Administração de Bases de Dados

Engenharia Informática – Universidade do Minho

Trabalho Prático – 2019/2020

Os grupos de trabalho devem ser constituídos por 3 (três) elementos, todos inscritos na Unidade Curricular. Será disponibilizado saldo na *Google Cloud Platform* para realizar este trabalho a cada um dos grupos. O resultado do trabalho é um relatório escrito. O relatório deve omitir considerações genéricas sobre as ferramentas utilizadas, focando a apresentação e justificação dos objetivos atingidos. O relatório deve ser entregue por apenas um dos elementos do grupo, identificando também o nome e número de todos os outros na capa. A entrega do relatório é feita na área da Unidade Curricular no *e-Learning*. A data limite é 3 de janeiro de 2020.

1 Contexto

O trabalho prático consiste na configuração, otimização e avaliação do *benchmark* TPC-C com alguns dados e interrogações adicionais. O TPC-C simula um sistema de bases de dados de uma cadeia de lojas, suportando a operação diária de gestão de vendas e stocks. As interrogações analíticas adicionais, baseadas na adaptação do TPC-H e listadas no Anexo A, poderão ser corridas ocasionalmente. A descrição dos *benchmarks* e da respetiva implementação está disponível no Anexo B.

2 Objetivos

Depois de instalar e configurar o benchmark *TPC-C*, obtendo uma **configuração de referência** em termos de número de *warehouses*, número de clientes e *hardware*, os objetivos do trabalho são:

1. Usando a configuração de referência, otimizar o desempenho da carga transacional tendo em conta, principalmente, os parâmetros de configuração do PostgreSQL.
2. Usando a configuração de referência, otimizar o desempenho das interrogações analíticas tendo em conta, principalmente, os respetivos planos e os mecanismos de redundância que estão a ser usados.
3. Propor e testar uma configuração usando replicação ou processamento distribuído com qualquer uma das ferramentas estudadas. Esta configuração deve atingir uma escala superior à da configuração de referência, com a qual deve ser comparada.

3 Notas

- Como medidas de desempenho da carga operacional (TPC-C) deve considerar-se principalmente o **debito máximo** atingível. Nas operações analíticas, deve considerar-se o **tempo de resposta**.

- Ao escolher a configuração de referência, devem ter em conta o saldo disponível na *Google Cloud* para efetuar todos os testes e o tempo que cada teste demora a correr. Se escolherem uma configuração demasiado grande, não conseguirão fazer todos os testes necessários. Por outro lado, se escolherem uma configuração demasiado pequena, vão ter poucas oportunidades de otimização.

- Poderão modificar as interrogações SQL e o código Java para atingir os objetivos 2 e 3. Devem explicar no relatório em que medida essas alterações preservam o funcionamento da aplicação original.

- Nos objetivos 2 e 3 não poderão considerar todas as otimizações possíveis nas suas várias combinações... Devem focar-se nas que consideram mais prometedoras e que mais vos interessam, justificando no relatório essas opções.

- Nos objetivos 2 e 3 é aceitável que não consigam uma melhoria de desempenho. Nesse caso, devem explicar porque é que a configuração de referência já era ótima. Por outro lado, a simples apresentação de uma melhoria de desempenho, não justificada, não é muito interessante.

- A utilização de ferramentas de monitorização e diagnóstico do PostgreSQL e do sistema operativo (e.g., pgbadger, iostat) valoriza o trabalho.

- A automatização da instalação e execução do *benchmark* permitirá obter resultados em maior quantidade, permitindo uma análise mais profunda e valorizando o trabalho.

- Devem também procurar estratégias para poupar recursos na *Google Cloud*, por exemplo, armazenando os dados em *Cloud Storage* e reutilizando-os, em vez de re-executar o `load.sh`. Tenham o cuidado de escolher as configurações mais baratas na *Google Cloud* (i.e., regiões nos EUA, *preemptible*, ...) e de não deixar máquinas virtuais ativas a consumir recursos desnecessariamente!

1- No ficheiro "EscadaTPC-C/pom.xml" nas linhas 16 e 17 substituir o '6' por '8'.

2- Correr na pasta "EscadaTPC-C" o comando "mvn package".

3- No ficheiro "EscadaTPC-C/tpc-c-0.1-SNAPSHOT-tpc-c/tpc-c-0.1-SNAPSHOT/etc/database-config.properties" substituir 'alfranio' pelo nosso nome de utilizador da Base de Dados ('joaomarques').

