Arquitetura Peer-to-Peer e Comunicação em Grupos

Sistemas Distribuídos e Mobile

Prof. Me. Gustavo Torres Custódio
gustavo.custodio@ulife.com.br

Arquitetura Peer-to-Peer e Comunicacao em Grupos

Comunicação de Grupos

Comunicação de Grupos

Arquiteturas Peer-to-Peer

Comunicação de Grupos

- **Grupo**

Comunicação de Grupos

- **Grupo**
  - Uma coleção de processos que agem juntos em um sistema, de forma que quando uma mensagem é enviada para o grupo, todos os membros a recebem.

Formas de Comunicação

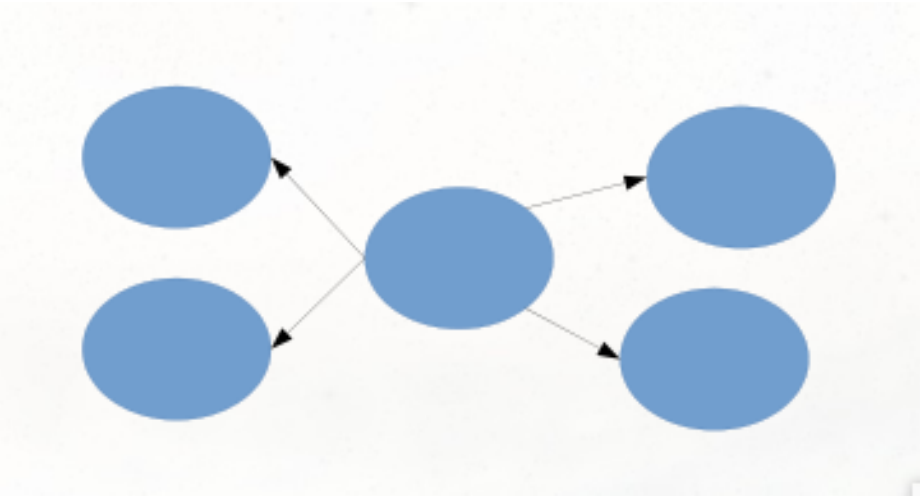
- Um para um (ponto a ponto);

Formas de Comunicação

- Um para um (ponto a ponto);
- Um para vários.

Ponto a Ponto

Um para Vários

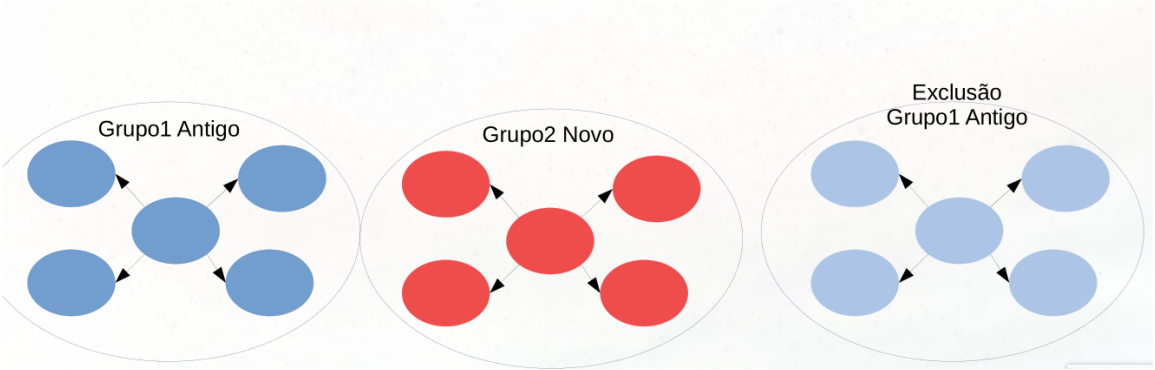


Comunicação de Grupos

- Grupos são dinâmicos;

Comunicação de Grupos

- Grupos são dinâmicos;
  - Novos grupos podem ser criados e grupos existentes podem ser eliminados.

