dos problemas principais presente no compartilhamento de dados estruturados em um PDMS, a identificação de mapeamentos semânticos entre esquemas, considere os esquemas exportados da Figura 3.8. Neste PDMS, cada ponto representa uma universidade e o objetivo é o compartilhamento de dados sobre projetos de pesquisa (exemplo adaptado de [Halevy et al. 2004]). As universidades podem representar seus dados através de esquemas e modelos distintos.

Considere que a UFPE deseja acessar o banco de dados da UFRGS e vice-versa. À princípio, sabemos que as universidades armazenam dados semanticamente equivalentes. Como a UFPE e a UFRGS participam de projetos em comum, é possível que as informações destes projetos estejam armazenadas de modo duplicado. Assumindo que foram definidos mapeamentos semânticos entre o esquema exportado pela UFPE e pela UFRGS, quando o ponto USP conectar-se ao sistema, os dados da UFRGS também podem ser acessados. Para isto, basta que sejam definidos novos mapeamentos semânticos entre o esquema da USP e o da UFPE.

UFPE	UFRGS
Áreas (id_area, nome, descrição)	Área (idarea, nome, descr)
Projetos (id_proj, id_area, nome)	Projeto (projnome, financiador)
Publicações (id_pub, título, ano)	ProjArea (projnome, idarea)
PublicaçõesAutor (id_pub, id_autor)	Pubs (idpub, projnome, título, local, a
PublicaçõesProjeto (id_pub, id_proj)	Autor (idpub, autor)
Pesquisador (id_pes, nome)	Membro (projnome, membro)
MembroProjeto (pes_id, proj_id, função)	
 USP	
Projeto (idproj, nome, desc)	•••
Aluno (idaluno, nome, sobrenome, status)	
Pesquisador (iddept, nome, sobrenome, rank, sala)	
MembroProjeto (idproj, idmembro)	
Artigo (idartigo, título, evento, ano)	
Autor (idautor, idartigo)	

Figura 3.8 – Exemplos de Esquemas Exportados, adaptado de [Halevy et al. 2004]

Diversos fatores inviabilizam a utilização de um esquema global único pelos PDMS:

- O comportamento dinâmico dos pontos acarretaria uma atualização constante do esquema global;
- Enquanto alguns pontos desejam compartilhar o esquema completo, outros estão interessados em compartilhar apenas uma parte;
- O esquema global único necessitaria ser armazenado de modo centralizado ou distribuído. No primeiro caso, o ponto que armazenasse o esquema global representaria um ponto único de falha no sistema e possivelmente necessitaria de manutenções contínuas para suportar o crescimento do sistema. Já se o esquema global fosse mantido de forma distribuída, seriam necessárias técnicas para garantir uma visão integrada do esquema global. Caso o esquema global fosse replicado talvez

nem todos os pontos tivessem capacidade de armazenamento suficiente. Além disso, seria preciso a utilização de técnicas para manter a consistência das réplicas.

Portanto, em um PDMS os mapeamentos são estabelecidos entre os próprios pontos. Conforme ilustrado na Figura 3.9, dois tipos de mapeamentos são utilizados pelos PDMS, **mapeamento local** e **mapeamento entre esquemas**. O mapeamento local define a equivalência semântica entre os elementos do esquema da fonte de dados e do esquema exportado. No mapeamento entre esquemas são estabelecidos relacionamentos semânticos entre os esquemas exportados por pontos distintos.

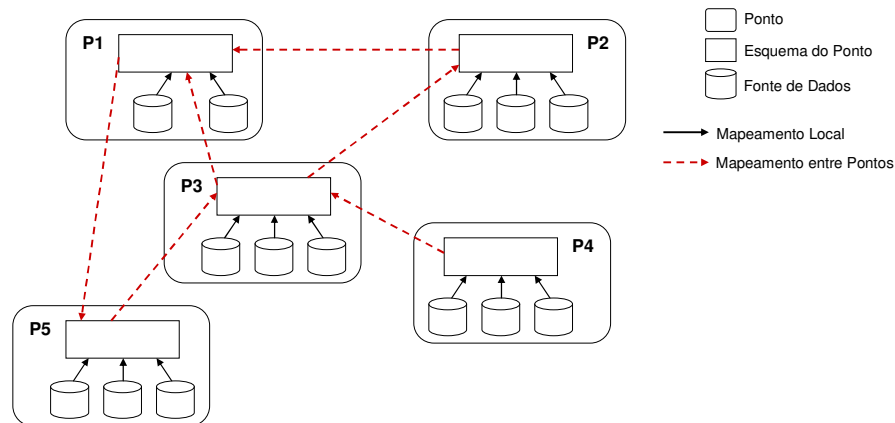


Figura 3.9 – Mapeamentos em um PDMS

Três tipos de abordagem para definição de mapeamentos entre esquemas são comumente utilizados: *pair mappings*, *peer-mediated mappings* e *super-peer mediated mappings* [Sung et al. 2005]. Independentemente da abordagem utilizada, a maioria dos PDMS tenta explorar relacionamentos de transitividade entre os esquemas dos pontos para compartilhamento e integração de dados [Halevy et al. 2004].

Na abordagem *pair mappings*, os mapeamentos são definidos entre pares de pontos. No exemplo da Figura 3.10a, adaptada de [Halevy et al. 2004], um *pair mapping* é um mapeamento definido entre as universidades UFPE e UFRGS. Os mapeamentos são armazenados no ponto que deseja acessar os dados do seu vizinho. O ponto USP pode acessar os dados armazenados no ponto UFRGS desde que sejam definidos *pair mappings* entre ambos, ou existam *pair mappings* entre USP e UFPE, e entre UFPE e UFRGS (transitividade). No Modelo Relacional Local [Bernstein et al. 2002], cada ponto define como seu esquema está relacionado com o esquema de outro ponto através de um conjunto de regras de tradução e fórmulas de coordenação, chamadas de *acquaintances*. A

definição das *acquaintances* entre pares de pontos forma uma rede semântica. Em [Rouse et al. 2006], os mapeamentos são estabelecidos entre pares de super-pontos.

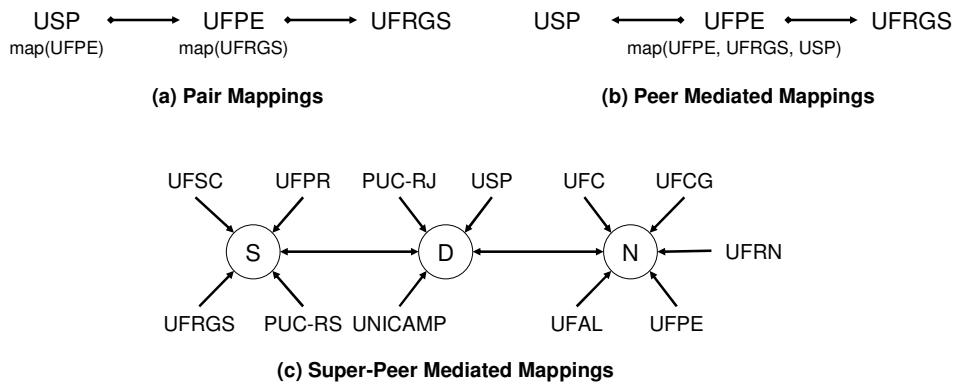


Figura 3.10 – Tipos de Mapeamentos entre Esquemas, adaptada de [Halevy et al. 2004]

Na abordagem *peer-mediated*, os pontos definem mapeamentos envolvendo esquemas de dois ou mais pontos. Os mapeamentos *peer-mediated* representam uma generalização dos mapeamentos *pair mappings*. Na Figura 3.10b, a UFPE disponibiliza um mapeamento representando os dados armazenados na UFPE, UFRGS e USP. Uma quarta universidade que deseje acessar os dados disponíveis nestes pontos, pode utilizar o esquema *peer mediated* disponível no ponto UFPE. Neste caso, o ponto UFPE não oferece dados e atua como um simples mediador entre os pontos UFRGS e USP. Como exemplo de PDMS que segue a abordagem *peer-mediated*, citamos o Piazza [Tatarinov et al. 2003].

A terceira abordagem para mapeamentos, chamada de *super-peer mediated mappings*, é utilizada por PDMS baseados na topologia super-ponto. Nestes sistemas, o super-ponto de cada cluster disponibiliza um esquema de mediação representando os dados compartilhados pelos outros pontos do cluster. Os mapeamentos ocorrem entre o esquema do super-ponto e o esquema dos pontos associados, e entre os esquemas dos próprios super-pontos. Na Figura 3.10c, o super-ponto S (Sul) armazena os mapeamentos entre seu esquema de mediação e os esquemas dos pontos UFRGS, UFPR, UFSC e PUC-RS. O super-ponto D (Sudeste) armazena os mapeamentos entre seu esquema de mediação e os dos pontos PUC-RJ, USP e UNICAMP. Por último, o super-ponto N (Nordeste) disponibiliza os mapeamentos entre seu esquema de mediação e os esquemas dos pontos UFPE, UFC, UFCG, UFAL e UFRN. Quando o ponto UFPE necessita consultar os dados do ponto UFC, os mapeamentos disponíveis no super-ponto N são utilizados. Se o ponto UFPE consulta os dados do ponto UFRGS, são utilizados os mapeamentos entre os super-pontos N e S, assim como os

mapeamentos entre o super-ponto S e o ponto UFRGS. O sistema Edutella [Nejdl et al. 2002] utiliza este tipo de mapeamento.

3.3.2 Processamento de Consultas

Um PDMS é composto por inúmeros pontos que disponibilizam seus dados através de um esquema exportado. As consultas submetidas em um ponto podem ser respondidas com os dados disponíveis no próprio ponto ou com dados armazenados nos demais pontos do sistema [Tatarinov et al. 2004]. No entanto, para que os dados dos outros pontos possam ser considerados, é necessário que os pontos sejam conectados através de **caminhos de mapeamentos**, isto é, um conjunto de mapeamentos entre esquemas de pontos distintos que faz com que uma consulta submetida em um ponto P_1 possa ser reformulada e executada em um ponto P_n . Note que P_1 e P_n não necessariamente precisam ser pontos adjacentes. Por exemplo, na Figura 3.11a uma consulta submetida em P_2 pode ser executada em P_4 desde que exista um caminho de mapeamento entre P_2 e P_1 , e P_1 e P_4 . Em geral, o processamento de consultas típico em um PDMS ocorre como exposto na Figura 3.11b:

1. Suponha que uma consulta Q é submetida em P_2 de acordo com o esquema de P_2 ;
2. Se P_2 possui dados que satisfazem Q , então Q é executada em P_2 ;
3. Em seguida, de acordo com os mapeamentos correspondentes, Q é reformulada para os vizinhos de P_2 (P_1 e P_3);
4. As consultas reformuladas são submetidas aos vizinhos de P_1 (P_4) e P_3 ($\{ \}$) e assim sucessivamente até que todos os pontos sejam alcançados ou algum limite seja atingido. Cada ponto pode responder a consulta inteira ou apenas uma parte dela;
5. Os pontos que receberam a consulta (P_1 , P_3 e P_4) retornam os dados para P_1 que, por sua vez, integra-os e apresenta ao usuário.

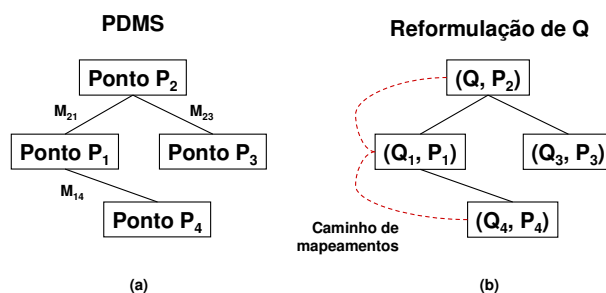


Figura 3.11 – Processamento de Consultas Típico em um PDMS

Durante o processamento de consultas, um ponto pode atuar como simples mediador, isto é, apenas repassa a consulta para outros pontos. Este tipo de ponto é bastante comum no sistema Piazza [Tatarinov et al. 2003], a ser descrito na subseção 3.4.1. Estes pontos não contribuem com dados porque disponibilizam apenas esquemas virtuais que possibilitam o estabelecimento de mapeamentos transitivos entre outros pontos que realmente armazenam dados.

Em um PDMS, um ponto só é acessível através dos mapeamentos. Com o acréscimo de pontos no sistema a quantidade de esquemas e mapeamentos também aumenta, o que pode ocasionar problemas como: caminhos de mapeamentos muito extensos (muitas reformulações de consulta), perda de semântica à medida que as consultas são repassadas (reformuladas) e restrições de tempo de resposta. Para reduzir o tamanho dos caminhos de mapeamentos, ontologias [Gruber 1993, Guarino 1998] podem ser utilizadas. Através de ontologias, é possível formar “atalhos” entre pontos inicialmente não-adjacentes, cujo conteúdo esteja semanticamente relacionado (ver Figura 3.12). Esta solução é utilizada em [Heese et al. 2005]. O uso de ontologias em PDMS é discutido de forma mais aprofundada no capítulo posterior.

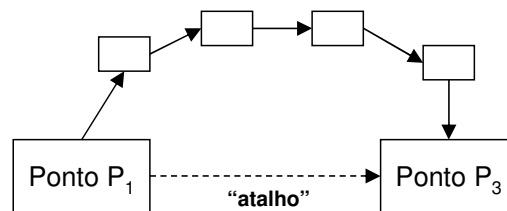


Figura 3.12 – Uso de Ontologias para definição de “atalhos” entre Pontos não-adjacentes

Os planos de consulta podem ser modificados ao trafegarem pela rede. Neste caso, cada ponto resolve parte do plano recebido e, em seguida, redireciona-o para outros pontos. [Papadimos et al. 2003] introduzem o conceito de “planos de consulta mutantes”, no qual os planos de consulta são inseridos na rede por um dos pontos. Qualquer ponto pode modificar o plano de consulta de duas maneiras: associar nomes de recursos a sua localização, ou fornecer dados referentes a algum recurso, se souber como. Os pontos podem reduzir o plano de consulta avaliando um sub-grafo do plano. Se o plano de consulta for totalmente avaliado, o mesmo é devolvido ao ponto solicitante, caso contrário é enviado para outro ponto capaz de continuar o processamento.

O trabalho de [Bernstein et al. 2002] propõe o Modelo Relacional Local, no qual o gerenciador de consultas de cada ponto utiliza regras para tradução de dados e

dependências semânticas para reformular consultas submetidas ao ponto do acordo com o esquema de outros pontos. No sistema Chatty Web [Aberer et al. 2003], os autores propõem a associação de medidas de qualidade aos mapeamentos entre esquemas utilizados em uma consulta. As medidas são atualizadas à medida que a consulta trafega na rede. A qualidade dos mapeamentos entre esquemas é medida pela similaridade sintática e semântica. A similaridade sintática refere-se à quantidade de informação perdida nas consultas quando atributos de um esquema não existem em outro esquema. A similaridade semântica refere-se ao nível de concordância semântica entre os esquemas, e é medida pelas transformações sofridas por uma consulta quando expressa em outro esquema.

3.4 Exemplos de PDMS

Diversas pesquisas têm sido desenvolvidas com o objetivo de propor soluções para os problemas referentes ao gerenciamento de dados estruturados, surgidos com a disseminação dos sistemas P2P. Na literatura, vários PDMS têm sido propostos para tratar destes problemas: Piazza [Tatarinov et al. 2003], PeerDB [Ooi et al. 2003a], XPeer [Bellahsene et al. 2004a], Edutella [Nejdl et al. 2002], APPA [Valduriez et al. 2004], Hyperion [Arenas et al. 2003] e Rosa-P2P [Brito 2005].

Na prática, cada PDMS trata apenas de um conjunto de problemas, por exemplo, identificação de mapeamentos semânticos e processamento de consultas. Esta seção oferece uma breve descrição de alguns dos principais PDMS, assim como as soluções propostas para os problemas anteriores. Ao final da seção, apresentamos um quadro comparativo entre os PDMS discutidos.

3.4.1 Piazza

Piazza [Tatarinov et al. 2003, Halevy et al. 2003a, Halevy et al. 2003b, Halevy 2004] é um PDMS desenvolvido pelas Universidades de Washington e da Pensilvânia. Seu objetivo é integrar fontes de dados heterogêneas e distribuídas de forma escalonável. Para possibilitar o compartilhamento de dados, o sistema utiliza uma estratégia de mapeamentos entre esquemas. Cada ponto disponibiliza seu esquema exportado a ser compartilhado com os demais pontos. O esquema representa a “visão do mundo” desejada pelo ponto.

A arquitetura do sistema é puramente descentralizada, não existindo a presença de um esquema global único. Os pontos interessados em compartilhar dados devem estabelecer mapeamentos entre seus esquemas. O conjunto desses mapeamentos define a semântica do sistema. Os usuários formulam consultas de acordo com seu esquema de preferência. Para atender à requisição, o sistema expande recursivamente os mapeamentos, recuperando dados armazenados em outros pontos.

Dois tipos de mapeamento são utilizados no Piazza (ver Figura 3.13): (i) *mapeamento entre pontos*: relaciona os esquemas disponibilizados por dois ou mais pontos, ou seja, define as correspondências entre as “visões do mundo” dos diferentes pontos; (ii) *descrição de armazenamento*: relaciona o esquema armazenado em um ponto com o esquema disponibilizado pelo ponto, isto é, mapeia os dados reais armazenados para a “visão do mundo” desejada pelo mesmo.

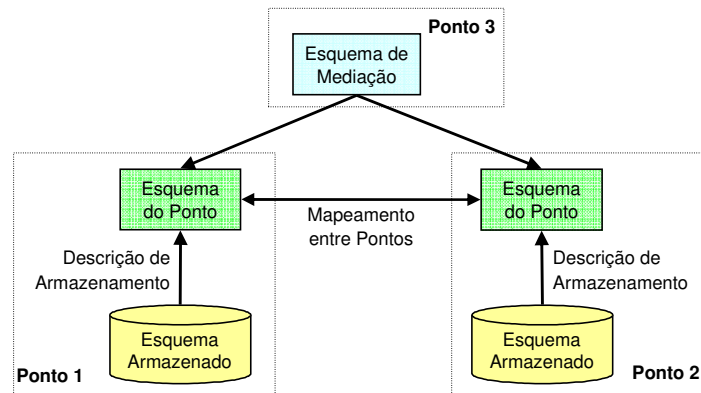


Figura 3.13 – Tipos de Mapeamento no Piazza [Tatarinov et al. 2003]

A Figura 3.14, adaptada de [Tatarinov et al. 2003], ilustra um exemplo sobre os tipos de mapeamento entre esquemas. O esquema (virtual) disponibilizado pelo ponto *DB-Projects* é mapeado nos esquemas exportados pelos pontos *UPenn* e *UW* (Figura 3.14a). Ainda, o esquema de *Stanford* é mapeado para o esquema de *Berkeley*. Com o novo mapeamento, do ponto *Stanford* para *UW* (Figura 3.14b), uma consulta submetida em qualquer ponto pode ser respondida acessando-se também os dados dos outros pontos. O acesso aos dados só é possível em decorrência da avaliação transitiva dos mapeamentos semânticos de *Stanford* para *DB-Projects* (Figura 3.14c). Analogamente, a presença de um novo esquema virtual (*Projects Mediation_n*) mapeando seu esquema para os esquemas dos pontos *Stanford* e *Berkeley*, possibilita a geração transitiva de mapeamentos entre *Projects Mediation₁* e *Projects Mediation_n*. Consultas passariam a poder ser respondidas por quaisquer pontos dos sistema (Figura 3.14d).

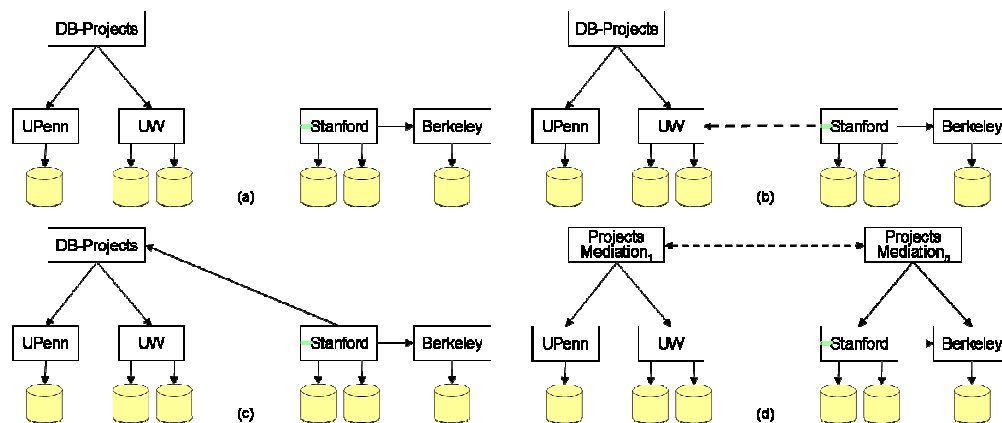


Figura 3.14 – Exemplo de Mapeamentos no Piazza, adaptado de [Tatarinov et al. 2003]

3.4.2 PeerDB

O sistema PeerDB [Ooi et al. 2003a] é um PDMS implementado sobre a plataforma BestPeer [Ng et al. 2002], voltada para o desenvolvimento de aplicações P2P baseada na topologia pura não-estruturada, desenvolvida pela Universidade de Singapura.

