

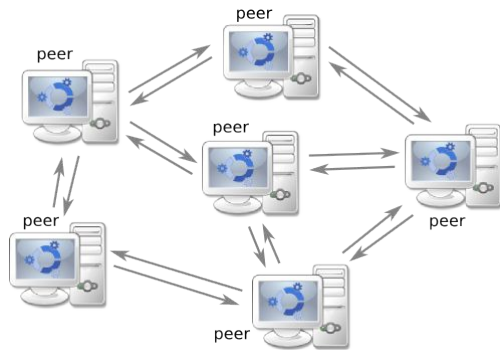
# Sistemas Distribuídos

Peer-To-Peer

## Peer to peer – What ?

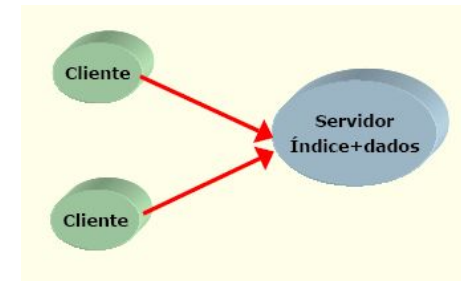


## Peer-To-Peer



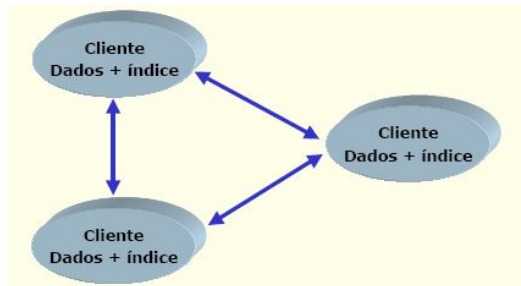
## Cliente - Servidor

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



## Peer to Peer puro

- No modo de operação puramente P2P tanto os dados quanto o índice são distribuídos.



## Peer to Peer

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

## Requisitos de 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
- Otimização das interações locais entre peers vizinhos
  - Idéia é buscar vizinhos mais “próximos”, evitando a latência da comunicação

## Requisitos de Peer to Peer

- 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
- 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 *SETI@home* que utiliza milhões de clientes Internet para procura de vida extraterrestre. É implementado como um *screen saver* (setiathome.ssl.berkeley.edu)
- 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.

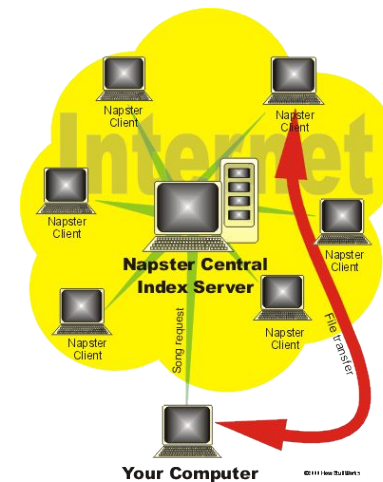
## Aplicações

- Aplicações colaborativas também costumam ser consideradas na categoria P2P. Entre estas aplicações se incluem os “Instant Messenger” e salas de chat.
- Nas aplicações colaborativas existe interação entre clientes em torno de uma atividade comum que podem ser jogos ou simulações.
- Um exemplo é o “White Board” que é uma aplicação onde cada cliente pode alterar desenhos ou textos e todos os demais visualizam e podem também fazer alterações.

## Estudo de Caso: Napster

- 1o sistema Peer to Peer a ser altamente popularizado
- Troca exclusiva de músicas, principalmente em formato MP3
- Funciona usando uma arquitetura centralizada
  - Servidor de índice, que concentra todas as pesquisas
  - Cada peer, ao ser iniciado, torna-se um servidor de arquivos
    - Exporta seus índices ao servidor central do Napster
    - Mantém a lista de todos os Peers disponíveis

