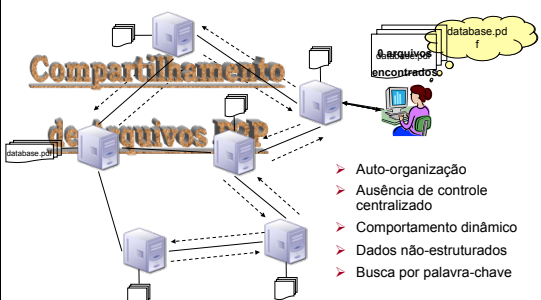


Gerenciamento de Dados em Sistemas P2P XXI SBBD – Florianópolis

Ana Carolina Salgado,
Carlos Eduardo Santos Pires,
e Bernadette Farias Lôscio
(acs,cesp)@cin.ufpe.br,
bernafarias@lia.ufc.br






Cenário Ponto-a-Ponto



Compartilhamento de arquivos P2P

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


 SBBD'06 - Gerenciamento de Dados em Sistemas P2P



Paradigma Ponto-a-Ponto


Compartilhamento de
serviços e
recursos computacionais
diretamente entre sistemas



 SBBD'06 - Gerenciamento de Dados em Sistemas P2P



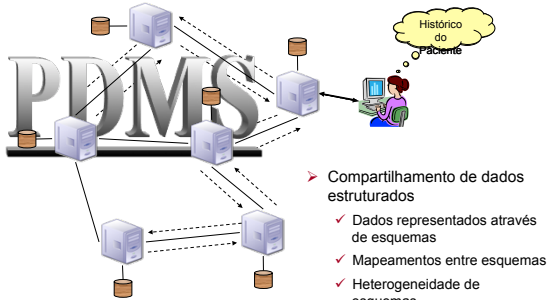
P2P – Questões Importantes

- Como reconhecer um outro ponto na rede?
- Como publicar informações para os demais pontos?
- Como identificar unicamente um ponto?
- Como compartilhar dados na rede?
- Como organizar os dados compartilhados?


 SBBD'06 - Gerenciamento de Dados em Sistemas P2P





Cenário PDMS (Peer Data Management System)



PDMS


- Compartilhamento de dados estruturados
- ✓ Dados representados através de esquemas
- ✓ Mapeamentos entre esquemas
- ✓ Heterogeneidade de esquemas



 SBBD'06 - Gerenciamento de Dados em Sistemas P2P



Novos Desafios

- **Localização dos Dados**
 - ✓ Pontos devem saber localizar dados em outros pontos
- **Processamento de Consultas**
 - ✓ Pontos devem localizar os dados e executar a consulta eficientemente
- **Integração de Dados**
 - ✓ A heterogeneidade da representação dos dados nos vários pontos deve ser resolvida
- **Consistência dos Dados**
 - ✓ A consistência deve ser mantida em caso de replicação e uso de *cache*


 SBBD'06 - Gerenciamento de Dados em Sistemas P2P



Agenda

- Paradigma Ponto-a-Ponto (P2P)
 - ✓ Características
 - ✓ Topologias de Rede
 - ✓ Propriedades dos Sistemas P2P
- PDMS – *Peer Data Management System*
 - ✓ Características
 - ✓ Gerenciamento de Dados em PDMS
 - ✓ Processamento de Consultas
- Desafios

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 7 UFC

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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 9 UFC

Terminologia

- Serviço
 - ✓ Funcionalidades oferecidas os 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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 10 UFC

Terminologia

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

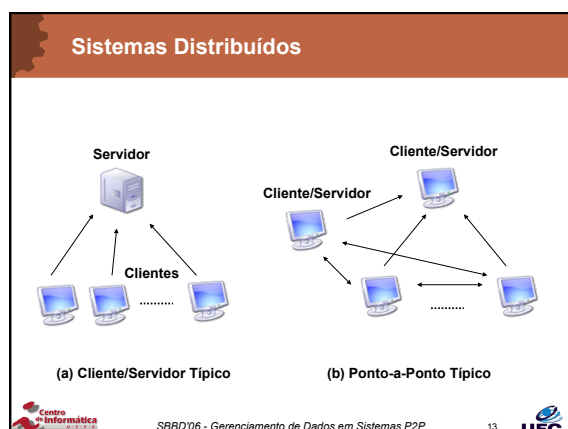
Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 11 UFC