No sistema PeerDB, cada ponto representa um sistema de gerenciamento de dados, proporcionando buscas baseadas em conteúdo. Os dados armazenados em um ponto são gerenciados através de um Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR), sendo compartilhados sem a presença de um esquema global e acessados através da linguagem SQL. Para auxiliar o processamento de consultas, o sistema utiliza agentes inteligentes.

A Figura 3.15 ilustra a arquitetura de um ponto PeerDB. Basicamente, ela é composta de 3 (três) camadas:

- *Camada P2P*: oferece serviços para compartilhamento de dados e descoberta de recursos na rede;
- *Camada de Agentes Inteligentes*: sistema multi-agente cuja função principal é disponibilizar um ambiente para que agentes móveis possam operar adequadamente nos vários pontos da rede. Cada ponto possui um agente principal (DBAgent) que gerencia as consultas dos usuários. O DBAgent é ainda responsável por monitorar estatísticas sobre pontos vizinhos e gerenciar políticas de reconfiguração de pontos na rede;

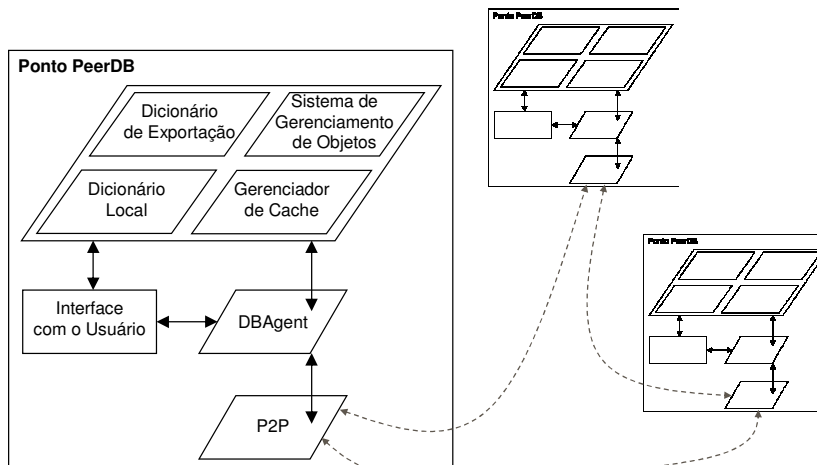


Figura 3.15 – Arquitetura de um Ponto PeerDB [Ooi et al. 2003a]

- *Camada de Gerenciamento de Dados:* possibilita a manipulação dos dados disponíveis no ponto. É subdividida em 4 (quatro) componentes:
 - *Sistema de Gerenciamento de Objetos:* oferece armazenamento, processamento e recuperação dos dados no ponto;
 - *Gerenciador de Cache:* proporciona o armazenamento temporário de resultados retornados pelos diversos pontos do sistema, com o objetivo de minimizar o tempo de resposta de consultas subseqüentes. O gerenciamento dos dados em cache é feito através de um algoritmo LRU;
 - *Dicionário Local:* contém metadados sobre o esquema disponível no ponto;
 - *Dicionário de Exportação:* contém metadados (por exemplo, palavras-chave) referentes às tabelas e colunas compartilhadas (ver Tabela 3.2). É um subconjunto dos metadados do Dicionário Local.

Quando uma consulta é submetida em um ponto, o DBAgent cria agentes auxiliares e envia-os ao conjunto de pontos vizinhos. Apenas os agentes auxiliares situados nos pontos aptos a responderem a consulta retornam metadados referentes às tabelas e colunas envolvidas na consulta. Em seguida, o DBAgent compara sintaticamente os metadados recebidos com os metadados do ponto onde a consulta foi submetida. Durante a etapa de comparação de metadados, o DBAgent pode interagir diretamente com o usuário para resolução de conflitos semânticos. Apenas após a escolha dos pontos é que o DBAgent reformula a consulta com base nos metadados recebidos e reenvia aos pontos selecionados. Finalmente, os dados recebidos são combinados e apresentados ao usuário final.

Tabela 3.2 – Palavras-Chave para Nomes de Relações e Atributos [Ng et al. 2003]

Ponto	Nome	Tipo	Palavras-Chave
P ₁	Kinases	Tabela	{Protein, Human}
	SeqID	Coluna	{Key, Identifier, ID}
	Length	Coluna	{Length}
	ProteinSeq	Coluna	{Sequence, Protein Sequence}
P ₂	Protein	Tabela	{Protein, Annexin, Zebrafish}
	SeqNo	Coluna	{Number, Identifier}
	Len	Coluna	{Length}
	Sequence	Coluna	{Sequence}
P ₃	ProteinKLen	Tabela	{Protein, Kinases, Length}
	ID	Coluna	{Number, Identifier}
	SeqLength	Coluna	{Length}
	ProteinKSeq	Relação	{Protein, Sequence}
	ID	Coluna	{Number, Identifier}
	Sequence	Coluna	{Sequence}
P ₄	Protein	Tabela	{Protein, Kinases, Annexin, ...}
	Name	Coluna	{Name}
	Char	Coluna	{Characteristics, Features, Functions}

A utilização de agentes visa evitar o tráfego desnecessário de dados, ou seja, com auxílio dos metadados, os pontos aptos a responder uma consulta são primeiramente identificados e, somente depois, os dados neles contidos são enviados para o ponto solicitante. Além disso, agentes possibilitam que o ponto atualize as estatísticas sobre os pontos vizinhos, facilitando a execução de consultas posteriores.

A escolha dos pontos é feita através de comparações sintáticas entre os nomes de tabelas e colunas da consulta e as palavras-chave contidas no Dicionário de Exportação dos pontos envolvidos. Um função de similaridade sintática é utilizada para atribuir pesos a cada comparação. Caso o valor retornado pela função seja maior do que um limite estipulado pelo próprio usuário (*threshold*), o ponto é considerado apto para responder à consulta. Considerando a Tabela 3.2, suponha que a seguinte consulta seja submetida no ponto P₁:

```
SELECT SeqId, proteinSeq FROM Kinases WHERE length > 30
```

Como uma das palavras-chave da tabela *Kinases* é *protein*, e *protein* também é uma palavra-chave da tabela *protein* pertencente a P₂ e das tabelas *ProteinKLen* e *ProteinKSeq* pertencentes a P₃, então todas essas tabelas satisfazem a tabela *Kinases* da consulta analisada. Analogamente, os atributos *SeqID*, *ProteinSeq* e *Length* possuem palavras-chave em comum com P₂ e P₃. Em relação a P₃, é necessária a utilização de uma junção para

reformulação da consulta. Para P_4 , existe apenas um “casamento” entre as palavras-chave da tabela, e não entre as das colunas. Dessa forma, P_4 é classificado abaixo de P_2 e P_3 .

O sistema PeerDB possui forte dependência do usuário, visto que durante processamento de consultas o usuário necessita confirmar os pontos que devem ser consultados. Além disso, a qualidade do resultado das consultas depende extremamente do processo de definição de palavras-chaves. A presença de termos homônimos (mesmo nome, porém significados distintos) pode comprometer o resultado das consultas.

3.4.3 XPeer

A proposta do sistema XPeer [Bellahsène et al. 2004a] é a realização de consultas em um PDMS ocultando a distribuição, localização e heterogeneidade das fontes de dados, e oferecendo tempos de resposta aceitáveis para os usuários.

O sistema XPeer baseia-se na topologia super-ponto [Bellahsène et al. 2004b]. Os pontos que compartilham um mesmo esquema são agrupados em um único cluster. Em cada cluster, o esquema compartilhado é gerenciado por um super-ponto. Conforme ilustrado na Figura 3.16, todos os pontos de um mesmo cluster comunicam-se diretamente com o super-ponto.

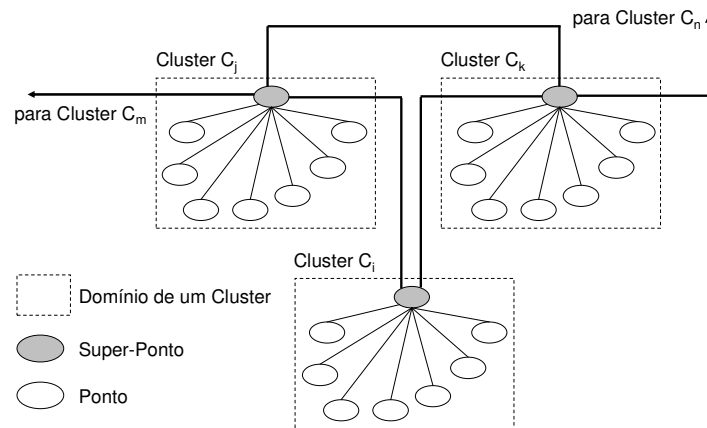


Figura 3.16 – Clusters no PDMS XPeer [Bellahsène et al. 2004a]

Os pontos associados a um super-ponto compartilham informações sobre um único domínio e por isso compartilham o mesmo esquema. Diferentes super-pontos podem agrupar pontos que compartilham informações sobre um mesmo domínio. Dessa forma, aplicações que desejem compartilhar informações sobre *hospitais*, por exemplo, podem precisar acessar mais de um cluster. Para atender esse tipo de necessidade, dois outros tipos de ponto são definidos no XPeer: pontos de domínio, responsáveis por

gerenciar um conjunto de super-pontos; e ponto global, cuja função é gerenciar o conjunto completo de pontos de domínio. Um domínio é definido como sendo um conjunto de clusters que compartilham uma mesma categoria de informação. Para cada domínio, existe um ponto de domínio, que serve como ponto de entrada para os pontos que desejam compartilhar informações sobre o domínio em questão. Um domínio é comumente chamado de comunidade semântica. No próximo capítulo, discutiremos mais detalhadamente sobre este assunto.

Assim como em qualquer sistema P2P, os pontos podem conectar-se e desconectar-se livremente, afetando apenas o super-ponto local. Para que um super-ponto conecte-se ao sistema, é necessário que o mesmo registre-se no domínio apropriado, fornecendo seu identificador, localização e esquema de mediação. O ponto de domínio, por sua vez, repassa as informações recebidas a todos os seus super-pontos. Em seguida, os demais super-pontos devolvem ao novo super-ponto seus respectivos identificadores, localizações e descrições. Quando um super-ponto desconecta-se do sistema, o ponto de domínio promove um dos pontos associados ao super-ponto que acabou de deixar o sistema para exercer o papel de novo super-ponto. O ponto de domínio notifica todos os outros super-pontos e remove os metadados do super-ponto de sua lista.

A presença de um esquema de mediação por cluster e o dinamismo do ambiente P2P sugerem que o formalismo LAV seja o mais indicada para especificação de mapeamentos entre os esquemas dos pontos e o esquema de mediação. A principal vantagem é que um ponto pode ser adicionado ou removido sem afetar o esquema dos outros pontos e o esquema de mediação. O único requisito necessário é a atualização da lista de pontos disponíveis. Quando um ponto deseja consultar o sistema inteiro, o ponto recebe um esquema de mediação referente aos esquemas dos clusters desejados. Esse processo é realizado seguindo o formalismo para especificação de mapeamentos GAV.

3.4.4 Edutella

Edutella [Nejdl et al. 2002, Löser et al. 2003b] é um sistema P2P para compartilhamento de recursos educacionais, desenvolvido através da plataforma JXTA. Os recursos compartilhados pelos pontos são descritos através de esquemas representados no formato RDF [W3C 2004]. RDF-QEL [Nejdl et al. 2003] é utilizada como linguagem de consulta. Os principais objetivos do sistema são: (i) suportar

múltiplos formatos de representação de esquemas (como Dublin Core e LOM); (ii) permitir uma melhor descrição dos recursos disponíveis nos pontos; (iii) proporcionar a elaboração de consultas mais complexas.

Para garantir uma melhor escalabilidade, o sistema adota o modelo de super-pontos. Os super-pontos são responsáveis por tarefas como roteamento de consultas e integração/mediação de metadados. Os super-pontos organizam-se de acordo com a topologia HyperCuP [Schlosser et al. 2003]. Esta topologia é considerada bastante eficiente para execução de consultas, ao propagá-las sem redundância.

Um esquema global é disponibilizado em cada super-ponto. Similarmente aos sistemas mediadores, os esquemas dos pontos associados a um super-ponto são mapeados para o esquema global do seu respectivo super-ponto. Os super-pontos são responsáveis ainda pela construção, armazenamento e manutenção de índices de roteamento. Um índice de roteamento contém metadados sobre o esquema dos pontos participantes do cluster. Os metadados são úteis para que os super-pontos possam direcionar as consultas recebidas diretamente para os pontos (ou outros super-pontos) capazes de respondê-las, melhorando assim o desempenho do sistema. Os super-pontos mantêm dois tipos de índice de roteamento: super-ponto/ponto (SP/P) e super-ponto/super-ponto (SP/SP).

Os índices de roteamento SP/P armazenam metadados sobre os pontos conectados ao super-ponto. Os metadados descrevem as principais características de um determinado ponto sob diferentes granularidades, e são enviados no momento em que o ponto conecta-se ao super-ponto. Para garantir que os índices permaneçam atualizados, sempre que ocorre uma alteração no conteúdo de um ponto, este notifica seu respectivo super-ponto, enviando uma mensagem. Caso um ponto se desconecte da rede, suas entradas no índice de roteamento são removidas.

Quando uma consulta é destinada a um super-ponto e o acesso ao seu índice de roteamento SP/P retorna zero entradas, significa que nenhum de seus respectivos pontos é capaz de responder a consulta recebida. Neste caso, o super-ponto utiliza seu índice de roteamento SP/SP para descobrir outros super-pontos aptos a atender à solicitação inicial. Os super-pontos selecionados recebem a consulta e utilizam seus índices SP/P para repassar a consulta para seus pontos capazes de respondê-las, reiniciando o processo. Os índices SP/SP armazenam resumos dos metadados extraídos dos índices SP/P.

Sempre que ocorre uma atualização no índice SP/P de um super-ponto, esta é propagada nos índices SP/SP dos demais super-pontos. Embora esse comportamento não seja considerado ideal em termos de desempenho, o custo das atualizações pode ser minimizado pelos seguintes fatores [Löser et al. 2003b]: (i) a quantidade de super-pontos no sistema é extremamente menor do que a de pontos; (ii) as atualizações podem ser feitas em *batch*, obedecendo intervalos de tempo pré-determinados; (iii) ao invés de ocorrerem atualizações sempre que houverem mudanças no conteúdo dos pontos e quando os pontos conectarem-se ao sistema, basta que sejam feitas quando acontecer o último caso.

Os super-pontos utilizam políticas de clusterização baseadas em regras para controlar a organização de rede e, conseqüentemente, diminuir a quantidade de consultas propagadas. Através das políticas é possível limitar a quantidade de pontos conectados a um super-ponto, agrupando-os de acordo com características estáticas (domínio, faixa de endereços IP e formato de representação do esquema) e dinâmicas (quantidade de recursos disponíveis e tempo de resposta para consultas).

As políticas de clusterização são definidas pelo administrador do super-ponto. Estas políticas baseiam-se em regras do tipo evento-condição-ação (ECA), onde os eventos podem ser a conexão/desconexão de um ponto ou a verificação pelo super-ponto da situação de um ponto já conectado. No exemplo da regra ECA a seguir, extraído de [Löser et al. 2003b], o super-ponto deseja agrupar apenas pontos cujo formato de representação de esquemas seja LOM ou DC, e sejam membros do domínio “cs.tu-berlin.de”.

```
ON (Event) Enter IF (
  ((Peer.Advertisement.Property query_schema="LOM") OR
   Peer.Advertisement.Property query_schema="DC")
)
AND
(Peer.Advertisement.Property peer_name
INCLUDE "cs.tu-berlin.de")
)
DO (Action) Approve(Peer)
ELSE (Action) Reject(Peer)
```

3.4.5 Outros PDMS

APPA [Valduriez et al. 2004, Akbarinia et al. 2006] é um PDMS cuja arquitetura é dividida em camadas e cada camada oferece um conjunto diferenciado de serviços. O sistema funciona em qualquer tipo de topologia de rede no sentido de atender aos requisitos de diferentes aplicações P2P. Suas camadas incluem: (i) *camada de rede*: oferece

serviços comuns a todas as redes P2P; (ii) *camada de serviços básicos*: disponibiliza serviços para conectividade, armazenamento e recuperação de metadados e comunicação entre pontos na rede; e (iii) *camada de serviços avançados*: oferece serviços essenciais para o compartilhamento de dados, como gerenciamento de esquemas, processamento de consultas e replicação de dados. Uma característica particular deste sistema é a forte utilização da replicação de dados para tornar o processamento de consultas mais eficiente.

O projeto Hyperion [Arenas et al. 2003, Bernstein et al. 2002] possui como objetivo principal o compartilhamento e a coordenação de dados entre pontos autônomos e independentes. Cada ponto corresponde a um um SGBD convencional acrescido de uma camada extra de interoperabilidade P2P. Para compartilhamento de dados, os pontos estabelecem relacionamentos dinâmicos (ou *acquaintances*), que se caracterizam pela definição direta de um ou mais mapeamentos semânticos entre dois pontos distintos [Kementsietsidis et al. 2003]. Os mapeamentos podem ser definidos no nível de esquema e/ou de dados. Para auxiliar o processo de criação de mapeamentos são utilizadas tabelas e expressões de mapeamento. A coordenação de dados envolve a reconciliação e integração de dados em tempo de consulta e a manutenção da consistência dos dados armazenados nos pontos.

ROSA-P2P [Brito 2005, Brito et al. 2005] é um sistema P2P para compartilhamento de conteúdo acadêmico, cuja arquitetura é inspirada no sistema Edutella. O sistema baseia-se na topologia super-ponto e apresenta estratégias para conectividade, balanceamento de carga e tolerância a falhas. Agrupamentos e agregações de pontos são utilizadas para otimizar o processamento de consultas. Vocabulários controlados são utilizados para atribuir valor semântico ao conteúdo compartilhado pelos pontos. Estes recursos são indispensáveis para localização de pontos, reescrita de consultas e resolução de conflitos semânticos.

3.4.6 Estudo Comparativo entre PDMS

Com a finalidade de destacar as similaridades e diferenças entre os vários PDMS discutidos na seção anterior, a Tabela 3.3 ilustra um resumo sobre as características principais destes PDMS. Determinadas características não foram encontradas na literatura para alguns dos PDMS. Neste caso, as células estão preenchidas com um hífen (“-”). O significado de cada característica analisada é apresentado a seguir:

- *Instituição*: nome da instituição de ensino que propôs o PDMS. Mais de uma instituição pode estar envolvida com o mesmo projeto;
- *Representação dos dados*: formato de representação de dados no PDMS;
- *Topologia*: topologia de rede adotada no PDMS. Alguns PDMS aceitam mais de uma topologia;
- *Integração semântica*: formato de representação das informações semânticas;
- *Mapeamentos*: modo como são estabelecidos os mapeamentos entre os esquemas dos pontos, assim como as estratégias adotadas;
- *Linguagem de consulta*: linguagem de consulta adotada pelo PDMS;
- *Processamento de consulta*: modo como são processadas as consultas, bem como as estratégias utilizadas;
- *Características particulares*: aponta uma ou mais características encontradas somente em um determinado PDMS.

Em relação ao modelo de rede, os PDMS estudados utilizam as topologias pura (estruturada e não-estruturada) e super-ponto. Não encontramos nenhum PDMS que utilizasse a topologia híbrida. Particularmente, o sistema APPA adapta-se a qualquer tipo de topologia. O modelo relacional e o formato RDF são utilizados como modelo de dados comum pela maioria dos PDMS. O mesmo ocorre com as linguagens de consulta XQuery e SQL. Em relação ao enriquecimento semântico de esquemas exportados são utilizados metadados, vocabulários controlados e ontologias. Os formalismos (LAV e GAV) utilizados pelos sistemas de integração de dados para definição de mapeamentos entre esquemas continuam sendo aplicados nos PDMS. Para otimizar o processamento de consultas diferentes técnicas são aplicadas: uso de agentes inteligentes, técnicas de aprendizagem, índices de roteamento, tabelas e expressões de mapeamento, entre outras.

