Duplicação

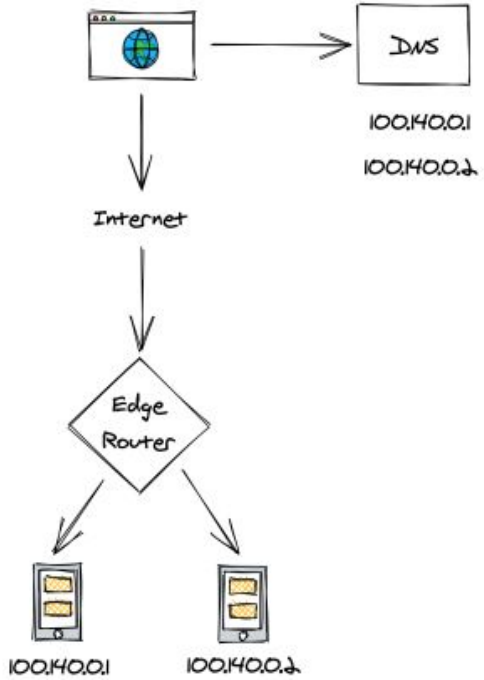
Universidade Tecnológica Federal do Paraná
Bacharelado em Ciência da Computação
Sistemas Distribuídos

Balanceamento de carga de rede

- **Introdução**
  - **Balanceamento de carga**
    - Modo mais fácil de aumentar a capacidade de um serviço;
    - O Balanceador age como um distribuidor da rede de servidores do serviço;
      - O cliente enxerga somente um IP virtual de um dos servidores;
    - Pode gerar gargalo nas dependências;
    - É necessário escolher um método de balanceamento;
      - Ex: Escolher 2 aleatórios e usar o menos ocupado;
  - **Descoberta de serviço**
    - Pode ser implementado via lista estática de servidores;
    - Ou por métodos dinâmicos como DNS;
      - Permite o escalonamento automático;
  - **Checagem de operação (*health check*)**
    - Método passivo ou ativo;

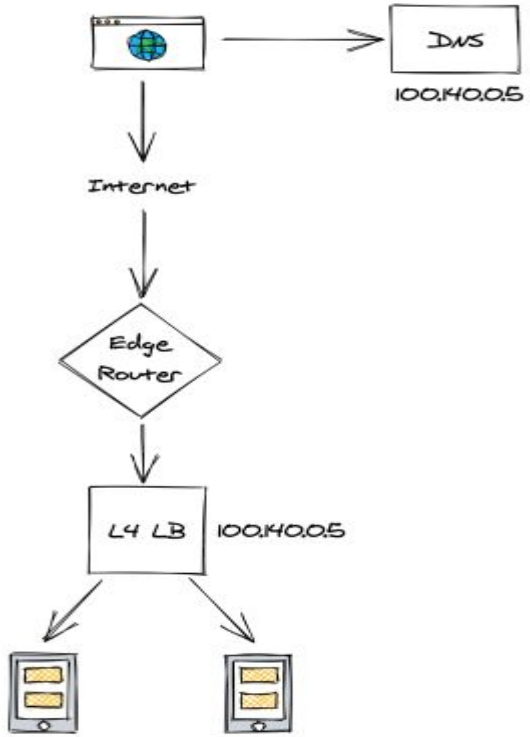
Balanceamento de carga de rede

- Balanceamento de carga com DNS
  - Intermediário entre cliente e serviço.
  - Forma mais básica de implementação de balanceamento de carga.
  - Atribuindo IPs de servidores acessíveis publicamente ao registro DNS do serviço para que os clientes escolham ao resolver o endereço DNS
    .
  - Não lida bem com falhas, se um dos dois servidores cair, o servidor DNS continuará a fornecer seu endereço IP, alheio à falha.



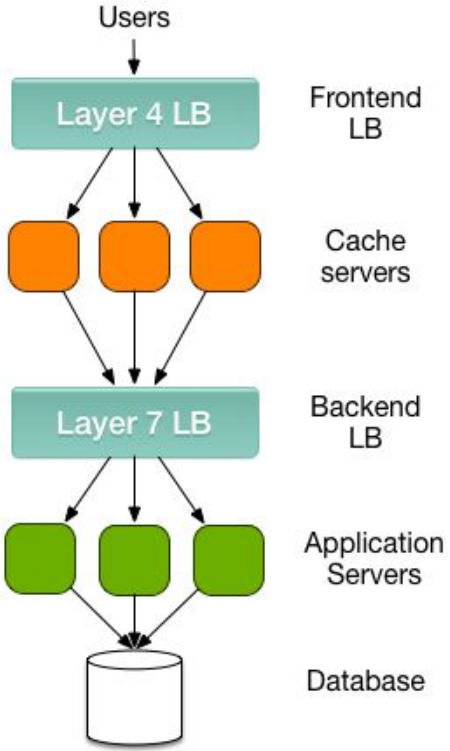
Balanceamento de carga de rede

- Balanceamento de carga na Camada de Transporte
  - Uma solução de balanceamento de carga mais flexível
  - Quando um cliente estabelece uma conexão TCP com o balanceador d carga, este seleciona um servidor do grupo e encaminha os pacotes entre o cliente e o servidor escolhido.
  - Desvantagem é que o balanceador de carga está apenas movendo byt sem saber o que realmente significam eles.