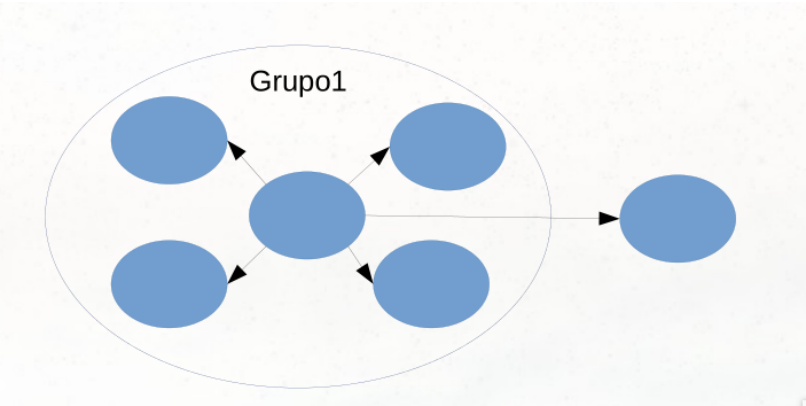


Comunicação de Grupos

- Grupos são dinâmicos;

Comunicação de Grupos

- Grupos são dinâmicos;
  - Um processo pode entrar para o grupo ou deixá-lo.

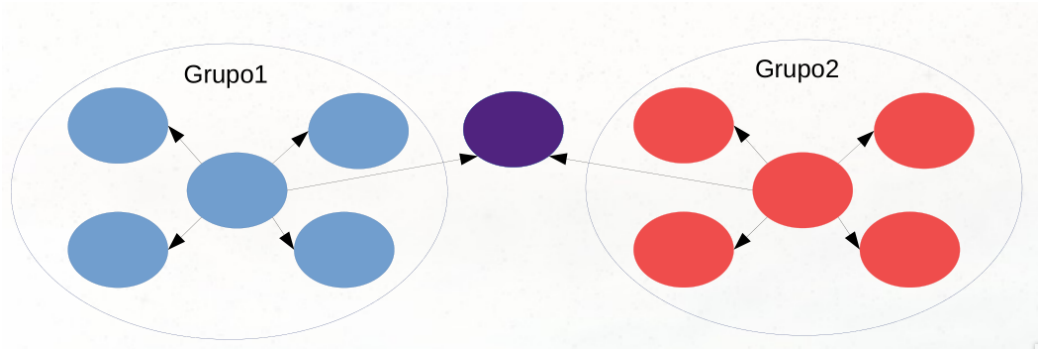


Comunicação de Grupos

- Grupos são dinâmicos;

Comunicação de Grupos

- Grupos são dinâmicos;
  - Um processo pode ser membro de diversos grupos simultaneamente.



Endereçamento

- *Multicasting*:

Endereçamento

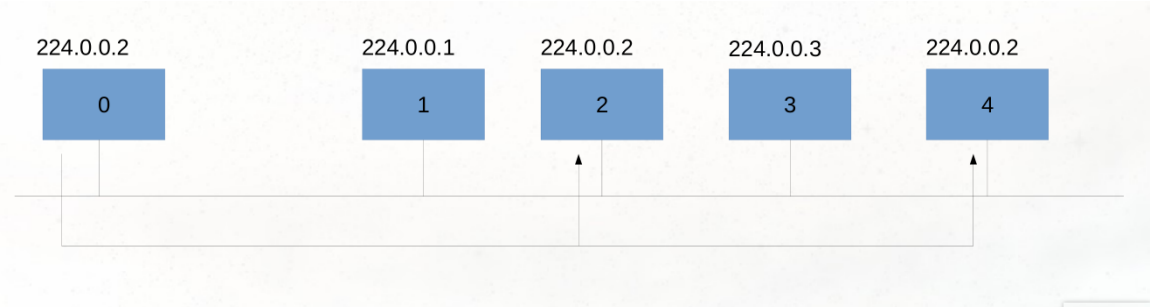
- *Multicasting*:
  - Endereço especial que múltiplas máquinas podem receber (224.x.x.x - 224.x.x.x).
  - Implementação é direta - basta atribuir a cada grupo um endereço *multicasting* diferente.
  - Envia/Recebe mensagem todos que possuem aquele endereço do grupo.