Tabela 3.3 – Quadro Comparativo entre PDMS

Projeto Característica	Instituição	Representação de Dados	Topologia de Rede	Integração Semântica	Mapeamentos entre Esquemas	Linguagem de Consulta	Processamento de Consultas	Características Particulares
Piazza	Universidades de Washington e Pensilvânia	XML, RDF e DAML-OIL	Pura	Metadados; Estabelecidos aos pares; Explora a propriedade de transitividade	Heurísticas e algoritmos; Mapeamentos armazenados em um repositório central; Fomalismo GLAV	XQuery	Consulta reescrita com base em mapeamentos; Técnicas de aprendizagem de máquina e exploração de casos anteriores	-
PeerDB	Universidade de Singapura	Relacional	Pura	Metadados (palavras-chave)	Através da estratégia de <i>relation-matching</i> , usando palavras-chaves para entidades e atributos	SQL	Agentes Inteligentes; <i>Relation-Matching</i> (Função de Similaridade); Forte dependência e interatividade com o usuário	Um ponto pode se auto-reconfigurar e definir seus vizinhos; Gerenciamento de cache
XPeer	Universidade de Montpellier	XML	Super-Ponto (Ponto, Super-Ponto, Ponto de Domínio e Ponto Global)	Esquemas dos peers devem se reportar a um esquema do super peer (cluster)	Formalismos LAV e GAV	XQuery	Consultas são reescritas pelos cluster peers e os resultados são integrados pelos domain peers via cluster peers	Gerenciamento de metadados através de um modelo de repositório
Edutella	Universidade de Hannover	XML e RDF	Super-Ponto (<i>HyperCup</i>)	Metadados; Índices de Roteamento	Mapeamentos armazenados nos super-pontos; Mecanismos de transformações Locais e Regras	RDF-QEL	Índices de roteamento SP/P e SP/SP para determinar pontos relevantes; Geração de plano de consulta a partir de índices, metadados e mapeamentos	Framework JXTA; <i>Rule-based clustering</i>
APPA	Universidade de Nantes	Relacional	Qualquer (Pura Não-Estruturada, Pura Estruturada, Super-Ponto)	Esquema de um ponto é especificado através de visões sobre um Common Schema Description	Formalismo LAV	XQuery	Estratégia de consulta considera dados replicados nos pontos; Processamento de consulta acontece em 4 fases: reformulação, matching, otimização e decomposição/execução	Arquitetura orientada a serviços
Hyperion	Universidades de Toronto e Ottawa	Relacional	Pura	Metadados	Tabelas e Expressões de Mapeamento; Mapeamentos entre dados	SQL	Reescrita através das Tabelas e Expressões de Mapeamento	-
Rosa-P2P	Instituto Militar de Engenharia (IME-RJ)	XML e RDF	Super-Ponto	Metadados e Vocabulários Controlados	A integração ocorre entre as próprias instâncias	-	Reenvio e reescrita de consultas segundo vocabulários controlados; uso de índices de roteamento	-

3.5 Considerações Finais

Neste capítulo, mostramos como ocorreu a evolução dos sistemas gerenciadores de dados distribuídos, desde os SGBDD até os atuais PDMS. Enfoque maior foi atribuído aos PDMS, onde discutimos suas características e 2 (dois) de seus principais problemas: identificação de mapeamentos semânticos e processamento de consultas. Discutimos também alguns dos PDMS mais importantes encontrados na literatura e apresentamos um quadro comparativo entre os mesmos.

Os problemas enfrentados pelos PDMS podem ser minimizados caso pontos que compartilhem conteúdos semanticamente similares sejam posicionados próximos uns dos outros. Neste sentido, uma estratégia bastante comum utilizada pelos sistemas P2P é a formação dinâmica de comunidades semânticas. No próximo capítulo mostramos a importância destas comunidades para os PDMS e os benefícios que podem proporcionar para a resolução de problemas como a identificação de mapeamentos semânticos e processamento de consultas. Outros PDMS que adotam esta estratégia são discutidos, incluindo suas soluções para formação e manutenção de comunidades semânticas.

Capítulo 4

Conectividade Baseada em Semântica em PDMS

A utilização de uma estratégia para definição dos vizinhos de um ponto é um fator importante para facilitar a identificação de mapeamentos semânticos e o processamento de consultas. Este capítulo trata da questão da conectividade em um PDMS baseada na semântica do conteúdo exportado pelos pontos. Neste tipo de estratégia, os pontos podem participar de um ou mais agrupamentos denominados comunidades semânticas. Uma comunidade semântica é um exemplo de rede *overlay* onde os pontos participantes compartilham dados semanticamente similares. O conceito é puramente lógico, ou seja, pontos pertencentes a domínios de conhecimento comuns tornam-se vizinhos, mesmo localizando-se distante fisicamente.

A seguir, apresentamos um resumo das pesquisas mais recentes desenvolvidas sobre a formação dinâmica de comunidades semânticas em sistemas P2P tradicionais e PDMS. Inicialmente, discutimos sobre a importância das comunidades semânticas e apresentamos os principais termos associados ao assunto. Em seguida, expomos uma proposta para utilização de comunidades semânticas em sistemas P2P dedicados ao compartilhamento de arquivos, e sua extensão para PDMS. Ainda, apresentamos uma descrição de alguns dos principais PDMS que utilizam o conceito de comunidades semânticas. Finalizamos o capítulo com uma análise comparativa destes PDMS e as considerações finais.

4.1 Motivação

Apesar de menos dinâmica do que nos sistemas P2P tradicionais dedicados ao compartilhamento de arquivos, o processo de conectividade em PDMS é considerado de extrema importância, especialmente porque os pontos podem compartilhar dados de domínios de conhecimento distintos. Nesse sentido, a conexão arbitrária de pontos em um PDMS é considerada ineficiente [Löser et al. 2003a, Castano et al. 2003] pois pontos com pouca ou talvez nenhuma informação compartilhada em comum podem tornar-se vizinhos. Conseqüentemente, pode ocorrer um aumento na transmissão de mensagens entre os pontos, comprometendo a escalabilidade do sistema. Além disso, diversas outras tarefas importantes podem ser prejudicadas [Heese et al. 2005]:

- *Identificação de mapeamentos semânticos*: mapeamentos de baixa qualidade podem ser produzidos;
- *Processamento de consultas*: a excessiva quantidade de pontos vizinhos faz com que os planos de consulta passem a considerar um número maior de pontos (fontes de dados). Conseqüentemente, ocorre um aumento no volume de dados retornados ao ponto que submeteu a consulta, o algoritmo de roteamento de consultas passa a gerar mais alternativas e o tempo de resposta das consultas aumenta. Além disso, resultados de consulta incompletos e/ou inconsistentes podem ser apresentados.

A conexão arbitrária de pontos pode ocasionar ainda um outro tipo de problema: o isolamento de pontos semanticamente similares na rede. Por exemplo, considere o exemplo ilustrado na Figura 4.1, extraído de [Castano et al. 2003]. Suponha que o ponto C conecta-se à rede compartilhando dados referentes a “livros de detetives”. Os vizinhos de C não compartilham nenhuma informação similar e, portanto, não estão interessados no assunto. Por sua vez, o ponto Y compartilha dados referentes a “livros de mistério” e está interessado em adquirir conhecimento sobre o assunto compartilhado pelo ponto C. Em termos número de encaminhamentos, os pontos C e Y podem ser considerados distantes. Em um cenário P2P o ponto C seria fatalmente isolado, visto que seus vizinhos não enviariam consultas referentes a “livros de detetives”. Além disso, os pontos C e Y teriam pouca chance de compartilhar diretamente seus dados em virtude da distância e de restrições de TTL. Uma forma de minimizar os problemas mencionados anteriormente é através da formação dinâmica de **comunidades semânticas** de pontos [Castano et al. 2005, Brito 2005, Comito et al. 2006, Herschel et al. 2005, Heese et al. 2005, Li et al. 2005, Rouse et al. 2006], discutida nas próximas seções.

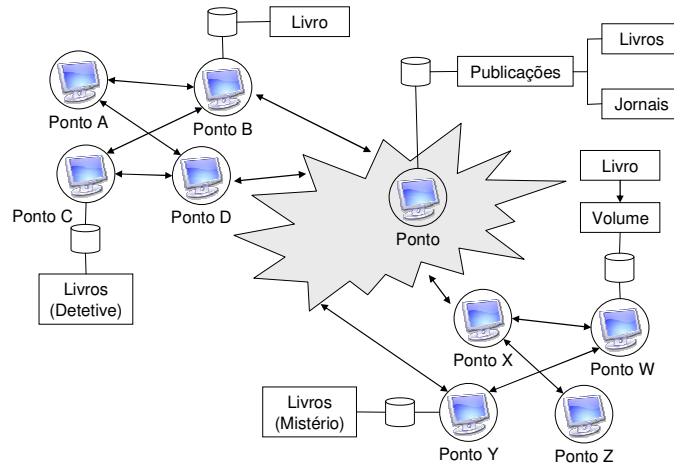


Figura 4.1 – Um cenário de rede P2P onde pontos semanticamente similares poderiam ficar isolados [Castano et al. 2003]

4.2 Comunidades Semânticas de Pontos

Comunidades semânticas podem ser definidas como “*um conjunto de pontos com interesse comum em algum assunto e que possuem alguma forma de organização*” [Castano et al. 2005]. As comunidades devem surgir de forma independente, a partir do momento em que um conjunto de pontos declara espontaneamente o interesse em compartilhar recursos em comum.

Cada comunidade deve tratar de um **interesse** (também chamado de tópico ou assunto) específico, por exemplo *educação*. Um interesse pode ser formalizado através de palavras-chave ou ontologias, e deve ser genérico o suficiente para incluir pontos significativos, entretanto é fundamental evitar tópicos muito genéricos que possam ocasionar ineficiência no processamento de consultas.

Políticas de clusterização [Löser et al. 2003a, Li et al. 2005] podem ser utilizadas para determinar as comunidades nas quais um ponto pode participar. Em alguns casos, tais políticas são projetadas através de regras do tipo evento-condição-ação. Quando o esquema exportado pelos pontos e o interesse de uma comunidade são representados através de ontologias, a técnica de *ontology matching* [Doan et al. 2002, Shvaiko et al. 2005, Zhdanova et al. 2006, Bruijn et al. 2004a] pode ser aplicada para determinar a participação dos pontos em comunidades [Castano et al. 2005].

Nas comunidades semânticas, os pontos podem organizar-se através das topologias de rede estudadas no Capítulo 2. Particularmente, em alguns sistemas uma

comunidade pode ser formada por um único cluster [Rouse et al. 2006, Nejdí et al. 2002] ou um conjunto de clusters [Brito 2005, Comito et al. 2006].

Para [Castano et al. 2005], os princípios básicos que possibilitam a formação de uma comunidade semântica são:

- *Descrição do ponto baseada em ontologias:* o esquema exportado por cada ponto deve ser representado por uma ontologia, ou seja, através de conceitos, propriedades e relacionamentos semânticos;
- *Interações baseadas em consultas:* para descoberta de pontos semanticamente similares, um ponto deve submeter “consultas de descoberta” aos demais pontos. Analogamente, um ponto deve responder às “consultas de descoberta” formuladas por outros pontos. O resultado retornado indica os pontos que podem participar de uma comunidade;
- *Capacidade de comparação semântica:* cada ponto deve possuir um componente responsável por determinar a correspondência semântica entre conceitos de ontologias e o grau em que ocorrem, por exemplo apenas entre nomes dos conceitos, ou entre nomes e propriedades.

Durante o processamento de consultas, cada consulta deve ser associada a um tópico e, em seguida, submetida apenas à(s) comunidade(s) que trata(m) do tópico em questão. Neste sentido, a noção de comunidade foi introduzida com o objetivo de proporcionar subsídios para cálculo do escopo de consultas. Antes de propagar uma consulta, o ponto realiza uma busca por comunidades de acordo com seus respectivos tópicos. O uso de comunidades semânticas minimiza a necessidade de um ponto ter que lidar individualmente com os demais pontos capazes de responder uma consulta.

Inúmeros sistemas P2P tradicionais e PDMS utilizam o conceito de comunidades semânticas para agrupamento dos pontos similares: H³ [Castano et al. 2005], Humboldt Discoverer [Herschel et al. 2005], Mapster [Rouse et al. 2006] e PARIS [Comito et al. 2006]. Nestes sistemas, o processo de conectividade dos pontos está diretamente relacionado com a formação, descoberta, manutenção e eliminação das comunidades. Um ponto ao conectar-se deve rapidamente encontrar uma ou mais comunidades adequadas. Neste sentido, o esquema de exportação disponibilizado pelos pontos é essencial. Caso não exista uma comunidade que trate do mesmo interesse do ponto, o ponto pode ser o “fundador” da comunidade. Os próximos pontos a se conectarem

podem ser agregados à comunidade. Ao desconectar-se, caso não exista nenhum outro ponto na comunidade, a mesma deve ser eliminada. A seguir, apresentamos uma das primeiras propostas de conectividade baseada na semântica do conteúdo exportado, para os sistemas P2P dedicados ao compartilhamento de arquivos.

4.3 Semantic Overlay Networks

Na topologia de rede pura não-estruturada, no momento que um ponto conecta-se ao sistema, a escolha dos seus vizinhos é feita normalmente de forma aleatória. Em virtude da diversidade de conteúdo compartilhado entre os pontos, é comum que pontos disponibilizando conteúdos semanticamente diferentes tornem-se vizinhos. Portanto, durante o processamento de consultas, mensagens desnecessárias podem ser enviadas para pontos irrelevantes, prejudicando o desempenho do sistema.

As Redes Overlay Semânticas (*Semantic Overlay Networks* - SON) representam uma tentativa de melhoria da topologia de rede pura não-estruturada, no que diz respeito à distribuição dos pontos [Crespo et al. 2002]. Nesta estratégia, durante a conexão dos pontos, o conteúdo é primeiramente classificado de acordo com a semântica associada. Como resultado desta classificação, os pontos com conteúdo semanticamente equivalentes são agrupados em clusters (cada cluster forma uma SON). Durante o processamento de consultas, dependendo do tipo de consulta, a troca de mensagens ocorre apenas entre os pontos de um mesmo cluster aumentando a probabilidade de que resultados mais significativos sejam retornados.

Para melhor ilustrar o funcionamento de uma SON, considere a Figura 4.2, adaptada de [Crespo et al. 2002], na qual os pontos compartilham arquivos de músicas. O sistema é formado por 8 (oito) pontos identificados pelas letras de A a H. As linhas entre dois pontos indicam que um ponto pode comunicar-se com o outro e vice-versa. Por sua vez, os retângulos representam os clusters. Dois pontos distintos só podem comunicar-se se pertencerem a um mesmo cluster. Por exemplo, os pontos A, B e C possuem músicas sobre o gênero *rock*. Similarmente, os pontos C, E e F compartilham músicas sobre *frevo*. Observe que um ponto de um cluster não necessariamente precisa conectar-se a todos os outros do mesmo cluster, por exemplo C e F. Além disso, pode haver sobreposição de pontos nos clusters, isto é, um ponto pode participar de mais de um cluster (por exemplo, o ponto C pertence aos clusters *frevo* e *rock*).

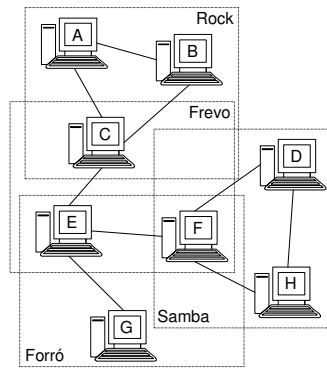


Figura 4.2 – Semantic Overlay Networks, adaptada de [Crespo et al. 2002]

A Figura 4.3 apresenta 2 (dois) exemplos de classificações para o domínio musical. Na Figura 4.3a, os arquivos (músicas) são classificados de acordo com o estilo (por exemplo, *rock* e *forró*) e pelo sub-estilo (por exemplo, *soft* e *dance*). Na Figura 4.3b, os arquivos (músicas) são classificados de acordo com a década em que foram lançadas.

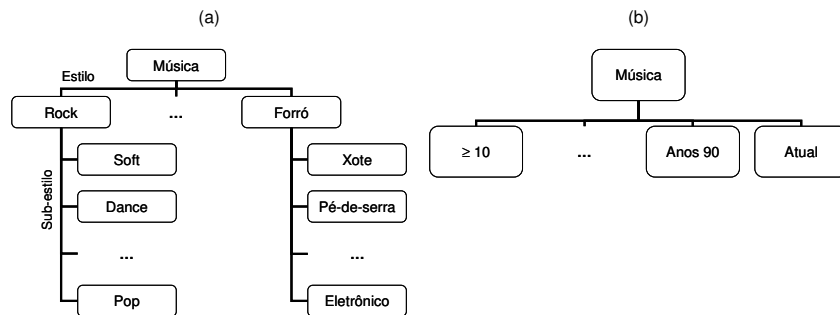


Figura 4.3 – Exemplos de Classificações

Os documentos e consultas são classificados em um ou mais conceitos-folha da classificação. Para determinar o conceito-folha são utilizadas **funções de classificação**. O uso de funções de classificação imprecisas pode fazer com que conceitos não-folha sejam retornados. Considerando a Figura 4.3a, um documento do sub-estilo *pop* pode ser classificado como estilo *rock*, caso a função não consiga determinar precisamente seu sub-estilo. Funções mal projetadas podem ainda retornar conceitos-folhas errados.

Como os documentos estão armazenados em pontos, na verdade são os pontos que são classificados. Uma SON é formada por um conjunto de pontos semanticamente similares. Cada SON é associada a um conceito da classificação. Formalmente, uma SON associada ao conceito *c* é chamada de *SON de c* ou SON_c . Por exemplo, na Figura 4.3a, 9 (nove) SON são visualizadas: 6 (seis) associadas a conceitos-folha (*soft*, *dance*, *pop*, *xote*, *pé-de-serra* e *eletrônico*), uma associada com *rock*, outra com *forró* e finalmente uma com *música*.

A escolha da(s) SON a(s) qual(is) um ponto deve se conectar ocorre mediante o resultado da classificação de seus documentos. Basicamente, duas estratégias podem ser utilizadas para conexão dos pontos a uma SON:

- *Bastante conservadora*: um ponto é inserido em SON_c caso possua pelo menos 1 (um) documento classificado em c ;
- *Menos conservadora*: um ponto é inserido em uma SON_c caso possua uma quantidade significativa de documentos classificados em c . Neste caso, a quantidade de pontos em uma SON é reduzida e poucos pontos precisam ser acessados durante uma consulta. Além disso, como o número de SON a qual um ponto pertence é reduzido, a quantidade de mensagens nas SON também é minimizada. A desvantagem desta estratégia é a possibilidade de que nem todos os documentos associados a uma consulta sejam encontrados.

As tarefas de definição de uma SON, conexão ao sistema e utilização das SON são ilustradas na Figura 4.4 e seus passos listados na Tabela 4.1.

Tabela 4.1 – Funcionamento de uma SON

Construção da SON	Conexão de um Ponto	Processamento de Consulta
<ol style="list-style-type: none"> 1. Diversas classificações candidatas são geradas a partir da análise dos documentos distribuídos nos pontos (ou a partir de uma amostra) e em seguida avaliadas; 2. Uma classificação é escolhida e armazenada em todos (ou apenas em alguns) os pontos do sistema; 3. As SON são definidas. 	<ol style="list-style-type: none"> 1. Para conectar-se ao sistema, o ponto “inunda” a rede com mensagens solicitando a classificação; 2. O ponto executa funções de classificação baseadas na classificação recebida em todos os seus documentos; 3. Um ponto classificador direciona o ponto as SON adequadas; 4. O ponto “conhece” os outros pontos de cada SON enviando mensagens ou através de um diretório central. 	<ol style="list-style-type: none"> 1. Um ponto, ao submeter uma consulta, primeiramente classifica-a determinando as SON relevantes; 2. O ponto envia a consulta para um ponto específico das SON selecionadas; 3. Nas SON o ponto específico propaga a consulta através de “inundação” ou índices (super-pontos).

Inúmeras questões precisam ser tratadas quando se deseja utilizar as SON. Estas questões são discutidas em [Crespo et al. 2002] e listadas a seguir:

- *Classificação*: não apenas o conteúdo compartilhado pelos pontos precisa ser classificado, mas também as consultas geradas pelos usuários;
- *Granularidade*: o nível de granularidade para a classificação deve ser definido apropriadamente. Um nível muito baixo pode aumentar o custo de manutenção do

sistema, enquanto que o contrário pode levar à descoberta de pontos menos relevantes;

- *Conteúdo*: critérios precisam ser definidos para que um ponto participe de um cluster. Se um ponto possui poucas músicas de *rock*, será que o mesmo deve participar de um cluster contendo pontos que compartilham milhares de músicas de *rock*?
- *Seleção de clusters*: qual a melhor estratégia para identificar clusters relevantes durante o processamento de consultas?