Classificação dos Sistemas Computacionais

```

graph TD
    SC[Sistemas Computacionais] --> SC1[Sistemas Centralizados]
    SC --> SC2[Sistemas Distribuídos]
    SC2 --> CS[Cliente-Servidor]
    SC2 --> P2P[Ponto-a-Ponto]
  
```

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 12 UFC



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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 14 UFC

Paradigma Ponto-a-Ponto (P2P)

➤ **Recursos compartilhados**

- ✓ Recursos computacionais
 - Espaço em disco
 - Processamento
- ✓ Recursos de rede
- ✓ Conteúdo

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 15 UFC

Paradigma Ponto-a-Ponto (P2P)

➤ **Sistemas P2P**

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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 16 UFC

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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 17 UFC

Sistemas Ponto-a-Ponto (P2P)

➤ **Tipos de Sistemas**

- ✓ Não Estruturados
- ✓ Estruturados

➤ **Não Estruturados**

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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 18 UFC

Sistemas Ponto-a-Ponto (P2P)

➤ 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
- ✓ Segurança

Topologias de Redes Ponto-a-Ponto

Topologias de Redes P2P

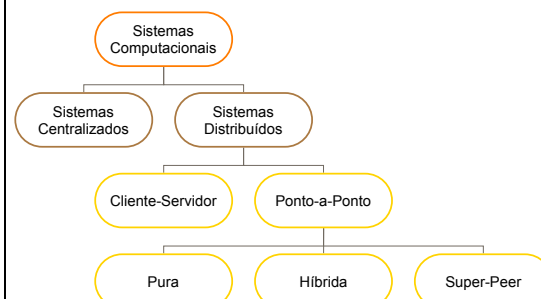
➤ Topologia

- ✓ Define a organização lógica dos pontos na rede

➤ Tipos [Fiorano 2003]

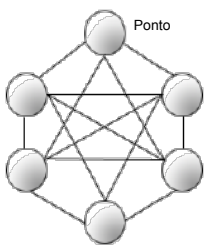
- ✓ Pura
- ✓ Híbrida
- ✓ *Super-Peer*

Topologias de Redes P2P



Topologia Pura

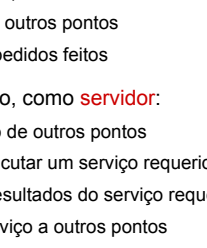
- Inexistência de um servidor ou repositório centralizado
- Todos os pontos são "iguais" e conectados entre si
- Busca
 - ✓ Não-Estruturada
 - Flooding
 - TTL (time-to-live)
 - ✓ Estruturada: DHT
- Sistemas: Gnutella, Freenet



Centro de Informática - Universidade Federal de Pernambuco (UFPE) | SBBD'06 - Gerenciamento de Dados em Sistemas P2P | 25 | UFC

Topologia Pura

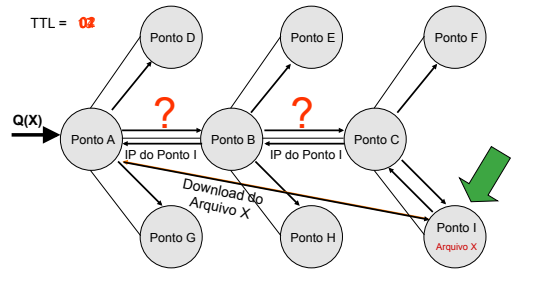
- Responsabilidades do ponto, como **cliente**:
 - ✓ Enviar pedidos de serviço a outros pontos
 - ✓ Receber as respostas dos pedidos feitos
- Responsabilidades do ponto, como **servidor**:
 - ✓ Receber pedidos de serviço de outros pontos
 - ✓ Processar os pedidos e executar um serviço requerido
 - ✓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



Centro de Informática - Universidade Federal de Pernambuco (UFPE) | SBBD'06 - Gerenciamento de Dados em Sistemas P2P | 26 | UFC

