

Sistemas Distribuídos - ESP625

Prof^a Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal de Ciência e Tecnologia da Bahia
Campus de Feira de Santana

carolsoko@ifba.edu.br

April 13, 2023

Paradigmas e Desafios em Sistemas Distribuídos

1 Paradigmas de Sistemas Distribuídos

- Paradigma “Hierárquico”
- Paradigma “Cache de CPU”
- Paradigma “Cliente-Servidor”
- Paradigma “Conjunto de Processadores”
- Paradigma “Orientado ao Fluxo de Dados”

2 Desafios em Sistemas Distribuídos

3 Referências

Paradigmas de Sistemas Distribuídos

Paradigmas de Sistemas Distribuídos

Os sistemas distribuídos podem ser classificados de acordo com cinco paradigmas:

Paradigmas de Sistemas Distribuídos

Os sistemas distribuídos podem ser classificados de acordo com cinco paradigmas:

1 Hierárquico

Paradigmas de Sistemas Distribuídos

Os sistemas distribuídos podem ser classificados de acordo com cinco paradigmas:

- 1 Hierárquico
- 2 Cache de CPU

Paradigmas de Sistemas Distribuídos

Os sistemas distribuídos podem ser classificados de acordo com cinco paradigmas:

- 1 Hierárquico
- 2 Cache de CPU
- 3 Cliente-Servidor

Paradigmas de Sistemas Distribuídos

Os sistemas distribuídos podem ser classificados de acordo com cinco paradigmas:

- 1 Hierárquico
- 2 Cache de CPU
- 3 Cliente-Servidor
- 4 Conjunto de Processadores

Paradigmas de Sistemas Distribuídos

Os sistemas distribuídos podem ser classificados de acordo com cinco paradigmas:

- 1 Hierárquico
- 2 Cache de CPU
- 3 Cliente-Servidor
- 4 Conjunto de Processadores
- 5 Orientado ao Fluxo de Dados

Paradigma “Hierárquico”

O Paradigma Hierárquico caracteriza-se por dispor os vários processadores de acordo com uma organização em árvore. É organizado de maneira que a capacidade computacional dos processadores seja maior à medida que esses processadores estejam mais próximos da raiz da árvore.

Paradigma “Hierárquico”

O Paradigma Hierárquico caracteriza-se por dispor os vários processadores de acordo com uma organização em árvore. É organizado de maneira que a capacidade computacional dos processadores seja maior à medida que esses processadores estejam mais próximos da raiz da árvore.

As funções tratadas pelos vários processadores são distribuídas de acordo com a capacidade computacional de cada processador, de modo que os processadores mais distantes da raiz tratam de serviços mais específicos e especializados, enquanto que os processadores mais próximos da raiz tratam de serviços mais globais.

Paradigma "Cache de CPU"

O Paradigma Cache de CPU caracteriza-se pelo fornecimento, ao usuário, de 2 níveis de capacidade computacional: o primeiro possui menor capacidade e o segundo maior.

Paradigma “Cache de CPU”

O Paradigma Cache de CPU caracteriza-se pelo fornecimento, ao usuário, de 2 níveis de capacidade computacional: o primeiro possui menor capacidade e o segundo maior.

Isto é conseguido conectando-se o usuário a um computador de menor capacidade, sendo que este estará conectado a um computador central de grande capacidade.

Paradigma “Cache de CPU”

O Paradigma Cache de CPU caracteriza-se pelo fornecimento, ao usuário, de 2 níveis de capacidade computacional: o primeiro possui menor capacidade e o segundo maior.

Isto é conseguido conectando-se o usuário a um computador de menor capacidade, sendo que este estará conectado a um computador central de grande capacidade.

O sistema operacional decidirá em que computador determinado serviço será realizado, de acordo com alguns dados, tais como, adequabilidade de cada máquina, custo relativo, taxa de transmissão entre as máquinas e carga de trabalho.

Paradigma “Cliente-Servidor”

O Paradigma Cliente-Servidor caracteriza-se pela existência de vários computadores clientes e vários computadores servidores conectados através de um subsistema de comunicação.

Paradigma “Cliente-Servidor”

O Paradigma Cliente-Servidor caracteriza-se pela existência de vários computadores clientes e vários computadores servidores conectados através de um subsistema de comunicação.

Os computadores clientes comandam a execução da aplicação, acessando, quando necessário, os computadores servidores, a fim de que esses realizem algumas funções específicas.

Paradigma “Cliente-Servidor”

O Paradigma Cliente-Servidor caracteriza-se pela existência de vários computadores clientes e vários computadores servidores conectados através de um subsistema de comunicação.

Os computadores clientes comandam a execução da aplicação, acessando, quando necessário, os computadores servidores, a fim de que esses realizem algumas funções específicas.

Servidores de arquivos, servidores de banco de dados e servidores de impressão são exemplos típicos de processos que residiriam em processadores servidores.

Paradigma “Cliente-Servidor”

Este paradigma apresenta algumas vantagens em relação ao Cache de CPU, tais como: não existe um computador central; existe a facilidade para o compartilhamento de periféricos caros; e existe economia em termos de armazenamento em memória secundária, pois dispositivos com alta capacidade de armazenamento apresentam um custo por byte armazenado inferior ao custo em dispositivos com baixa capacidade.

Paradigma “Conjunto de Processadores”

O Paradigma Conjunto de Processadores caracteriza-se pela existência de um conjunto de processadores disponíveis a todos os usuários, os quais estão conectados diretamente ao subsistema de comunicação.