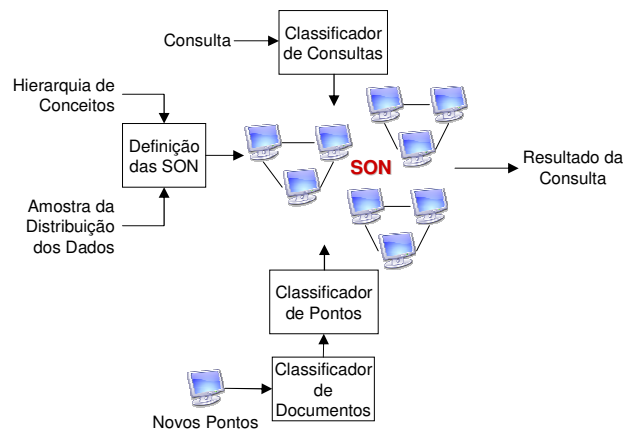


Figura 4.4 – Processo de Construção das SON [Crespo et al. 2002]

O trabalho de [Li et al. 2005] utiliza o conceito de SON para propor uma topologia de rede mista para sistemas P2P de compartilhamento de arquivos (ver Figura 4.5). Cada ponto de uma rede DHT representa uma comunidade semântica distinta e atua como ponto de entrada. Uma comunidade é composta por múltiplos pontos que compartilham arquivos semanticamente similares e é organizada segundo a topologia pura não-estruturada. Políticas de clusterização baseadas em ontologias são aplicadas para determinar se um ponto pode participar de uma comunidade.

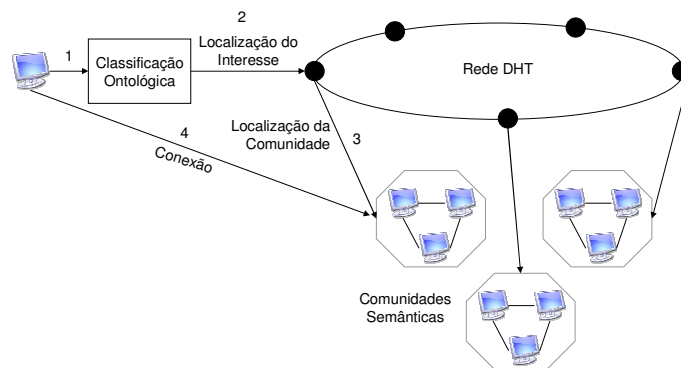


Figura 4.5 – Arquitetura do Sistema [Li et al. 2005]

A rede DHT tem como objetivo tornar mais rápido o processo de descoberta de recursos e auxiliar no processamento de consultas. Os pontos localizados na rede DHT são estáveis e apresentam boa largura de banda em relação aos outros pontos da comunidade. Um ponto passa a participar da rede DHT somente nos seguintes casos: (i) atenda os requisitos de capacidade física; (ii) permanece estável por um determinado intervalo de tempo; (iii) o processamento na comunidade está intenso. O protocolo Chord é utilizado para estruturar a rede DHT.

Dentro de cada comunidade, as consultas são propagadas através do modelo de inundação. Uma consulta que está relacionada com pontos de outras comunidades é resolvida através dos seguintes passos:

1. Definir a categoria da consulta. Esta tarefa é feita pelo ponto de entrada (localizado na rede DHT) da comunidade na qual a consulta foi originada;
2. Identificar as demais comunidades na rede DHT relacionadas com a categoria da consulta;
3. Enviar a consulta para o ponto de entrada das comunidades identificadas;
4. Propagar a consulta nas comunidades;
5. Integrar os resultados no ponto de entrada de cada comunidade;
6. Devolver os resultados para o ponto de entrada da comunidade original que, por sua vez, integra-os novamente e devolve ao ponto inicial.

Em [Vazirgiannis et al. 2006], são discutidos alguns dos principais requisitos que um sistema P2P precisa atender para utilizar o conceito de SON. Um deles diz respeito à descentralização de metadados sobre os pontos, isto é, não deve existir um ponto com conhecimento global sobre o sistema. Assim, para formação de uma SON os pontos precisam realizar diversas buscas na rede no sentido de descobrir outros pontos que compartilham conteúdo semanticamente similar. Em geral, esta não é considerada um tarefa simples, especialmente quando se utiliza a topologia pura não-estruturada.

4.4 Semantic Overlay Clusters

Semantic Overlay Clusters (SOC) representam uma evolução das SON [Löser et al. 2003a]. Como vimos anteriormente, as SON são redes *overlay* utilizadas por sistemas P2P de compartilhamento de arquivos. Durante a conexão dos pontos, o conteúdo compartilhado é primeiramente classificado de acordo com uma taxonomia. Os pontos

com conteúdo semanticamente equivalente formam uma SON. Em cada SON a organização dos pontos é estabelecida de acordo com a topologia pura não-estruturada.

De acordo com a Tabela 4.2, as SOC são redes *overlay* formadas por pontos que desejam compartilhar dados estruturados. Normalmente, os pontos são organizados de acordo com a topologia super-ponto. O agrupamento de pontos em clusters é feito com base nos esquemas exportados pelos pontos. Um modelo de dados comum é utilizado para descrever as características dos pontos (por exemplo, esquema exportado e aspectos de qualidade). Cada cluster está associado a uma política de clusterização, que define explicitamente os pontos que podem participar do cluster. Quando um ponto deseja conectar-se ao sistema, ele envia sua descrição a um super-ponto arbitrário.

Tabela 4.2 – Comparativo entre Semantic Overlay Networks e Semantic Overlay Clusters

Rede Overlay	Conteúdo Compartilhado	Topologia	Mecanismo de Classificação
SON	Não-Estruturado (por exemplo, arquivos)	Pura Não-Estruturada	Taxonomias
SOC	Estruturado (por exemplo, banco de dados)	Super-ponto	Ontologias

Conforme ilustrado na Figura 4.6, a descrição do ponto e a política de clusterização do cluster são utilizadas por uma função de comparação que indica se o ponto pode ou não participar do cluster. A função é armazenada nos super-pontos e pode ser auxiliada por ontologias. Cada super-ponto pode aplicar uma função de comparação diferente. A descrição do ponto é propagada para todos os outros super-pontos que, por sua vez, aplicam sua função de comparação própria. Um exemplo de sistema baseado em SOC é o Edutella, discutido no capítulo anterior.

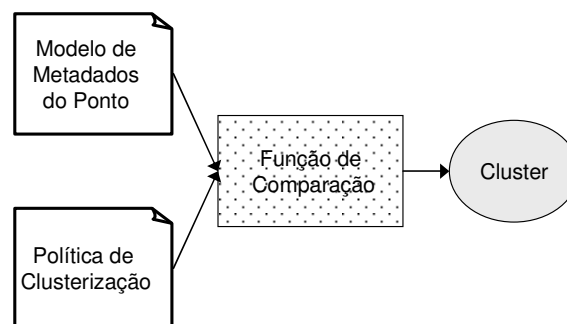


Figura 4.6 – Abordagem de Clusterização Baseada em um Modelo de Metadados [Löser et al. 2003a]

4.5 PDMS com Conectividade Baseada em Semântica

Nesta seção, apresentamos um resumo das pesquisas mais recentes desenvolvidas sobre a utilização de comunidades semânticas em PDMS. Alguns dos principais PDMS que tratam desta questão são discutidos e suas estratégias apresentadas. Os PDMS discutidos incluem H³ [Helios 2007, Castano et al. 2005], PARIS [Comito et al. 2006], Humboldt Discoverer [Heese et al. 2005, Herschel et al. 2005] e Mapster [Rouse et al. 2006]. No final desta seção, é feita uma análise comparativa destes PDMS.

4.5.1 Framework H³

H³ é um framework para compartilhamento de dados em sistemas P2P [Helios 2007]. O framework propõe a construção de uma rede *overlay* na qual os dados (por exemplo, relacionais e documentos XML) dos pontos são representados através de ontologias. Um ponto pode armazenar informações sobre as ontologias dos pontos vizinhos. Durante o processamento de consultas, os mapeamentos entre as ontologias dos pontos são explorados. Para evitar que pontos semanticamente similares fiquem isolados na rede os pontos são associados a comunidades semânticas adequadas.

Conforme ilustrado na Figura 4.7, a arquitetura de um ponto no Framework H³ é composta basicamente de duas camadas:

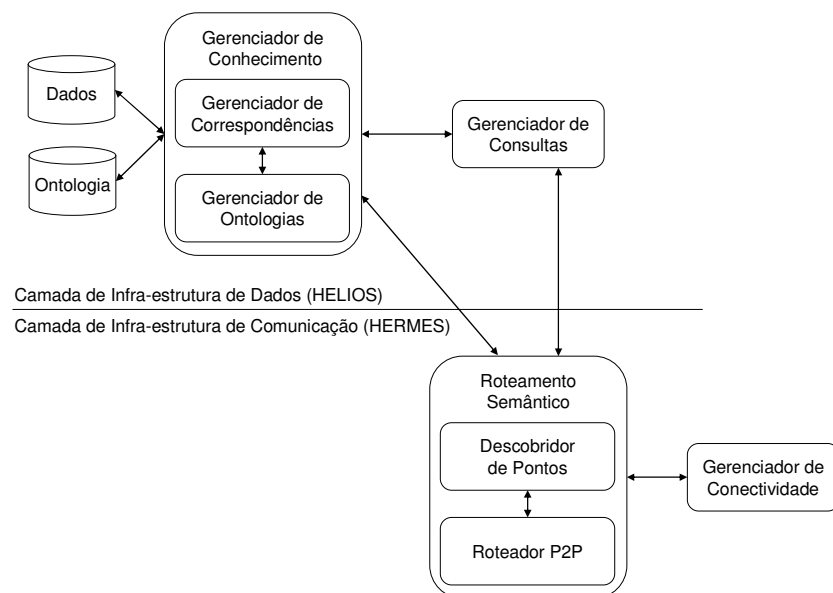


Figura 4.7 – Arquitetura de um Ponto H³ [Castano et al. 2003]

- *Camada de dados (HELIOS)*: fornece a infra-estrutura de software necessária para o compartilhamento de dados. Ontologias são utilizadas para representar o conhecimento de cada ponto (incluindo os dados compartilhados pelo ponto e o conhecimento que o mesmo possui sobre outros pontos) e para interação entre os pontos, proporcionando recuperação de conteúdo e aquisição/extensão de conhecimento de acordo com modelos de consulta pré-definidos e técnicas de *ontology matching* [Doan et al. 2002, Zhdanova et al. 2006]. A camada de dados é subdividida em 3 (três) componentes:
 - *Gerenciador de Consultas*: recebe uma consulta, executa seu processamento e integra os resultados;
 - *Gerenciador de Correspondências*: realiza a comparação entre os conceitos de uma consulta enviada a um ponto e os conceitos de sua ontologia;
 - *Gerenciador de Ontologias*: responsável por representar os dados compartilhados por um ponto através de uma ontologia e pela adição de novos conceitos na ontologia.

- *Camada de comunicação (HERMES)*: oferece a infra-estrutura de software necessária para comunicação entre os pontos e roteamento de consultas baseado em semântica. É subdividida nos seguintes componentes:
 - *Gerenciador de Conectividade*: serviço utilizado por um ponto para descoberta de outros pontos na mesma comunidade. É também usado para notificar os pontos vizinhos sobre a desconexão de um ponto;
 - *Descobridor de Pontos*: utilizado durante o roteamento de consultas. Explora os serviços do Gerenciador de Conhecimento para descobrir se existem pontos com conceitos iguais ou similares aos conceitos incluídos em uma consulta;
 - *Roteador P2P*: caracteriza o conjunto de vizinhos de um ponto em termos de distância física ou similaridade semântica, e estabelece correspondências entre os vizinhos e o ponto em questão. Envia as consultas para os pontos selecionados pelo Descobridor de Pontos. Caso nenhum ponto seja retornado é feito um *broadcast* da consulta.

Em [Castano et al. 2005] os autores apresentam uma estratégia para formação dinâmica das comunidades semânticas de pontos no Framework H³. Cada comunidade é identificada através um identificador de comunidade (*Community Identifier – CID*). Um

assunto é representado através de uma ontologia e identificado por um cartão de identificação (*Identity Card* – ICard). Um ponto pode participar de múltiplas comunidades e armazena o CID e o ICard de cada comunidade que participa. Não é necessária a participação de pontos diferenciados (por exemplo, um super-ponto) durante o processo de descoberta e formação das comunidades.

A técnica de *ontology-matching* é utilizada para definir a participação dos pontos nas comunidades. Ao receber um ICard, o ponto executa uma correspondência entre a ontologia recebida e sua ontologia (*ontology matching*). As ontologias são representadas através de um modelo próprio (H-Model [Castano et al. 2004]), capaz de suportar múltiplos formalismos de especificação de ontologias (por exemplo, OWL, RDF(S) e UML). O algoritmo H-Match [Castano et al. 2004] é utilizado para realização do processo de *ontology matching*. Um *threshold* é definido para selecionar apenas as correspondências ontológicas que apresentam maior nível de similaridade semântica. O algoritmo considera as características linguísticas e conceituais dos conceitos e pode ser executado em 4 (quatro) graus diferentes. Cada grau considera um elemento da ontologia ou uma combinação deles, conforme mostrado na Tabela 4.3.

Tabela 4.3 – Graus suportados pelo algoritmo H-Match

Grau	Elemento(s)
<i>Surface</i>	Nome dos conceitos
<i>Shallow</i>	Nome e propriedades dos conceitos
<i>Deep</i>	Nome e relacionamentos entre os conceitos
<i>Intensive</i>	Nome, relacionamentos entre os conceitos e valor das propriedades

A formação de uma comunidade semântica ocorre quando um ponto (denominado ponto fundador), executa um processo preliminar de sinalização semântica, composto de 4 (quatro) etapas:

1. *Publicação do ICard*: o ponto fundador (P_f) define uma CID e um ICard descrevendo o interesse da comunidade, além de uma série de restrições à qual os pontos precisam satisfazer para participar da comunidade como, por exemplo: número mínimo de participantes e *threshold* para correspondências ontológicas. Em seguida, P_f define uma “mensagem de convite” incluindo a CID, a ICard, um valor de TTL, o grau para correspondência ontológica e o *threshold* a ser aplicado no processo de *ontology matching*. A mensagem é enviada a todos os vizinhos de P_f ;

2. *Identificação de membros*: o processo de *ontology matching* entre o ICard e a ontologia do ponto é iniciado em cada ponto que recebe a mensagem. Os pontos (P_i), que apresentam correspondências ontológicas acima do *threshold*, devolvem a P_f através de uma “mensagem de interesse” os trechos relevantes de sua ontologia. Independente do resultado produzido durante o processo de *ontology matching*, os pontos P_i propagam a “mensagem de convite” para seus vizinhos, obedecendo o valor do TTL;
3. *Aprovação da requisição*: ao receber as “mensagens de interesse”, P_f avalia quais pontos interessados podem participar da comunidade. A avaliação é feita através de novo processo de *ontology matching*, desta vez entre os trechos de ontologia recebidos e a ontologia de P_f . Os pontos aceitos recebem uma “mensagem de aprovação”;
4. *Validação da comunidade*: P_f verifica se as restrições definidas na etapa de Publicação do ICard estão satisfeitas. Em caso positivo, uma “mensagem de validação” é enviada aos pontos aceitos e a comunidade é formada. Caso contrário, P_f aborta a formação da comunidade. Os pontos aceitos aguardam a “mensagem de validação” até um intervalo pré-definido de tempo. Quando a mensagem não é recebida, a formação da comunidade é cancelada.

Um exemplo de formação de uma comunidade semântica é ilustrado na Figura 4.8. A ontologia de P_f pertence ao domínio de publicações acadêmicas. Assuma que P_f deseja formar uma comunidade semântica com um ICard contendo o conceito *publicação* e as propriedades *autor* e *título*. P_f define uma “mensagem de convite”, incluindo os seguintes itens: ICard, TTL = 1, grau de *ontology matching* = *deep* e *threshold* = 0.5. A mensagem é enviada para os vizinhos de P_f (pontos A, B e C). Como o valor de TTL é 1, a mensagem é repassada para os pontos D e G.

Os pontos que receberam a mensagem iniciam o processo de *ontology matching* entre a ontologia recebida (ICard) e sua ontologia local. Os pontos A e C pertencem ao domínio de turismo e portanto não possuem nenhum conceito em comum com a ontologia do ICard. Assim, estes pontos não precisam responder à mensagem recebida e são descartados. O resultado do processo de *ontology matching* nos pontos B, D e G é apresentado na Tabela 4.4. Como estão acima do valor do *threshold* (0.5), os pontos B, D e G possuem conceitos relevantes com a ontologia do ICard. Estes pontos enviam uma “mensagem de interesse” para P_f e o processo de formação continua com as etapas seguintes.

Tabela 4.4 – Resultado do Processo de *Ontology Matching*

Ponto	Resultado
B	AS (Publicação, Conferência_Anais) = 0.56 AS (Publicação, Workshop_Anais) = 0.56
D	AS(Publicação, Artigo) = 0.86
G	AS(Publicação, Publicação) = 0.85

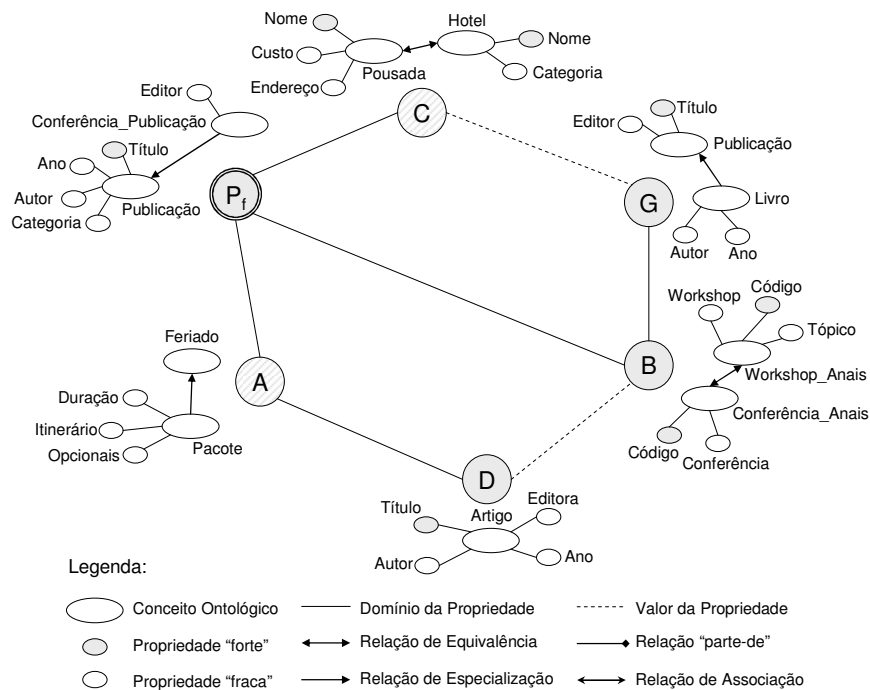


Figura 4.8 – Exemplo da formação de uma comunidade semântica [Castano et al. 2005]

4.5.2 PARIS

PARIS (*Peer-to-Peer Architecture for Data Integration Systems*) é um PDMS que utiliza uma rede *overlay* baseada em semântica para compartilhamento de dados entre fontes heterogêneas, autônomas e dinâmicas [Comito et al. 2006]. O sistema utiliza uma topologia de rede mista (pura não-estruturada e pura estruturada). Cada ponto representa uma fonte de dados e armazena os mapeamentos entre seu esquema exportado e os esquemas exportados pelos pontos vizinhos.

Em relação às comunidades semânticas, 4 (quatro) tipos podem ser formadas: (i) *local*: formada por pontos que compartilham um mesmo esquema exportado (por

exemplo, todos os pontos da comunidade A); (ii) *diretamente semânticas*: formada por comunidades locais que possuem mapeamentos semânticos diretos (por exemplo, comunidades A e C); (iii) *transitivamente semânticas*: formada por comunidades locais que possuem mapeamentos transitivos (por exemplo, comunidades A e B); e (iv) *externas*: composta pelos pontos restantes (por exemplo, comunidades E e F).

A rede *overlay* do sistema PARIS é ilustrada na Figura 4.9. As comunidades locais são organizadas em redes não-estruturadas. Em cada comunidade local, os pontos com melhores características físicas (simbolizados na cor preta) também participam da rede estruturada, implementada segundo o protocolo Chord. Os pontos (na cor branca) que necessitam enviar mensagens para pontos de outras comunidades, precisam comunicar-se primeiramente com o ponto da rede DHT de sua comunidade.

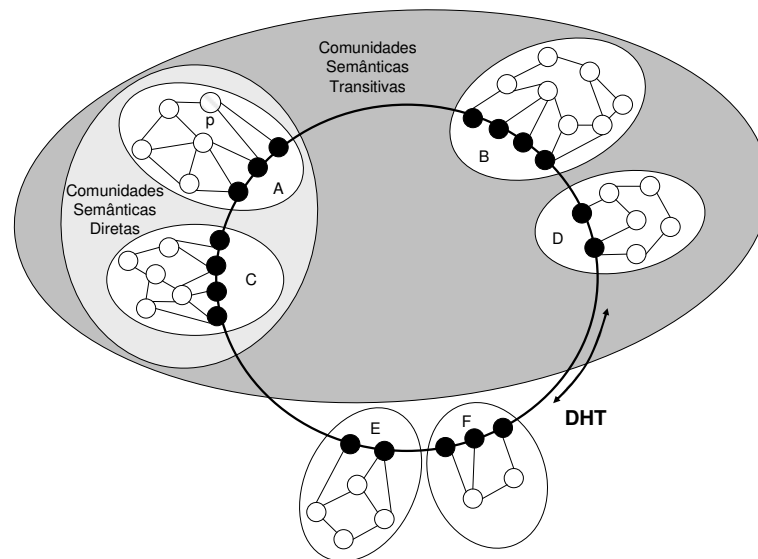


Figura 4.9 – Rede Overlay Semântica do PDMS PARIS

Como o esquema de cada comunidade local é idêntico, um ponto ao tentar conectar-se ao sistema necessita comparar seu esquema exportado com o esquema de cada comunidade local. Para facilitar esta tarefa cada esquema é mapeado em um número identificador único e o protocolo de identificação de pontos na rede Chord é aplicado. Caso a comunidade seja encontrada, o ponto recebe o endereço de um dos pontos da rede DHT da referida comunidade. Este ponto decide se o ponto solicitante deve ou não fazer parte da rede DHT. Caso não exista uma comunidade com esquema idêntico ao do ponto solicitante, este forma uma nova comunidade e conecta-se à rede DHT.

