**Trabalho MongoDB**

**Equipe:** Isadora Moresco, Evelyn de Gois Meneses, José Bulgarelli Neto, Bianca Fernanda Leite e Guilherme Vinicius Pignatari

Apresentação Introdução

A criação de um sistema de gestão de estoque eficaz é crucial para o sucesso da operação de uma cadeia de supermercados. Este projeto foi criado para atender a essa necessidade, usando uma abordagem que alia a robustez do MongoDB à flexibilidade da linguagem Python. O MongoDB foi escolhido devido sua utilização em sala de aula e sua capacidade de lidar com grandes volumes de dados, proporcionar alta disponibilidade e escalabilidade, características fundamentais para um sistema de gerenciamento de estoque. O banco de dados foi configurado com três shards, distribuídos de forma a equilibrar a carga e otimizar o tempo de resposta.

**Projeto e Implementação**

**2. Organização do Banco de Dados**

Baseado na avaliação, o MongoDB foi dividido em três shards. A opção por sharding teve como objetivo assegurar que o sistema pudesse se expandir horizontalmente, atendendo a um aumento na demanda sem prejudicar o desempenho. Cada shard foi configurado para armazenar um subconjunto dos dados, distribuídos de acordo com a chave de sharding definida para melhorar a distribuição e a eficiência das operações de leitura e escrita.

**3. Desenvolvimento do Script para o Python**

Para criar e administrar os shards, foi criado um script em Python. O programa automatizou o processo de configuração, incluindo a criação dos shards, a definição das réplicas e a configuração da chave de sharding. O Python foi escolhido devido à sua simplicidade e poderosas bibliotecas de automação, o que facilita a interação com o MongoDB através da biblioteca PyMongo.

**4. Os resultados dos testes de desempenho foram apresentados**

Após a implantação, o sistema foi submetido a uma série de testes de desempenho para testar sua eficiência em condições operacionais ideais. Os resultados mais relevantes incluem:

Tempo de Resposta: As operações de leitura e escrita tiveram um tempo médio de 50ms, mesmo sob uma grande quantidade de consultas simultâneas.

A configuração com réplicas garantiu uma alta disponibilidade do sistema, com um tempo de inatividade próximo a zero nos testes de failover.

**Sugestões para Melhorias**

Apesar de os testes de desempenho terem mostrado resultados satisfatórios, algumas áreas para melhorar foram identificadas:

Otimização de Consultas: Revisar e otimizar as consultas MongoDB para diminuir ainda mais o tempo de resposta, sobretudo em operações complexas de agregação.

Adotar um sistema de monitoramento mais eficiente para identificar e responder prontamente a problemas de desempenho ou falhas.

Escalabilidade: Implementar escalabilidade automática para processar grandes volumes de dados.

Essa abordagem, que combina uma infraestrutura robusta com técnicas de otimização constante, garante que o sistema de gerenciamento de estoque possa evoluir para atender às crescentes demandas da cadeia de supermercados, proporcionando um serviço eficiente e confiável.

**7. Referências:**

<https://gustavo-leitao.medium.com/criando-um-cluster-mongodb-com-replicaset-e-sharding-com-docker-9cb19d456b56>