## Napster



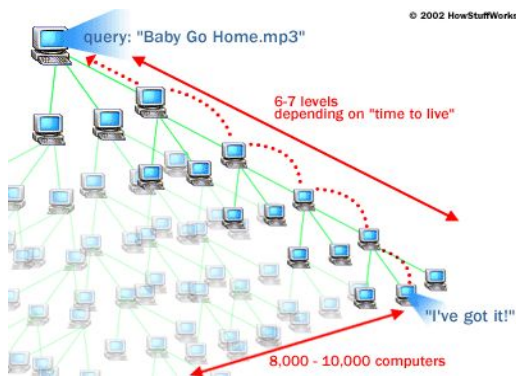
## Napster

- O 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-chaves da busca
- A troca é feita entre os clientes
- Servidor do Napster funciona como um “Binder”

## Estudo de caso 2: Rede GNUtella

- Funciona no padrão P2P puro
  - Alta disponibilidade
  - Alta dispersão
  - Alto nível de balanceamento de carga
  - Praticamente impossível de ser eliminada
- Não há garantia de que o arquivo exista em um dos N peers atingíveis a partir do peer de origem
- Cada peer funciona como:
  - Cliente
  - Servidor
  - Gateway – realizando “forward” das mensagens
- Busca?
  - TTL – Time To Live

## Rede Gnutella



## Rede 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 até 6 ou 7)
- O peer vizinho realiza a consulta localmente e encaminha a consulta 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.



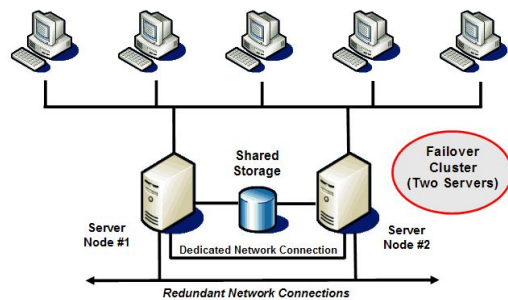
## Rede GNUTella

- Questionamentos
  - Legalidade do uso do software
  - Compartilhamento de conteúdo protegido por Direitos Autorais
- Comparação Napster X Gnutella
  - Napster – Centralizado
    - Ausência do servidor central – ausência do serviço
  - GNUTella – Distribuído
    - Critérios de busca distribuídos entre os diversos nós do sistema
    - Ausência de coordenação global
    - Altíssima disponibilidade

## Cluster

## Cluster: Definição

- De forma geral, é um aglomerado de máquinas conectadas em uma rede local ou dedicadas.



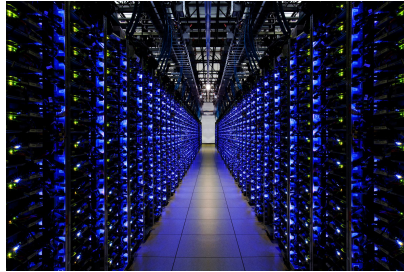
## Exemplos



## Cluster

- Características:

- Hardware
- Software
- Autenticação
- Recursos



- NOWs (Network of Workstations) algumas vezes não são consideradas clusters.
- No nosso contexto, consideraremos cluster como qualquer aglomerado de máquinas em rede local com serviços básicos de rede (ssh, nfs, etc)

## Cluster

- Nível de usuário:

- políticas de utilização
- gerenciadores de recursos
- interface com o usuário

- Nível de administração:

- gerência de hardware
- gerência de software

## Cluster: Exemplos

- **Cluster para Alta Disponibilidade**

- A Alta Disponibilidade se refere a sistemas que praticamente não param de funcionar. Usados em aplicações de missão crítica, eles costumam ter meios eficientes de proteção e de detecção de falhas;

- **Cluster para Balanceamento de Carga**

- Balanceamento de Carga se refere à distribuição equilibrada de processamento aos nós do cluster. É muito usado na Internet, em servidores de e-mail, comércio eletrônico e em sistemas de lojas.

## Cluster: Exemplos

- **Cluster para Processamento de Alto Desempenho**

- Neste tipo de cluster, a principal finalidade é o processamento de aplicações computacionalmente custosas.

- **Cluster para Distribuição dos Dados**

- Neste tipo de cluster, os dados ficam distribuídos entre os nós do cluster. Pode-se implementar redundância, tolerância a falhas, etc.