4.5.3 Humboldt Discoverer

Humboldt Discoverer [Herschel et al. 2005, Heese et al. 2005] é um PDMS estendido cujo objetivo é facilitar a descoberta de pontos e o processamento de consultas. O sistema utiliza uma topologia mista (pura não-estruturada e pura estruturada), na qual os esquemas exportados pelos pontos são indexados através de tecnologias ligadas à Web Semântica [Berners-Lee et al. 2001]. Conforme ilustrado na Figura 4.10, a extensão do PDMS ocorre através da adição das seguintes dimensões:

- *Dimensão semântica:* oferece um caminho alternativo para os mapeamentos semânticos entre esquemas. Consiste de um conjunto de ontologias e mapeamentos semânticos entre as mesmas. Além dos mapeamentos entre esquemas exportados, um ponto pode possuir mapeamentos entre seu esquema e as ontologias;
- *Dimensão web:* quando um ponto submete uma consulta, um índice semântico é utilizado para localização de pontos relevantes para a consulta. Em geral, os pontos relevantes identificados não possuem inicialmente mapeamentos semânticos com o ponto que submeteu a consulta. Cada ponto mantém uma base de conceitos que indexa o esquema de seus vizinhos;
- *Dimensão de qualidade:* influencia o processamento de consultas em todas as dimensões. Na camada do PDMS e na dimensão semântica, a qualidade do resultado da consulta depende do mapeamento entre os pontos. À medida que uma consulta é repassada entre os pontos, a qualidade dos mapeamentos é registrada. Na dimensão web uma medida é definida para localizar e classificar os pontos relevantes para uma consulta.

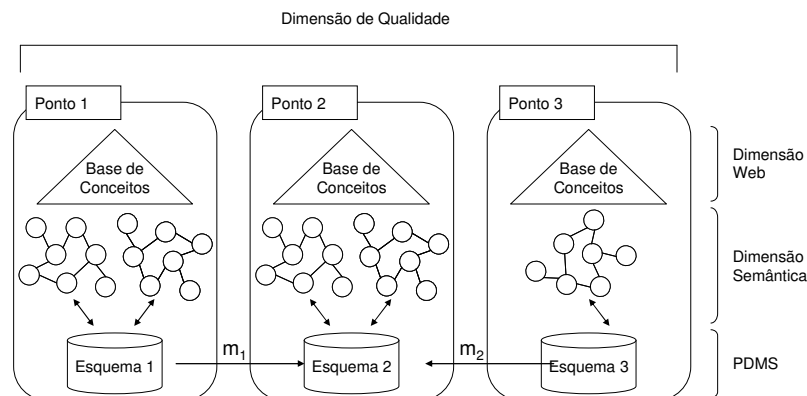


Figura 4.10 – Arquitetura estendida do PDMS Humboldt Discoverer [Herschel et al. 2005]

O processamento de consultas é dividido em duas fases distintas. Na primeira fase, a consulta é processada na camada PDMS utilizando-se os mapeamentos (relacionais). Caso não sejam suficientes, a próxima fase é iniciada e a consulta é processada utilizando-se mapeamentos ontológicos. A consulta é traduzida para um grafo de consulta e a Base de Conceitos é utilizada para descoberta de pontos que possam oferecer respostas para a consulta.

O PDMS oferece um índice semântico distribuído para localização de pontos (ver Figura 4.11). Através deste índice, um ponto pode localizar outros pontos que inicialmente seriam inalcançáveis através de simples mapeamentos entre esquemas. O índice é composto de 3 (três) camadas de pontos com níveis de granularidade incrementais:

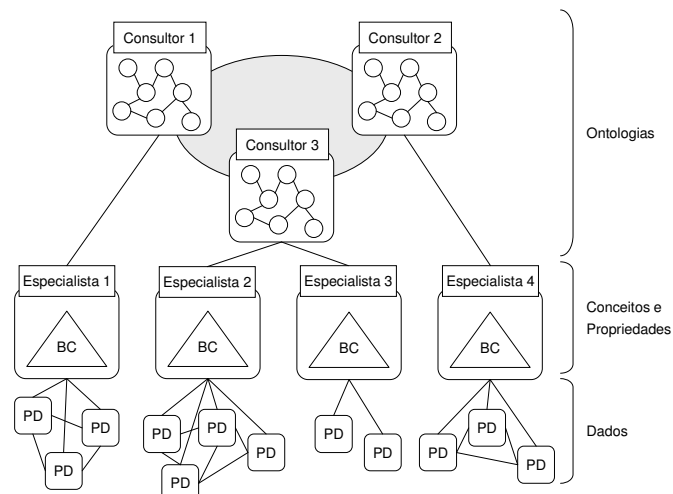


Figura 4.11 – Índice Semântico em 3 Camadas [Herschel et al. 2005]

- *Pontos consultores*: são organizados segundo a topologia DHT e armazenam uma lista de endereços de pontos especialistas (ver a seguir) que abrangem uma determinada ontologia O , onde a função *hash* da rede DHT aplicada a URI da ontologia resulta no endereço de um ponto consultor. Formalmente, $b(URI(O)) = endereço(ponto\ consultor)$. Como os pontos consultores apresentam boa disponibilidade, o custo de manutenção da rede DHT é relativamente baixo. Além disso, as buscas são rápidas e confiáveis;
- *Pontos especialistas*: são responsáveis por manter uma ou mais ontologias. Mantêm uma lista dos pontos relacionados com as ontologias de sua responsabilidade e oferecem funcionalidades de processamento de consultas sobre as ontologias. Atuam como super-pontos em relação aos pontos de dados (ver a seguir), armazenando metadados (por exemplo, conceitos compartilhados e características físicas dos pontos de

dados). Como um ponto especialista pode suportar mais de uma ontologia, ele pode comunicar-se com vários pontos consultores;

- *Pontos de dados*: representam as fontes de dados que desejam compartilhar conteúdo. Quando um ponto de dados deseja conectar-se ao sistema, primeiramente ele comunica-se com um ponto consultor para descobrir os pontos especialistas aos quais ele pode conectar-se. Em seguida, o ponto de dados passa suas informações para o ponto especialista e recebe o endereço de outros pontos de dados conhecidos pelo ponto especialista.

Inicialmente, um ponto sempre conecta-se à rede como ponto de dados. Caso o ponto permaneça conectado por um período de tempo previamente estipulado e ofereça recursos computacionais satisfatórios, pode-se tornar candidato a ponto especialista. Analogamente, os pontos especialistas podem tornar-se pontos consultores, caso apresentem um grau de confiança adequado e possuam largura de banda suficiente.

Como os pontos conectam-se dinâmica e automaticamente a um PDMS, vários fatores podem comprometer a qualidade dos mapeamentos: má qualidade dos dados compartilhados pelas fontes, estabelecimento de mapeamentos incorretos ou incompletos. Diferentemente dos PDMS tradicionais, neste PDMS um ponto de dados não precisa manter necessariamente mapeamentos com outros pontos de dados para que possa ser considerado durante o processamento de consultas. Os mapeamentos são estabelecidos entre pontos situados em níveis de hierarquia diferentes (por exemplo, pontos de dados e especialistas, e pontos especialistas e consultores).

4.5.4 Mapster

A maioria das técnicas de *schema matching* existentes foi desenvolvida para sistemas em ambientes pequenos e estáticos. O sistema Mapster [Rouse 2006, Rouse et al. 2006] tem como objetivo principal mostrar como as técnicas de *schema matching* podem ser incorporadas em um ambiente dinâmico como um PDMS, baseado na topologia super-ponto.

Os pontos representam bancos de dados relacionais e são distribuídos nos clusters de acordo com o respectivo domínio de interesse. Cada domínio é gerenciado por um super-ponto que disponibiliza um esquema de mediação, construído a partir dos esquemas exportados por seus respectivos pontos. A participação de um novo ponto em um cluster é definida pelo super-ponto, que mapeia o esquema exportado do ponto no

esquema de mediação através de técnicas de *schema matching*. O algoritmo de construção e manutenção do esquema de mediação é apresentado em [Rouse 2006]. A Figura 4.12 ilustra um exemplo de uma rede Mapster. Os círculos maiores representam domínios (clusters). Cada domínio contém vários pontos e um único super-ponto. As setas mais grossas representam a comunicação entre os super-pontos dos domínios.

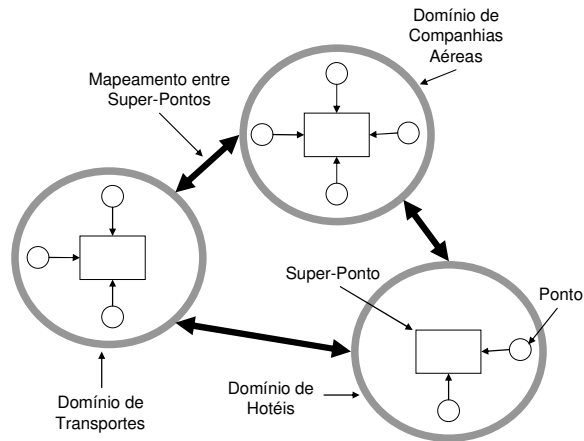


Figura 4.12 – Exemplo de uma Rede no PDMS Mapster [Rouse et al. 2006]

O sistema Mapster propõe uma estratégia de tolerância a falhas e balanceamento de carga na qual um domínio pode possuir múltiplos super-pontos. O esquema de mediação é replicado em cada super-ponto do domínio. No caso de falha em um super-ponto, os demais pontos do domínio não são desconectados. Além disso, o processamento de consultas pode ser distribuído entre os múltiplos super-pontos. A escolha dos super-pontos é feita com base nos critérios discutidos em [Rouse 2006]. Um dos super-pontos é escolhido como super-ponto principal, sendo responsável por manter o esquema de mediação e distribuir as réplicas para os outros super-pontos. O super-ponto principal é o único ponto visível fora do domínio.

Para conectar-se ao sistema, um ponto deve inicialmente procurar por um domínio apropriado. Informações sobre domínios existentes são publicadas periodicamente por cada super-ponto principal. Se um domínio não é encontrado, o ponto deve criar um novo domínio e assumir o papel de super-ponto principal. Nesse caso, o esquema de mediação é o próprio esquema do ponto e o super-ponto principal deve publicar informações sobre seu domínio para futuros pontos que desejem conectar-se ao sistema. A conexão de um ponto a um domínio pré-existente resulta no estabelecimento de mapeamentos semânticos entre o ponto e o super-ponto principal.

4.5.5 Análise Comparativa dos PDMS

Em relação às comunidades semânticas, o processo de formação e manutenção é descrito apenas para o framework H³. Os demais sistemas apenas informam que utilizam este tipo de estratégia, porém não entram em maiores detalhes. Ainda, dos sistemas anteriores, apenas o PARIS estende o conceito de comunidade, propondo a idéia de “comunidades de comunidades”.

Nos sistemas H³ e PARIS, a topologia de rede utilizada é a pura não-estruturada, portanto a busca por pontos semanticamente similares ocorre através da técnica de inundação, considerada ineficiente por muitos autores quando se trata de sistemas P2P em larga escala. Em relação ao PDMS Humboldt Discoverer, o processamento de consultas poderia ser melhorado, caso os metadados dos pontos consultores fossem replicados nos pontos especialistas, visto que as consultas não necessitariam ser repassadas até os pontos consultores (rede DHT). No sistema Mapster, os mapeamentos são estabelecidos diretamente entre os esquemas exportados pelos pontos, sem o auxílio de técnicas de Web Semântica, como ontologias. No que diz respeito ao sistema PARIS, os pontos de uma mesma comunidade local precisam possuir exatamente o mesmo esquema exportado, um requisito pouco comum quando se trata de sistemas dinâmicos e escalonáveis como é o caso de um PDMS.

4.6 Considerações Finais

Neste capítulo, enfatizamos a importância da utilização de comunidades para agrupamento de pontos em sistemas P2P de acordo com a semântica associada ao conteúdo compartilhado pelos pontos. A noção de comunidade semântica foi proposta inicialmente para os sistemas P2P tradicionais dedicados ao compartilhamento de arquivos e, posteriormente, estendida para os PDMS. Atualmente, inúmeros PDMS têm utilizado esta estratégia para facilitar a descoberta de recursos na rede, melhorar a geração de mapeamentos semânticos e otimizar o processamento de consultas.

No capítulo seguinte, apresentamos nossa proposta de tese, a qual inclui uma nova arquitetura de PDMS, baseada numa topologia mista (pura estrutura e super-ponto), além de uma estratégia para formação e manutenção de comunidades semânticas no sistema proposto.

Capítulo 5

Proposta de Tese

Conforme discutido nos capítulos anteriores, nos últimos anos diversos PDMS têm sido propostos na literatura. Na tentativa de tratar os principais problemas introduzidos pelas características do ambiente P2P, a maioria destes sistemas abstraem a infra-estrutura de rede e tratam o PDMS como um simples grafo formado por pontos interconectados.

Em nosso trabalho, propomos uma nova arquitetura que facilite a execução de tarefas primordiais em um PDMS como, por exemplo, a descoberta de recursos, a identificação de mapeamentos semânticos e o processamento de consultas. Neste sentido, faz-se necessária uma estratégia de conectividade baseada em semântica. A semântica associada ao conteúdo compartilhado pelos pontos é utilizada para determinar a participação dos pontos em comunidades semânticas e clusters semânticos.

Nos sistemas P2P, a conectividade dos pontos está diretamente relacionada com outras propriedades: balanceamento de carga e tolerância a falhas. Portanto, nesta proposta também são incluídas estratégias para estas questões. A seguir, apresentamos uma visão geral do PDMS proposto, assim como a arquitetura interna dos pontos do sistema. Em seguida, discutimos detalhadamente sobre as estratégias propostas.

5.1 O SPEED

O Sistema P2P de Gerenciamento de Dados Baseado em Semântica (ou *Semantic PEEr-to-Peer Data Management System* - SPEED) [Pires et al. 2006] utiliza uma topologia de rede mista (DHT e super-ponto) de modo a explorar as vantagens oferecidas por ambas as topologias. Uma rede DHT é utilizada para auxiliar um ponto a encontrar outros pontos com interesses comuns, formando comunidades semânticas. Nas

comunidades, os pontos são organizados de acordo com a topologia super-ponto. Nas próximas seções, apresentamos uma visão geral da arquitetura do SPEED e, em seguida, a arquitetura interna dos pontos.

5.1.1 Visão Geral da Arquitetura

Conforme ilustrado na Figura 5.1, 3 (três) tipos diferentes de pontos são considerados na arquitetura do SPEED: pontos de dados, pontos de integração e pontos semânticos. Um **ponto de dados** é uma fonte de dados que compartilha dados com outros pontos de dados no sistema. Na Figura 5.1, os pontos I_1D_1 e I_1D_2 são exemplos de pontos de dados. Os pontos de dados são agrupados em **clusters semânticos**, de acordo com seus respectivos interesses. Um **interesse** é uma ontologia que representa os dados exportados por um ponto de dados. Sua principal utilidade é para o agrupamento dos pontos semanticamente relacionados. Um ponto pode participar de um ou mais clusters semânticos.

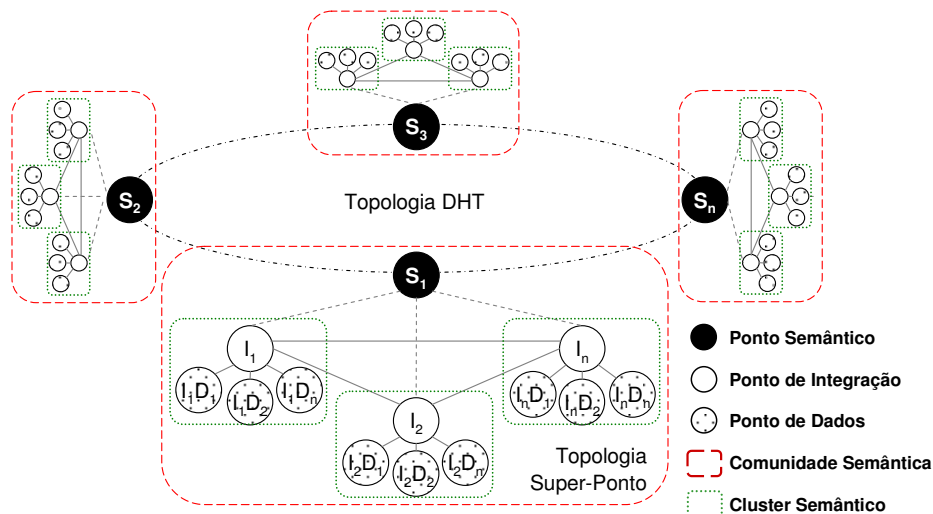


Figura 5.1 – Visão Geral da Arquitetura do SPEED

Em cada cluster semântico existe um tipo especial de ponto, caracterizado por apresentar um maior poder computacional, denominado **ponto de integração**. Os pontos de integração são pontos de dados com maior disponibilidade, largura de banda, poder de processamento e capacidade de armazenamento. Estes pontos são responsáveis por tarefas como gerenciamento de metadados dos pontos de dados, processamento de consultas e integração de dados. Por exemplo, o ponto I_1 é o ponto de integração do cluster semântico formado pelos pontos de dados I_1D_1 , I_1D_2 e I_1D_n .

Os pontos de integração comunicam-se com um **ponto semântico**, cuja função principal é oferecer uma ontologia sobre um domínio de conhecimento particular, por exemplo *educação*. Além disso, os pontos semânticos são responsáveis por tarefas como gerenciamento de metadados dos pontos de integração. Os metadados podem ajudar um ponto que deseja conectar-se à comunidade a descobrir clusters aos quais pode participar. Portanto, uma **comunidade semântica** é composta por clusters com interesses semânticos similares. Para que um ponto (de dados) possa conectar-se ao sistema, é necessária a identificação da comunidade semântica e dos clusters semânticos apropriados. Na Figura 5.1, S_1 é um exemplo de ponto semântico. Em nossa abordagem, o ponto de integração nomeia seu respectivo cluster semântico, enquanto que o ponto semântico nomeia sua respectiva comunidade semântica (por exemplo, o cluster semântico I_1 e a comunidade semântica S_1). Considerando nosso exemplo, a comunidade semântica S_1 é formada pelos clusters I_1 , I_2 e I_n .

No SPEED, a conexão de um ponto ocorre sempre pela rede DHT. Esta rede é formada exclusivamente por pontos semânticos, caracterizados por apresentarem altos índices de confiabilidade, largura de banda e disponibilidade. A função principal da rede DHT é facilitar a descoberta de recursos, auxiliando pontos que desejam se conectar a encontrarem de modo eficiente outros pontos já conectados com interesses similares. Neste sentido, a semântica associada ao conteúdo compartilhado pelos pontos é fator determinante para formação das comunidades semânticas.

Nas comunidades, os pontos são organizados segundo a topologia super-ponto. A utilização de clusters semânticos proporciona um ambiente mais adequado para aplicação de técnicas para identificação de mapeamentos semânticos, por exemplo *ontology matching* [Doan et al. 2002, Zhdanova et al. 2006]. Na topologia pura (não-estruturada e estruturada) um ponto pode definir mapeamentos entre esquemas com uma quantidade ilimitada (e possivelmente não relacionados semanticamente) de outros pontos. Na topologia super-ponto, se considerarmos a utilização de clusters semânticos, os mapeamentos são estabelecidos entre pontos semanticamente equivalentes. Logo, mapeamentos de melhor qualidade tendem a ser produzidos. No sistema proposto, os mapeamentos são estabelecidos entre o ponto de integração e seus respectivos pontos de dados (no nível de cluster), e entre os pontos de integração (no nível de comunidade).

Com a utilização da topologia super-ponto, o processamento de consultas torna-se mais eficiente, visto que cada comunidade semântica é dividida em clusters semânticos

e cada ponto de integração armazena metadados sobre as informações disponíveis nos seus pontos de dados associados. As consultas são direcionadas apenas para os clusters relevantes para respondê-la. A topologia super-ponto possibilita ainda que a heterogeneidade dos pontos participantes seja melhor explorada. Por exemplo, tarefas que demandam um volume maior de processamento são atribuídas aos pontos que oferecem os melhores recursos computacionais, ou seja, os pontos de integração.