Endereçamento

- *Multicasting*:

Endereçamento

- *Multicasting*:



Endereçamento

- *Broadcasting*:

Endereçamento

- *Broadcasting*:
  - Pacotes contendo certos endereços são enviados para todas as máquinas.
  - Menos eficiente do que *multicasting*.
    - todas as máquinas recebem as mensagens enviadas por *broadcasting*.
  - o software precisa verificar se o pacote é para essa máquina.

Endereçamento

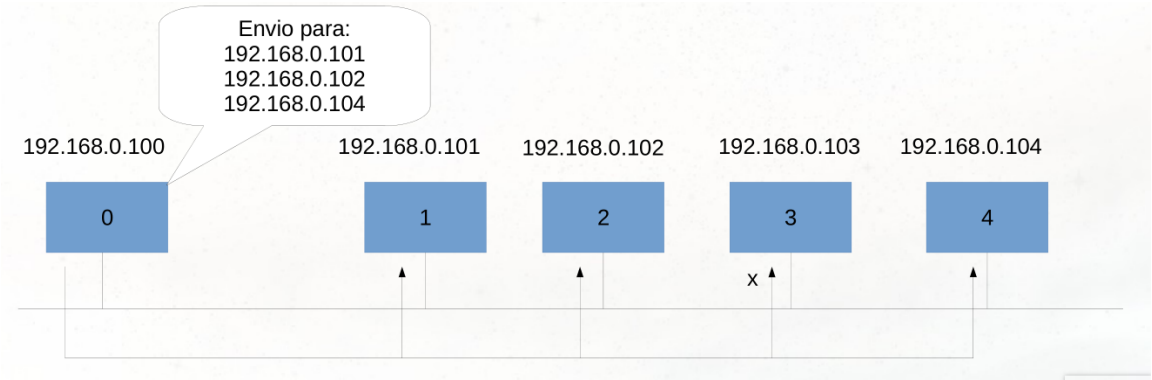
- *Broadcasting*:
  - Pacotes contendo certos endereços são enviados para todas as máquinas.
  - Menos eficiente do que *multicasting*.
    - todas as máquinas recebem as mensagens enviadas por *broadcasting*.
  - o software precisa verificar se o pacote é para essa máquina.
- Somente um pacote é necessário para atingir todos os membros do grupo.

Endereçamento

- *Broadcasting*:

Endereçamento

- *Broadcasting*:



Endereçamento

- *Unicasting*:

Endereçamento

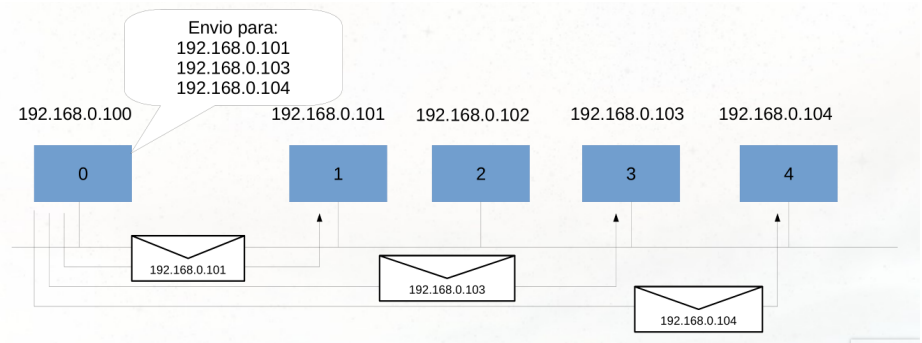
- *Unicasting*:
  - Transmissão separada de pacote para cada membro do grupo.
  - n membros, n pacotes necessários.

Endereçamento

- *Unicasting*:

Endereçamento

- *Unicasting*:



Tipos de Grupos

- Grupo Fechado;

Tipos de Grupos

- Grupo Fechado;
- Grupo Aberto;

Tipos de Grupos

- Grupo Fechado;
- Grupo Aberto;
- Grupo Hierárquico;

Tipos de Grupos

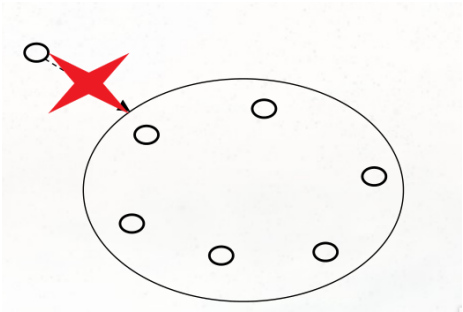
- Grupo Fechado;
- Grupo Aberto;
- Grupo Hierárquico;
- Grupo de Iguais.