Busca na Topologia Pura Não-Estruturada

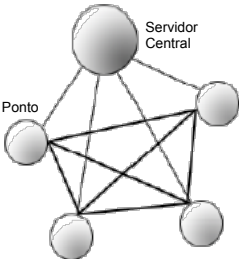
TTL = 02



Centro de Informática - Universidade Federal de Pernambuco (UFPE) | SBBD'06 - Gerenciamento de Dados em Sistemas P2P | 27 | UFC

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



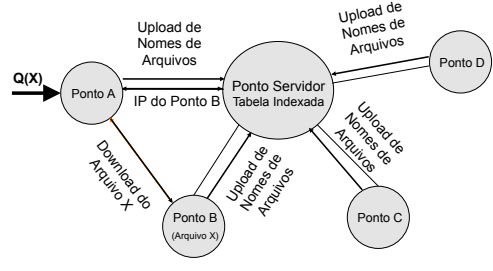
Centro de Informática - Universidade Federal de Pernambuco (UFPE) | SBBD'06 - Gerenciamento de Dados em Sistemas P2P | 28 | UFC

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 requerido 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

Centro de Informática - Universidade Federal de Pernambuco (UFPE) | SBBD'06 - Gerenciamento de Dados em Sistemas P2P | 29 | UFC

Busca na Topologia Híbrida



Centro de Informática - Universidade Federal de Pernambuco (UFPE) | SBBD'06 - Gerenciamento de Dados em Sistemas P2P | 30 | UFC

Topologia Super-Peer

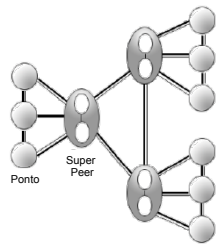
Considera:

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

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 31 UFC

Topologia Super-Peer

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



Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 32 UFC

Busca na Topologia Super-Peer

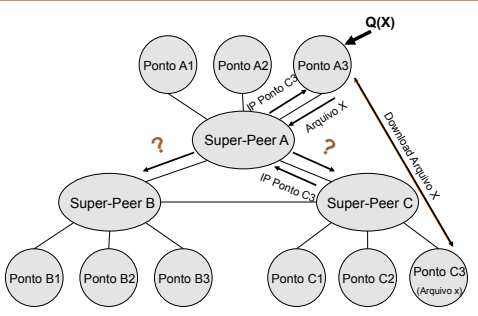
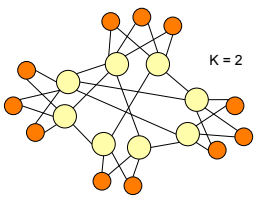


Diagrama de busca na topologia Super-Peer. Um ponto q(x) quer baixar um arquivo X. Ele se conecta a Super-Peer A. Super-Peer A não tem o arquivo, então pergunta para Super-Peer B e Super-Peer C. Super-Peer C indica que o arquivo está em Ponto C3. Super-Peer A então se conecta a Ponto C3 para fazer o download.

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 33 UFC

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



Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 34 UFC

Comparativo entre Topologias

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

[Fiorano 2003]

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 35 UFC

Propriedades dos Sistemas P2P

Principais Propriedades dos Sistemas P2P

- Conectividade
- Auto-Organização
- Descentralização
- Escalabilidade
- Roteamento
- ...

[Walkerdine 2002]

Conectividade

- *Ad-hoc* e dinâmica
- Envolve
 - ✓ Conexão
 - ✓ Desconexão (normal, falha)
- Conexão de um ponto na rede
 - ✓ Feita através 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

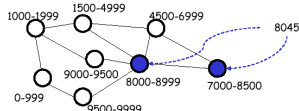
- Tentativa de melhorar os algoritmos de roteamento dos sistemas P2P não-estruturados
- Itens (arquivos) são distribuídos entre os pontos de acordo com um algoritmo
 - ✓ Pontos não escolhem os itens à vontade
 - ✓ Uso de replicação para garantir disponibilidade

Roteamento – Modelo DHT

- **Função hash**
 - ✓ Mapeia **um ponto** em um identificador único
 - $h("172.17.166.99") \rightarrow 8400$
 - ✓ Mapeia **um item** (arquivo) em um identificador único
 - $h("TutorialP2P.ppt") \rightarrow 8045$
 - ✓ 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: Kademia
 - ✓ Hiper-cubo: CAN
 - ✓ Híbrida: Pastry (pode trabalhar como anel)

Roteamento – Modelo DHT

Chord

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 49 UFC

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 através de ontologias
 - ✓ Música, filmes, artigos científicos, ...

