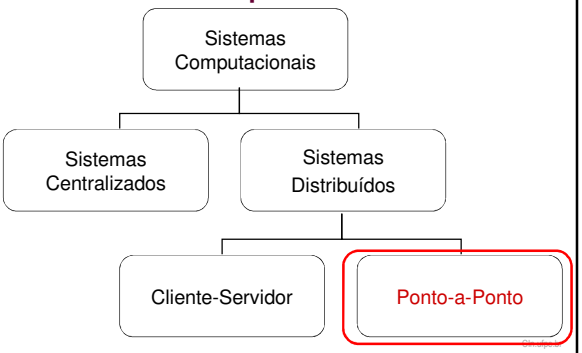
Integração de Dados e Warehousing

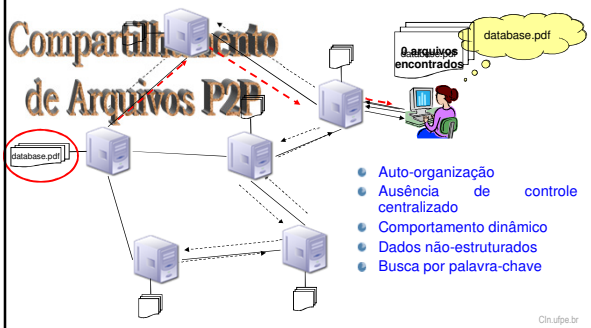
Introdução a PDMS

Fernando Fonseca
Ana Carolina

Classificação dos Sistemas Computacionais



Cenário Ponto-a-Ponto

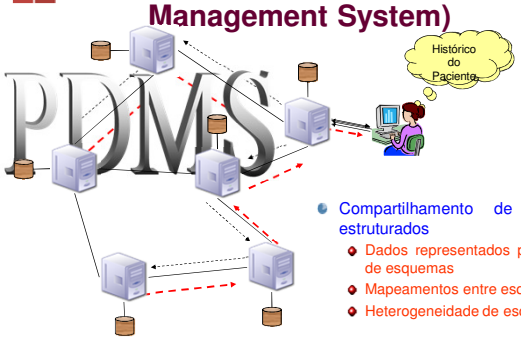


- Auto-organização
- Ausência de controle centralizado
- Comportamento dinâmico
- Dados não-estruturados
- Busca por palavra-chave

Paradigma Ponto-a-Ponto

Compartilhamento de serviços e recursos computacionais diretamente entre sistemas

Cenário PDMS (Peer Data Management System)



- Compartilhamento de dados estruturados
  - Dados representados por meio de esquemas
  - Mapeamentos entre esquemas
  - Heterogeneidade de esquemas

Paradigma Ponto-a-Ponto

Terminologia

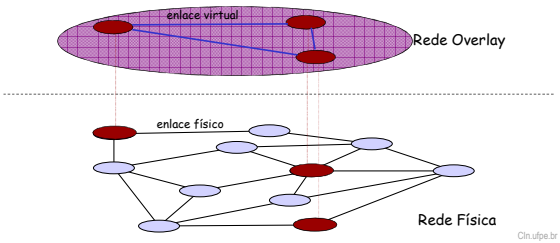
- Peer ≡ Ponto ≡ Nó
  - Componente de uma rede P2P
  - Pode assumir o papel de cliente e servidor
- Cluster
  - Agrupamento de pontos com interesses específicos
    - Exemplo: cluster semântico
- Topologia de rede e localização dos dados
  - Estruturada
  - Não-estruturada

Terminologia

- Serviço
  - Funcionalidades oferecidas pelos pontos
    - Transferência de conteúdo
    - Disponibilização de status
  - Motivação para agrupamento de pontos em uma rede P2P
- Anúncio
  - Forma de comunicar a disponibilidade de um recurso por um ponto

Terminologia

- Rede overlay
  - Rede virtual criada sobre uma rede já existente
  - Não é exatamente igual à rede física



Paradigma Ponto-a-Ponto (P2P)

- Consiste de uma ampla rede de pontos computacionais (ou nós de informação) interconectados que cooperam uns com os outros, trocando serviços e informações
- Cada ponto compartilha recursos com os outros pontos e se beneficia dos recursos dos demais
  - Recursos computacionais
    - Espaço em disco
    - Processamento
  - Recursos de rede
  - Conteúdo

Paradigma Ponto-a-Ponto (P2P)

