

# CONCEITOS

## Sistemas Distribuídos

Eduardo Furlan Miranda

Adaptado de: PEREIRA, C. S. Sistemas Distribuídos.  
Londrina: EDE SA, 2019. ISBN 978-85-522-1443-4.

# Sistemas Distribuídos: Conceitos, Classificação e Sincronização

2/11

## Entendendo a Base da Interconectividade e Robustez

- Contexto: No cenário tecnológico atual, sistemas computacionais estão cada vez mais interligados em rede.
- Relevância: É fundamental compreender como esses sistemas funcionam, suas classificações e a importância da sincronização para garantir a operação correta de aplicações distribuídas.
- Problemas comuns: Um exemplo de problema é a sincronização de relógios, que pode causar inconsistências como horários equivocados em e-mails.

# O Que São Sistemas Distribuídos?

- Definição: um sistema distribuído é um conjunto de computadores que são interligados via rede, mas que, para o usuário final das aplicações, aparenta ser um sistema único.
  - Pode parecer uma única máquina ou um único software.
- Funcionamento Independente: os computadores que fazem parte desses sistemas têm funcionamento independente, agindo por si próprios.
  - Frequentemente, os sistemas e hardwares dessas máquinas podem ser totalmente diferentes (heterogêneos).
- Motivação Principal: o desejo de compartilhar recursos.
  - Recursos abrangem desde componentes de hardware (discos, impressoras) até entidades de software (arquivos, bancos de dados, objetos de dados).
- Assimile: um sistema distribuído é formado por vários nós/máquinas que executam uma função, de modo que, para o usuário final, aparentam ser uma única máquina.

# Exemplos Cotidianos de Sistemas Distribuídos

4/11

- Acesso à Internet: Ao abrir um navegador e acessar uma página, você já está usando um sistema distribuído.
- Serviços Comuns: As redes são a base de muitos serviços cotidianos que consideramos naturais:
  - Internet e World Wide Web.
  - Pesquisa na web.
  - Jogos online.
  - E-mails.
  - Redes sociais.
  - E-Commerce.
- Jogos Multiplayer Online: A maioria utiliza a arquitetura cliente-servidor e são exemplos de sistemas distribuídos.
  - Exemplo: Em um jogo de guerra online, o jogador X tem seu avatar, uma imagem autoritária (cópia do servidor) e um terceiro avatar (cópia enviada aos outros jogadores), ilustrando a arquitetura cliente-servidor.

# Arquitetura Cliente-Servidor e Middleware

- Arquitetura Cliente-Servidor: É a mais utilizada em sistemas distribuídos.
  - Apresenta recursos compartilhados (hardware e software) para permitir que milhares de clientes acessem e utilizem esses recursos como se houvesse uma comunicação direta.
- Robustez dos Sistemas Distribuídos: Sistemas distribuídos podem ser considerados uma solução mais robusta em comparação com sistemas puramente de rede.
- Middleware: Parte dessa robustez é alcançada graças a um componente conhecido como middleware.
  - Definição: É uma camada de software situada entre as aplicações e o sistema operacional.
  - Sua função é facilitar a comunicação e o gerenciamento em ambientes distribuídos.

# Classificação: Computação em Cluster

- Definição: Um conjunto de máquinas com hardware semelhantes, ou seja, possuem características homogêneas.
  - As máquinas são ligadas por rede local (LAN).
- Características de Software:
  - Geralmente, o sistema operacional entre as máquinas é equivalente.
  - Um único programa funciona de forma paralela, subdividido em partes menores, executado em cada nó do cluster.
  - Objetivo: Obter aumento significativo de desempenho e executar tarefas em menos tempo.
- Acoplamento: As máquinas são geralmente fortemente acopladas em suas ligações.
  - Muitas vezes, podem até compartilhar a mesma memória RAM entre várias máquinas.
- Nó Mestre: Há sempre uma máquina chamada nó mestre ou máquina principal.
  - Funções do nó mestre: Gerencia o funcionamento da aplicação entre todos os nós, faz a interface com o usuário, aloca tarefas e administra a fila de tarefas.