# GRID Computing

26

## Grid: Definição

- Tipo de sistema paralelo e distribuído que permite o compartilhamento, seleção e agregação de recursos autônomos geograficamente distribuídos dinamicamente e em tempo de execução.

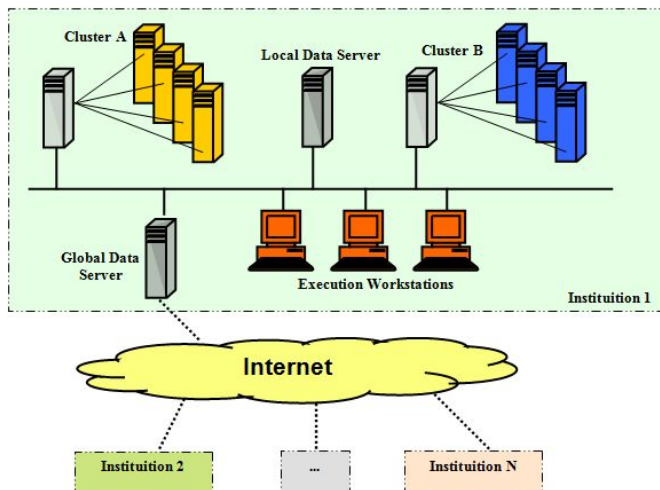
## Grid: Definição

- De acordo com Ian Foster, um Grid é um sistema que:
  - Coordena recursos que não estão sujeitos a um controle centralizado.
  - Utiliza protocolos e interfaces padronizados, abertos e de propósito geral.

## Grid: Requisitos

- Um sistema Grid precisa ser:
  - Confiável
  - Persistente
  - Barato
  - Pervasivo

## Grid: Exemplo de Arquitetura



Prof. Emerson Paduan: emerson@paduan.pro.br

## Diferença entre Cluster e Grid

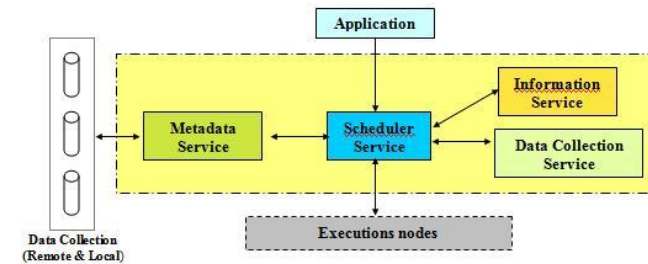
- A maneira como os recursos são gerenciados
  - Em Cluster há um gerenciador de recursos centralizado que aloca todos os recursos do cluster e assim todos os nós trabalham conjuntamente.
  - Em Grid cada nó possui seu próprio gerenciador de recursos e não tem como objetivo prover a visão de que faça parte de um só sistema.

Prof. Emerson Paduan: emerson@paduan.pro.br

## Diferença entre Cluster e Grid

- Conectividade
- Heterogeneidade
- Dispersão geográfica
- Compartilhamento
- Múltiplos domínios administrativos
- Controle distribuído

## Grid: Exemplo de Integração de Serviços





## Grids

- Nível de usuário:
  - políticas de utilização
  - gerenciadores de recursos locais
  - Gerenciadores de recursos globais
  - Monitoração
  - Autenticação
  - Certificação
  - interface com o usuário
- Nível de administração (local e global):
  - gerência de hardware
  - gerência de software

## Grids: Necessidades

- Gerenciador de Recursos
- Comunicação
- Serviço de Informação
- Serviço de Segurança
- Escalonamento
- Balanceamento de Carga

## Grids: Ferramentas Utilizadas

- Hoje em dia, as principais ferramentas open-source utilizadas são:
- Globus Toolkit:
  - Gerenciamento dos Recursos
  - Autenticação
  - Serviços de Segurança
- Condor
  - Escalonamento
  - Balançamento de Cargas

## Plataformas de computação paralela e distribuída

- Execução eficiente de aplicações intensivas em dados ou computação
- Tipos de ambientes:
  - HPC (High Performance Computing)
  - HTC (High Throughput Computing)
- Exs de apps HPC: meteorologia, processamento matemático em geral
- Exs de apps HTC: bioinformática, finanças, etc