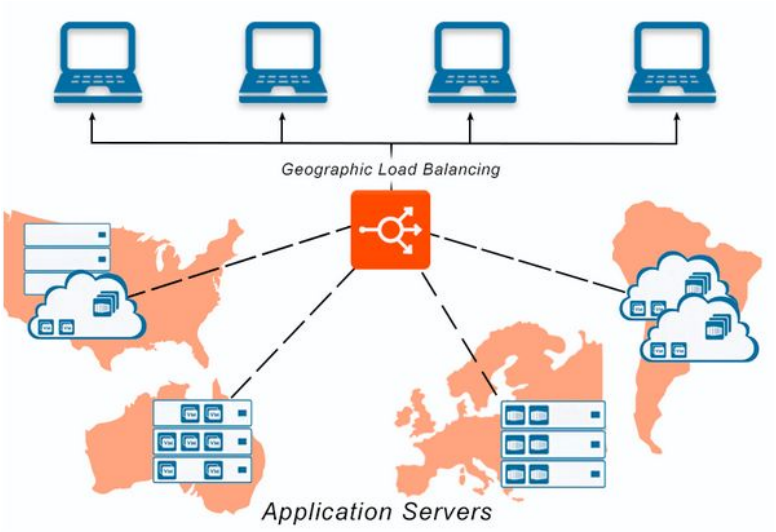
Balanceamento de carga de rede

- Balanceamento de carga na Camada de Aplicação
  - Proxy Reverso HTTP
  - Tem acesso a dados do pacote
  - Pode Tomar decisões com base no conteúdo.



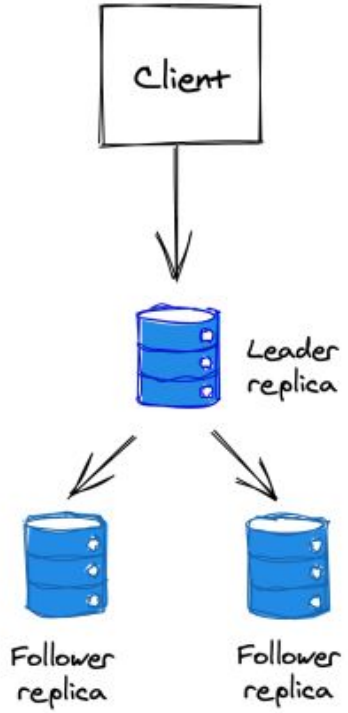
Balanceamento de carga de rede

- Balanceamento de carga Geográfica
  - Redireciona pacotes com base na localização
  - Pelo IP determina a localização do cliente
  - Redireciona o tráfego para a região mais perto do cliente



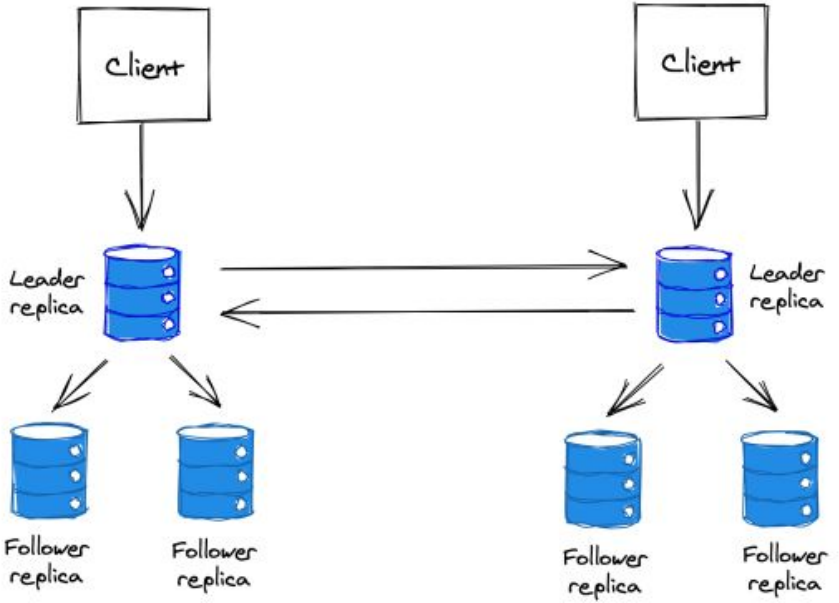
Replicação

- Coordenação para escalar
- Guardar cópias em vários nós
- Dados estáticos x dinâmicos
- Replicação com único líder
  - Método mais comum
  - Clientes enviam dados exclusivamente para o líder
  - Assíncrona
    - Responde o cliente antes da replicação acabar
    - Rápida, mas não é tolerante a falhas
    - Problema de consistência levando à perda de dados.
  - Síncrona
    - Espera a replicação acontecer para todos seguidores
    - Problema de desempenho
    - Se uma réplica for extremamente lenta, cada solicitação será afetado por ela



