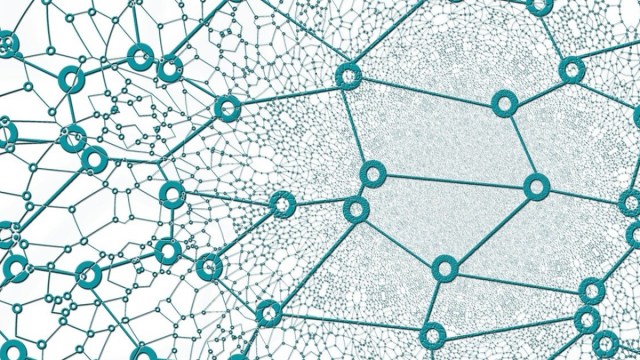


Licenciatura em Engenharia Multimédia

Computação na Nuvem

**A computação distribuída, princípios gerais do funcionamento**



Viorel Cojocari nº 21467

Vasco Sarrico nº 21459

Vanessa Brioso nº 21466

Professor Pedro Almeida

Turma Pós-Laboral

Ano letivo 2021 / 2022

Índice

[I. Introdução 3](#_Toc131615092)

[Definição de computação distribuída 4](#_Toc131615093)

[Importância da computação distribuída 5](#_Toc131615094)

[II. Arquiteturas de sistemas distribuídos 6](#_Toc131615095)

[Redes ponto a ponto 7](#_Toc131615096)

[Sistemas cliente-servidor 8](#_Toc131615097)

[Sistemas em cluster 9](#_Toc131615098)

[III. Compartilhamento de recursos em sistemas distribuídos 10](#_Toc131615099)

[Uso de vários computadores para processamento e armazenamento de dados 11](#_Toc131615100)

[Comunicação entre computadores 11](#_Toc131615101)

[IV. Segurança e confiabilidade em sistemas distribuídos 12](#_Toc131615102)

[Replicação de dados 13](#_Toc131615103)

[Deteção e tratamento de falhas 14](#_Toc131615104)

[V. Protocolos de comunicação em sistemas distribuídos 15](#_Toc131615105)

[Sincronização de tarefas 15](#_Toc131615106)

[Transmissão e recepção de dados 17](#_Toc131615107)

[VI. Conclusão 18](#_Toc131615108)

[Benefícios da computação distribuída 19](#_Toc131615109)

[Desafios e oportunidades futuras 20](#_Toc131615110)

# I. Introdução

## Definição de computação distribuída

Computação distribuída é um modelo de processamento de dados em que um sistema computacional é composto por múltiplos componentes autónomos, como computadores, dispositivos móveis, servidores, entre outros, que trabalham juntos de forma coordenada para realizar uma tarefa computacional.

Neste modelo, cada componente do sistema é responsável por uma parte da computação, e a comunicação e coordenação entre eles é realizada por meio de uma rede de comunicação, como a internet. Esta abordagem permite que tarefas complexas sejam divididas em partes menores, que podem ser processadas em paralelo, aumentando assim a eficiência e a velocidade de processamento.

A computação distribuída é utilizada em diversas áreas, como processamento de grandes volumes de dados, inteligência artificial, sistemas de controle de tráfego aéreo, entre outras aplicações que requerem alta capacidade de processamento e coordenação entre múltiplos componentes.

## Importância da computação distribuída

Algumas das principais razões pelas quais a computação distribuída é importante são:

1.     Escalabilidade: Isto significa que é possível aumentar a capacidade de processamento de um sistema sem ter que substituí-lo completamente por um novo sistema, o que pode ser mais caro e complexo.

2.     Redundância: A computação distribuída também permite que o sistema tenha redundância em caso de falhas em algum componente. Isso significa que, se um componente falhar, outros componentes do sistema podem continuar processando a tarefa, garantindo assim a continuidade do processamento.

3.     Eficiência: A computação distribuída permite que tarefas sejam divididas em partes menores, que podem ser processadas em paralelo, aumentando a eficiência e a velocidade de processamento. Isso é especialmente importante em tarefas que requerem alto poder de processamento, como inteligência artificial, simulações complexas e análise de grandes volumes de dados.

4.     Flexibilidade: A computação distribuída permite que diferentes componentes do sistema tenham diferentes funções e responsabilidades. Isso significa que é possível criar sistemas mais flexíveis e adaptáveis, que podem ser ajustados para atender às necessidades específicas de uma determinada tarefa ou aplicação.