- Sistemas P2P
  - Pontos possuem relativamente as mesmas características e funções
  - Pontos trocam mensagens por meio dos seus *links* lógicos sem a interferência de um coordenador (ponto servidor)
  - Pontos são organizados em uma rede lógica (*Overlay Network*) no nível da aplicação

Paradigma Ponto-a-Ponto (P2P)

- Principais características
  - Sem coordenação central
  - Sem repositório central
  - Sem local único de falha ou gargalo
  - Nenhum ponto tem visão global do sistema
  - Todos os dados e serviços são acessíveis de qualquer ponto
  - Pontos são autônomos
  - Pontos e conexões não são confiáveis

Sistemas Ponto-a-Ponto (P2P)

- Tipos de Sistemas
  - Não Estruturados
    - Sem restrição de localização dos dados
    - Principal aplicação: compartilhamento de arquivos
    - Busca por palavra-chave
    - Alta disponibilidade de arquivos (réplicas nos pontos)

Sistemas Ponto-a-Ponto (P2P)

- Tipos de Sistemas (Cont.)
  - Estruturados
    - Referenciados como *Distributed Hash Tables (DHT)*
    - Alta escalabilidade
    - Boa cobertura e alta precisão
    - Dois aspectos importantes
      - Busca aos dados
      - Acesso aos dados

} camada virtual de rede *(overlay network)*

Paradigma Ponto-a-Ponto (P2P)

- Vantagens
  - Poder computacional (recursos dos demais pontos)
  - Pontos com diferentes papéis (cliente ou servidor)
  - Compartilhamento de recursos
    - Melhor desempenho, tolerância a falhas (replicação)
  - Autonomia dos pontos participantes
    - Ausência de administração
  - Escalabilidade (e.g. KaZaA com ~3-4 milhões de usuários)

Paradigma Ponto-a-Ponto (P2P)

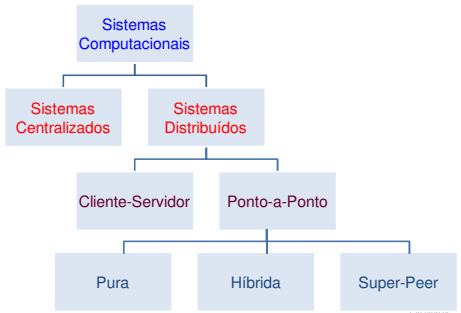
- Desvantagens
  - Ausência de tratamento semântico na troca de dados
  - Problemas com disponibilidade e consistência
  - Falta de estratégia para distribuição dos dados
  - Pode prejudicar o desempenho de pontos
  - Ausência de administração centralizada
  - Usuários responsáveis por gerenciar seus próprios recursos
  - Problemas de Segurança

Topologias de Redes Ponto-a-Ponto

Topologias de Redes P2P

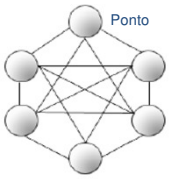
- Topologia
  - Define a organização lógica dos pontos na rede
- Tipos
  - Pura
  - Híbrida
  - *Super-Peer*

Topologias de Redes P2P



Topologia Pura

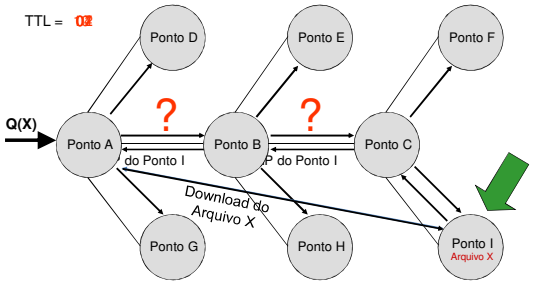
- Inexistência de um servidor ou repositório centralizado
- Todos os pontos são "iguais" e conectados entre si
- Busca
  - Não-Estruturada
    - *Flooding*
    - TTL (*time-to-live*)
  - Estruturada: DHT
- Sistemas: Gnutella, Freenet



Topologia Pura

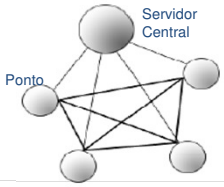
- Responsabilidades do ponto, como cliente
  - Enviar pedidos de serviço a outros pontos
  - Receber as respostas dos pedidos feitos
- Responsabilidades do ponto, como servidor
  - Receber pedidos de serviço de outros pontos
  - Processar os pedidos e executar um serviço requerido
  - Enviar a resposta com os resultados do serviço requerido
  - Propagar os pedidos de serviço a outros pontos