Replicação

- Replicação com múltiplos líderes
  - Mais de um nó pode aceitar escrita
  - Utilizado quando a quantidade de escritas é muito alta ou quando o líder deve ficar em várias localidades
  - Muito complexa
  - Principal problema é o conflito de escritas
    - Pode ser resolvido pelo próprio cliente
    - Pode ser resolvido de forma automáti-ca



Replicação

- Replicação sem líderes
  - Precisa satisfazer uma invariante:
    - W+R > N
    - W: Número de réplicas necessário para aceitar o pedido de escrita
    - R: Número de réplicas lidas para descobrir a mais recente
    - N: Número de réplicas que o dado tem
  - A escrita sempre é enviada às N réplicas em paralelo
  - Também precisa de uma resolução de conflito
  - Performance depende da quantidade de escritas e leituras realizadas
  - Ainda mais complexa

Caching

- Reduz a carga nos servidores, aumentando a performance do acesso aos dados.
- Políticas:
- Tratamento de erro: duas formas de tratamento:
  - Side cache: cliente trata o erro.
  - Inline cache: cache trata o erro.
- Eliminação de dados não acessados recentemente:
  - LRU: Elimina os dados não utilizados com frequência
  - TTL: Elimina dados com marcações de tempo
- A eliminação não ocorre imediatamente. Pode ocorrer no próximo acesso ao cache.

Caching

- Clientes (*in-process cache*): Cada cliente tem uma fração de memória dedicada ao serviço remoto, como tabelas hash de tamanho limitado.
- Gera problemas de inconsistência por caches diferentes obtendo dados diferentes uns dos outros.

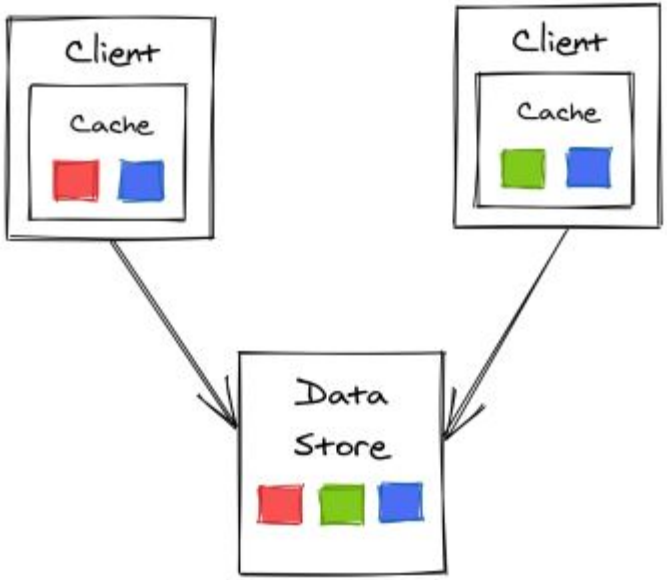


Figure 14.8: In-process cache

Caching

- Serviços (*out-of-process cache*): Cada porção de clientes se comunicam com um servidor cache.
- Esse tipo de serviço gera menos inconsistências na cache, porém gera mais problemas na recuperação dos dados caso haja uma falha no servidor.
- Aumento no tempo de acesso (latência)

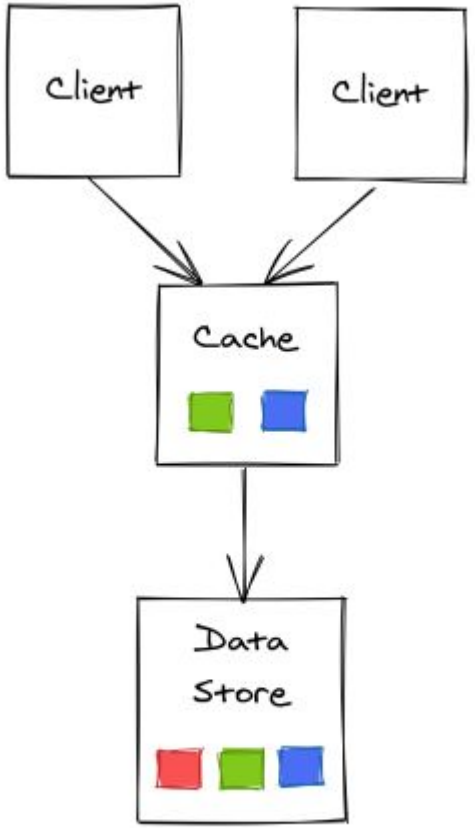


Figure 14.9: Out-of-process cache

Referências

- VITILLO, Roberto. Understanding Distributed Systems. Fevereiro, 2021.