

Aplicação com TPC-DS e comparação com TPC-H usado na SmartIX

Igor Alves e João Vitor Pioner



Introdução

- Artigo base: SmartIX: A database indexing based on reinforcement learning;
 - Autores: Gabriel Paludo Licks, Julia Colleoni Couto, Priscilla de Fátima Miehe, Renata De Paris, Duncan Dubugras Ruiz, Felipe Meneguzzi;
- O objetivo deste trabalho é adaptar o Smartix para aceitar um outro Benchmark, que no nosso caso é o TPC-DS;
- TPC-DS foi escolhido por ser um benchmark feito pela mesma empresa que do usado no artigo original, e por ser mais moderno que o TPC-H;



DBMS e Indexagem

- DBMS é um software feito para gerenciar bancos de dados e facilitar a organização de dados de maneira eficiente;
- Indexagem é uma técnica de gerenciamento de database que se utiliza de índices para otimizar operações com uso de chaves;
- Esta técnica facilita consideravelmente a procura de dados usando apenas uma chave dada.
- É uma técnica muito útil em melhorar a eficiência da consulta, mas se ter muitos índices lentifica as operações de inserção, remoção e atualização.



Reinforcement Learning

- É o método de aprendizagem de máquina mais semelhante à aprendizagem humana;
- É um método caracterizado pela tentativa e erro;
- Objetivo: Ter o maior número de recompensas no menor tempo possível assim criando uma política que maximiza a utilidade de cada estado;
- O agente fica “caminhando” entre estados e observando e coletando recompensas em volta;
- Reinforcement Learning é uma interpretação para aprender a política de agente ideal;



Q Learning

- Algoritmo utiliza uma tabela (Q-table) que mapeia qual o valor (ou qualidade) de uma ação em determinado estado. Assim o agente precisa apenas consultar a tabela para escolher uma ação;
- Um agente baseado em Q Learning possui um comportamento ativo;
- Os valores da Q-Table são calculados realizando uma fase de exploração dos estados. Nessa etapa, o agente escolhe ações aleatórias e com base nas recompensas recebidas a tabela é atualizada seguindo a equação:

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

Sendo s o estado, a a ação, α a taxa de aprendizagem, r é a recompensa recebida e γ um desconto aplicado aos valores de ação do novo estado, já que essas ações não têm garantia de acontecerem.