Busca na Topologia Pura Não-Estruturada



Topologia Híbrida

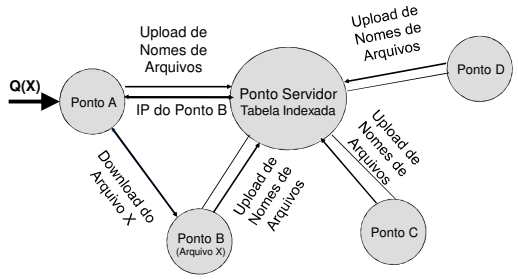
- Existência de um ou mais servidores centrais
- Informações de controle são armazenadas e fornecidas por um servidor central
- Gerência facilitada
- Servidor central representa um ponto único de falha
- Sistema: Napster



Topologia Híbrida

- Responsabilidades do ponto, como cliente
  - Registrar no servidor seus serviços disponíveis
  - Enviar ao servidor pedidos de busca por serviços e receber respostas contendo listas de pontos com os serviços desejados
  - Enviar a outros pontos pedidos de serviço e receber as respostas destes pedidos
  - Processar e executar os serviços requeridos e enviar repostas a quem fez o pedido
- Responsabilidades do ponto, como servidor
  - Registrar serviços disponíveis nos pontos
  - Receber pedidos de busca por serviços disponíveis, buscar por esses serviços e enviar respostas com as localizações dos serviços desejados

Busca na Topologia Híbrida

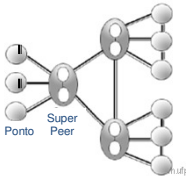


Topologia *Super-Peer*

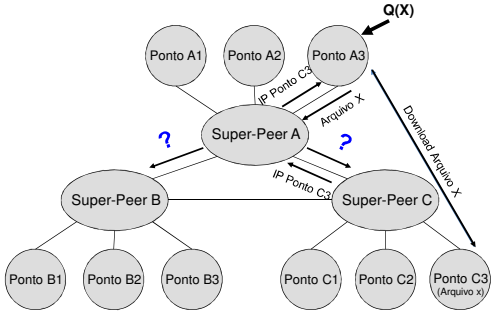
- Considera
  - Heterogeneidade dos pontos
  - Muitos pontos em conexões de baixa capacidade e alta instabilidade
  - Poucos pontos em conexões de alta capacidade e baixa instabilidade

Topologia *Super-Peer*

- Pontos heterogenêneos
- Organização hierárquica
- Grupos de pontos comunicam-se com outros grupos por meio de *super-peers*
  - Cada *super-peer* indexa as informações armazenadas no seu conjunto de pontos
- Sistemas: KaZaA, Morpheus

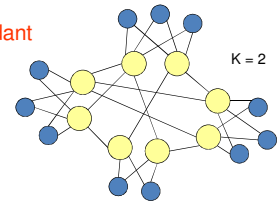


Busca na Topologia *Super-Peer*



Desafios da Topologia *Super-Peer*

- Qual a taxa ideal de pontos por *super-peer*?
- Como os *super-peers* devem conectar-se entre si?
  - Topologia estruturada ou não-estruturada?
- Variação
  - K-redundant



Comparativo entre Topologias

Arquitetura	Segurança (Pontos Maliciosos)	Consistência (Dados)	Escalabilidade (Entrar e Sair)	Confiabilidade (Ponto de Falha)
P2P Pura	👎	👎	👍	👍
P2P Híbrida	👍	👍	👍	👎
Super-Peer	👍	👍	👍	👍

Propriedades dos Sistemas P2P

Principais Propriedades dos Sistemas P2P

- Conectividade
- Auto-organização
- Descentralização
- Escalabilidade
- Roteamento
- …

Conectividade

- *Ad-hoc* e dinâmica
- Envolve
  - Conexão
  - Desconexão (normal, falha)
- Conexão de um ponto na rede
  - Feita por meio de outro que já esteja participando
- Alguns pontos podem atuar como *entry points*
- Pontos relacionados devem ficar "próximos" uns dos outros

Auto-Organização

- Capacidade dos pontos se realocarem na rede após a ocorrência de um evento
  - Conexão
  - Desconexão e/ou Falha
  - *Timeout*
- A inexistência de uma administração centralizada faz com que a reorganização de rede P2P fique ao encargo dos próprios pontos