Na arquitetura do sistema SPEED, os pontos semânticos necessitam apresentar altos índices de disponibilidade e largura de banda. Além disso, durante o processo de conexão, caso um ponto não consiga identificar uma comunidade semântica apropriada, o mesmo não poderá conectar-se ao sistema.

Em virtude da topologia de rede mista adotada, alguns requisitos necessitam ser considerados:

- Política de substituição de pontos de integração, em caso de falha ou desconexão;
- Presença de uma estratégia de atualização de metadados nos pontos semânticos e pontos de integração, em virtude da conectividade dinâmica dos pontos;
- Estratégia para balanceamento de carga dos clusters semânticos;
- Necessidade de redundância do ponto semântico, visto que o mesmo representa um ponto único de falha no sistema.

Conforme vimos anteriormente o sistema SPEED é composto de 3 (três) tipos de ponto: pontos de dados, pontos de integração e pontos semânticos. A seguir, apresentamos os componentes da arquitetura interna de cada um destes pontos.

5.1.2 Arquitetura Interna dos Pontos

Considera-se um **ponto de dados** um ponto qualquer (por exemplo, um simples computador ou um servidor) que pode conectar-se e/ou desconectar-se frequentemente da rede P2P. Um ponto de dados corresponde a uma fonte cujos dados são compartilhados com outros pontos mediante o estabelecimento de mapeamentos semânticos. Seus componentes arquiteturais incluem:

- *Camada Ponto-a-Ponto*: utilizada para comunicação com um ponto de integração;
- *Wrapper*: traduz as consultas elaboradas através da linguagem de consulta comum para a linguagem de consulta aplicada no ponto de dados, e vice-versa. É também

responsável pela tradução dos dados do modelo de dados da fonte para o formato do modelo de dados comum;

- *Fonte de Dados*: repositório contendo todos os dados disponíveis no ponto. É importante salientar que apenas os dados representados no esquema exportado podem ser acessados por outros pontos.

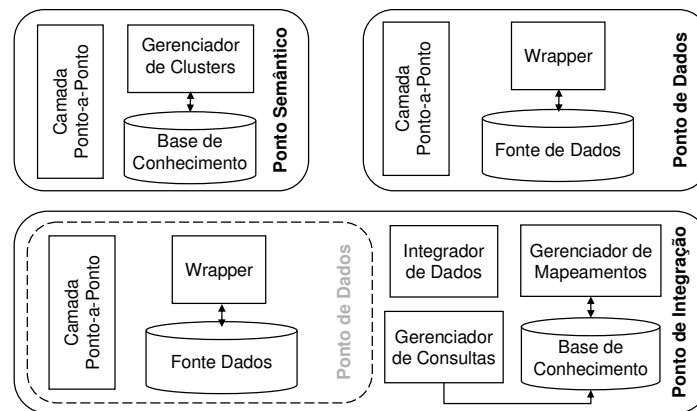


Figura 5.2 – Componentes Arquiteturais dos Pontos do SPEED

Em geral, um ponto ao conectar-se assume o papel de ponto de dados. Entretanto, em algumas situações (por exemplo, cluster semântico sobrecarregado ou cluster semântico com apenas um ponto) um ponto pode ser conectado diretamente como ponto de integração. Se um ponto é conectado como ponto de dados, dependendo de seus recursos computacionais, o mesmo pode mudar de papel na rede e tornar-se um ponto de integração em qualquer um dos clusters que participa. Isto pode ocorrer para substituir um ponto de integração que acabou de desconectar-se, por exemplo. Assim, os componentes arquiteturais de um ponto de dados e um ponto de integração são basicamente os mesmos, porém quando um ponto desempenha o papel de ponto de dados os componentes referentes ao papel de ponto de integração ficam desabilitados.

Um **ponto de integração** é um ponto de dados diferenciado em um cluster semântico por oferecer melhores recursos computacionais. Este ponto possui um conhecimento detalhado sobre os pontos de dados do cluster. Conforme visto na Figura 5.2, um ponto de integração também pode armazenar uma fonte de dados e, conseqüentemente, seus dados podem ser considerados durante o processamento de consultas. Além dos componentes arquiteturais já apresentados anteriormente, os outros componentes de um ponto de integração são:

- *Camada Ponto-a-Ponto*: utilizada para comunicação com os pontos de dados do mesmo cluster, pontos de integração da mesma comunidade semântica, e com o ponto semântico;
- *Base de Conhecimento*: armazena informações úteis para processos como identificação de mapeamentos semânticos, processamento de consultas e integração de dados como, por exemplo, o esquema exportado pelos pontos de dados do cluster e os mapeamentos semânticos;
- *Gerenciador de Consultas*: executa as consultas enviadas para o ponto de integração. Quando uma consulta é recebida, este componente consulta a base de conhecimento para identificar quais os pontos de dados capazes de respondê-la (pontos relevantes). Em seguida, o gerenciador de consultas reformula a consulta original, produz novas consultas e as envia para os pontos de dados do cluster relevantes para a consulta. Em paralelo, o gerenciador de consultas propaga a consulta original para os demais pontos de integração da mesma comunidade semântica. Os esquemas exportados e os mapeamentos semânticos são utilizados para reformulação de consultas;
- *Integrador de Dados*: integra o resultado das consultas retornados pelos pontos de dados e pontos de integração;
- *Gerenciador de Mapeamentos*: responsável pela geração e manutenção dos mapeamentos semânticos entre os esquemas exportados pelos pontos de dados. Na subseção 5.2.1, veremos que o esquema exportado por um ponto é mapeado para uma representação ontológica. Assim, no sistema SPEED os mapeamentos semânticos correspondem na verdade a relacionamentos entre conceitos de ontologias.

Um **ponto semântico** atua como um servidor de conhecimento (por exemplo, ontologias e metadados) na hierarquia dos pontos. Estes pontos oferecem um conjunto de ontologias pertencentes a um domínio de conhecimento em particular, por exemplo *educação*. As ontologias são utilizadas principalmente para: (i) *descoberta de recursos*: facilitar a localização de pontos que compartilham conteúdo semanticamente similar; (ii) *conectividade*: atuar como ponto de entrada e distribuir os pontos em comunidades e clusters semânticos apropriados; (iii) *identificação de mapeamentos semânticos*: identificação de equivalências semânticas entre os esquemas exportados pelos pontos; e (iv) *processamento de consultas*: roteamento de consultas apenas para clusters relevantes. Seus componentes arquiteturais incluem:

- *Camada Ponto-a-Ponto*: responsável pela comunicação com os pontos de integração da mesma comunidade semântica, e com os outros pontos semânticos;
- *Base de Conhecimento*: armazena informações úteis para manutenção da comunidade semântica, como ontologias e descrição dos pontos de integração participantes da comunidade;
- *Gerenciador de Clusters*: identifica se um ponto de dados pode participar da comunidade. Caso positivo, ajuda o ponto a descobrir o(s) cluster(s) semântico(s) mais apropriado(s). O esquema exportado pelo ponto de dados é utilizado em ambos os casos. Além disso, auxilia a auto-organização e balanceamento de carga dos clusters semânticos.

Nas próximas seções apresentamos nossas propostas de conectividade baseada em semântica, balanceamento de carga nos clusters semânticos e tolerância a falhas no SPEED. As tarefas de identificação de mapeamentos semânticos e processamento de consultas não fazem parte do escopo deste trabalho e estão sendo desenvolvidas em paralelo [Souza 2007].

5.2 Conectividade no SPEED

No sistema SPEED, basicamente o domínio semântico define como os pontos são organizados na rede. Os pontos semânticos são distribuídos ao longo da rede DHT e servem como ponto de entrada para sua respectiva comunidade. Nossa proposta de conectividade prevê a utilização de ontologias para organizar os pontos de dados e pontos de integração em comunidades semânticas e clusters semânticos. Por questões de nomenclatura, a partir daqui, um ponto (de dados) que deseja conectar-se ao sistema será chamado de **ponto solicitante** (PS).

Durante o processo de conexão, um ponto solicitante precisa primeiramente descobrir uma comunidade semântica na rede DHT. Uma vez encontrada a comunidade, são identificados na topologia super-ponto um ou mais clusters semânticos nos quais o ponto solicitante pode participar. O esquema exportado pelo ponto solicitante é de fundamental importância para descoberta da comunidade e dos clusters semânticos, visto que seus conceitos são comparados com os conceitos presentes no esquema exportado de outros pontos.

Como os pontos semânticos são pontos estáveis, ou seja, que oferecem boa disponibilidade e recursos computacionais, assumimos que a estrutura da rede DHT é

raramente modificada e o custo para sua construção e manutenção é relativamente baixo. Portanto, nossa proposta de conectividade preocupa-se principalmente com a organização dos pontos na rede super-ponto, ou seja, com a formação e manutenção dos clusters semânticos.

5.2.1 Interesse de um Ponto Solicitante

O interesse de um ponto é a representação semântica de seu conteúdo exportado. O interesse é fator determinante para descoberta de comunidades semânticas e clusters semânticos. A representação do conteúdo compartilhado baseia-se em padrões de Web Semântica (particularmente ontologias), e engloba os conceitos, atributos e relacionamentos tratados pelo ponto. Um conceito possui atributos e pode estar relacionado com outros conceitos.

Em geral, o interesse de um ponto é extraído a partir de seu esquema exportado. Para isso, pode ser necessária uma etapa de pré-processamento para normalização sintática dos nomes dos termos presentes no esquema exportado. A Figura 5.3 mostra um exemplo simples de um esquema exportado e seu respectivo interesse. O esquema exportado é formado por 4 (quatro) entidades as quais são mapeadas em conceitos de uma ontologia. Na literatura, existem diversas técnicas de mapeamento de modelos de dados comuns (por exemplo, relacional e orientado a objeto) para ontologias [Lakshmanan et al. 2003]. Em nosso trabalho, pretendemos realizar um estudo das principais propostas de mapeamento e selecionar a mais adequada.

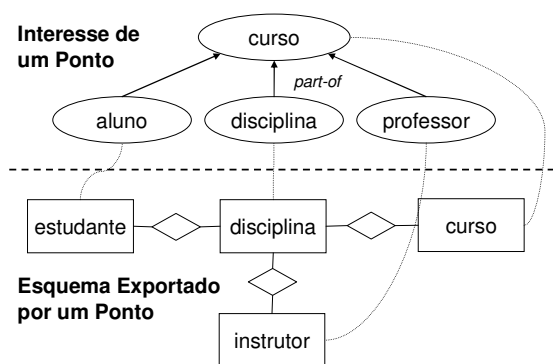


Figura 5.3 – Representação do Interesse de um Ponto

Conforme dito anteriormente, a conexão de um ponto solicitante inicia-se com a descoberta da comunidade semântica e, em seguida, com a descoberta de um ou mais clusters semânticos aos quais o ponto solicitante pode ser conectado. Em ambos os casos, técnicas de comparação ontológica [Bruijn et al. 2004a, Shvaiko et al. 2005,

Maedche et al. 2002] podem ser aplicadas para determinar a inclusão do ponto solicitante na comunidade semânticas e nos clusters semânticos. Uma das técnicas é a aplicação de **funções de comparação ontológica**. Tais funções recebem como entrada duas ontologias (por exemplo, O_1 e O_2) e produzem como resultado um valor que indica o grau de similaridade semântica entre as ontologias (GS). Além do agrupamento semântico de pontos, as funções de comparação ontológica também podem ser utilizadas para o roteamento de consultas em sistemas P2P [Wang et al. 2005].

Neste trabalho, pretendemos realizar um estudo detalhado sobre as principais técnicas de comparação ontológica com o objetivo de selecionar a mais apropriada para o sistema SPEED, ou ainda adaptar uma ou mais técnicas existentes. A seguir, mostramos os processos de descoberta de comunidade semântica e de clusters semânticos.

5.2.2 Descoberta da Comunidade Semântica

A conexão de um ponto solicitante inicia-se com o envio de seu interesse para um dos pontos semânticos localizados na rede DHT que, por sua vez, pode repassá-lo para os demais pontos semânticos (ver Figura 5.4). Para implementação da rede DHT, pretendemos utilizar um dos protocolos de rede pura estruturada discutidos: Chord, CAN, Pastry ou Tapestry. É objeto de estudo deste trabalho analisar o protocolo mais adequado.

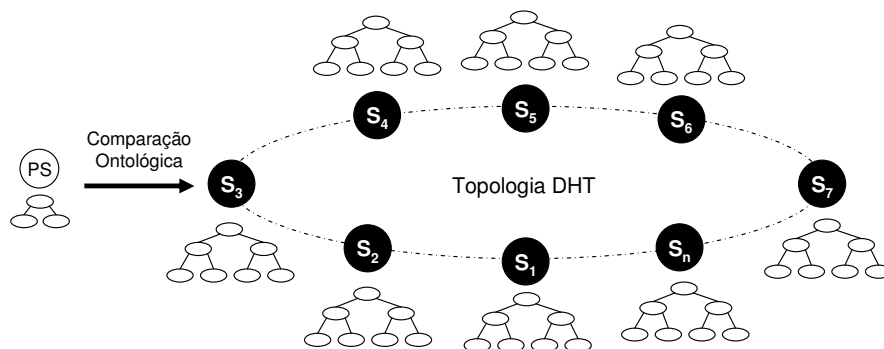


Figura 5.4 – Descoberta de uma Comunidade Semântica através de Comparação Ontológica

Em seguida, uma função de comparação ontológica é aplicada em cada ponto semântico da rede DHT. A função recebe como entrada o interesse do ponto solicitante (uma ontologia) e a ontologia de domínio disponível no ponto semântico. A comunidade na qual o ponto solicitante irá participar será aquela que apresente o maior grau de similaridade. Um limite de comunidade pode ser definido de modo a desconsiderar comunidades que apresentem um grau de similaridade baixo. Em uma primeira versão,

um ponto solicitante pode participar de apenas uma comunidade semântica. Ainda, partimos do princípio que a ontologia do ponto semântico é uma ontologia padrão do domínio, ou seja, representa a maior parte dos conceitos associados a um domínio de conhecimento.

Alguns PDMS (Humboldt Discoverer e PARIS) cuja arquitetura baseia-se na topologia pura estruturada implementada através do protocolo Chord, propõem a aplicação de uma função *hash* sobre a URI da ontologia do ponto solicitante. Como resultado, a função *hash* fornece o endereço de um ponto semântico cujo interesse é semanticamente equivalente ao do ponto solicitante. Embora os sistemas não ofereçam detalhes sobre tais funções *hash*, esta solução também será estudada neste trabalho.

5.2.3 Descoberta do(s) Cluster(s) Semântico(s)

Nas comunidades semânticas, o interesse dos pontos também é utilizado para organizar os mesmos em um nível ainda menor de granularidade semântica: os clusters semânticos. Cada cluster está associado a um conjunto de conceitos ontológicos (ontologia do cluster) que representa os dados compartilhados pelos pontos (de dados e de integração) daquele cluster. Este conjunto de conceitos é, na verdade, um subconjunto dos conceitos da ontologia do ponto semântico. A ontologia do cluster é armazenada no ponto de integração.

Analogamente à descoberta de comunidade, para descoberta de clusters uma função de comparação ontológica também é aplicada para cada cluster da comunidade semântica. Desta vez a função recebe como entrada o interesse do ponto solicitante (uma ontologia) e a ontologia do cluster (ver Figura 5.5). Como resultado, a função produz um valor indicando o grau de similaridade semântica entre as ontologias. Um limite de cluster pode ser definido para descartar clusters que apresentem baixo grau de similaridade semântica. Em seguida, o ponto solicitante é conectado a todos os clusters cujo resultado da função de comparação ontológica é maior que o valor do limite de cluster. Os conceitos do interesse semântico do ponto solicitante são adicionados à ontologia do cluster, caso ainda não existam. Neste caso, pode ser aplicado um processo de *merge* entre ontologias [Bruijn et al. 2004b].

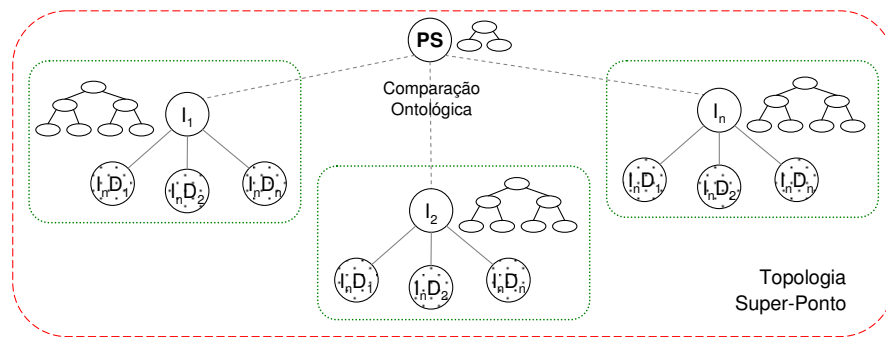


Figura 5.5 – Descoberta de Clusters Semânticos através de Comparação Ontológica

Por outro lado, caso o resultado retornado por cada cluster seja inferior a um determinado limite de cluster, significa dizer que nenhum dos clusters semânticos daquela comunidade trata do interesse do ponto solicitante. Neste caso, o ponto solicitante irá formar um novo cluster. A ontologia do novo cluster será formada inicialmente pelo interesse do ponto solicitante e evoluirá com a conexão de outros pontos a este cluster.

Para detalhar melhor a descoberta de clusters, considere o exemplo de uma ontologia do domínio de *educação* ilustrada na Figura 5.6. Observe que *escola* e *instituição educacional* representam o mesmo conceito e, por isso, são ligados através do relacionamento de sinonímia. Os conceitos *estudante*, *disciplina* e *professor* são parte de *curso* (relacionamento de meronímia). Analogamente, *curso* é formado por *estudante*, *disciplina* e *professor* (relacionamento de holonímia). O conceito *estudante* pode ser especializado em *graduação* ou *pós-graduação* (relacionamento de hipernímia), e vice-versa (relacionamento de hiponímia). Neste trabalho, consideramos os seguintes relacionamentos semânticos: sinonímia, meronímia, holonímia, hipernímia e hiponímia [WordNet 2007].

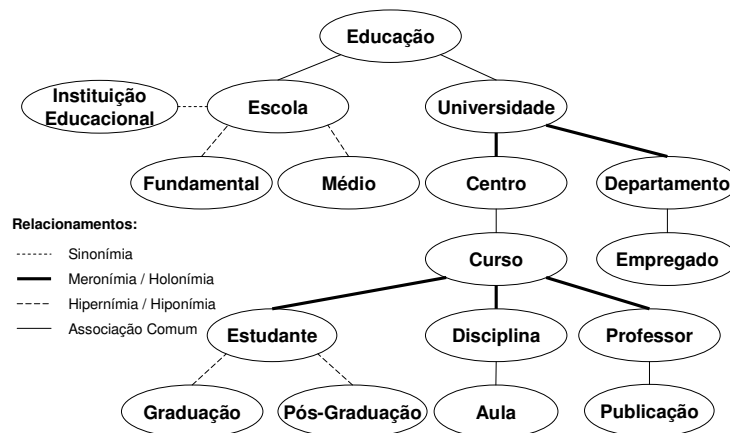


Figura 5.6 – Parte de uma Ontologia do Domínio “Educação”

Dando continuidade ao exemplo, considere os interesses semânticos de 6 (seis) pontos solicitantes (P_1, P_2, P_3, P_4, P_5 e P_6) ilustrados na Figura 5.7. Assuma ainda que os pontos em questão fazem parte de uma mesma comunidade (*educação*) e que esta encontra-se inicialmente vazia, ou seja, sem nenhum cluster (e conseqüentemente sem nenhum ponto).

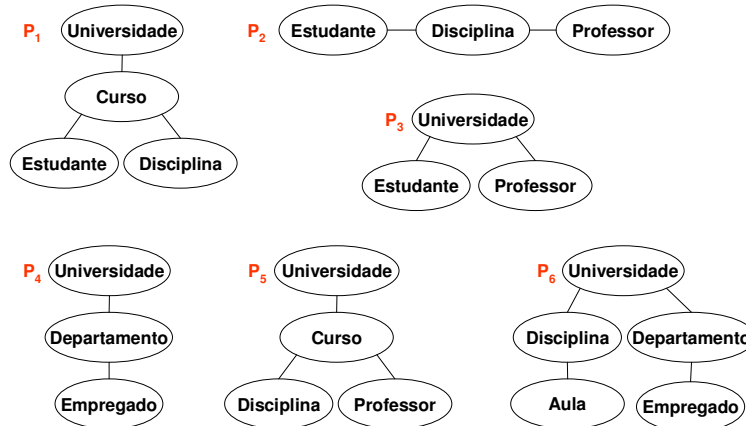


Figura 5.7 – Interesse Semântico dos Pontos Solicitantes P_1 até P_6

Na Tabela 5.1, ilustramos o passo-a-passo para a conexão dos pontos P_1, P_2, P_3, P_4, P_5 e P_6 , nesta ordem. Para cada conexão, apresentamos o estado parcial da formação dos clusters semânticos (C_n). Assuma que o valor do limite de cluster é 0.5. Note que: (i) na conexão de P_1 , a função de comparação ontológica não é executada pois P_1 é o primeiro ponto da comunidade e, portanto, irá compor o cluster C_1 ; (ii) a participação de P_2 no cluster C_1 provoca um *merge* entre as ontologias de P_2 e de C_1 ; (iii) como o resultado da função de comparação ontológica entre as ontologias de P_4 e de C_1 é inferior ao limite de cluster, P_4 é alocado em um novo cluster (C_2); (iv) P_6 é alocado em C_1 e C_2 , pois o resultado retornado pela função de comparação ontológica para ambos os clusters é superior ao valor do limite de cluster.