Paradigma “Conjunto de Processadores”

O Paradigma Conjunto de Processadores caracteriza-se pela existência de um conjunto de processadores disponíveis a todos os usuários, os quais estão conectados diretamente ao subsistema de comunicação.

Os serviços dos usuários serão designados dinamicamente a um dos processadores. Nesse paradigma, poderão existir ainda processadores com funções específicas.

Paradigma “Conjunto de Processadores”

Este paradigma é mais otimizado que o Cliente-Servidor, pois evita que um processador seja dedicado exclusivamente a um usuário ou a um conjunto de usuários.

Paradigma “Conjunto de Processadores”

Este paradigma é mais otimizado que o Cliente-Servidor, pois evita que um processador seja dedicado exclusivamente a um usuário ou a um conjunto de usuários.

Dessa forma, é possível uma distribuição balanceada dos serviços requisitados pelos usuários entre os vários elementos processadores, evitando que exista grande diferença de carga entre esses elementos processadores.

Paradigma “Orientado ao Fluxo de Dados”

O paradigma Orientado ao Fluxo de Dados é muito mais radical que os demais. Neste paradigma, um programa é descrito por um membro da rede e designado a um elemento processador da rede.

Paradigma “Orientado ao Fluxo de Dados”

O paradigma Orientado ao Fluxo de Dados é muito mais radical que os demais. Neste paradigma, um programa é descrito por um membro da rede e designado a um elemento processador da rede.

Os resultados gerados por cada um destes elementos são enviados a outros elementos processadores como mensagens.

Paradigma “Orientado ao Fluxo de Dados”

O paradigma Orientado ao Fluxo de Dados é muito mais radical que os demais. Neste paradigma, um programa é descrito por um membro da rede e designado a um elemento processador da rede.

Os resultados gerados por cada um destes elementos são enviados a outros elementos processadores como mensagens.

As instruções são executadas assincronamente, dependendo apenas da disponibilidade dos dados.

Paradigma “Orientado ao Fluxo de Dados”

Isto significa dizer que uma máquina otimizada apresentará o seu máximo desempenho.

Paradigma “Orientado ao Fluxo de Dados”

Isto significa dizer que uma máquina otimizada apresentará o seu máximo desempenho.

Este paradigma objetiva alcançar um paralelismo de granularidade fina, ou seja, os processos que executam em paralelo são aproximadamente do tamanho de uma instrução de uma máquina convencional. [TANENBAUM e STEEN 2007]

Desafios em Sistemas Distribuídos

Desafios em Sistemas Distribuídos

Os desafios em Sistemas Distribuídos envolvem latência, escalabilidade, compreensão sobre APIs de redes, complexidade dos algoritmos, tolerância a falhas, entre outros.

Desafios em Sistemas Distribuídos

Os desafios em Sistemas Distribuídos envolvem latência, escalabilidade, compreensão sobre APIs de redes, complexidade dos algoritmos, tolerância a falhas, entre outros.

Conforme os sistemas aumentam e tornam-se mais distribuídos, os casos de borda que eram teóricos tornaram-se ocorrências regulares.

Desafios em Sistemas Distribuídos

Os desafios em Sistemas Distribuídos envolvem latência, escalabilidade, compreensão sobre APIs de redes, complexidade dos algoritmos, tolerância a falhas, entre outros.

Conforme os sistemas aumentam e tornam-se mais distribuídos, os casos de borda que eram teóricos tornaram-se ocorrências regulares.

O desenvolvimento de serviços distribuídos, como redes confiáveis de telefone de longa distância, ou serviços da Amazon Web Services (AWS), é muito complexo.

Desafios em Sistemas Distribuídos

A computação distribuída também é mais estranha e menos intuitiva do que outras formas de computação, devido a dois problemas correlacionados.

Desafios em Sistemas Distribuídos

A computação distribuída também é mais estranha e menos intuitiva do que outras formas de computação, devido a dois problemas correlacionados.

Falhas independentes

Desafios em Sistemas Distribuídos

A computação distribuída também é mais estranha e menos intuitiva do que outras formas de computação, devido a dois problemas correlacionados.

Falhas independentes e não determinismo causam os problemas mais impactantes nos sistemas distribuídos. Além das falhas de computação típicas com as quais a maioria dos programadores está acostumado, as falhas em sistemas distribuídos podem ocorrer de várias outras maneiras.

Desafios em Sistemas Distribuídos

A computação distribuída também é mais estranha e menos intuitiva do que outras formas de computação, devido a dois problemas correlacionados.

Falhas independentes e não determinismo causam os problemas mais impactantes nos sistemas distribuídos. Além das falhas de computação típicas com as quais a maioria dos programadores está acostumado, as falhas em sistemas distribuídos podem ocorrer de várias outras maneiras.

E o que é pior: é impossível saber sempre se algo apresentou falha.

Desafios em Sistemas Distribuídos

MAIORES DESAFIOS EM SISTEMAS DISTRIBUÍDOS

- 1 Tolerância a Falha devido à ocorrência de Falhas Independentes
- 2 Imprevisibilidade e não determinismo
- 3 Manutenção de Sistemas de Tempo-Real Críticos e Não Críticos
- 4 Confiabilidade
- 5 Disponibilidade
- 6 Escalabilidade

Referências

Referências



TANENBAUM, A.; STEEN, M. V. *Sistemas Distribuídos - Princípios e Paradigmas*. 2. ed. [S.l.]: Prentice Hall, 2007. v. 1.