5.     Redução de custos: A computação distribuída pode ser uma opção mais econômica do que investir em um único sistema de alto desempenho. Isso porque, ao distribuir o processamento entre vários componentes, é possível utilizar recursos existentes e mais baratos, em vez de ter que investir em um único sistema de alto desempenho, que pode ser mais caro e difícil de manter.

# II. Arquiteturas de sistemas distribuídos

As arquiteturas de sistemas distribuídos podem ser classificadas de acordo com a forma como os componentes são organizados e como as comunicações entre eles são realizadas. As principais arquiteturas de sistemas distribuídos são:

1. Cliente-Servidor: Nesta arquitetura, os serviços são fornecidos por um servidor centralizado e os clientes fazem solicitações para aceder esses serviços. Esta arquitetura é comum em sistemas de banco de dados, onde os clientes acedem a informações armazenadas no servidor.
2. Peer-to-Peer: Nesta arquitetura, todos os componentes são iguais e colaboram para realizar a tarefa. Cada componente é tanto um cliente como um servidor, e comunicam uns com os outros diretamente para compartilhar recursos ou trocar informações. Esta arquitetura é comum em sistemas de compartilhamento de arquivos, como o BitTorrent.
3. Arquitetura em camadas: Nesta arquitetura, os componentes são organizados em camadas, onde cada camada tem uma função específica. A comunicação entre as camadas é realizada de forma estruturada e organizada, com cada camada a fornecer serviços à camada superior. Esta arquitetura é comum em sistemas de rede, onde as camadas podem incluir a camada de aplicação, a camada de transporte e a camada de rede.
4. Arquitetura orientada a serviços: Nesta arquitetura, os serviços são fornecidos como unidades independentes, que podem ser combinadas para formar sistemas mais complexos. Cada serviço é autónomo e pode ser acedido por meio de uma interface bem definida. Essa arquitetura é comum em sistemas de comércio eletrônico, onde serviços como pagamentos, envio de pedidos e gestão de estoque podem ser fornecidos por diferentes fornecedores.
5. Arquitetura de micro serviços: Nesta arquitetura, o sistema é dividido em micro serviços independentes, onde cada um executa uma tarefa específica. Cada micro serviço é autónomo e pode ser escalado independentemente. A comunicação entre os micro serviços é realizada por meio de APIs e protocolos bem definidos. Esta arquitetura é comum em sistemas modernos de computação em nuvem e em aplicações web escaláveis.

## Redes ponto a ponto

As redes ponto a ponto (peer-to-peer, ou P2P) são uma forma de conexão de computadores em que todos os dispositivos são considerados iguais, e cada um pode comunicar diretamente com qualquer outro dispositivo conectado à rede, sem a necessidade de um servidor central.

Essas redes são bastante utilizadas para compartilhamento de arquivos, pois permitem que os user’s possam baixar e enviar arquivos diretamente uns aos outros, sem a necessidade de um servidor intermediário. No entanto, as redes P2P também podem ser usadas para outras finalidades, como jogos online, mensagens instantâneas e compartilhamento de recursos, como impressoras e conexão à internet.

Existem vários tipos de redes P2P, cada uma com suas próprias características. Alguns exemplos incluem:

1. Rede P2P pura: Nesta rede, todos os dispositivos são iguais e se comunicam diretamente uns com os outros. Não há um servidor central que gira as conexões.
2. Rede P2P híbrida: Nesta rede, alguns dispositivos atuam como servidores centrais, mas apenas para coordenar as conexões entre os outros dispositivos. Esses servidores não armazenam conteúdo.
3. Rede P2P baseada em tracker: Nesta rede, um servidor central, chamado de tracker, é responsável por coordenar as conexões entre os dispositivos. Os dispositivos comunicam diretamente uns com os outros para compartilhar arquivos, mas usam o tracker para encontrar outros dispositivos na rede.

As redes P2P têm algumas vantagens em relação às redes centralizadas, como maior eficiência no uso de recursos e maior resistência a falhas. No entanto, também podem ser mais difíceis de gerir e podem apresentar problemas de segurança e privacidade, como a possibilidade de compartilhamento de conteúdo ilegal e a exposição de informações pessoais.

## Sistemas cliente-servidor

O sistema cliente-servidor é uma arquitetura de rede em que um ou mais dispositivos, chamados de clientes, se conectam a um ou mais dispositivos, chamados de servidores, para compartilhar recursos, dados e serviços.