Grupo Fechado

- Não permitem que processos fora do grupo enviem mensagens para os membros.

Grupo Fechado

- Não permitem que processos fora do grupo enviem mensagens para os membros.



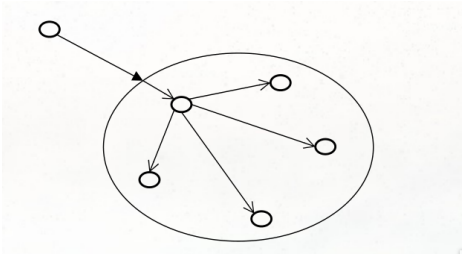
- Exemplo: processamento paralelo

Grupo Aberto

- Permitem que processos externos ao grupo (processos que não pertencem ao grupo) possam enviar mensagens para os membros do grupo.

Grupo Aberto

- Permitem que processos externos ao grupo (processos que não pertencem ao grupo) possam enviar mensagens para os membros do grupo.



- Exemplo: servidores replicados

Grupo Hierárquico

- Possui um **coordenador** que realiza decisões a respeito dos membros.

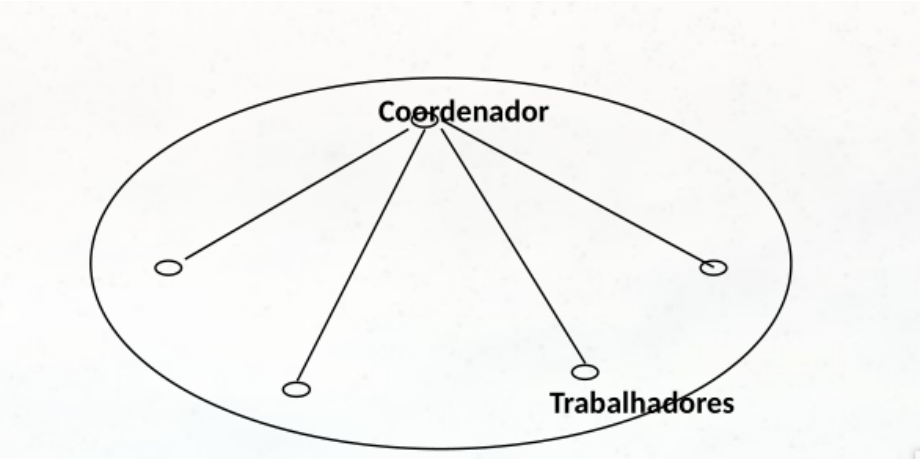
Grupo Hierárquico

- Possui um **coordenador** que realiza decisões a respeito dos membros.
- Vantagem:
  - decisões mais rápidas.

Grupo Hierárquico

- Possui um **coordenador** que realiza decisões a respeito dos membros.
- Vantagem:
  - decisões mais rápidas.
- Desvantagem:
  - a perda do coordenador interrompe o grupo (ponto único de falha).

Grupo Hierárquico



Grupo Hierárquico

- Servidor de Grupo:

Grupo Hierárquico

- Servidor de Grupo:
  - Recebe todas as requisições e mantém um banco de dados completo sobre todos os grupos e seus membros.

Grupo Hierárquico

- Servidor de Grupo:
  - Recebe todas as requisições e mantém um banco de dados completo sobre todos os grupos e seus membros.
  - Se o servidor cair, o gerenciamento é interrompido.
    - Único ponto de falha.

Grupos de Iguais

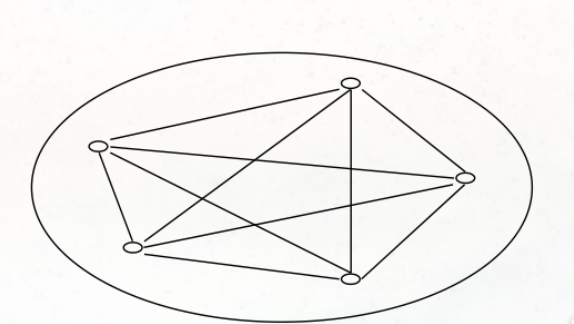
- Gerenciamento Distribuído:

Grupos de Iguais

- Gerenciamento Distribuído:
  - Todos os membros do grupo fazem a mesma coisa.
  - Sistema completamente distribuído.

