

ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

Aula 05: Arquiteturas (parte 3)

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

11/03/2024

Organizações alternativas

Distribuição vertical

Decorre da divisão de aplicações distribuídas em três camadas lógicas e da execução dos componentes de cada camada em um servidor (máquina) diferente.

Distribuição horizontal

Um cliente ou servidor pode ser dividido fisicamente em partes logicamente equivalentes, mas cada parte opera em sua própria parte do conjunto de dados completo.

Arquiteturas peer-to-peer

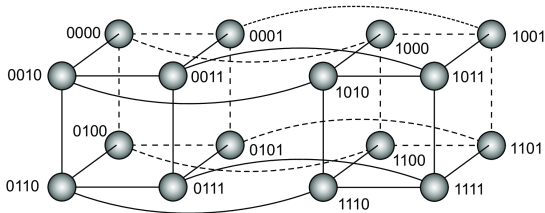
Os processos são todos iguais: as funções que precisam ser executadas são representadas por todos os processos \Rightarrow cada processo atua como cliente e servidor ao mesmo tempo.

P2P Estruturado

- Sistema organizado em uma topologia conhecida, de forma que a topologia é utilizada para procurar dados de forma eficiente.
- Faz uso de um **índice livre de semântica**: cada item de dados é associado unicamente a uma chave, por sua vez usada como índice.
- Sistema P2P responsável por armazenar pares (*chave*, *valor*).
- Prática comum: usar uma **função hash** (DHT – distributed hash table)

P2P Estruturado: exemplo 1

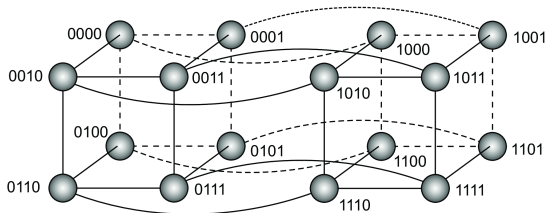
Hipercubo



Procurar um dado d com a **chave** $k \in \{0, 1, 2, \dots, 2^4 - 1\}$ significa **rotear** a solicitação para o nó com **identificador** k .

P2P Estruturado: exemplo 1

Hipercubo



Procurar um dado d com a **chave** $k \in \{0, 1, 2, \dots, 2^4 - 1\}$ significa **rotear** a solicitação para o nó com **identificador** k .

Porém...

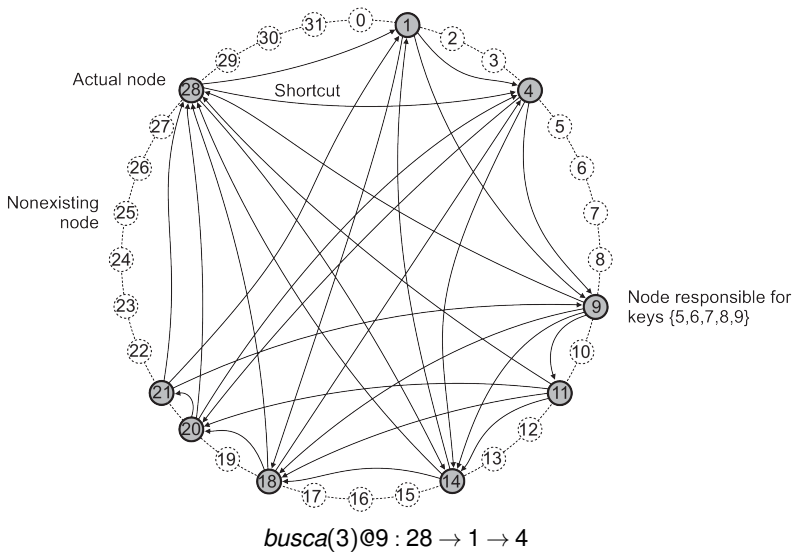
Difícil adicionar ou remover nós.