SmartIX

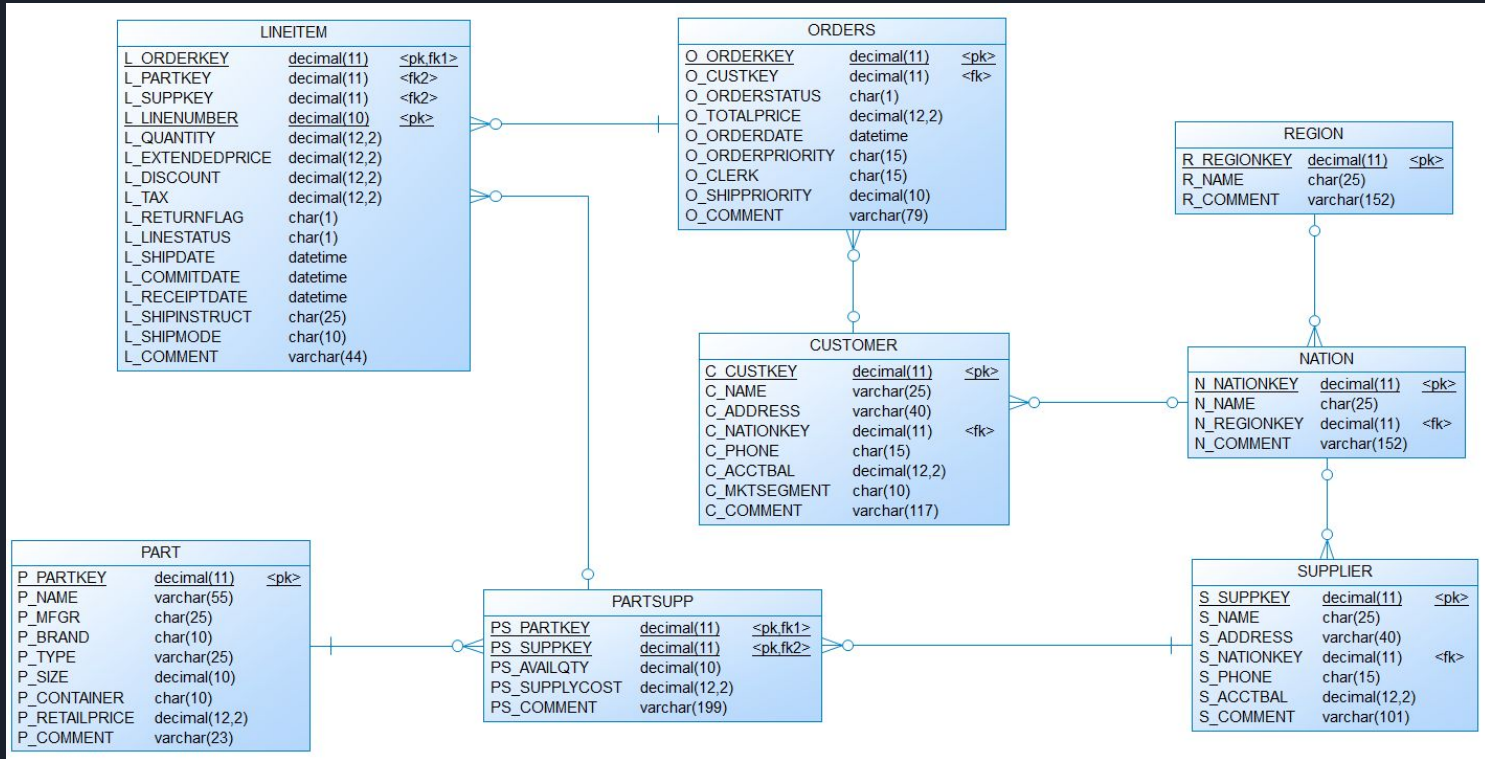
- SmartIX é um agente baseado em Q learning que automaticamente escolhe índices em banco de dados relacionais;
- SmartIX foi criado para abstrair as funções do administrador que envolve uma análise frequente em todas as colunas candidatas e verificando aqueles que provavelmente irá melhorar a configuração da database;
- O agente implementado é baseado em reinforcement learning para assim poder explorar seu espaço possível de configuração de índices;
- Para avaliar o desempenho do agente é utilizado o banco de dados provido pelo TPC-H e seu protocolo de benchmarking ;



TPC-H

- TPC-H é um benchmark para controlar a performance do banco de dados;
- Modelo relacional do TPC-H:
 - Region: continentes do mundo;
 - Nation: país do mundo;
 - Customer: uma pessoa que compra parte do supridor;
 - Supplier: supridor que vende as peças;
 - Part: a peça disponível pelo supridor;
 - Partsupp: relação entre a parte e o supridor;
 - Orders: dados relacionados a compra de peças;
 - Lineitem: todos os detalhes dos pedidos de cada client

TPC-H - Representação do ER



TPC-H - Cálculo de Performance

- Para avaliar o desempenho do agente é usado a medida QphH(consultas por hora):

$$QphH@Size = \sqrt{Power@Size \times Throughput@Size}$$

$$Power@Size = \frac{3600}{\sqrt[24]{\pi_{i=1}^{22} QI(i,0) \times \pi_{j=1}^2 RI(j,0)}} \times SF$$

Calcula a velocidade que o DBMS responde a consulta

$$Throughput@Size = \frac{S \times 22}{T_s} \times 3600 \times SF$$

Calcula a habilidade do sistema em fazer a muitas consultas no menor tempo possível



TPC-DS

- O TPC-DS usa as vantagens do TPC-H e do TPC-R para transformá-los num benchmark DSS moderno;
- Normalmente aplicado na indústria que deverá transformar dados externos e operacionais em inteligência de indústrias e modela tarefas de suporte de decisão de fornecedores de produtos de vendas típicas;
- Criado para ajudar na leitura em relacionar intuitivamente aos componentes do benchmark, sem acompanhar o segmento da indústria para minimizar a relevância do benchmark;