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 50 UFC

Roteamento - SON

Semantic Overlay Network

- Virtual, abstrata, camada independente de pontos selecionados

Vantagens

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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 51 UFC

Roteamento - SON

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

[Garcia-Molina et al. 2002]

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 52 UFC

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 ton

Estilos

```

Música
├── Rock
│   ├── Sub-estilos
│   └── Soft
├── Jazz
│   ├── New Orleans
│   ├── Fusion
│   └── Bop
└── Dance
    └── Pop
                    
```

Tons

```

Música
├── Warm
├── Exciting
└── Sweet
                    
```

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 53 UFC

SON: Critérios para Definição da Hierarquia

- Documentos devem ser **associados a conceitos** para que o ponto seja associado a(s) SON correspondente(s)
- Pontos devem possuir documentos em um **número reduzido de categorias**
- **Algoritmo de classificação** eficiente

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 54 UFC

SON: Estratégias para Alocação de Pontos

- Estratégias conservadora
 - ✓ Aloque um ponto em SON_c , se o mesmo possui algum documento classificado no conceito c

↓

Produz muitos links
- Estratégias menos conservadora
 - ✓ Aloque um ponto em SON_p , caso o mesmo possua uma quantidade de documentos "significativa" classificados no conceito c

↓

Evita que sejam encontrados todos os documentos

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 55 UFC

Outras Propriedades de Sistemas P2P

- **Autonomia**
 - ✓ Um ponto não deve ter controle sobre os recursos compartilhados por outro ponto
- **Anonimato**
 - ✓ Capacidade do sistema ocultar a identificação do usuário
- **Descoberta de Recursos**
 - ✓ Os pontos devem comunicar a disponibilidade de recursos através de anúncios

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 56 UFC

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**, ou **mapeamentos** entre os esquemas dos pontos

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 58 UFC

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

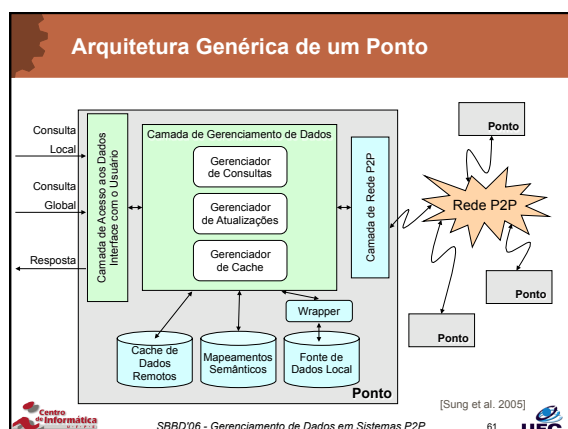
São uma evolução natural dos **Sistemas de Integração de Dados** substituindo seu único esquema lógico (mediação) por uma **coleção de mapeamentos semânticos** entre os esquemas individuais de cada ponto

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 59 UFC

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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 60 UFC



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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 62 UFC

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
- **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

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 63 UFC

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)

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 64 UFC

Pontos Positivos dos PDMS

- **Inexistência de Ponto Único de Falha**
 - ✓ Dados altamente distribuídos usando *cache* local
 - ✓ Se um ponto falha, as consultas podem continuar
- **Pouca administração**
- **Escopo dos dados**
 - ✓ Grande quantidade de dados de diferentes tipos
- **Replicação**
 - ✓ Dados podem ser replicados nos pontos
 - ✓ Resultados de consultas podem ir para *cache* local
 - ✓ Acesso rápido aos dados locais se pontos originais não disponíveis

Centro de Informática
SBBD'06 - Gerenciamento de Dados em Sistemas P2P 65 UFC

Gerenciamento de Dados em PDMS

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

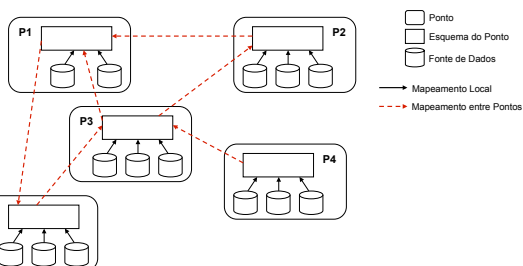
- Mapeamentos em um PDMS
 - ✓ Pontos em um PDMS são relacionados através de mapeamentos
 - ✓ O conjunto de mapeamentos define a rede semântica de um PDMS
 - ✓ A otimização da rede semântica considera
 - Eliminação de mapeamentos redundantes
 - Redução do diâmetro do PDMS (para reduzir a perda de informação 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