Descentralização

- Dados e metadados estão distribuídos entre os pontos
- Não existe um servidor central responsável por tarefas como
  - Reorganização da rede
  - Armazenamento de metadados
- Próprios pontos devem ser responsáveis por tais tarefas
- Inexistência de ponto único de falha

Escalabilidade

- Capacidade da rede P2P crescer sem ficar sobrecarregada
- Sistema cliente-servidor
  - Administradores podem estender ou rebalancear os recursos computacionais para compensar o crescimento da rede
- Sistema P2P
  - Soluções devem estar embutidas em cada ponto

Escalabilidade

- Depende da topologia adotada
  - Híbrida
    - Dificuldade em tratar a escalabilidade
    - Pontos centrais podem necessitar de balanceamento e/ou expansão física do *hardware* para compensar o crescimento da rede
    - Preocupação com os custos de manutenção dos pontos centrais
    - Contra-exemplo: Napster mostrou-se robusto e eficiente
  - Pura
    - Sobrecarga de troca de mensagens para descoberta de novos pontos e buscas na rede
  - *Super-Peer*
    - Divisão e/ou fusão (*coalesce*) de *clusters*

Roteamento

- Principais mecanismos de roteamento para redes P2P
  - Híbrido
  - *Flooding* (ou inundação): modelo descentralizado não-estruturado
  - Tabela Hash Distribuída (DHT): modelo descentralizado estruturado
  - *Semantic Overlay Network* (SON)

Roteamento – Flooding

- Problemas
  - Excesso de mensagens
  - Mensagens duplicadas
  - Valor ideal de TTL
    - TTL alto: sobrecarga na rede
    - TTL baixo: nenhum resultado encontrado
- Variações
  - Busca informada: uso de cache local
  - Busca informada com replicação
  - Aprofundamento iterativo: múltiplos valores crescentes para TTL

Roteamento – Modelo DHT

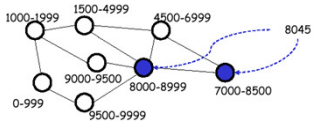
- Tentativa de melhorar os algoritmos de roteamento dos sistemas P2P não-estruturados
- Itens (arquivos) são distribuídos entre os pontos de acordo com um algoritmo
  - Pontos não escolhem os itens à vontade
  - Uso de replicação para garantir disponibilidade

Roteamento – Modelo DHT

- Função hash
  - Mapeia um ponto em um identificador único
    - h('172.17.166.99') → 8400
  - Mapeia um item (arquivo) em um identificador único
    - h('TutorialP2P.ppt') → 8045
  - Qualquer função aleatória de *hash* "boa" é suficiente
    - Padrão SHA-1 (colisão praticamente impossível)
- Faixa de resultados da função *hash* é distribuída pela rede

Roteamento – Modelo DHT

- Cada ponto é responsável por armazenar itens cujo identificador é igual ou próximo ao identificador do ponto

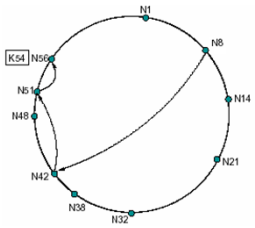


- Dado um identificador, um ponto deve ser capaz de encaminhar a consulta para o ponto cujo identificador mais se aproxima

Roteamento – Modelo DHT

- Para cada objeto, o(s) ponto(s) cuja faixa "cobre" o objeto deve ser alcançável por um caminho "curto"
  - De qualquer outro ponto
- Abordagens
  - Chord, CAN, Pastry, Tapestry, ...
  - Diferem na escolha do algoritmo de roteamento (determina a geometria da rede)
- Geometrias
  - Anel: Chord
  - Árvore: Pastry, Tapestry
  - XOR: Kademlia
  - Hipercubo: CAN
  - Híbrida: Pastry (pode trabalhar como anel)

Roteamento – Modelo DHT



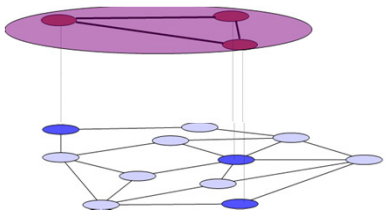
Chord

Roteamento

- Ineficiência de consultas no modelo de inundação (escalabilidade)
- Consultas no modelo DHT
  - Escalonável, porém "pobre"
  - Não permite
    - Consultas por aproximação
    - Consultas por faixa
- Uma consulta deve ser enviada apenas para os pontos aptos a respondê-la
- Em geral, é possível representar o conteúdo compartilhado por meio de ontologias
  - Música, filmes, artigos científicos, …