Os clientes enviam solicitações aos servidores, que processam as solicitações e enviam as respostas de volta aos clientes. Esta arquitetura é amplamente utilizada em redes de computadores, em que muitos dispositivos precisam de se conectar a um ou mais servidores para aceder a recursos compartilhados, como arquivos, bases de dados, impressoras e serviços de aplicações.

Algumas das principais características do modelo cliente-servidor incluem:

1. Centralização: O servidor é o ponto central da rede, responsável por gerir os recursos e os serviços que são compartilhados com os clientes.
2. Escalabilidade: A arquitetura cliente-servidor permite que mais clientes sejam adicionados à rede conforme a procura aumenta, sem afetar o desempenho ou a disponibilidade dos recursos.
3. Segurança: Os servidores podem ser configurados para controlar o acesso aos recursos compartilhados, permitindo que apenas user’s autorizados tenham acesso a esses recursos.
4. Desempenho: A arquitetura cliente-servidor pode melhorar o desempenho da rede, pois o servidor pode executar operações complexas e pesadas em recursos, deixando os clientes livres para realizar tarefas mais leves.

No entanto, o modelo cliente-servidor também pode apresentar alguns desafios, como a necessidade de manter e atualizar o software do servidor, a complexidade da configuração e a possível sobrecarga do servidor quando muitos clientes se conectam simultaneamente.

## Sistemas em cluster

Um sistema em cluster é um conjunto de computadores interconectados que trabalham juntos para fornecer alta disponibilidade, escalabilidade e desempenho para aplicações e serviços. Esses sistemas são usados em serviços web, bases de dados, renderização de imagens, processamento de dados e simulações científicas.

Num cluster, cada computador é um “nó” e possui um sistema operativo e recursos próprios, como processadores, memória e armazenamento.

Existem vários tipos de clusters, como:

1. Cluster de alto desempenho (HPC): usado para computação científica, engenharia e outros tipos de aplicativos que exigem grande poder de processamento e memória. Estes clusters podem ter centenas ou até milhares de nós.
2. Cluster de Load Balancer: usado para distribuir o tráfego de rede entre vários servidores, a fim de garantir que a carga seja equilibrada e os recursos não sejam sobrecarregados.
3. Cluster de alta disponibilidade (HA): usado para fornecer serviços ininterruptos, mesmo em caso de falha de um ou mais nós do cluster.
4. Cluster de armazenamento: usado para fornecer armazenamento compartilhado numa rede de computadores.

Os sistemas em cluster oferecem várias vantagens, incluindo:

* Alta disponibilidade: se um nó falhar, o serviço ainda pode ser executado noutros nós do cluster.
* Escalabilidade: o cluster pode ser facilmente expandido adicionando mais nós.
* Desempenho: a capacidade de processamento e memória pode ser compartilhada entre os nós do cluster, melhorando o desempenho do sistema.

No entanto, a configuração e manutenção de um sistema em cluster podem ser complexas e exigir um conhecimento técnico avançado. Além disso, os custos de hardware e software podem ser significativos, especialmente em clusters de alta performance.

# III. Compartilhamento de recursos em sistemas distribuídos

## Uso de vários computadores para processamento e armazenamento de dados

## Comunicação entre computadores

# IV. Segurança e confiabilidade em sistemas distribuídos

## Replicação de dados

## Deteção e tratamento de falhas

# V. Protocolos de comunicação em sistemas distribuídos

Os protocolos de comunicação são conjuntos de regras e padrões que governam a comunicação entre computadores em sistemas distribuídos. Eles são essenciais para garantir que os diferentes computadores em um sistema distribuído possam se comunicar de forma eficaz e coordenada para executar uma tarefa ou compartilhar dados.

Os protocolos de comunicação em sistemas distribuídos incluem várias etapas que permitem que os computadores se comuniquem uns com os outros. Essas etapas podem incluir:

1. Estabelecimento da conexão: o protocolo define como um computador estabelece uma conexão com outro.
2. Transmissão de dados: o protocolo define como os dados são transmitidos de um computador para outro.
3. Recebimento de dados: o protocolo define como um computador recebe os dados enviados por outro.
4. Tratamento de erros: o protocolo define como os erros de comunicação são tratados.
5. Encerramento da conexão: o protocolo define como uma conexão é encerrada após a conclusão da tarefa ou transmissão de dados.

Existem vários protocolos de comunicação amplamente usados em sistemas distribuídos, incluindo TCP/IP, HTTP, FTP, SMTP, entre outros. Cada protocolo é projetado para atender a uma finalidade específica e possui diferentes níveis de confiabilidade, segurança e eficiência.