# Classificação: Computação em Grid

- Definição: Um conjunto de máquinas com características diferentes, ou seja, possuem características heterogêneas.
  - O hardware e os sistemas operacionais podem ser de fabricantes distintos.
- Estrutura: Essencialmente, um sistema de computação em grid interliga vários clusters.
  - Pode ser visto como um conjunto de dois ou mais clusters, cada um responsável por um tipo de pesquisa.
- Finalidade: Criados para executar diferentes tarefas, de certa maneira relacionadas entre si.
  - Formam um centro de pesquisas de caráter multidisciplinar.
- Exemplo: O CineGrid é um exemplo de grid, trabalhando no desenvolvimento de ferramentas colaborativas multimídia e interligando centros de pesquisa em várias partes do mundo.

# Principais Diferenças: Cluster vs. Grid

Característica	Computação em Cluster	Computação em Grid
Homogeneidade	Homogênea: Máquinas com hardware e, na maioria das vezes, sistemas operacionais semelhantes.	Heterogênea: Máquinas com hardware e sistemas operacionais diferentes, podendo ser de fabricantes distintos.
Finalidade	Criados para executar tarefas específicas que necessitam de alto poder de processamento e levariam muito tempo em um computador convencional (ex: treinamento de redes neurais).	Criados para executar diferentes tarefas relacionadas, formando centros de pesquisas multidisciplinares. Interliga vários clusters.
Acoplamento	Geralmente fortemente acoplados, podendo compartilhar memória RAM.	Menos acoplados, atuando como uma rede de recursos mais distribuída.



# A Importância da Sincronização de Relógios

9/11

- Problema Real: É comum receber um e-mail de resposta com um horário anterior ao e-mail original, ilustrando um problema de sincronização de relógios.
- Necessidade Crucial: Para o funcionamento correto de uma aplicação distribuída, os relógios de todas as máquinas devem estar sincronizados.
- Impacto na Operação:
  - Serviços como acesso remoto e autenticação de usuários podem não funcionar adequadamente caso haja uma diferença muito grande no horário entre as máquinas cliente e servidora.
  - Pode levar a inconsistências de dados e falhas operacionais.
- Solução: A sincronização horária é uma das maneiras mais utilizadas e populares para sistemas formados por múltiplos computadores sincronizarem suas ações.
  - O protocolo Network Time Protocol (NTP) é a base para essa sincronização.

# Network Time Protocol (NTP)

- Propósito: Protocolo mais utilizado para a sincronização do relógio de máquinas locais (desktops, notebooks, servidores) e demais dispositivos de rede.
- Funcionamento: Utiliza o User Datagram Protocol (UDP), operando na porta 123.
- Referência Horária: A referência horária é dada por sistemas de altíssima precisão, como os relógios atômicos.
- Hierarquia de Servidores (Estratos):
  - Estrato 0: Os sistemas de alta precisão (relógios atômicos).
  - Estrato 1: Servidores conectados diretamente a relógios atômicos.
  - Estrato 2: Servidores conectados aos de estrato 1, formando uma segunda camada.
  - Esta hierarquia se estende até o estrato 15.
- Cálculo da Latência: O NTP é projetado para verificar o atraso (latência) entre a máquina cliente e o servidor.
  - A máquina cliente consulta um servidor NTP para verificar seu atraso (ou adiantamento) em relação à referência horária, usando timestamps ( $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ ) para ajustar precisamente o relógio local.

# Sincronização de Relógios na Prática

- Configuração no Windows 10:
  - Abrir o Prompt de Comando (CMD) como Administrador.
  - Inserir o comando para apontar para o servidor NTP: `w32tm /config /syncfromflags:manual /manualpeerlist:0.pool.ntp.org`.
  - Reiniciar o serviço de data e hora: `net stop w32time net start w32time`.
  - Forçar uma sincronização: `w32tm /resync /rediscover`.
    - Uma mensagem confirmará a conclusão bem-sucedida.
- Configuração no Linux (Ubuntu 18.04):
  - Pelo Terminal, editar o arquivo de configuração NTP com o comando: `sudo nano /etc/ntp.conf`.
  - Inserir as informações do servidor NTP (ex: `pool.ntp.br`) adicionando a linha: `pool pool.ntp.br`.
  - Reiniciar o serviço NTP: `sudo service ntp restart`.
    - Isso sincroniza as informações de data e hora na máquina Linux.