Roteamento - SON

- Semantic Overlay Network
  - Virtual, abstrata, camada independente de pontos selecionado

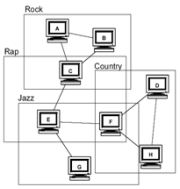


Roteamento - SON

- Vantagens
  - Introduz visões semânticas sobre a rede física
  - Mediação e integração (correspondências, reescrita de consultas)
  - Reduz a quantidade de mensagens na rede

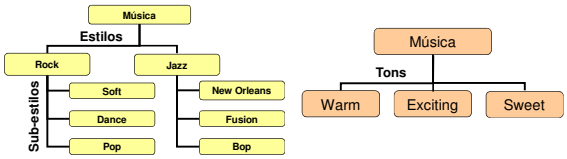
Roteamento - SON

- Pontos agrupados em *clusters*
- *Overlap* de *clusters*
- Consultas enviadas apenas para *clusters* relevantes
- *Clusters* irrelevantes são descartados



Roteamento - SON

- SON associada ao conceito de hierarquia de classificação
- Exemplos
  - 9 SON para classificação de músicas por estilo
  - 4 SON para classificação de músicas por tom



PDMS

Peer Data Management System

Sistemas de Gerenciamento de Dados Ponto-a-Ponto (PDMS)

- Sistema de Gerenciamento de Dados
  - Com arquitetura descentralizada
  - Facilmente extensível
    - Na qual qualquer usuário pode contribuir com
      - Novos dados
      - Novos esquemas
      - Mapeamentos entre os esquemas dos pontos

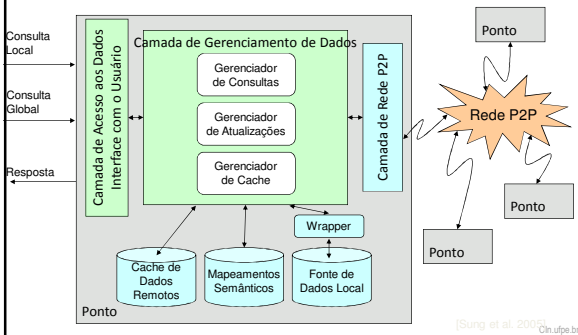
Sistemas de Gerenciamento de Dados Ponto-a-Ponto (PDMS)

São uma evolução natural dos
Sistemas de Integração de Dados
substituindo seu único esquema lógico (mediação) por uma
coleção de mapeamentos semânticos
entre os esquemas individuais de cada ponto

Sistemas de Gerenciamento de Dados Ponto-a-Ponto (PDMS)

- Algumas características
  - Autonomia: controle sobre os dados locais
  - Dinamismo: pontos e recursos podem entrar e sair a qualquer momento
  - Descentralização: cada ponto é independente dos outros
  - Cooperação: compartilhamento de recursos e serviços entre os pontos
  - Esquema local do BD: cada ponto tem seu esquema (ausência de esquema global)
  - Dados: podem estar incompletos, indisponíveis ou inconsistentes

Arquitetura Genérica de um Ponto



[Sung et al. 2005]

Sistemas de Gerenciamento de Dados Ponto-a-Ponto (PDMS)

- Gerenciamento de dados distribuídos
- Compartilhamento de dados em larga escala
- Solução depende fortemente da topologia adotada
- Impraticável a existência de esquema de mediação único

Problemas de um Esquema de Mediação Único

- Conflito com as propriedades dos sistemas P2P
  - Dinamismo
    - Atualização do esquema de mediação a cada conexão e/ou desconexão
  - Autonomia
    - Nem todos os pontos querem compartilhar todos os dados

Problemas de um Esquema de Mediação Único

- Conflito com as propriedades dos sistemas P2P (Cont.)
  - Escalabilidade
    - Onde armazenar um esquema de mediação único?
      - Centralizado
        - Ponto único de falha
        - Investimento em hardware e conectividade
      - Distribuído
        - Técnicas para garantir uma visão integrada do esquema de mediação
        - Esquema replicado: problemas de armazenamento e consistência