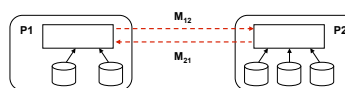
➤ Mapeamentos em um PDMS



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

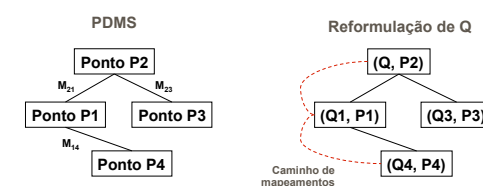
PDMS – Processamento de consultas

- 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})
 - $P2.OfertasEmprego(código, descrição, empresa, data) \subseteq P1.Ofertas(código, descrição, organização, dataInício)$

PDMS – Processamento de Consultas

➤ Exemplo

Esquema do Ponto P1: Ofertas (código, descrição, organização, dataInício)

Esquema do Ponto P2: OfertasEmprego(código, descrição, empresa, área, data)

Mapeamento local

Relações locais

Centro de Informática - UFPA

SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

79

UFC

PDMS – Processamento de Consultas

➤ Exemplo 1

➤ Consulta submetida em P2:

- ✓ $Q \leftarrow \text{OfertasEmprego}(\text{código}, \text{'Desenvolvimento'}, \text{'Microsoft'}, \text{'TI'}, \text{'01/10/2006'})$

➤ 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

Esquema do Ponto P1: Ofertas (código, descrição, organização, dataInício)

Esquema do Ponto P2: OfertasEmprego(código, descrição, empresa, área, data)

Relações locais: q_1' , q_2'

Relações locais: q_1 , q_2

Centro de Informática - UFPA

SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

80

UFC

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

Esquema do Ponto P1: Ofertas (código, descrição, organização, dataInício)

Esquema do Ponto P2: OfertasEmprego(código, descrição, empresa, área, data)

Relações locais: r_1' , r_2'

Relações locais: r_1 , r_2

Centro de Informática - UFPA

SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

81

UFC

PDMS – Processamento de Consultas

➤ Exemplo 2

➤ Consulta submetida em P1:

- ✓ $Q \leftarrow \text{Ofertas}(\text{código}, \text{'Desenvolvimento'}, \text{'Microsoft'}, \text{'01/10/2006'})$

➤ 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_{21})
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

Esquema do Ponto P1: Ofertas (código, descrição, organização, dataInício)

Esquema do Ponto P2: OfertasEmprego(código, descrição, empresa, área, data)

Relações locais: q_1 , q_2

Relações locais: q_1' , q_2'

Centro de Informática - UFPA

SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

82

UFC

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

Esquema do Ponto P1: Ofertas (código, descrição, organização, dataInício)

Esquema do Ponto P2: OfertasEmprego(código, descrição, empresa, área, data)

Relações locais: r_1 , r_2

Relações locais: r_1' , r_2'

Centro de Informática - UFPA

SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

83

UFC

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

Centro de Informática - UFPA

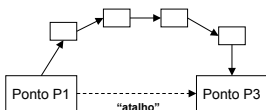
SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

84

UFC

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* frequentemente 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: *cache* 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

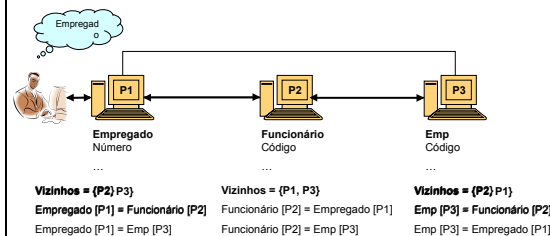
- Difícil de ser prevista
 - ✓ Conectividade dinâmica
 - ✓ Descentralização (conhecimento parcial)
- Topologia Híbrida
 - ✓ Pontos descrevem seus dados durante a conexão
 - ✓ Metadados armazenados em um repositório central