P2P Estruturado: exemplo 2

Chord

- Os nós são organizados logicamente em um anel. Cada nó tem um **identificador** de m bits.
- Cada item de dados é hashado para uma **chave** de m bits.
- O item de dados com a chave k é armazenado no nó com o menor identificador $id \geq k$, chamado de **sucessor** da chave k .
- O anel é estendido com vários **links de atalho** para outros nós.

Exemplo: Chord



P2P Não Estruturado

Essência

Cada nó mantém uma lista ad hoc de vizinhos. A sobreposição resultante se assemelha a um **grafo aleatório**: uma aresta $\langle u, v \rangle$ existe apenas com uma certa probabilidade $\mathbb{P}[\langle u, v \rangle]$.

Busca

- **Inundação (flooding)**: o nó requisitante u passa a solicitação d para todos os vizinhos. A solicitação é ignorada quando o nó receptor já a viu antes. Caso contrário, v busca localmente por d (recursivamente). Pode ser limitado por um **Tempo de Vida**: um número máximo de saltos.
- **Caminhada aleatória (random walk)**: o nó requisitante u passa a solicitação d para um vizinho escolhido aleatoriamente, v . Se v não tiver d , ele encaminha a solicitação para um de *seus* vizinhos escolhidos aleatoriamente, e assim por diante.

Inundação versus caminhada aleatória

Modelo

Suponha N nós e que cada item de dados é replicado em r nós escolhidos aleatoriamente.

Caminhada aleatória

$\mathbb{P}[k]$ probabilidade de encontrar o item após k tentativas:

$$\mathbb{P}[k] = \frac{r}{N} \left(1 - \frac{r}{N}\right)^{k-1}.$$

S (“tamanho da busca”) é o número esperado de nós que precisam ser sondados:

$$S = \sum_{k=1}^N k \cdot \mathbb{P}[k] = \sum_{k=1}^N k \cdot \frac{r}{N} \left(1 - \frac{r}{N}\right)^{k-1} \approx N/r \text{ para } 1 \ll r \leq N.$$

Inundação versus caminhada aleatória

Inundação

- Inunda para d vizinhos escolhidos aleatoriamente
- Após k etapas, alguns $R(k) = d \cdot (d-1)^{k-1}$ terão sido alcançados (assumindo que k é pequeno).
- Com fração r/N de nós tendo dados, se $\frac{r}{N} \cdot R(k) \geq 1$, teremos chances razoáveis de ter encontrado o item de dados.

Comparação

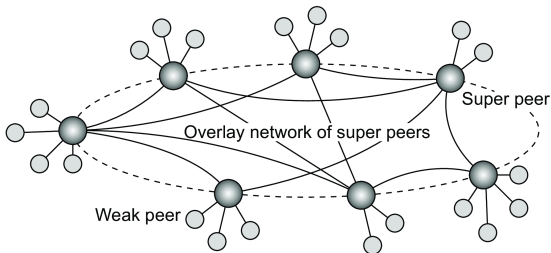
- Se $r/N = 0.001$, então $S \approx 1000$
- Com inundação e $d = 10, k = 4$, contatamos 7290 nós.
- As caminhadas aleatórias são mais eficientes em comunicação, mas podem demorar mais para encontrar o resultado.

Redes de super-pares (super-peer)

Essência

Às vezes, faz sentido quebrar a simetria em redes puramente peer-to-peer:

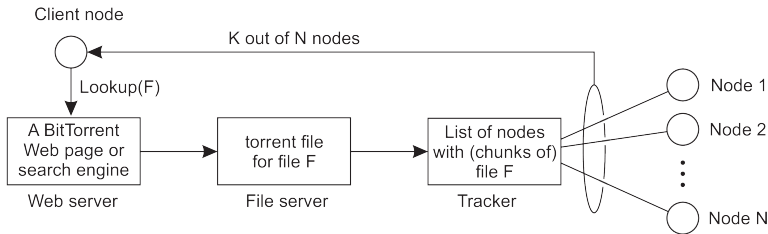
- Ao buscar em sistemas P2P não estruturados, ter **servidores de índice** melhora o desempenho
- Decidir onde armazenar dados muitas vezes pode ser feito de maneira mais eficiente por meio de **intermediários (brokers)**.



