

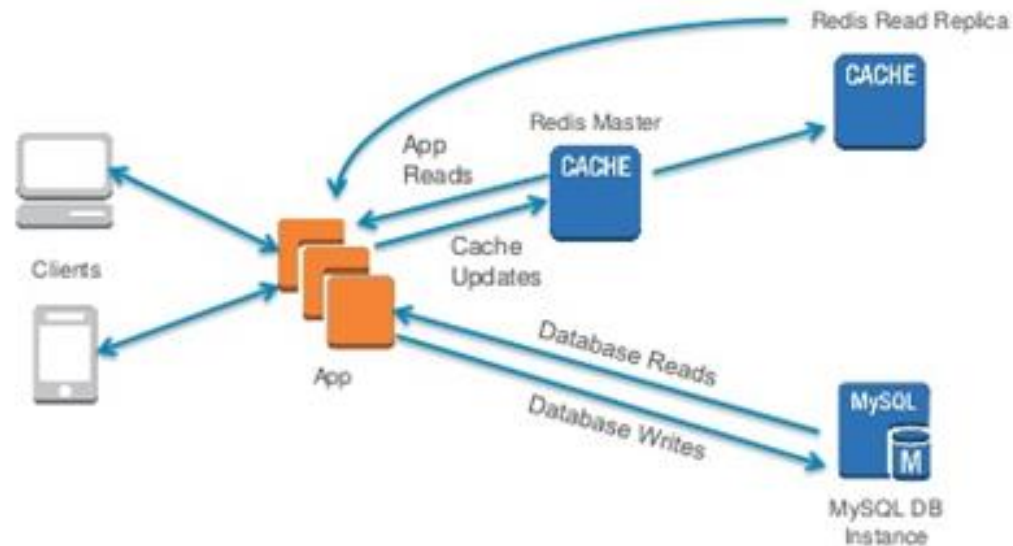
MBA⁺

Redis

- Banco de Dados NoSQL.
- Utiliza armazenamento do tipo chave-valor (*key-value*).
- Não possui *schema*, então não utiliza internamente estruturas como tabelas, registros ou colunas.
- Fácil de realizar escalonamento horizontal (é fácil e rápido aumentar o *cluster*), aumentando a performance conforme o tamanho do *cluster*.