A ontologia do ponto semântico pode auxiliar o processo de comparação ontológica entre o interesse semântico de um ponto solicitante e a ontologia de um cluster. Uma consulta à ontologia do ponto semântico pode ajudar a definir se um ponto solicitante irá formar um novo cluster ou se o mesmo irá participar de um cluster já existente. Outra alternativa é simplesmente criar os clusters (ou seja, sem auxílio da ontologia do ponto semântico) e, periodicamente, executar um processo para verificar se as ontologias de dois ou mais clusters podem ser unificadas (neste caso, é necessário um *merge* entre ontologias dos clusters).

Tabela 5.1 – Formação de Clusters na Comunidade Semântica “Educação”

Ponto Conectado	Comparação Ontológica	Clusters Formados
P ₁	Não é executada	C ₁ = {P ₁ } O(C ₁) = {universidade, curso, estudante, disciplina}
P ₂	GS(O(P ₂), O(C ₁)) = 0.7	C ₁ = {P ₁ , P ₂ } O(C ₁) = {universidade, curso, estudante, disciplina, professor}
P ₃	GS(O(P ₃), O(C ₁)) = 1.0	C ₁ = {P ₁ , P ₂ , P ₃ } O(C ₁) = {universidade, curso, estudante, disciplina, professor}
P ₄	GS(O(P ₄), O(C ₁)) = 0.3	C ₁ = {P ₁ , P ₂ , P ₃ } O(C ₁) = {universidade, curso, estudante, disciplina, professor} C ₂ = {P ₄ } O(C ₂) = {universidade, departamento, empregado}
P ₅	GS(O(P ₅), O(C ₁)) = 1.0 GS(O(P ₅), O(C ₁)) = 0.2	C ₁ = {P ₁ , P ₂ , P ₃ , P ₅ } O(C ₁) = {universidade, curso, estudante, disciplina, professor} C ₂ = {P ₄ } O(C ₂) = {universidade, departamento, empregado}
P ₆	GS(O(P ₆), O(C ₁)) = 0.7 GS(O(P ₆), O(C ₁)) = 0.7	C ₁ = {P ₁ , P ₂ , P ₃ , P ₅ , P ₆ } O(C ₁) = {universidade, curso, estudante, disciplina, departamento, professor, aula} C ₂ = {P ₄ , P ₆ } O(C ₂) = {universidade, departamento, empregado}

A freqüente desconexão dos pontos de dados e dos pontos de integração pode resultar em um alto índice de atualizações da ontologia do cluster. Portanto, nossa proposta inicial é modificar a ontologia do cluster somente quando houverem conexões de pontos solicitantes. Esta solução pode ainda beneficiar possíveis reconexões.

A seguir, apresentamos nossa proposta para manutenção de clusters semânticos. A manutenção está diretamente relacionada com as propriedades dos sistemas P2P auto-gerenciamento e balanceamento de carga. A proposta visa minimizar o impacto de mudanças na estrutura da rede ocasionadas pela volatilidade dos pontos (de dados e de integração), tentando preservar a distribuição semântica dos pontos.

5.3 Balanceamento de Carga nos Clusters Semânticos

O comportamento dinâmico dos pontos de dados e pontos de integração pode provocar situações em que um cluster semântico possui mais pontos de dados do que outro na mesma comunidade. Neste caso, faz-se necessário redistribuir os pontos de

dados entre os clusters existentes, obedecendo o interesse dos pontos de dados em questão, na medida do possível. Além disso, podem ocorrer situações em que o interesse semântico de inúmeros pontos (de dados e de integração) coincida, fazendo com que estes pontos concentrem-se em um único cluster semântico. Portanto, é necessária a formação de um novo cluster semanticamente equivalente.

A definição do parâmetro **tamanho médio do cluster** é importante no sentido de que, se a maioria dos pontos conectados são pontos de integração (presença de muitos clusters), a comunidade terá um comportamento similar ao de uma rede P2P pura. Assim, durante o processamento de consultas inúmeros pontos de integração serão consultados. Por outro lado, caso exista uma quantidade reduzida de pontos de integração (presença de poucos clusters com muitos pontos de dados), o cluster terá um comportamento similar ao de um sistema centralizado. Neste caso, os pontos de integração podem ficar sobrecarregados.

O valor ideal para o tamanho médio do cluster depende principalmente da quantidade de pontos que podem estar conectados simultaneamente ao sistema, o que não é uma tarefa simples de ser prevista. Além disso, o valor pode variar de acordo com o tipo de aplicação a ser desenvolvida. Em nossa proposta, pretendemos utilizar um parâmetro para especificar limites mínimo e máximo para o tamanho do cluster semântico. O limite inferior é definido como $\text{tamanho médio do cluster} - \text{parâmetro}$, enquanto o limite máximo é igual ao $\text{tamanho médio do cluster} + \text{parâmetro}$. Ambos os parâmetros podem ser configurados pelo administrador do sistema.

Regras do tipo evento-condição-ação (ECA) podem ser aplicadas para manter o tamanho médio dos clusters semânticos. Dependendo dos eventos e das condições, uma ou mais ações podem ser disparadas. Três tipos de evento são considerados para as regras ECA (ver Tabela 5.2): solicitação de conexão de um ponto (de dados) a um cluster, desconexão/falha de ponto de dados de um cluster e desconexão/falha de um ponto de integração de um cluster. A condição consiste basicamente em avaliar se o tamanho atual do cluster permanece entre os limites inferior e superior. São exemplos de ações que podem ser executadas: a conexão de um ponto solicitante (neste caso, a condição é avaliada como “falsa”) e a eleição de um novo ponto de integração (condição “verdadeira”).

Tabela 5.2 – Exemplos de Regras ECA para Manter o Tamanho Médio dos Clusters

Evento	Condição	Ação
Conexão de um ponto (de dados) a um cluster	Falsa	Conectar ponto de dados
	Verdadeira	Formar novo cluster semântico Conectar ponto de dados
Desconexão de um ponto de dados	Falsa	Desconectar ponto de dados
	Verdadeira	Desconectar ponto de dados Redistribuir pontos
Desconexão de um ponto de integração	Falsa	Desconectar ponto de integração Eleger novo ponto de integração
	Verdadeira	Desconectar ponto de integração Eleger novo ponto de integração Redistribuir pontos

A regra da Figura 5.8 é disparada quando um ponto solicitante requisita conexão a um cluster já existente. Se o tamanho do cluster (incluindo o novo ponto) está entre os limites inferior e superior, o ponto de dados é simplesmente conectado. Caso contrário, um novo cluster semântico é formado com a presença do novo ponto. Excepcionalmente, o ponto solicitante é conectado como ponto de integração e poderá ser substituído quando um outro ponto for conectado ao mesmo cluster. A substituição de um ponto de integração é discutida na próxima seção.

```

ON data peer connection (data peer, cluster, community);
IF NOT (cluster average size - threshold ≤ size(cluster) + 1 ≤ cluster
average size + threshold) THEN
    Form new cluster(community);
connect(data peer, cluster);

```

Figura 5.8 – Exemplo de Regra ECA para Manutenção do Tamanho Médio dos Clusters Semânticos

5.4 Tolerância a Falhas

Quando um ponto de integração desconecta-se do sistema ou apresenta falha, é necessária a eleição de um novo ponto de integração. Os pontos candidatos a novo ponto de integração são os pontos de dados do mesmo cluster que apresentam uma série de características relevantes. Tais características incluem os recursos físicos disponíveis nos pontos de dados, por exemplo memória física, espaço em disco, poder de processamento e largura de banda. Além disso, o comportamento dos pontos de dados

enquanto estão conectados ao sistema deve ser um outro fator relevante. Logo, características subjetivas também são consideradas: disponibilidade, tempo de resposta e quantidade de dados.

Para medir a capacidade de um ponto de dados, pretendemos utilizar a **função de cálculo de capacidade de um ponto**, proposta em [Zhuang et al. 2004]:

$$capacity(p) = \sum_{i=1}^n w_i * v_i(p)$$

Na função $capacity(p)$, $v_i(p)$ é o valor da i -ésima característica para o ponto p analisado e $w_i(d)$ é o peso da respectiva característica. Ambas as características (físicas e subjetivas) devem ser consideradas pela função $capacity(p)$. O peso de uma característica é atribuído pelo administrador do sistema.

Além da análise da capacidade dos pontos de dados candidatos, a estratégia de eleição de um novo ponto de integração também pode considerar o interesse semântico destes pontos. Por exemplo, um novo ponto de integração pode ser aquele que apresente o maior número de conceitos comuns com outros pontos do mesmo cluster, maior quantidade de conceitos exportados e/ou maior quantidade de pontos semanticamente relacionados.

5.5 Contribuições Esperadas

Uma das contribuições deste trabalho é a arquitetura do sistema SPEED [Pires et al. 2006]. Durante as pesquisas realizadas, nenhum dos PDMS estudados apresentou uma arquitetura mista baseada nas topologias de rede pura estruturada (DHT) e super-ponto. A rede DHT é formada exclusivamente por pontos semânticos, caracterizados por apresentarem bons índices de confiabilidade, largura de banda e disponibilidade. Sua função principal é facilitar a descoberta de recursos, auxiliando pontos que desejam se conectar a encontrarem outros pontos já conectados com interesses semânticos similares.

Na topologia super-ponto, a utilização de clusters semânticos proporciona um ambiente mais adequado para aplicação das técnicas de identificação de mapeamentos semânticos. O processamento de consultas torna-se mais eficiente, visto que as consultas são direcionadas apenas para os clusters relevantes. Além disso, a topologia super-ponto possibilita ainda que a heterogeneidade dos pontos participantes seja melhor explorada. Neste sentido, as demais contribuições deste trabalho podem ser descritas como se segue.

Dentro do contexto do sistema SPEED, propomos estratégias para conectividade de pontos, balanceamento de carga e tolerância a falhas. Para todas as estratégias, pretendemos desenvolver modelos, técnicas e algoritmos que explorem principalmente a semântica associada ao conteúdo exportado pelos pontos que desejam participar do sistema.

Em relação à conectividade de pontos, as contribuições deste trabalho dizem respeito ao desenvolvimento de soluções para a descoberta de comunidades semânticas, e descoberta, formação e manutenção de clusters semânticos. Para tratar as questões relacionadas à conectividade dos pontos foram estabelecidos os seguintes objetivos:

- Definição de um protocolo para a rede DHT do sistema SPEED;
- Estudo de viabilidade da ontologia do cluster, no que se refere à manutenção (em virtude da volatilidade dos pontos);
- Escolha de um algoritmo de mapeamento do esquema exportado por um ponto no formato da fonte de dados (por exemplo, relacional) para uma representação ontológica (interesse semântico do ponto solicitante);
- Definição de uma função de comparação ontológica que verifique a equivalência semântica entre o interesse semântico de um ponto solicitante e: (i) a ontologia disponível em um ponto semântico; e (ii) a ontologia do cluster disponível em um ponto de integração.

No tocante ao balanceamento de carga, as contribuições referem-se ao desenvolvimento de soluções que possibilitem a redistribuição de pontos de dados em clusters semânticos existentes, tentando preservar a semântica do conteúdo exportado pelos pontos. Para tratar os problemas relacionados com esta questão foram definidos os seguintes objetivos:

- Regras para manutenção do tamanho médio dos clusters. Neste sentido, propomos a aplicação de regras ECA disparadas quando o tamanho de um cluster estiver fora dos limites inferior e superior permitidos;
- Algoritmo de balanceamento de carga que considere a formação de novos clusters (para clusters sobrecarregados) e a fusão de clusters existentes (para clusters com poucos pontos de dados). O processo de *merge* entre ontologias pode ser aplicado em ambos os casos.

No que diz respeito à tolerância a falhas, as contribuições referem-se ao desenvolvimento de uma solução que possibilite a substituição de um ponto de integração. Para tratar esta questão, definimos como objetivo a especificação e a implementação de um algoritmo para substituição de um ponto de integração.

Em nossa proposta inicial, pretendemos atribuir a um dos pontos de dados do cluster a função de ponto de integração candidato. Este ponto mantém uma réplica da base de conhecimento do ponto de integração, servindo como ponto redundante. Periodicamente, a base de conhecimento do ponto de integração é replicada para o ponto de integração candidato. Em caso de falha ou desconexão neste ponto, o ponto de integração candidato assume seu papel. Em caso de falha ou desconexão do ponto de integração candidato um dos pontos de dados é escolhido para ser o novo ponto de integração candidato.

5.6 Metodologia

Para atender os objetivos de nossa proposta, pretendemos desenvolver o trabalho conforme as etapas descritas a seguir.

- **Etapa 1: Formalização da Arquitetura Proposta e seus Componentes**
 - Nesta etapa pretendemos definir um formalismo que represente os componentes do sistema SPEED (por exemplo, pontos, comunidade semântica e cluster semântico). O formalismo visa facilitar a especificação de algoritmos a serem desenvolvidos nas próximas etapas;
- **Etapa 2: Identificação de Metadados**
 - Especificar as informações a serem armazenadas na base de conhecimento dos pontos semânticos e pontos de integração.
- **Etapa 3: Definição da Estratégia de Conectividade de Pontos**

Durante esta etapa serão realizadas as seguintes tarefas:

 - Definição de formalismos referentes às regras de conectividade do sistema. Por exemplo, um ponto de dados pode participar de mais de um cluster na mesma comunidade;
 - Especificação do fluxo de transmissão de mensagens entre os pontos durante o processo de conectividade de um ponto solicitante;

- Estudo detalhado dos protocolos de rede DHT existentes e definição do protocolo mais apropriado para o sistema SPEED. No estudo, pretendemos analisar ainda as funções *hash* para ontologias, utilizadas em outros PDMS;
 - Análise de algoritmos de mapeamento de um esquema exportado (por exemplo, esquema relacional) para uma representação ontológica. Em seguida, será feita a escolha do algoritmo mais adequado para o sistema SPEED;
 - Estudo e definição de estratégias para formação e manutenção de clusters. Esta tarefa prevê a análise de técnicas para realizar *merge* entre ontologias considerando a volatilidade dos pontos de dados e de integração;
 - Definição de uma função de comparação ontológica que possibilite verificar a equivalência semântica entre duas ontologias, onde uma ontologia é a do interesse semântico de um ponto solicitante e a outra é a ontologia de um ponto semântico ou a ontologia de um cluster;
 - Implementação de protótipo para validação da estratégia de conectividade.
- **Etapa 4: Definição da Estratégia de Balanceamento de Carga**
 - Definição de formalismos referentes às regras de balanceamento de carga;
 - Especificação do fluxo de transmissão de mensagens entre os pontos durante o processo de balanceamento de carga nos clusters;
 - Implementação de regras ECA para manter o tamanho médio dos clusters;
 - Especificação de um algoritmo para balanceamento de carga entre clusters semânticos. A redistribuição dos pontos pode levar à formação de novos clusters e/ou à fusão de clusters existentes.
 - Implementação de protótipo para validação da estratégia de balanceamento de carga.
- **Etapa 5: Definição da Estratégia de Tolerância a Falhas**
 - Definição de formalismos referentes às regras de tolerância a falhas no sistema;
 - Especificação do fluxo de transmissão de mensagens entre os pontos durante o processo de substituição de um ponto de integração;
 - Especificação de um algoritmo para substituição do ponto de integração por um ponto candidato em um cluster semântico;

- Implementação de protótipo para validação da estratégia de tolerância a falhas.
- **Etapa 6: Execução de Testes e Validação dos Resultados**
 - Definição dos cenários para realização de testes, incluindo quantidade de pontos (semânticos, de integração e de dados), esquemas exportados e parâmetros do sistema (por exemplo, tamanho médio do cluster e limite de comparação ontológica);
 - Definição de critérios e métricas para avaliação das estratégias propostas (por exemplo, quantidade de mensagens transmitidas entre os pontos para a conexão de um ponto solicitante);
 - Execução dos testes e análise dos resultados.
- **Etapa 7: Escrita da Tese**
 - Esta etapa deverá ser realizada em paralelo com as outras etapas e compreende a escrita final da tese. Como resultado desta etapa, além da tese final, artigos científicos serão submetidos para publicação.

5.7 Cronograma

Nesta seção é apresentado o cronograma para a realização do trabalho proposto de acordo com o cronograma definido pelo Programa de Pós-Graduação da UFPE. O cronograma apresenta as atividades já realizadas e a serem desenvolvidas.

Atividades Realizadas (Março/2005 a Março/2007)

- *Atividade 1:* Créditos em Disciplinas;
- *Atividade 2:* Levantamento do Estado da Arte;
- *Atividade 3:* Estudo de PDMS existentes e Definição da Arquitetura do Sistema SPEED;
- *Atividade 4:* Preparação da Qualificação e Proposta de Tese de Doutorado;
- *Atividade 5:* Apresentação da Qualificação e Proposta de Tese de Doutorado.

Atividades a serem realizadas (Abril/2007 a Março/2009)

- *Atividade 6:* Definição das estratégias de conectividade de pontos, balanceamento de carga e tolerância a falhas;

- *Atividade 7:* Desenvolvimento de Algoritmos;
- *Atividade 8:* Implementação de Protótipos;
- *Atividade 9:* Validação de Protótipos;
- *Atividade 10:* Escrita da Tese de Doutorado e Submissão de Artigos Científicos;
- *Atividade 11:* Defesa da Tese de Doutorado.

Atividades	2005									
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Atividade 1										
Atividade 2										

Atividades	2006											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Atividade 3												
Atividade 4												

Atividades	2007											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Atividade 4												
Atividade 5												
Atividade 6												

Atividades	2008											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Atividade 6												
Atividade 7												
Atividade 8												
Atividade 9												
Atividade 10												

Atividades	2009											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Atividade 10												
Atividade 11												

Referências Bibliográficas

- [Aberer et al. 2002a] Aberer, K., Hauswirth, M. (2002) “An Overview on Peer-to-Peer Information Systems”. In Proc. of Workshop on Distributed Data and Structures (WDAS’02), Paris, France.
- [Aberer et al. 2002b] Aberer, K., and Hauswirth, M. (2002) “Semantic Gossiping”. In Database and Information Systems Research for Semantic Web and Enterprises, Invitational Workshop, University of Georgia, Amicalola Falls and State Park, Georgia.
- [Aberer et al. 2003] Aberer, K., Cudré-Mauroux, P., Hauswirth, M. (2003) “The chatty web: Emergent semantics through gossiping”. In WWW’03: Proc. of the 12th International Conference on World Wide Web. ACM Press, pp. 197-206.
- [Akbarinia et al. 2006] Akbarinia, R., Martins, V. (2006) “Data Management in the APPA P2P System”. In Proc. of the 7th International Conference on High Performance Computing for Computational Science (VECPAR’06), Rio de Janeiro, Brazil.
- [Anceaume et al. 2003] Anceaume, E., Gradinariu, M., Roy, M. (2003) “Self-Organizing Systems Case Study: Peer-to-Peer Systems”. In Proc. of the 17th International Symposium on Distributed Computing (DISC’03). Sorrento, Italy.
- [Andersen et al. 2001] Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R. (2001) “Resilient Overlay Networks. In Proc. of the 18th ACM Symposium on Operating Systems Principles, Banff, Canada.
- [Arenas et al. 2003] Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R. J., Mylopoulos, J. (2003) “The Hyperion Project: From Data Integration to Data Coordination”. In SIGMOD Record, Special Issue on Peer-to-Peer Data Management, 32(3):53-58.
- [Backx et al. 2002] Backx, P., Wauters, T., Dhoedt, B., Demeester, P. (2002) “A Comparison of Peer-to-Peer Architectures”. In Proc. of the Eurescom Summit.
- [Belian 2005] Belian, R.B. (2005) “Integração Semântica de Informações”. Thesis Proposal. Federal University of Pernambuco (UFPE).
- [Bellahsène et al. 2004a] Bellahsène R., King, N., Roantree M. (2004) “Services for Large Scale P2P Networks”. European Research Consortium for Informatics and Mathematics News Journal (ERCIM’04).
- [Bellahsène et al. 2004b] Bellahsène, Z., Roantree, M. (2004) “Querying Distributed Data in a Super-Peer Based Architecture”. In Proc. of the 15th International Conference on Database and Expert Systems Applications (DEXA’04), Volume 3180 of LNCS, pp. 296-305, Zaragoza, Spain.
- [Berners-Lee et al. 2001] Berners-Lee, T., Hendler, J., Lassila, O. (2001) “The Semantic Web”. Scientific American, 284(5):34-43.
- [Bernstein et al. 2002] Bernstein, P., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I. (2002) “Data Management for Peer-to-Peer Computing: A Vision”. In Proc. of the 5th International Workshop on the Web and Databases, Wisconsin, USA.