Grupos de Iguais

- Gerenciamento Distribuído:
  - Todos os membros do grupo fazem a mesma coisa.
  - Sistema completamente distribuído.



Problemas - Grupos

- Quando um processo deixa o grupo, ele envia uma mensagem de "*goodbye*" para todos os outros membros:

Problemas - Grupos

- Quando um processo deixa o grupo, ele envia uma mensagem de "*goodbye*" para todos os outros membros:
- Problema:

Problemas - Grupos

- Quando um processo deixa o grupo, ele envia uma mensagem de "*goodbye*" para todos os outros membros:
- Problema:
  - Se um membro falha, ele deixa o grupo sem o "*goodbye*".
  - Os outros membros têm que descobrir isso de alguma forma e removê-lo do grupo.

Atomicidade

- O que é?
  - Quando uma mensagem é enviada para um grupo, ela chega corretamente para todos os membros ou não chega para nenhum.

Atomicidade

- Uma maneira de ter certeza que todos os destinatários receberam a mensagem é implementar o envio do ACK para cada mensagem recebida.

Atomicidade

- Uma maneira de ter certeza que todos os destinatários receberam a mensagem é implementar o envio do ACK para cada mensagem recebida.
  - ACK = sinal enviado quando todos os dados são recebidos.
  - Se não houver falhas, esse método funciona.

Atomicidade

- Com falhas:

Atomicidade

- Com falhas:
  - Processo que enviar mensagem inicia temporizadores e envia retransmissões quando necessário.

Atomicidade

- Com falhas:
  - Processo que enviar mensagem inicia temporizadores e envia retransmissões quando necessário.
  - Quando um processo recebe uma mensagem que ainda não viu, envia mensagem para todos os membros do grupo.
    - problema: sobrecarga.

Ordenando Mensagens

- *Global Time Ordering* (GTO)

Ordenando Mensagens

- *Global Time Ordering* (GTO)
  - todas as mensagens chegam na ordem exata em que foram ordenadas:
    - não é fácil de ser implementado.

Ordenando Mensagens

- *Consistent Time Ordering* (CTO)

Ordenando Mensagens

- *Consistent Time Ordering* (CTO)
  - se duas mensagens são enviadas em tempos próximos, o sistema escolhe uma como a "primeira" e enviar a todos os membros do grupo segundo esta ordem.
    - é garantido que as mensagens cheguem a todos os membros do grupo na mesma ordem; que podem não ser a ordem real.

Overlapping Groups

- Um processo (ou um conjunto de processos) pode pertencer a vários grupos simultaneamente.

Overlapping Groups

- Um processo (ou um conjunto de processos) pode pertencer a vários grupos simultaneamente.
- Problema:
  - pode se encontrar inconsistências na questão de ordenação de mensagens.

Overlapping Groups

- Mensagens podem chegar em ordens diferentes, mesmo implementando-se GTO.

Overlapping Groups

- Mensagens podem chegar em ordens diferentes, mesmo implementando-se GTO.
- Isso acontece por conta de sincronização de relógios nos diversos grupos.
  - diferenças mínimas de horários interferem no funcionamento do sistema.

Exercícios

- Pense na lógica para a implementação dos grupos:

Exercícios

- Pense na lógica para a implementação dos grupos:
  (a) Grupo aberto;
  (b) Grupo fechado;
  (c) Grupo Hierárquico.

Exercícios

- Pense na lógica para a implementação dos grupos:
  (a) Grupo aberto;
  (b) Grupo fechado;
  (c) Grupo Hierárquico.
  - Dicas:
    - O que será necessário?
    - Quais estruturas serão necessárias?
    - Como será feita a comunicação - pensando no tipo de endereçamento?

Arquitetura Peer-to-Peer e Comunicacao em Grupos

Arquiteturas Peer-to-Peer

Conceito

- Aplicações *peer-to-peer* distribuem a informação entre seus nós membros em lugar de concentrar tudo em um servidor único.