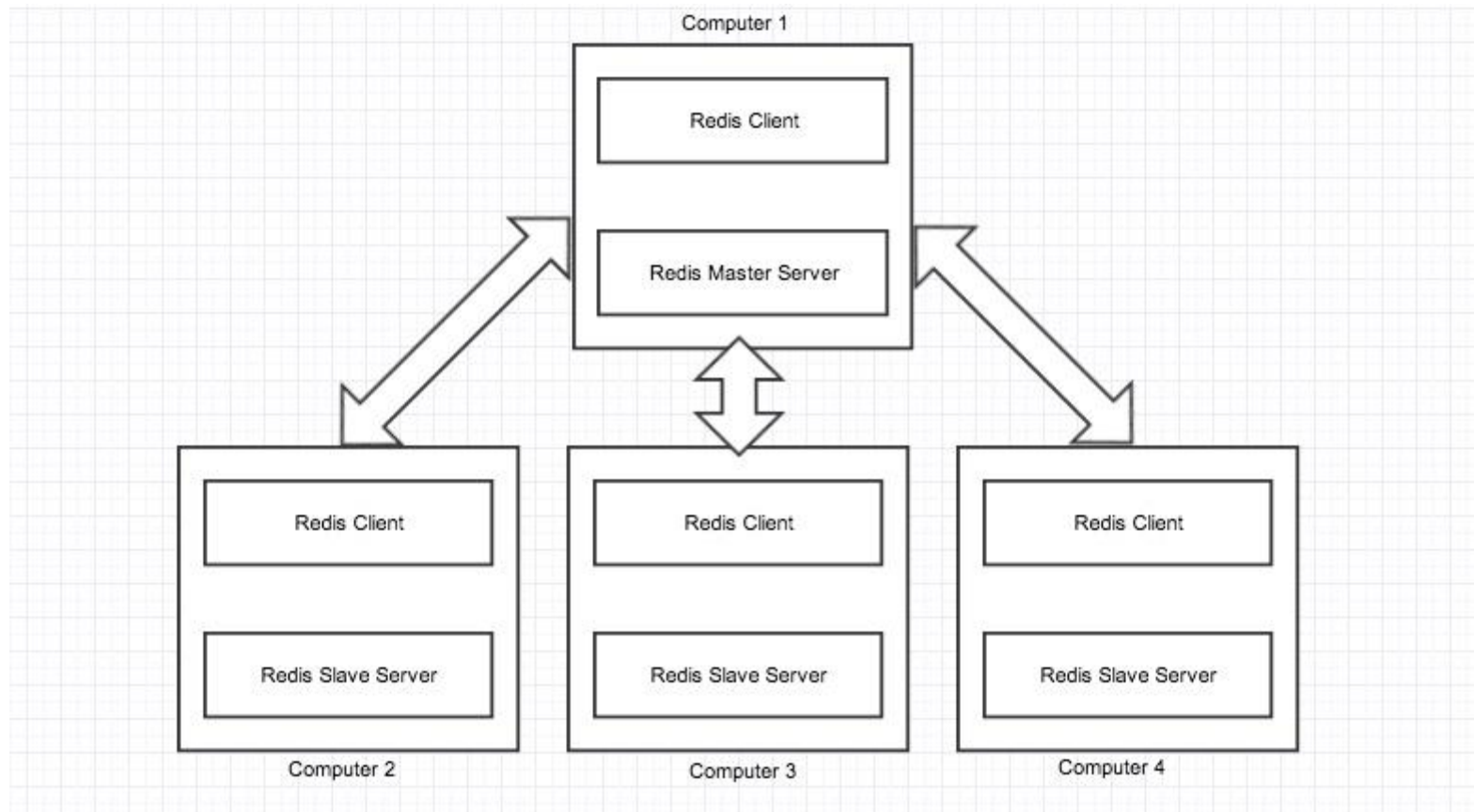
- Lembrete: bancos de dados relacionais costumam apresentar facilidade para escalonamento vertical (a performance aumenta com o aumento da capacidade de uma máquina, como aumentar I/O, CPU ou RAM).
- A estrutura interna de chave-valor permite acesso extremamente rápido as informações, mas é impeditivo para acesso a estruturas de dados mais complexas ou maiores em termos de tamanho.

Redis: Architecture



- Redis Server: responsável pelo armazenamento das informações em memória e cuida do gerenciamento das informações.
- Redis Client: responsável pelo acesso ao Server, pode ser o próprio client do Redis ou uma API de interfaceamento.
- O Server e o Client podem ou não estarem na mesma máquina.

Redis [Replicação]



Redis [Vantagens / Desvantagens]

- O armazenamento das informações do banco é feito em RAM (em memória), tornando o tempo de acesso as informações bem rápido.
- No entanto, memória RAM possui capacidade bem mais limitada do que disco e é bem mais cara, logo o Redis não serve para armazenar arquivos grandes e nem binários.
- É bem performático especialmente para gerenciar pequenos blocos de informação ou de texto.
- É possível fazer persistência do banco em disco, em troca de uma pequena penalidade em performance (*flush*).

- *Caching* (será explorado no próximo lab).
- Leitura massiva de dados (tempo de leitura em RAM é bem menor do que tempo de escrita) que não são atualizados com frequência.
- Armazenamento de dados de sessão (reduz *delay*).

- Tipos de Dados que são suportados de maneira mais “natural”:
 - String
 - List
 - Set
 - Sorted Set
 - Hash
- Lembrando que a chave deve ser sempre uma String, mas o valor pode ser de outros tipos.

- SET: armazena um chave-valor. Exemplo:

 > SET chave valor
 OK (resposta sucesso)
- MSET: armazena múltiplos pares de chave-valor. Exemplo:

 > MSET chave1 valor1 chave2 valor2 chave3 valor3
 OK

Redis [Comandos GET / MGET]

- GET: recupera um chave-valor. Exemplo:

 > GET chave
 "valor"
- MGET: recupera múltiplos pares de chave-valor. Exemplo:

 > MGET chave1 chave2 chave3
 1) "valor1"
 2) "valor2"
 3) "valor3"

MBA⁺

Copyright © 2019 Prof. Rafael Tsuji Matsuyama

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).