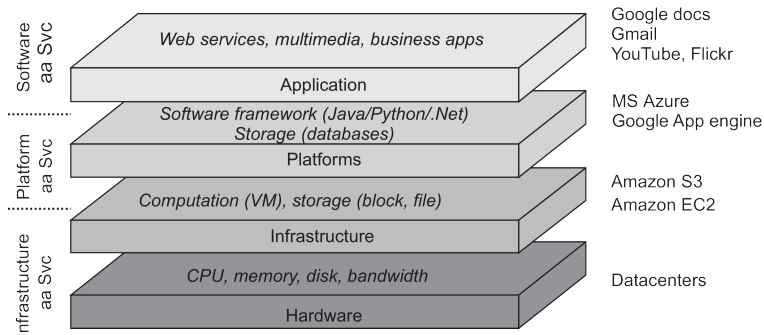
Colaboração: O caso do BitTorrent

Princípio: procurar por um arquivo F

- Procurar arquivo em um diretório global \Rightarrow retorna um **arquivo torrent**
- O arquivo torrent contém uma referência para um **rastreador (tracker)**: um servidor que mantém um registro preciso de nós **ativos** que possuem (partes de) F .
- P pode se juntar ao **enxame (swarm)**, obter uma parte gratuitamente e depois trocar uma cópia dessa parte por outra com um peer Q também no enxame.



Computação em nuvem



Computação em nuvem

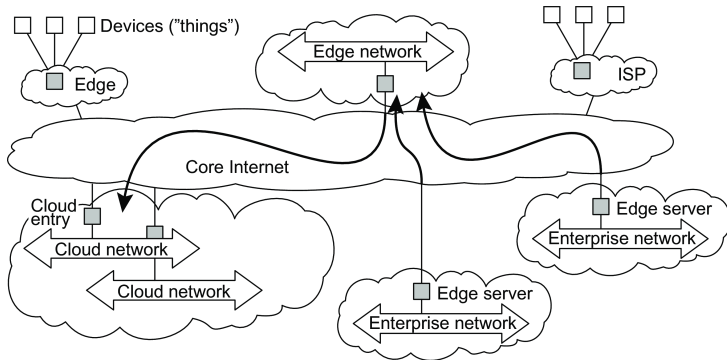
Fazer uma distinção entre quatro camadas

- **Hardware:** Processadores, roteadores, sistemas de energia e de resfriamento. Os clientes normalmente não enxergam estes elementos.
- **Infraestrutura:** Implanta técnicas de virtualização. Gira em torno de alocar e gerenciar dispositivos de armazenamento virtual e servidores virtuais.
- **Plataforma:** Fornece abstrações de nível mais alto para armazenamento e similares. Exemplo: o sistema de armazenamento Amazon S3 oferece uma API para que arquivos (criados localmente) sejam organizados e armazenados nos chamados **buckets**.
- **Aplicação:** Aplicações reais, como suítes de escritório (processadores de texto, planilhas eletrônicas, aplicativos de apresentação). Comparável à suíte de aplicativos fornecidos com os sistemas operacionais.

Arquitetura de servidor de borda

Essência

Sistemas implantados na Internet onde os servidores são colocados **na borda** da rede: a fronteira entre redes empresariais e a Internet real.



Motivos para ter uma infraestrutura de borda

Argumentos comuns (e frequentemente equivocados)

- **Latência e largura de banda:** Especialmente importante para certas aplicações em tempo real, como aplicações de realidade aumentada/virtual. Muitas pessoas subestimam a latência e a largura de banda para a nuvem.
- **Confiabilidade:** A conexão com a nuvem muitas vezes é assumida como sendo pouco confiável, o que geralmente é uma falsa suposição. Pode haver situações críticas em que garantias de conectividade extremamente altas são necessárias.
- **Segurança e privacidade:** A suposição implícita muitas vezes é que, quando os recursos estão próximos, eles podem ser melhor protegidos. A prática mostra que essa suposição geralmente é falsa. No entanto, lidar de forma segura com operações de dados na nuvem pode ser mais complicado do que dentro de sua própria organização.