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



PDMS – Localização de Dados

- Topologia Super-Peer
 - ✓ Índices semânticos

(a) Cluster Semântico (b) Comunidade Semântica

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 91 UFC

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

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 92 UFC

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 metadados (*Super-peer* → Servidor de Backup)

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 93 UFC

PDMS – Qualidade de Dados

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

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 94 UFC

PDMS – Qualidade de Dados



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

Centro de Informática SBBD'06 - Gerenciamento de Dados em Sistemas P2P 95 UFC

Desafios

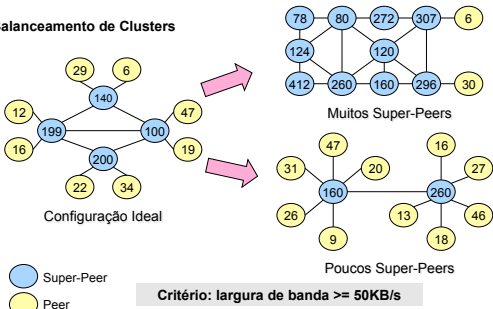
Localização dos Dados



- Balancear:
 - localização dos dados X tempo de resposta
- Diversificar o **tipo de acesso**: não apenas baseado em chaves
- Estratégia para otimização:
 - ✓ armazenar em **locais estratégicos**
 - ✓ usar para **melhorar o desempenho** de consultas

Localização dos Dados

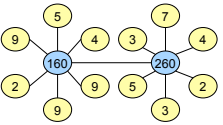
Balanceamento de Clusters





Localização dos Dados

- **Critérios para Balanceamento**
 - ✓ Disponibilidade
 - ✓ Características físicas (CPU, memória, ...)
 - ✓ Tempo de resposta
- Critérios podem ser incompatíveis





Disponibilidade
x
Largura de Banda

- **Problema**
 - ✓ Pontos podem ser promovidos mesmo sem condições físicas ideais



Integração de Dados

- **Mapeamento entre esquemas**
 - ✓ Linguagens para especificação de mapeamentos
 - ✓ Facilidade no gerenciamento de mapeamentos: modelo de representação
 - ✓ Mapeamentos semânticos: continuam um desafio
- **Eficiência e otimização**
 - ✓ Evitar caminhos redundantes (escolha dos melhores)
 - ✓ Propagar atualizações de forma eficiente
 - ✓ Composição de mapeamentos sem perda: se A -> B e B -> C então A-> C



Processamento de Consultas

- **Linguagem de consulta** apropriada
- **Mapeamentos** estruturais e semânticos adequados
- Planos de consultas: **consultas mutantes**
 - ✓ Direcionar para a localização correta, ou
 - ✓ Fornecer os dados diretamente (se sabe como obtê-los)
- **Índices** de roteamento X DHT
- **Regras** de tradução de dados e dependências semânticas
- Avaliação da **qualidade dos resultados** das consultas

Consistência dos Dados

- Oferecer um grau de consistência razoável
- Políticas de replicação e **caching (Freshness)**
 - ✓ Manter a continuidade dos dados (dinamismo dos pontos)
 - ✓ Conservar réplicas consistentes
- Tolerância a falhas
- Segurança

Projeto SPEED

(Semantic PEER Data Management System)


I EPI - 2011

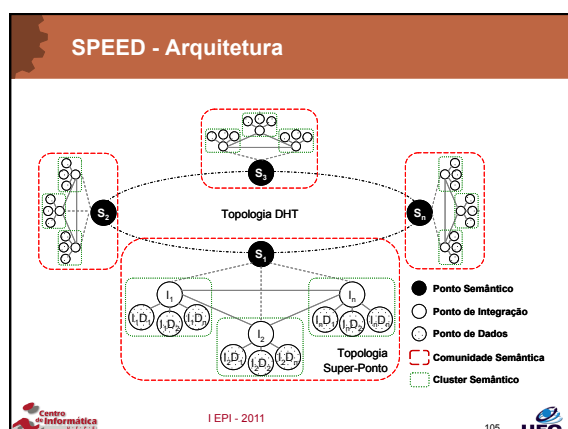
103 

O Sistema SPEED

- O **SPEED - Semantic PEER-to-Peer Data Management System** é um sistema gerenciador de dados para ambiente P2P
- Adota uma **topologia mista**, empregando os conceitos de super-ponto e DHT.
- Utiliza **semântica** como base para o desenvolvimento e gerenciamento de seus serviços
- Representa os esquemas das fontes ligadas aos pontos de dados por **ontologias**