Pontos Positivos dos PDMS

- Não existe esquema global
  - Manutenção
- Mapeamentos definidos da forma mais conveniente (pontos "próximos")
- Consultas são elaboradas de acordo com o esquema do ponto
  - Resultados vêm de qualquer lugar do sistema
- PDMS x Compartilhamento de Arquivos
  - Dados possuem semântica mais "rica"
  - Não são tão dinâmicos (conexão/desconexão)

Gerenciamento de Dados em Sistemas P2P

Aspectos do Gerenciamento de Dados em Sistemas P2P

- Mapeamentos entre Esquemas
- Processamento de Consultas
- Consistência de Dados
- Localização de Dados
- Conectividade entre Pontos
- Tolerância a Falhas
- Qualidade de Dados

PDMS – Mapeamentos entre Esquemas

- Mapeamentos
  - Estabelecem relacionamentos entre esquemas
  - Em sistemas de integração de dados são estabelecidos entre o esquema de mediação e as fontes de dados
  - Em PDMS são estabelecidos entre os pontos
  - A qualidade dos mapeamentos possui forte influência no resultado das consultas
- Principais formalismos
  - Global-as-view (GAV)
  - Local-as-view (LAV)

PDMS – Mapeamentos entre Esquemas

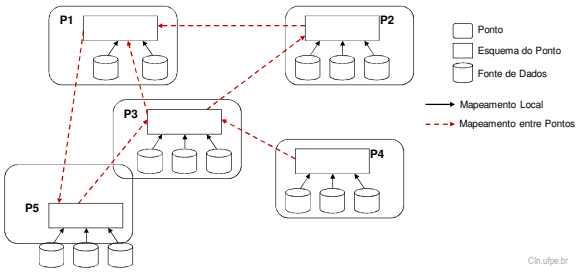
- Mapeamentos em um PDMS
  - Pontos em um PDMS são relacionados através de mapeamentos
  - O conjunto de mapeamentos define a rede semântica de um PDMS
  - A otimização da rede semântica considera
    - Eliminação de mapeamentos redundantes
    - Redução do diâmetro do PDMS (para reduzir a perda de informação na reformulação de consultas)
    - Identificação de pontos semânticos inacessíveis

PDMS – Mapeamentos entre Esquemas

- Mapeamentos em um PDMS
  - Vantagem de um PDMS
    - Quando um novo ponto entra no sistema será preciso fornecer mapeamentos para um pequeno número de pontos já existentes
    - Em termos de esquema, os pontos existentes devem ser similares ao novo ponto
  - Dada uma consulta submetida a um ponto, o sistema reformulará a consulta de acordo com os esquemas dos pontos vizinhos

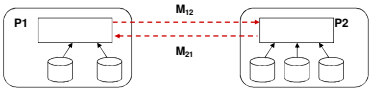
PDMS – Mapeamentos entre Esquemas

- Mapeamentos em um PDMS



PDMS – Mapeamentos entre Esquemas

- Diferentes formalismos podem ser usados para a definição dos mapeamentos (GAV, LAV)
  - Exemplos
    - Mapeamentos locais: LAV
    - Mapeamentos entre pontos: GAV
- Os mapeamentos podem ser direcionados



PDMS – Mapeamento de Dados

- Mapeamentos entre dados são necessários quando seus valores (formatos) são diferentes
- Tabelas de mapeamento podem ser usadas para especificar a correspondência entre valores de atributos
  - Associam valores dentro de um único domínio ou entre domínios
  - Representam o conhecimento de especialistas
  - Ex.:

Para	Origem
Fortaleza	FOR
Porto Alegre	POA
Recife	REC

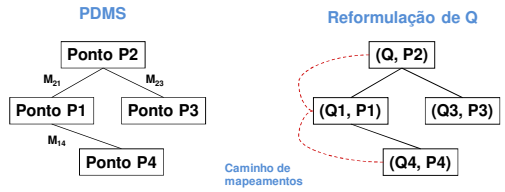
PDMS – Processamento de consultas

- Cada ponto tem um esquema associado
- Pontos são conectados através de "caminhos de mapeamentos"
- Pontos podem ser servidores de dados, mediadores entre pontos e clientes que submetem consultas

PDMS – Processamento de Consultas

1. Uma consulta Q é submetida a um ponto P1 de acordo com o esquema de P1
2. Se P1 possui seus próprios dados, então o PDMS recupera a resposta de Q a partir de P1
3. Em seguida, de acordo com os mapeamentos correspondentes, Q será reformulada para os vizinhos de P1
4. As consultas reformuladas são submetidas aos vizinhos de P1 e assim sucessivamente

