Intelligent Cleaning System

TPC-H e Benchmarking SGDs

Mestrado em Engenharia Informática
Sistemas de Gestão de Dados 2023/2024

Document version: 1.0

Bruno Sequeira 2020235721, brunosequeira@student.dei.uc.pt
Rui Santos 2020225542, rpsantos@student.dei.uc.pt

Universidade de Coimbra

1 Introdução

No mundo contemporâneo, onde os dados se tornaram um ativo extremamente valioso e omnipresente em quase todos os setores da sociedade, a capacidade de gerir grandes quantidades de dados de forma eficiente tornou-se uma prioridade fundamental. Empresas, organizações governamentais, instituições académicas e até mesmo indivíduos enfrentam o desafio de lidar com volumes massivos de dados, muitas vezes provenientes de diversas fontes e em formatos diversos.

Nesse contexto, os sistemas de gestão de base de dados (SGBDs) desempenham um papel crucial. Essas ferramentas oferecem estruturas e mecanismos para armazenar, organizar, recuperar e manipular dados de maneira eficaz. No entanto, à medida que a quantidade e a complexidade dos dados aumentam, os desafios enfrentados pelos SGBDs também crescem exponencialmente.

O objetivo deste projeto é explorar e entender como os motores de base de dados tradicionais lidam com grandes volumes de dados e consultas complexas de forma eficiente. Para alcançar esse objetivo, iremos utilizar o TPC-H como benchmarking. O TPC-H é um conjunto padrão de consultas de benchmarking projetado para avaliar o desempenho de SGBDs em cenários de análise de dados complexos.

Este projeto se concentrará em dois dos SGBDs mais amplamente utilizados e estabelecidos no mercado: MySQL e PostgreSQL. Ambos são sistemas de código aberto com uma vasta base de utilizadores e uma longa história de desenvolvimento. Ao comparar o desempenho desses dois SGBDs em relação ao benchmark TPC-H, pretendemos obter insights valiosos sobre suas capacidades e limitações no processamento de grandes quantidades de dados e consultas complexas.

Ao finalizar este projeto, esperamos não apenas aumentar nossa compreensão sobre como os motores de base de dados tradicionais lidam com grandes volumes de dados e consultas complexas, mas também fornecer insights práticos e acionáveis para otimizar o desempenho e a eficiência do gestão de dados em ambientes do mundo real.

2 Especificações e Configurações do Computador

Nesta seção, vamos nos aprofundar nas especificações e configurações do hardware do computador que utilizamos para realizar nossos testes de desempenho e análises comparativas entre diferentes sistemas de gestão de base de dados (SGBDs). A escolha adequada do hardware é fundamental para garantir resultados precisos e significativos nas nossas experiências.

(AGORA CONTIGO RUI)

3 Criação de Dados

Para realizar nossos testes de desempenho e análises comparativas entre diferentes SGBDs, como PostgreSQL e MySQL, decidimos gerar um conjunto de dados TPC-H usando o dbgen. Este conjunto de dados terá aproximadamente 25GB de tamanho, proporcionando uma carga de trabalho significativa para os sistemas em teste.

O dbgen é uma ferramenta versátil que nos permite configurar diversos parâmetros, como o fator de escala, para ajustar o tamanho do conjunto de dados gerado de acordo com nossos requisitos específicos. Além disso, ele produz dados realistas e consistentes, conforme definido pelas especificações do TPC-H.

Após instalação dessa feramenta, utilizamos o seguinte comando **dbgen -s 25**.

O dbgen gerará 8 arquivos .tbl (dados de tabela).

4 Plano de Execução

Para cada motor (mysql e postgres) iremos seguir o seguinte plano:

1. Tempos de Importação dos dados, sem as chaves(PK's e FK's).
2. Tempos de Execução de cada query, sem as chaves(PK's e FK's).
3. Tempos de Importação de criação de cada chave PK e FK.
4. Tempos de Execução da pesquisa de cada query, com as chaves (PK's e FK's).
5. Explain plan de uma query rápida e de uma lenta.
6. Explain plan mysql vs postgres para uma query lenta.

Os quatro primeiros passos esses tempos irão ser executados 5 vezes, sendo no final apresentado o valor de todos, a cold run finalizando com a média total.

5 Scripts e código utilizado

Inicialmente criamos vários scripts, dos quais foram todos submetidos na entrega. Iremos agora explicar a função de cada script.

Estes scripts estão divididos por cada motor de pesquisa, mas contêm o mesmo nome.

- **CreateDB.sh** este script irá criar uma nova base de dados, com nome "tpch-cloud".
- **CreateTables.sh** irá criar todas as 8 tabelas do tpc-h (customer, lineitem, nation, orders, part, partsupp, region e supplier).
- **ImportData.sh** irá importar os dados anteriormente gerados pelo database generator para a base de dados "tpch-cloud", colocando o resultado do tempo de execução de cada tabela no ficheiro Results/ImportTime.txt
- **DropColumnExtra.sh** irá eliminar de todas as tabelas a ultima coluna, isto porque foi necessário criar uma nova columa por tabela para que a importação fosse concluída com sucesso.
- **CreatePK.sh** criação das chaves primárias (PK's), colocando o tempo de criação no ficheiro Results/PKCreation.txt
- **CreateFK.sh** criação das chaves estrangeiras (FK's), colocando os resultados no ficheiro Results/FKCreation.txt
- **Search.py** este código em python usa multithreading para executar consultas SQL na base de dados PostgreSQL/MySQL. Ele cria várias threads(1 a 5) que executam consultas SQL em paralelo, controlando o acesso a uma lista compartilhada de consultas e registrando os resultados em um arquivo de texto. O tempo de execução de cada query é colocado num ficheiro com nome Results/ExecutionTime.txt .

6 Resultados - Tabelas

7 Análise de Resultados