I EPI - 2011


104 



SPEED - Principais Resultados

- Cluster Semântico de Peers
 - ✓ Simulador da Rede P2P
- Reformulação de Consultas entre Peers
 - ✓ Personalizadas
- Operações sobre Ontologias
 - ✓ Matching – SemMatcher
 - ✓ Sumário - OWLSum
 - ✓ Merge


I EPI - 2011

106 

SPEED – Trabalhos em Andamento

- Roteamento de Consultas
- Manutenção dos Clusters Semânticos
 - Balanceamento de carga
 - Tolerância a falhas
- Framework de Gerenciamento de Ontologias
- Qualidade de Informação em PDMS

I EPI - 2011

107 



acs@cin.ufpe.br – Ana Carolina Salgado
 cesp@cin.ufpe.br – Carlos Eduardo Pires
 bernafarias@lia.ufc.br – Bernadette Lôscio
 www.cin.ufpe.br/~cesp/tutorialSBBD

SBB'D'06 - Gerenciamento de Dados em Sistemas P2P

108 

Referências Bibliográficas [PDMS]

- Sung, L. G. A., Ahmed, N., Blanco, R., Li, H., Soliman, M. A., and Hadaller, D. (2005) "A Survey of Data Management in Peer-to-Peer Systems". School of Computer Science, University of Waterloo.
- Valdúriez, P., Pacitti, E. (2004) "Data Management in Large Scale P2P Systems", Proc of VECPAR
- Halevy, H.Y., Ives, Z.G. (2005) "Schema Mediation for Large-Scale Semantic Data Sharing", VLDB Journal, V. 14, N.1

Referências Bibliográficas [Integração de Dados]

- WIEDERHOLD, G. (1992) "Mediators in the Architecture of Future Information Systems". IEEE Computer, 25(3):38-49.
- Lenzerini, M. (2004) "Principles of Peer-to-Peer Data Integration", citeseer.ist.psu.edu/lenzerini04principles.html
- Ruzzi, M. (2004) "Data Integration: state of the art, new issues and research plan, Dipartimento di Informatica e Sistemistica.

Referências Bibliográficas [P2P]

- Walkerdine, J., Melville, L. and 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.
- Milošević, 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.
- Rocha, J., Domingues, M., Callado, A., Souto, E., Silvestre, G., Kamiński, C., Sadok, D. (2004) "Peer-to-Peer: Computação Colaborativa na Internet". Mini-curso 22º Simpósio Brasileiro de Redes de Computadores. Gramado, RS.

Referências Bibliográficas [Conectividade]

- Li, J. and 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) Vol. 1, p. 278
- Zhuang, Z., Liu, Y., and Xiao, L. (2004) "Dynamic Layer Management in Super-Peer Architectures". In Proc. of the International Conference on Parallel Processing (ICPP'04) - Volume 00 p. 29-36

Referências Bibliográficas [Topologias]

- Backx, P., Wauters, T., Dhoedt, B., and Demeester, P. (2002) "A Comparison of Peer-to-Peer Architectures". In Proc. of the Eurescom Summit.
- Fiorano. (2003) "Super-Peer Architectures for Distributed Computing". White Paper, Fiorano Software, Inc.

Referências Bibliográficas [GridxP2P]

- Talia, D. and Trunfio, P. (2003) "Toward a Synergy Between P2P and Grids". In IEEE Internet Computing, Vol. 07, No. 4, pp. 96, 94-95.
- Foster, I. (2002) "The Grid: A New Infrastructure for 21st Century Science". Physics Today, 55 (2). 42-47.
- Foster, I. and 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.

Referências Bibliográficas [SON]

- Garcia-Molina, H., Crespo, A. (2002) "Semantic Overlay Networks for P2P Systems". Technical Report, Stanford University.
- Vazirgiannis, M., Nørnvåg, K., and Doukeridis, 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-2006, co-located with ICEIS 2006), Paphos, Cyprus.

SON: Funcionamento