Orquestração de borda

Gerenciar recursos na borda pode ser mais complicado do que na nuvem

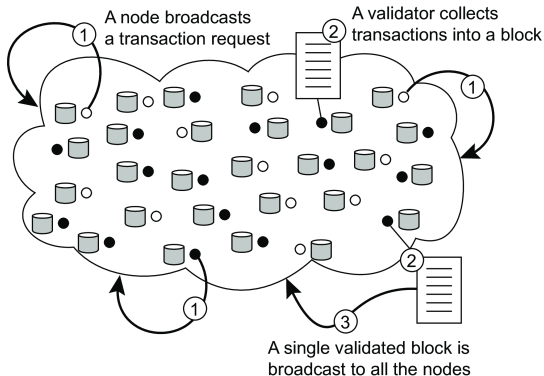
- **Alocação de recursos:** precisamos garantir a disponibilidade dos recursos necessários para realizar um serviço.
- **Posicionamento de serviço (placement):** precisamos decidir **quando** e **onde** colocar um serviço. Isso é especialmente relevante para aplicativos móveis.
- **Seleção de borda:** precisamos decidir qual infraestrutura de borda deve ser usada quando um serviço precisa ser oferecido. A mais próxima pode não ser a melhor.

Observação

Ainda há muita euforia sobre infraestruturas e computação de borda, mas se toda essa euforia faz sentido ainda está por ser visto.

Blockchains

Princípio de funcionamento de um sistema blockchain

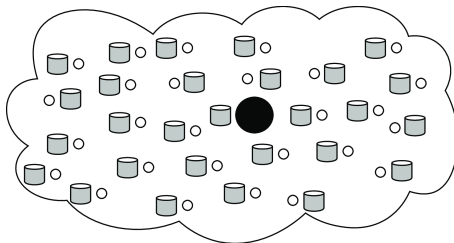


Observações

- Os blocos são organizados em uma cadeia **append-only**
- Cada bloco na blockchain é **imutável** \Rightarrow replicação maciça
- A verdadeira dificuldade reside em quem está autorizado a anexar um bloco a uma cadeia

Anexando um bloco: consenso distribuído

Solução centralizada

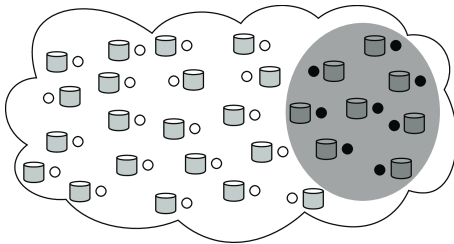


Observação

Uma única entidade decide qual validador pode prosseguir e anexar um bloco. Não se encaixa nos objetivos de projeto das blockchains.

Anexando um bloco: consenso distribuído

Solução distribuída (com permissão)

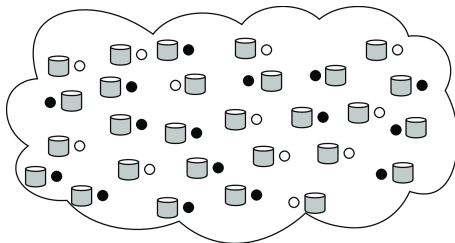


Observação

- Um grupo selecionado, relativamente pequeno, de servidores chega coletivamente a um consenso sobre qual validador pode prosseguir.
- Nenhum desses servidores precisa ser confiável, contanto que aproximadamente dois terços se comportem de acordo com suas especificações.
- Na prática, apenas algumas dezenas de servidores podem ser acomodados.

Anexando um bloco: consenso distribuído

Solução descentralizada (sem permissão)



Observação

- Os participantes se envolvem coletivamente em uma **eleição de líder**. Apenas o líder eleito tem permissão para anexar um bloco de transações validadas.
- Eleição de líder descentralizada em larga escala e que é justa, robusta, segura, etc, está longe de ser trivial.