PDMS – Processamento de Consultas



PDMS – Processamento de Consultas

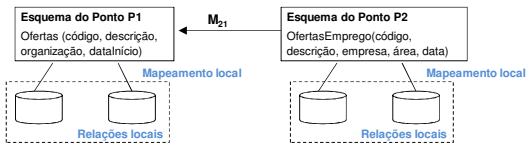
- A técnica de reformulação de consulta dependerá do formalismo para definição dos mapeamentos
  - *View Unfolding* (Abordagem GAV)
  - *View Rewriting* (Abordagem LAV)

PDMS – Processamento de Consultas

- Exemplo
  - Ponto P1
    Ofertas (código, descrição, organização, dataInício)
  - Ponto P2
    OfertasEmprego(código, descrição, empresa, área, data)
  - Mapeamento entre P2 e P1 (M_{21})
    P2.OfertasEmprego(código, descrição, empresa, data) ⊆
    P1.Ofertas (código, descrição, organização, dataInício)

PDMS – Processamento de Consultas

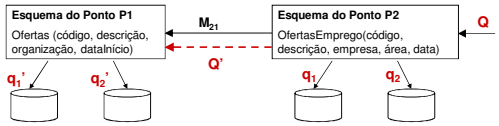
- Exemplo



PDMS – Processamento de Consultas

Exemplo 1

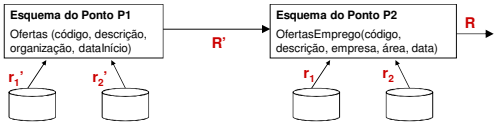
- Consulta submetida em P2:
  - Q ← OfertasEmprego(código, 'Desenvolvimento', 'Microsoft', 'TI', '01/10/2006')
- Passos de execução da consulta:
  1. P2 usa os mapeamentos locais para reformular Q de acordo com as relações locais de P2
  2. P2 passa Q para P1 de acordo com o mapeamento (M_{21})
  3. A reformulação de consultas é *view unfolding*
  4. P1 usa os mapeamentos locais para reformular Q' de acordo com as relações locais de P1



PDMS – Processamento de Consultas

Exemplo 1

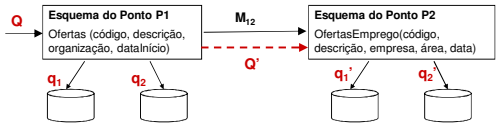
- Passos de execução da consulta:
  5. Como não existem mais mapeamentos o processo de reformulação é finalizado
  6. Em seguida as consultas são avaliadas nas relações locais, seus resultados integrados e retornados para P2



PDMS – Processamento de Consultas

Exemplo 2

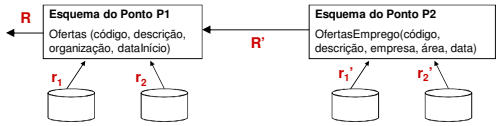
- Consulta submetida em P1:
  - Q ← Ofertas(código, 'Desenvolvimento', 'Microsoft', '01/10/2006')
- Passos de execução da consulta:
  1. P1 usa os mapeamentos locais para reformular Q de acordo com as relações locais de P1
  2. P1 passa Q para P2 de acordo com o mapeamento (M_{12})
  3. A reformulação de consultas é *view rewriting*
  4. P2 usa os mapeamentos locais para reformular Q' de acordo com as relações locais de P2



PDMS – Processamento de Consultas

Exemplo 2

- Passos de execução da consulta:
  5. Como não existem mais mapeamentos o processo de reformulação é finalizado
  6. Em seguida as consultas são avaliadas nas relações locais, seus resultados integrados e retornados para P1

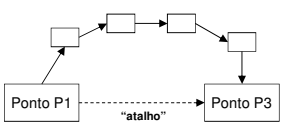


PDMS – Processamento de Consultas

- Em um PDMS um ponto só é acessível através dos mapeamentos
- Com o aumento do número de pontos no sistema o número de esquemas e mapeamentos aumenta
  - Caminhos de mapeamentos muito extensos (muitas reformulações)
  - Perda de semântica à medida que as consultas são repassadas (reformuladas)
  - Restrições de tempo de resposta

PDMS – Processamento de Consultas

- Ontologias podem ser usadas para reduzir o tamanho dos caminhos de mapeamentos
  - Pontos similares devem ser detectados
  - "Atalhos" podem ser definidos entre os pontos