TPC-DS - Diagrama ER



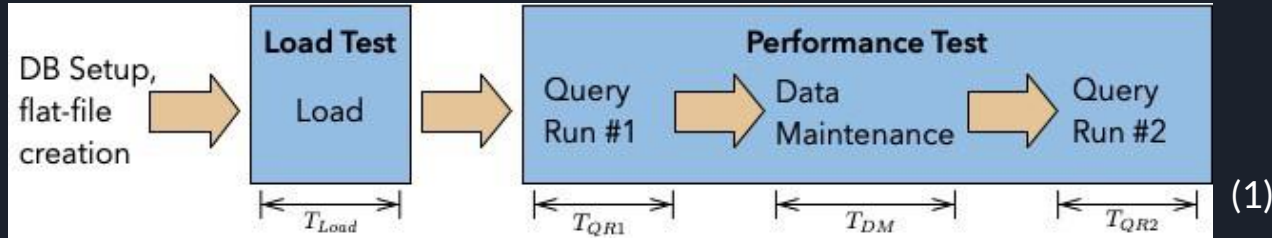
- Duas Fact Tables:
Store_Sales e
Store>Returns
- Dois Associated
Dimension Tables:
Customer e Store.



TPC-DS - Dataset e Workload

- Tanto TPC-DS quanto TPC-H se utilizam do conceito de fator de escala (SF) para controlar o tamanho da Database;
- O TPC-DS possui escalas de domínio e da tupla;
 - Escala de domínio se refere a escala das dimension tables que normalmente não são escalonadas de maneira linear;
 - Escala de tupla se refere aos números de tuplas nos fact tables;
- O TPC-DS possui duas simulações do workload separadas: O Query workload e o Data Maintance workload;
 - Query faz as operações na database;
 - Data Maintance sincroniza os dados da database com os dados registrados dentro do TPC-DS.

TPC-DS - Execução e métrica



$$Q_{phDS@SF} = SF \cdot 3600 \cdot \frac{198 \cdot S}{(T_{QR1} + T_{DM} + T_{QR2} + 0.01 \cdot S \cdot T_{Load})} \quad (2)$$

1. Representa o processo de medida do tempo gasto em cada etapa;
2. Fórmula para medir o desempenho da database ;
 - a. SF é o fator escala ;
 - b. Os valores T são os tempo que foram medidos no processo(1);
 - c. S é o valor de query streams per query run;



Descrição do Trabalho


- No plano original: Montar uma database, linkar a database com o TPC-DS, popular a database com os dados teste do TPC-DS, linkar o banco e o benchmark no código e por fim executar e começar a treinar a IA;
- Para a tentativa, foi usado o Ubuntu 18.04 LTS para o SO;
- Na realidade:
 - Para montar a database foi usado o Postgres 10;
 - Inserção das tabelas do TPC-DS para dentro da database;
 - Linkar o Postgres no TPC-DS
 - Infelizmente, o teste parou aqui;
 - Muitos bugs(relatados no Caderno de Pesquisa)

Outros testes

TPC Benchmark™ DS Metrics

Total System Cost (RMB)	TPC-DS Throughput (QphDS@10000GB)	Price/Performance (RMB/QphDS@10000GB)	Availability Date
¥1,126,006.68	18,998,559	¥0.06	As of Publication

- Teste do Alibaba com o TPC-DS
- Escala de 10.000GB
- 4 Query streams
- 396 Queries

	Dell PowerEdge R6415 using Exasol 6.2	TPC-H Rev. 2.18.0 TPC-Pricing Rev. 2.4.0		
		Report Date July 8, 2019		
Total System Cost	Composite Query per Hour Metric		Price / Performance	
\$809,230	8,667,578 QphH@10000GB		\$ 0.10 \$ / QphH@10000GB	
Database Size	Database Manager	Operating System	Other Software	Availability Date
10,000GB	Exasol 6.2	CentOS 7.6	None	July 8, 2019

- Teste da Dell com o TPC-H
- Escala de 10.000GB
- 12 Query streams



Conclusão

- Sobre o projeto, infelizmente não teve muito para mostrar, mas acreditamos que criamos um começo para resolver o problema proposto aqui para uma futura tentativa;
- Em uma eventual nova tentativa, usar uma configuração de máquina diferente e talvez uma aproximação diferente para resolver o problema;

Obrigado!