4-No ficheiro "EscadaTPC-C/tpc-c-0.1-SNAPSHOT-tpc-c/tpc-c-0.1-SNAPSHOT/etc/workload-config.properties" modificar todos os valores relevantes para os testes de carga.

5- Na pasta base do git copiar o ficheiro "start_db.sh" para a mesma diretoria onde se encontra o "EscadaTPC-C", de seguida abrir o ficheiro "start_db.sh" e alterar o conteúdo da variável 'PROJECT_DIR' para o caminho completo desde o '/' até ao 'EscadaTPC-C', incluindo este último no conteúdo da variável. Para funcionar corretamente o caminho especificado não pode conter espaços.

6- Executar o comando "initdb -D data".

7- Executar o comando "postgres -D data -k.".

8- Abrir um novo terminal e navegar até à diretoria onde se encontra a pasta 'EscadaTPC-C' e o ficheiro 'start_db.sh'.

9- Executar o comando "bash start_db.sh".

10- Caso tenha aparecido o erro "./load.sh: line 4: /bin/java: No such file or directory" e o erro "./run.sh: line 6: /bin/java: No such file or directory" aceder a ambos os ficheiros na diretoria "EscadaTPC-C/target/tpc-c-0.1-SNAPSHOT-tpc-c/tpc-c-0.1-SNAPSHOT/" e nos ficheiros alterar para a diretoria correta do JAVA. Depois navegar até à diretoria "EscadaTPC-C/target/tpc-c-0.1-SNAPSHOT-tpc-c/tpc-c-0.1-SNAPSHOT/" e correr os comandos "./load.sh" e "./run.sh".

11- Se os comandos ".load.sh" e "./run.sh" derem erro verificar se a Base de Dados se encontra ativa.

A Interrogações analítcas

A.1

```
select  su_name, su_address
from  supplier, nation
where  su_suppkey in
          (select  mod(s_i_id * s_w_id, 10000)
           from     stock, order_line
           where    s_i_id in
                    (select i_id
                     from item
                     where i_data like 'c%')
             and ol_i_id=s_i_id
             and extract(second from ol_delivery_d) > 50
           group by s_i_id, s_w_id, s_quantity
           having   2*s_quantity > sum(ol_quantity))
  and su_nationkey = n_nationkey
  and n_name = 'GERMANY'
order by su_name;
```

A.2

```
select  i_name,
 substr(i_data, 1, 3) as brand,
 i_price,
 count(distinct (mod((s_w_id * s_i_id),10000))) as supplier_cnt
from  stock, item
where  i_id = s_i_id
 and i_data not like 'z%'
 and (mod((s_w_id * s_i_id),10000) not in
(select su_suppkey
 from supplier
 where su_comment like '%bean%'))
group by i_name, substr(i_data, 1, 3), i_price
order by supplier_cnt desc;
```

A.3

```
select  n_name,
 sum(ol_amount) as revenue
from  customer, orders, order_line, stock, supplier, nation, region
where  c_id = o_c_id
  and c_w_id = o_w_id
  and c_d_id = o_d_id
  and ol_o_id = o_id
  and ol_w_id = o_w_id
  and ol_d_id= o_d_id
  and ol_w_id = s_w_id
  and ol_i_id = s_i_id
  and mod((s_w_id * s_i_id),10000) = su_suppkey
  and ascii(substr(c_state,1,1))-ascii('a') = su_nationkey
  and su_nationkey = n_nationkey
  and n_regionkey = r_regionkey
  and r_name = 'EUROPE'
group by n_name
order by revenue desc;
```

A.4

```
select  c_last, c_id o_id, o_entry_d, o_ol_cnt, sum(ol_amount)
from  customer, orders, order_line
where  c_id = o_c_id
 and c_w_id = o_w_id
 and c_d_id = o_d_id
 and ol_w_id = o_w_id
 and ol_d_id = o_d_id
 and ol_o_id = o_id
group by o_id, o_w_id, o_d_id, c_id, c_last, o_entry_d, o_ol_cnt
having  sum(ol_amount) > 200
order by sum(ol_amount) desc, o_entry_d
```

B Referências

1. Código fonte e instruções de instalação:
   https://github.com/jopereira/EscadaTPC-C
   https://gist.github.com/jopereira/4086237

2. Documentação:
   http://www.tpc.org/tpcc/
   https://db.in.tum.de/research/projects/CHbenCHmark/index.shtml?lang=en