PDMS – Processamento de Consultas

- Otimização
  - Seguir por todos os caminhos leva a ineficiências
    - O algoritmo pode seguir muitos caminhos que resultam em reformulações redundantes (consultas desnecessárias)
    - A aplicação repetitiva de *query unfolding* freqüentemente resulta em consultas redundantes.
  - Evitar a execução de uma consulta de forma redundante – que retorne um subconjunto de resultados de uma consulta executada anteriormente (*query containment*)

PDMS – Consistência de Dados

- Problema que surge em qualquer cenário onde exista a duplicação de dados
- Principais cenários: *caching* e replicação
- Benefícios da abordagem P2P trazem novos desafios
  - *Caching*: garantir que os dados da *cache* de um ponto estejam consistentes com os dados dos seus pontos vizinhos
  - Replicação: a propagação de atualizações nos dados torna-se uma tarefa bastante complexa
    - Grande número de pontos
    - Indisponibilidade dos pontos

PDMS – Localização de Dados

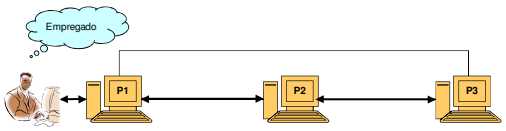
- Difícil de ser prevista
  - Conectividade dinâmica
  - Descentralização (conhecimento parcial)
- Topologia Híbrida
  - Pontos descrevem seus dados durante a conexão
  - Metadados armazenados em um repositório central

PDMS – Localização de Dados

- Topologia Pura
  - Não-Estruturada
    - Descoberta de Recursos
      - Transitividade (mapeamentos)
    - Aproximação de Pontos
      - Estatísticas e probabilidade para aproximação de pontos "distantes"
  - Estruturada
    - *Semantic Overlay Networks*
- Topologia *Super-Peer*
  - Pontos são agrupados em *clusters* (e comunidades)
  - Utilização de índices (SP-P e SP-SP)

PDMS – Localização de Dados

- Topologia Pura Não-Estruturada



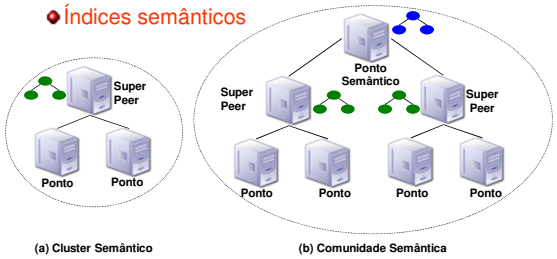
Vizinhos = {P2} P3}
Empregado [P1] = Funcionário [P2]
Empregado [P1] = Emp [P3]

Vizinhos = {P1, P3}
Funcionário [P2] = Empregado [P1]
Funcionário [P2] = Emp [P3]

Vizinhos = {P2} P1}
Emp [P3] = Funcionário [P2]
Emp [P3] = Empregado [P1]

PDMS – Localização de Dados

- Topologia Super-Peer
  - Índices semânticos



(a) Cluster Semântico (b) Comunidade Semântica

PDMS – Conectividade

- Dinâmica e *ad-hoc*
  - Dinamismo é menor do que em outros sistemas P2P, e.g. compartilhamento de arquivos
- Importância da alocação eficiente de pontos
  - Mapeamentos entre esquemas
    - Qualidade dos mapeamentos
  - Processamento de consultas
    - Escolha dos pontos aptos a responderem consultas
- Topologia Pura
  - Definição dos vizinhos iniciais
- Topologia *Super-Peer*
  - *Clusters* e comunidades semânticas para agrupar

PDMS – Tolerância a Falhas

- Substituição de *super-peers*
  - Eleição
- Fusão de *clusters*
- Servidores de *Backup*
  - Alternativa para evitar políticas de substituição
  - Selecionado entre os pontos
  - Cópia periódica dos metatados (*Super-peer* → Servidor de Backup)

PDMS – Qualidade de Dados

- Aspectos de qualidade
  - Dados das fontes
  - Mapeamentos entre esquemas
    - Incompletos
    - Incorretos
  - Plano da Consulta
  - Resultado das consultas

PDMS – Qualidade de Dados

- Alguns Critérios:
  - Disponibilidade
  - Consistência
  - Relevância
  - Completude
    - Extensão: conjunto de objetos
    - Intenção: esquema
  - Cobertura
  - Densidade
  - Tempo de Resposta

} **Web**