Conceito

- Aplicações *peer-to-peer* distribuem a informação entre seus nós membros em lugar de concentrar tudo em um servidor único.
- Não há a necessidade de nenhum elemento coordenador ou centralizador de recursos ou políticas.

Conceito

- Aplicações *peer-to-peer* distribuem a informação entre seus nós membros em lugar de concentrar tudo em um servidor único.
- Não há a necessidade de nenhum elemento coordenador ou centralizador de recursos ou políticas.
- Existe um certo grau de anonimato para o proprietário do recurso.

Conceito

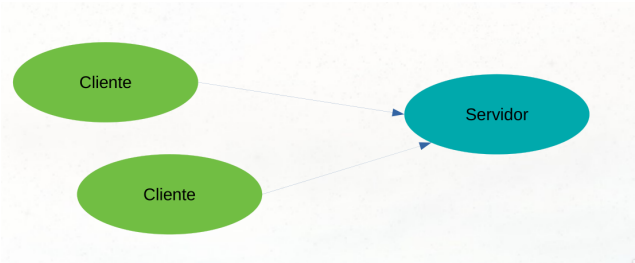
- Todos os nós membros possuem a mesma capacidade de compartilhar informação com os demais membros da rede (todos seriam ao mesmo tempo clientes e servidores de dados).

Conceito

- Todos os nós membros possuem a mesma capacidade de compartilhar informação com os demais membros da rede (todos seriam ao mesmo tempo clientes e servidores de dados).
- Cada usuário torna seu repositório de informações disponível para distribuição e pode estabelecer conexão direta com outro usuário.

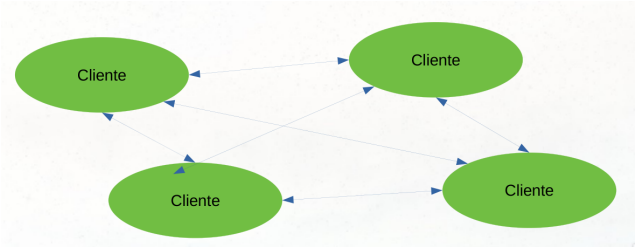
Cliente/Servidor

- No modo de operação cliente/servidor tem-se o acesso aos dados e aos índices centralizados no servidor.



Peer-to-Peer Puro

- No modo de operação puramente P2P, tanto os dados quanto os índices são distribuídos.



Requisitos Peer-to-Peer

- Escalabilidade Global.

Requisitos Peer-to-Peer

- Escalabilidade Global.
  - Imensas quantidades de *hosts* conectados à rede.
  - Milhares de objetos e dezenas de milhares de *hosts*.

Requisitos Peer-to-Peer

- Escalabilidade Global.
  - Imensas quantidades de *hosts* conectados à rede.
  - Milhares de objetos e dezenas de milhares de *hosts*.
- Balanceamento de carga.

Requisitos Peer-to-Peer

- Escalabilidade Global.
  - Imensas quantidades de *hosts* conectados à rede.
  - Milhares de objetos e dezenas de milhares de *hosts*.
- Balanceamento de carga.
  - Distribuição igualitária entre os *peers*;
  - Possibilidade de download de diferentes *peers*, em função de sua carga.

Requisitos Peer-to-Peer

- Otimização das interações locais entre *peers* vizinhos.

Requisitos Peer-to-Peer

- Otimização das interações locais entre *peers* vizinhos.
  - A ideia é buscar vizinhos mais “próximos”, evitando a latência de comunicação.

Requisitos Peer-to-Peer

- Otimização das interações locais entre *peers* vizinhos.
  - A ideia é buscar vizinhos mais "próximos", evitando a latência de comunicação.
- Dinamicidade dos hosts.

Requisitos Peer-to-Peer

- Otimização das interações locais entre *peers* vizinhos.
  - A ideia é buscar vizinhos mais "próximos", evitando a latência de comunicação.
- Dinamicidade dos hosts.
  - Peers podem entrar e sair do sistema a qualquer momento.
  - Quando entram, devem ser integrados ao sistema global.
  - Quando saem (voluntariamente ou não), o próprio sistema deve detectar e adequar a nova carga.

Requisitos Peer-to-Peer

- Segurança dos dados em um ambiente heterogêneo.