Os protocolos de comunicação são essenciais para garantir a coordenação e a integração de sistemas distribuídos, permitindo que os computadores trabalhem juntos de forma eficaz e eficiente para atender aos objetivos da tarefa ou do sistema.

## Sincronização de tarefas

A sincronização de tarefas é um aspecto importante dos protocolos de comunicação em sistemas distribuídos, que permite que diferentes computadores trabalhem juntos de forma coordenada para executar uma tarefa específica. Em sistemas distribuídos, as tarefas geralmente são divididas em partes menores e executadas em diferentes computadores, que precisam ser sincronizados para garantir que a tarefa seja concluída com sucesso.

Existem vários protocolos de sincronização de tarefas que podem ser usados em sistemas distribuídos, incluindo:

1. Protocolos de bloqueio: esses protocolos são usados para controlar o acesso a recursos compartilhados em sistemas distribuídos. Eles permitem que apenas um computador acesse um recurso compartilhado por vez, evitando conflitos e garantindo que o recurso seja usado corretamente.
2. Protocolos de transação: esses protocolos são usados para garantir a integridade dos dados em sistemas distribuídos, permitindo que as transações sejam executadas em diferentes computadores de forma coordenada. Eles garantem que as transações sejam concluídas com sucesso ou que sejam totalmente desfeitas caso ocorra um erro.
3. Protocolos de eleição: esses protocolos são usados para selecionar um líder ou coordenador em um grupo de computadores em um sistema distribuído. O líder é responsável por coordenar as tarefas e garantir que todos os computadores estejam trabalhando juntos de forma coordenada.
4. Protocolos de sincronização de relógio: esses protocolos são usados para sincronizar os relógios de diferentes computadores em um sistema distribuído, permitindo que eles coordenem suas ações e eventos com precisão.

A sincronização de tarefas é essencial para garantir que as tarefas em sistemas distribuídos sejam concluídas de forma eficaz e eficiente. Os protocolos de sincronização de tarefas ajudam a garantir que os diferentes computadores em um sistema distribuído possam trabalhar juntos de forma coordenada para concluir uma tarefa com sucesso.

## Transmissão e recepção de dados

A transmissão e recepção de dados são aspectos fundamentais dos protocolos de comunicação em sistemas distribuídos. Esses protocolos são projetados para permitir que os computadores se comuniquem entre si, transmitindo e recebendo dados de diferentes tipos e formatos.

A transmissão de dados envolve o envio de informações de um computador para outro em um sistema distribuído. Para que isso aconteça, os dados precisam ser codificados em um formato que possa ser entendido pelos computadores de destino e enviados por meio de uma rede de comunicação. Existem vários protocolos de transmissão de dados que podem ser usados em sistemas distribuídos, incluindo TCP/IP, HTTP, FTP, SMTP, entre outros.

A recepção de dados envolve o processo de receber informações enviadas por outro computador em um sistema distribuído. Para que isso aconteça, o computador receptor precisa estar conectado à rede de comunicação e ter um protocolo de recepção de dados configurado para receber e interpretar os dados corretamente. O protocolo de recepção de dados pode incluir verificações de integridade, como checksums ou hashes, para garantir que os dados recebidos sejam precisos e completos.

Além disso, em sistemas distribuídos, a transmissão e recepção de dados podem ser afetadas por vários fatores externos, como latência de rede, perda de pacotes, congestionamento de rede e outros problemas que podem afetar a qualidade da transmissão de dados. Por esse motivo, os protocolos de comunicação em sistemas distribuídos precisam ser projetados para lidar com esses desafios e garantir que os dados sejam transmitidos e recebidos com segurança e eficácia.

# VI. Conclusão

A computação distribuída é uma abordagem para a construção de sistemas de computação que permite que múltiplos computadores trabalhem juntos de forma coordenada para executar tarefas complexas. Essa abordagem oferece várias vantagens em relação aos sistemas centralizados, incluindo maior escalabilidade, maior disponibilidade, melhor desempenho e maior resiliência a falhas.

No entanto, a computação distribuída também apresenta desafios significativos em termos de gerenciamento de recursos, sincronização de tarefas, transmissão e recepção de dados, segurança e outros aspectos críticos. Para lidar com esses desafios, são necessários protocolos e algoritmos específicos que permitem que os computadores em um sistema distribuído trabalhem juntos de forma coordenada e eficiente.