- [Brito 2005] Brito, G. (2005) “Integração de Objetos de Aprendizagem no Sistema ROSA-P2P”. MSc Dissertation, IME-RJ, Brasil.
- [Brito et al. 2005] Brito, G., Moura, A. M. (2005) “ROSA-P2P: a Peer-to-Peer System for Learning Objects Integration on the Web”. In Proc. of the 11th Brazilian Symposium on Multimedia and the Web (WebMedia’05), Poços de Caldas, Minas Gerais, Brazil. pp. 1-9.
- [Bruijn et al. 2004a] Bruijn, J., Polleres, A. (2004) “Towards an Ontology Mapping Specification Language for The Semantic Web”. Digital Enterprise Research Institute (DERI) Technical Report.
- [Bruijn et al. 2004b] Bruijn, J., Martin-Recuerda, F., Manov, D., Ehrig, M. (2004) “D4.2.1 State-of-the-art-survey on Ontology Merging and Aligning” v1. SEKT Project deliverable D4.2.1.
- [Castano et al. 2003] Castano, S., Ferrara, A., Montanelli, S., Pagani, E., Rossi, G.P. (2003) “Ontology-Addressable Contents in P2P Networks”. In Proc. of WWW’03 1st SemPGRID Workshop, Budapest, Hungary.
- [Castano et al. 2004] Castano, S., Ferrara, A., Montanelli, S., Racca, G. (2004) “Semantic Information Interoperability in Open Networked Systems”. In Proc. of the International Conference on Semantics of a Networked World (ICSNW’04), Paris, France.
- [Castano et al. 2005] Castano, S., Montanelli, S. (2005) “Semantic Self-Formation of Communities of Peers”. In Proc. of the ESWC Workshop on Ontologies in Peer-to-Peer Communities, Heraklion, Greece.
- [Chawathe et al. 2003] Chawathe, Y., Ratnasam, S., Breslau, L., Lanhan, N., Shenker, S. (2003) “Making Gnutella-like P2P Systems Scalable”. In Proc. of ACM SIGCOMM’03.
- [Comito et al. 2006] Comito, C., Patarin, S., Talia, D. (2006) “A Semantic Overlay Network for P2P Schema-Based Data Integration”. In Proc. of the 11th IEEE Symposium on Computers and Communications (ISCC’06), pp. 88-94.
- [Crespo et al. 2002] Crespo, A., Garcia-Molina, H. (2002) “Semantic Overlay Networks for P2P Systems”. Technical Report, Stanford University.
- [Dinesh 2004] Dinesh, V. (2004) “Legitimate Applications of Peer-To-Peer Networks”. Wiley-Interscience, Volume 13-5.
- [Doan et al. 2002] Doan, A., Madhavan, J., Domingos, P., Halevy, A. (2002) “Learning to map between ontologies on the semantic web”. In Proc. of the 11th International WWW Conference, Honolulu, Hawaii. pp. 662-673.
- [Doval et al. 2003] Doval, D., O’Mahony, D. (2003) “Overlay Networks: A Scalable Alternative for P2P”. IEEE Internet Computing, vol. 07, no. 4, pp. 79-82.
- [Elmagarmid et al. 1999] Elmagarmid, A., Rusinkeiwicz, M., Sheth, A. (1999) “Management of Heterogeneous and Autonomous Database Systems”. Morgan Kaufmann Publishers, 1st Edition.
- [Emule 2007] Emule. The Emule Project. <http://emule-project.net/home/perl/help.cgi?l=30>, , último acesso em 20/5/2008

- [Fiorano 2003] Fiorano Software. (2003) “Super-Peer Architectures for Distributed Computing”. White Paper, Fiorano Software, Inc.
- [Foster 2002] Foster, I. “The Grid: A New Infrastructure for 21st Century Science”. *Physics Today*, 55 (2). 42-47.
- [Foster et al. 2003] Foster, I., Iamnitchi, A. (2003) “On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing”. In Proc. of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS’03), Berkeley, USA.
- [Freenet 2007] Freenet. The Free Network Project. <http://freenetproject.org/>, último acesso em 20/5/2008
- [Furtado 2005] Furtado, P. (2005) “Flexible Networked Data Management on P2P”. In Proc. of the 16th Internacional Workshop on Database and Expert Systems Applications (DEXA’05). pp. 1127-1131.
- [Globus 2007] The Globus Alliance. <http://www.globus.org/>, último acesso em 20/5/2008
- [Gnutella 2007] Gnutella Website. <http://www.gnutella.com/>, último acesso em 20/5/2008
- [Gong 2002] Gong, L. (2002) “Project JXTA: a Technology Overview”. Technical Report, Sun Microsystems, Inc. Palo Alto, CA, USA.
- [Groove 2007] Groove Networks. <http://www.groove.net/home/index.cfm>, , último acesso em 20/5/2008
- [Gruber 1993] Gruber, T. (1993) “A Translation Approach to Portable Ontologies. Knowledge Acquisition”, Volume 5, n.2, p.199-220.
- [Guarino 1998] Guarino, N. (1998) “Formal Ontology and Information Systems”. In Proc. of FOIS’98, Trento, Italy. Pp. 3-15.
- [Halevy 2000] Halevy, A. Y. (2000) “Theory of Answering Queries Using Views”. *SIGMOD Record*, Vol. 29, No. 4, pp. 40-47.
- [Halevy et al. 2003a] Halevy, A. Y., Ives, Z. G., Mork, P., Tatarinov, I. (2003) “Piazza: Data Management Infrastructure for Semantic Web Applications”. In Proc. of the 12th International World Wide Web Conference (WWW’03), Budapest, Hungary.
- [Halevy et al. 2003b] Halevy, A. Y., Ives, Z., Suciu, D., Tatarinov, I. (2003) “Schema Mediation in Peer Data Management Systems”. In Proc. of the International Conference on Data Engineering (ICDE’03), Bangalore, India. pp. 505-516.
- [Halevy et al. 2004] Halevy, A., Ives, Z., Mork, P., Tatarinov, I. (2004) “Piazza: Mediation and Integration Infrastructure for Semantic Web Data”. *Journal of Web Semantics*, Vol. 1(2), pp. 155-175.
- [Heese et al. 2005] Heese R, Herschel S, Naumann F, Roth A. (2005) “Self-Extending Peer Data Management”. In GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW’05), Karlsruhe, Germany.
- [Helios 2007] The HELIOS Project Website. <http://islab.dico.unimi.it/helios/>, último acesso em 20/5/2008.
- [Herschel et al. 2005] Herschel, S., Heese, R. (2005) “Humboldt Discoverer: A Semantic P2P index for PDMS”. In Proc. of the International Workshop Data Integration and the Semantic Web.

- [Johnstone et al. 2005] Johnstone, S., Sage, P., Milligan P. (2005) “iXChange – A Self-Organising Super Peer Network Model”. In Proc. of the 10th IEEE Symposium on Computers and Communications (ISCC’05), Volume 00, pp. 164-169.
- [Katchaounov 2003] Katchaounov, T. (2003) “Query Processing for Peer Mediator Databases”. PhD thesis, Department of Information Technology, Uppsala University, Sweden
- [Kementsietsidis et al. 2003] Kementsietsidis, A., Arenas, M., Miller, R. J. (2003) “Managing Data Mappings in the Hyperion Project”. In Proc. of the International Conference on Data Engineering (ICDE’03), Bangalore, India.
- [Kim 1995] Kim, W. (1995) “Modern Database Systems: The Object Model, Interoperability, and Beyond”. ACM Press and Addison-Wesley.
- [Kimball 2002] Kimball, R. (2002) “The Data Warehouse Toolkit”. 2^a Edição. Editora Campus.
- [Korth et al. 2006] Korth, H. F., Silberschatz, A. (2006) “Sistemas de Bancos de Dados”, Makron Books, 5^a Edição, Editora Campus.
- [Lakshmanan et al. 2003] Lakshmanan, L., Sadri, F. (2003) “Information Integration and the Semantic Web”. In Proc. of the IEEE Data Eng. Bull. 26(4): 19-25.
- [Ledlie et al. 2002] Ledlie, J., Taylor, J., Serban, L., Seltzer, M. (2002) “Self-organization in peer-to-peer systems”. In Proc. of the 10th European SIGOPS Workshop.
- [Leibowitz et al. 2003] Leibowitz, N., Ripeanu, M., Wierzbicki, A. (2003) “Deconstructing the KaZaA Network”. In Proc. of the 3rd IEEE Workshop on Internet Applications (WIAPP’03), San Jose, CA, USA.
- [Leuf 2002] Leuf, B. (2002) “Peer-to-Peer: Collaboration and Sharing over the Internet”. Pearson Education, 1st Edition.
- [Levy 2000] Levy, A. Y. (2000) “Logic-based techniques in data integration”. In Jack Minker, editor, Logic-Based Artificial Intelligence, Kluwer Academic Publishers, Dordrecht. pp. 575-595.
- [Li et al. 2004] Li, J., Vuong, S. (2004) “An Efficient Clustered Architecture for P2P Networks”. In Proc. of the 18th International Conference on Advanced Information Networking and Applications (AINA’04) Volume 1, pp. 278.
- [Li et al. 2005] Li, J., Vuong, S. (2005) “Ontology-Based Clustering and Routing in Peer-to-Peer Networks”. In Proc. of the 6th International Conference on Parallel and Distributed Computing, Applications and Technologies, Dalian, China.
- [Lin 2004] Lin, Y. (2004) “Peer-to-Peer Systems”. Distributed Database Systems, School of Computer Science, McGill University.
- [Lóscio 2003] Lóscio, B. F. (2003) “Managing the Evolution of XML-based Mediation Queries”. PHD Thesis, Federal University of Pernambuco, Brazil.

- [Löser et al. 2003a] Löser, A., Naumann, F., Siberski, W., Nejd, W., Thaden, U. (2003) "Semantic Overlay Clusters within Super-Peer Networks". In Proc. of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB 2003, Berlin, Germany.
- [Löser et al. 2003b] Löser, A, Siberski, W., Wolpers, M., Nejd, W. (2003) "Information Integration in Schema-Based Peer-to-Peer Networks". In Proc. of the Conference on Advanced Information Systems Engineering.
- [Lv et al. 2002] Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S. (2002) "Search and Replication in Unstructured Peer-to-Peer Networks". In Proc. of the 16th ACM International Conference on Supercomputing (ICS'02), New York, USA.
- [Maedche et al. 2002] Maedche, A., Staab, S. (2002) "Measuring Similarity between Ontologies". In Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management, Sigüenza, Spain. pp. 251-263.
- [Milojicic et al. 2002] Milojicic, D., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., Xu, Z. (2002) "Peer-to-Peer Computing". Technical Report HPL-2002-57, HP Labs.
- [Montresor 2004] Montresor, A. (2004) "A Robust Protocol for Building Superpeer Overlay Networks". In Proc. of the 4th International Conference on Peer-to-Peer Computing (P2P'04), Zurich, Switzerland. pp. 202-209.
- [Morpheus 2007] Morpheus. <http://www.morpheus.com/>, último acesso em 02/01/2007
- [Napster 2007] Napster. <http://www.napster.com/>, último acesso em 02/01/2007
- [Nejd et al. 2002] Nejd, W., Wolf, B., Qu, B., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palm, M., Risch, T. (2002) "Edutella: a P2P Network Infrastructure Based on RDF". WWW'02, Honolulu, Hawaii, USA.
- [Nejd et al. 2003] Nejd, W., Siberski, W., Sintek, M. (2003) "Design Issues and Challenges for RDF- e Schema-Based Peer-to-Peer Systems". ACM SIGMOD Record Volume 32, Issue 3, pp.41-46.
- [Ng et al. 2002] Ng, W. S., Ooi, B. C., and Tan, K.-L. (2002) "BestPeer: A Self-Configurable Peer-to-Peer System". In Proc. of the 18th International Conference on Data Engineering, San Jose, CA, USA. pp. 272.
- [Ng et al. 2003] Ng, W. S., Ooi, B. C., Tan, K.-L., Zhou, A. (2003) "PeerDB: A P2P-based System for Distributed Data Sharing". In Proc. of the International Conference on Data Engineering (ICDE'03). 633-644.
- [Ooi et al. 2003a] Ooi, B. C., Shu, Y., Tan, K.-L. (2003) "Relational Data Sharing in Peer-based Data Management Systems". ACM SIGMOD Record, 32 (3), 59-64.

- [Ooi et al. 2003b] Ooi, B. C., Tan, K.-L., Zhou, A., Goh, C. H., Li, Y., Liau, C. Y., Ling, B., Ng, W. S., Shu, Y., Wang, X., Zhang, M. (2003) "PeerDB: Peering into Personal Databases". In SIGMOD Conference 2003: 659
- [Oram 2001] Oram, A. (2001) "Peer-to-Peer: Harnessing the Power of Disruptive Technologies". O'Reilly.
- [OurGrid 2007] OurGrid. <http://www.ourgrid.org/>, último acesso em 02/01/2007
- [Özsu et al. 2001] Özsu, M. T., Valduriez, P. (1999) "Principles of Distributed Database Systems". Prentice Hall, 2nd Edition.
- [Papadimos et al. 2003] Papadimos, V., Maier, D., Tufte, K. (2003) "Distributed Query Processing and Catalogs for Peer-to-Peer Systems". In Proc. of Conference on Innovative Data Systems Research (CIDR'03), Asilomar, CA, USA.
- [Pires et al. 2006] Pires, C. E. S., Lóscio, B. F., Salgado, A. C. (2006) "Gerenciamento de Dados em Sistemas P2P". In Proc. of the 21st Brazilian Symposium on Databases (SBBD'06), Florianópolis, Brazil. pp. 310.
- [Rahm et al. 2001] Rahm, E., Bernstein, P. A. (2001) "A survey of approaches to automatic schema matching". In VLDB Journal 10(4): 334-350.
- [Ratnasamy et al. 2001] Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S. (2001) "A Scalable Content-Addressable Network". In ACM SIGCOMM, pp. 161-172.
- [Rocha et al. 2004] Rocha, J., Domingues, M., Callado, A., Souto, E., Silvestre, G., Kamienski, C., Sadok, D. (2004) "Peer-to-Peer: Computação Colaborativa na Internet". Mini-curso 22^o Simpósio Brasileiro de Redes de Computadores (SBRC'04). Gramado, RS, Brasil.
- [Rouse 2006] Rouse, C. (2006) "Schema Matching in a Peer-to-Peer Database System". Masters Thesis, University of Cape Town, South Africa.
- [Rouse et al. 2006] Rouse, C., Berman, S. (2006) "A Scalable P2P Database System with Semi-Automated Schema Matching". In Proc. of the 26th IEEE International Conference Workshops on Distributed Computing Systems (ICDCSW'06). pp. 78.
- [Rowstron et al. 2001] Rowstron, A., Druschel, P. (2001) "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems". In Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms. pp. 329-350.
- [Schlosser et al. 2003] Schlosser, M., Sintek, M., Decker, S., e Nejd, W. (2003) "HyperCuP - Hypercubes, Ontologies and Efficient Search on P2P Networks". In Proc. of the 1st Workshop on Agents and P2P Computing, Bologna, Italy.
- [Schollmeier et al. 2002] Schollmeier, R. (2001) "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". In Proc. of the 1st International Conference on Peer-to-Peer Computing (P2P '01), Linköping, Sweden, pp.101-102.
- [Sen et al. 2004] Sen, S., Wang, J. (2004) "Analyzing Peer-to-Peer Traffic Across Large Networks". IEEE/ACM Transactions on Networking, 12(2):219-232.

- [Seti 2007] SETI@home. <http://www.setiathome.ssl.berkeley.edu/>, último acesso em 20/5/2008
- [Sheth et al. 1990] Sheth, A. P., Larson, J. (1990) “A federated database system for managing distributed, heterogeneous, and autonomous databases”. *Computing Surveys*, Vol. 22, No. 3, pp. 183-236.
- [Shirky 2000] Shirky, C. (2000) “What is P2P... And What Isn't?” <http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html>, último acesso em 20/5/2008
- [Shvaiko et al. 2005] Shvaiko, P., Euzenat, J. (2005) “A Survey of Schema-Based Matching Approaches”. *Journal on Data Semantics IV*: 146-171.
- [Singh 2001] Singh, M. P. (2001) “Peering at Peer-to-Peer Computing”. *IEEE Internet Computing*, Vol. 5, No. 1.
- [Souza 2007] Souza, D. (2007) “Reformulação de Consultas Baseada em Semântica para PDMS”. Thesis Proposal. Federal University of Pernambuco (UFPE).
- [Sterling et al. 1995] Sterling, T., Savarese, D., Becker, D. J., Dorband, J. E., Ranawake, U. A., Packer, C. V. “BEOWULF: A parallel workstation for scientific computation”. In *Proc. of the 24th International Conference on Parallel Processing, Oconomowoc, WI*. pp. 11-14.
- [Stoica et al. 2001] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H. (2001) “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”. In *Proc. of the ACM SIGCOMM'01, San Diego*, pp.149-160.
- [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.
- [Talia et al. 2003] Talia, D., Trunfio, P. (2003) “Toward a Synergy Between P2P and Grids”. In *IEEE Internet Computing*, 7(4):94-95.
- [Tatarinov et al. 2003] Tatarinov, I., Ives, Z., Madhavan, J., Halevy, A., Suci, D., Dalvi, N., Dong, X., Kadiyska, Y., Miklau, G., Mork, P. (2003) “The Piazza Peer Data Management Project”. In *Proc. of the ACM SIGMOD Record*, 32(3):47-52.
- [Tatarinov et al. 2004] Tatarinov, I., Halevy, A. (2004) “Efficient query reformulation in peer-data management systems”. In *Proc. of SIGMOD Conference*, 539-550.
- [Touch 2001] Touch, J. (2001) “Overlay Networks”. *Computer Networks*, 3(2-3):115-116.
- [Ullman 1997] Ullman, J. D. (1997) “Information Integration Using Logical Views”. In *Proc. of the 6th International Conference on Database Theory (ICDT'97), Delphi, Greece*. pp. 19-40.
- [Valduriez et al. 2004] Valduriez, P., Pacitti, E. (2004) “Data Management in Large-Scale P2P Systems”. In *Proc. of Int. Conf. on High Performance Computing for Computational Science (VecPar'04), Valencia, Spain*.

- [Vazirgiannis et al. 2006] Vazirgiannis, M., Nørnvåg, K., Doulkeridis, C. (2006) “Peer-to-Peer Clustering for Semantic Overlay Network Generation”. In Proc. of the 6th International Workshop on Pattern Recognition in Information Systems (PRIS’06), Paphos, Cyprus.
- [Walkerdine et al. 2002] Walkerdine, J., Melville, L., Sommerville, I. (2002) “Dependability Properties of P2P Architectures”. In Proc. of the 2nd International Conference on Peer to Peer Computing (P2P’02), Linköping, Sweden, pp.173-174.
- [Wang et al. 2005] Wang, J. Z., Ali, F. (2005) “An efficient ontology comparison tool for semantic Web applications”. In Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence, France. pp. 372-378.
- [Wiederhold 1992] Wiederhold, G. (1992) “Mediators in the Architecture of Future Information Systems”. IEEE Computer, 25(3):38–49.
- [WordNet 2007] WordNet: a lexical database for the English language. <http://wordnet.princeton.edu/>, último acesso em 20/5/2008
- [W3C 2004] W3C. RDF/XML Syntax Specification (Revised), 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>, último acesso em 20/5/2008
- [Yang et al. 2002] Yang, B., Garcia-Molina, H. (2002) “Efficient Search in Peer-to-Peer Networks”. In Proc. of CDCS’02, Vienna, Austria.
- [Yang et al. 2003] Yang, B., Garcia-Molina, H. (2003) “Designing a Super-Peer Network”. In Proc. of International Conference on Data Engineering (ICDE’03), Bangalore, India.
- [Zhao et al. 2001] Zhao, B. Y., Kubiawicz, J. D., Joseph, A. D. (2001) “Tapestry: an Infrastructure for Fault-Tolerant Wide-Area Location and Routing”. Technical Report UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley.
- [Zhdanova et al. 2006] Zhdanova, A. V., Shvaiko, P. (2006) “Community-Driven Ontology Matching”. In Proc. of the 3rd European Semantic Web Conference (ESWC’06), Budva, Montenegro.
- [Zhuang et al. 2004] Zhuang, Z., Liu, Y., Xiao, L. (2004) “Dynamic Layer Management in Super-Peer Architectures”. In Proc. of the International Conference on Parallel Processing (ICPP’04), Volume 00, pp. 29-36.