Requisitos Peer-to-Peer

- Segurança dos dados em um ambiente heterogêneo.
  - Autenticação, criptografia, necessidade de membros da “Rede P2P”.

Requisitos Peer-to-Peer

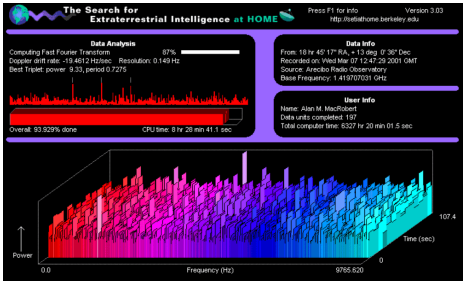
- Segurança dos dados em um ambiente heterogêneo.
  - Autenticação, criptografia, necessidade de membros da “Rede P2P”.
- Anonimato, capacidade de negação e resistência à censura.

Requisitos Peer-to-Peer

- Segurança dos dados em um ambiente heterogêneo.
  - Autenticação, criptografia, necessidade de membros da “Rede P2P”.
- Anonimato, capacidade de negação e resistência à censura.
  - Capacidade de negar o compartilhamento de um arquivo.
  - Possibilidade de não realizar download de conteúdo protegido.

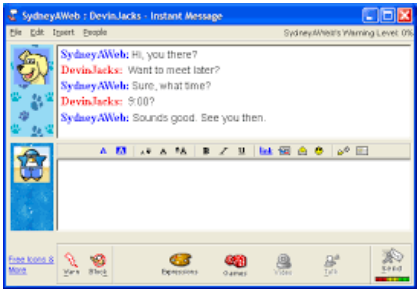
Aplicações

- Aplicações de Computação Distribuída muitas vezes se enquadram na categoria P2P tal como WETIhome que utiliza milhões de clientes Internet para procurar de vida extraterrestre.
  - **setiathome.ssl.berkeley.edu**.



Aplicações

- Aplicações colaborativas também costumam ser consideradas na categoria P2P.
  - "*Instant Messenger*" e salas de chat.



Aplicações

- Nas aplicações colaborativas existe interação entre clientes em torno de uma atividade comum que podem ser jogos ou simulações.

Aplicações

- Nas aplicações colaborativas existe interação entre clientes em torno de uma atividade comum que podem ser jogos ou simulações.
  - "*White Board*" - Aplicação onde cada cliente pode alterar desenhos ou textos e todos os demais visualizam e podem também fazer alterações.

Aplicações

- Outras aplicações nesta categoria incluem sistemas para:

Aplicações

- Outras aplicações nesta categoria incluem sistemas para:
  - modelagem financeira;
  - bioinformática;
  - teste de desempenho Web.

Aplicações

- Outras aplicações nesta categoria incluem sistemas para:
  - modelagem financeira;
  - bioinformática;
  - teste de desempenho Web.
- Estes sistemas aproveitam o tempo ocioso da máquina dos clientes para realizar computações de forma distribuída.

Estudo de Caso: Napster

- Primeiro sistema *peer-to-peer* a ser altamente popularizado;
- Troca exclusiva de músicas, principalmente mp3.

Estudo de Caso: Napster

- Primeiro sistema *peer-to-peer* a ser altamente popularizado;
- Troca exclusiva de músicas, principalmente mp3.

Napster

- Funciona usando uma arquitetura centralizada.

Napster

- Funciona usando uma arquitetura centralizada.
- Servidos de índica, que concentra todas as pesquisas.

Napster

- Funciona usando uma arquitetura centralizada.
- Servidos de índica, que concentra todas as pesquisas.
  - Cada *peer*, ao ser iniciado, torna-se um servidor de arquivos:
    - Exporta seus índices para o servidor central do Napster.
    - Mantém a lista de todos os peers disponíveis.

Napster

- Cliente que deseja realizar uma pesquisa envia a *query* ao Servidor Central do Napster.

Napster

- Cliente que deseja realizar uma pesquisa envia a *query* ao Servidor Central do Napster.
- Este identifica o *peer* que contém a música com as palavras-chave da busca.