No geral, a computação distribuída é uma abordagem importante para a construção de sistemas de computação de grande escala e complexidade. À medida que as demandas por recursos computacionais continuam a crescer, a computação distribuída provavelmente se tornará cada vez mais importante para a construção de sistemas eficientes e escaláveis que possam lidar com as necessidades do mundo moderno.

## Benefícios da computação distribuída

A computação distribuída oferece uma série de benefícios em relação aos sistemas centralizados, incluindo:

1. Escalabilidade: a computação distribuída permite que os recursos computacionais sejam facilmente adicionados ou removidos do sistema, o que permite que o sistema cresça ou diminua de acordo com as necessidades.
2. Melhor desempenho: a computação distribuída pode executar tarefas mais rapidamente do que os sistemas centralizados, porque as tarefas podem ser distribuídas em vários computadores e executadas em paralelo.
3. Disponibilidade: a computação distribuída pode oferecer maior disponibilidade do sistema, porque se um computador falhar, outros computadores podem assumir as tarefas que estavam sendo executadas pelo computador que falhou.
4. Redundância: a computação distribuída pode ser usada para criar sistemas redundantes, que podem ser usados para garantir que as informações e os dados estejam sempre disponíveis.
5. Flexibilidade: a computação distribuída pode ser usada para criar sistemas flexíveis, que podem ser adaptados às necessidades específicas de diferentes usuários ou aplicações.
6. Custo-efetividade: a computação distribuída pode ser mais econômica do que os sistemas centralizados, porque permite que os recursos computacionais sejam compartilhados e utilizados de forma mais eficiente.
7. Resiliência a falhas: a computação distribuída é mais resiliente a falhas do que os sistemas centralizados, porque os dados e as tarefas podem ser distribuídos em vários computadores e redundâncias podem ser criadas para garantir que o sistema continue funcionando mesmo em caso de falhas.

No geral, a computação distribuída pode ser uma solução eficaz para lidar com os desafios computacionais modernos, oferecendo maior escalabilidade, desempenho, disponibilidade, flexibilidade e resiliência a falhas.

## Desafios e oportunidades futuras

Embora a computação distribuída apresente muitos benefícios, também existem desafios e oportunidades futuras a serem considerados. Alguns desses desafios e oportunidades incluem:

1. Gerenciamento de recursos: gerenciar recursos distribuídos em vários computadores pode ser complexo e desafiador. Futuras soluções de gerenciamento de recursos devem ser desenvolvidas para garantir que os recursos sejam alocados de forma eficiente e otimizada.
2. Segurança: com tantos dispositivos conectados a um sistema distribuído, a segurança é uma grande preocupação. A computação distribuída deve ser protegida contra invasões e ataques cibernéticos, o que pode ser difícil devido à natureza distribuída da computação.
3. Sincronização de tarefas: garantir que as tarefas sejam executadas em ordem correta e de forma síncrona pode ser um desafio em sistemas distribuídos. Novos protocolos e algoritmos devem ser desenvolvidos para garantir que as tarefas sejam sincronizadas de forma eficiente.
4. Transmissão e recepção de dados: o envio e o recebimento de dados em sistemas distribuídos podem ser problemáticos. As soluções futuras devem se concentrar em melhorar a eficiência da transmissão e garantir que os dados cheguem ao destino corretamente e de forma segura.
5. Análise de dados em tempo real: a computação distribuída pode ser usada para análise de dados em tempo real, mas pode ser desafiador garantir que a análise seja precisa e rápida o suficiente. Soluções futuras devem se concentrar em melhorar a eficiência da análise de dados em tempo real.
6. Oportunidades para a IoT: a computação distribuída é fundamental para a Internet das Coisas (IoT), permitindo que dispositivos se comuniquem uns com os outros e realizem tarefas complexas. A computação distribuída pode ser usada para melhorar a eficiência da IoT e permitir novas aplicações e serviços.
7. Aumento da capacidade de processamento: com a computação distribuída, é possível aumentar significativamente a capacidade de processamento disponível para executar tarefas complexas. Isso pode levar a novas aplicações e serviços que não eram possíveis anteriormente.

No geral, a computação distribuída oferece muitas oportunidades para melhorar a eficiência e a capacidade de processamento dos sistemas de computação modernos, mas também apresenta desafios que precisam ser superados. O futuro da computação distribuída dependerá da capacidade de enfrentar esses desafios e aproveitar as oportunidades apresentadas.