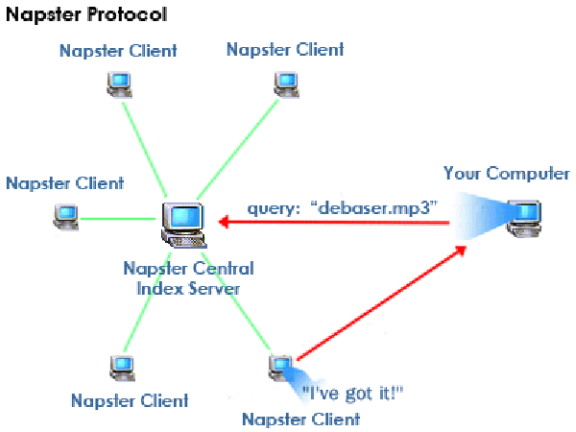
Napster

- Cliente que deseja realizar uma pesquisa envia a *query* ao Servidor Central do Napster.
- Este identifica o *peer* que contém a música com as palavras-chave da busca.
- A troca é feita entre os clientes.

Napster

- Cliente que deseja realizar uma pesquisa envia a *query* ao Servidor Central do Napster.
- Este identifica o *peer* que contém a música com as palavras-chave da busca.
- A troca é feita entre os clientes.
- O servidor do Napster funciona como um “Binder”.

Napster


Napster Protocol
Napster Client
Napster Client
Napster Client
Your Computer
query: "debaser.mp3"
Napster Central
Index Server
Napster Client
"I've got it!"
Napster Client

Estudo de Caso: GNUtella

Estudo de Caso: GNUtella

- Funciona no padrão P2P puro:
  - Alta disponibilidade.
  - Alta dispersão.
  - Praticamente impossível de ser eliminada.

Estudo de Caso: GNUtella

- Cada *peer* funciona como:

Estudo de Caso: GNUtella

- Cada *peer* funciona como:
  - Cliente;
  - Servidor;
  - Gateway - realizando “forward” das mensagens.

Estudo de Caso: GNUtella

- Cada *peer* funciona como:
  - Cliente;
  - Servidor;
  - Gateway - realizando “forward” das mensagens.
- Busca?
  - TTL - *Time to Live*.

GNUtella

- Cada *peer* conhece, pelo menos, 1 *peer* vizinho.
- A consulta é feita pelo *peer* de origem e a ela é atribuída um TTL (em geral 6 ou 7).
- O *peer* vizinho realiza a consulta localmente e a encaminha para seus vizinhos, incrementando o TTL.
- A consulta é propagada até que o TTL atinja seu limite e, então, conforme os resultados são colhidos, a resposta é enviada.
- A partir do momento em que o *peer* de origem encontra um *peer* que possui o arquivo, a troca é realizada.

GNUtella

Figure 2:Gnutella's Topology

GNUtella

- Questionamentos:
  - Legalidade do uso do software;
  - Compartilhamento de conteúdo protegido por Direitos Autorais.

GNUtella

- Questionamentos:
  - Legalidade do uso do software;
  - Compartilhamento de conteúdo protegido por Direitos Autorais.
- Comparação Napster $\times$ Gnutella:
  - Napster - centralizado, sem servidor, sem serviço.
  - Gnutella - Distribuído.
    - Ausência de coordenação global.
    - Altíssima disponibilidade.

Exercícios

- Projete uma arquitetura peer-to-peer para um sistema distribuído que necessita fazer um backup a cada 1 hora em todas as máquinas presentes na rede. Considere que a rede possui 8 máquinas.

Exercícios

- Projete uma arquitetura peer-to-peer para um sistema distribuído que necessita fazer um backup a cada 1 hora em todas as máquinas presentes na rede. Considere que a rede possui 8 máquinas.
  - Considere que a rede possui 8 máquinas.
  - Use o draw.io

Referências

- Sistemas Distribuídos: Princípios e Paradigmas - Tanenbaum
  - Cap 13 - Sistemas distribuídos baseados em coordenação
- Sistemas Distribuídos: Conceitos e Projetos - Coulouris
  - Cap 6 - Comunicação Indireta
  - Cap 10 - Sistemas Peer-to-peer

Conteúdo

https://gustavotcustodio.github.io/sdmobile.html

Obrigado

gustavo.custodio@ulife.com.br