Instituto Federal Goiano - Campus Ceres
Bacharelado em Sistemas de Informação
Prof. Me. Ronneesley Moura Teles

Anny Karoliny Moraes Ribeiro
Eduardo Oliveira Silva
Gleyson Israel Alves
José Gilvan Jacinto Junior
Lara Caroline Damaceno Faria

Estudo da Diferença de Performance com Índices em Tabelas

Novembro
2017

Sumário

Estudo da diferença de performance com índices em tabelas

1 Introdução

Ao trabalhar com um banco de dados é imprescindível realizar consultas em uma grande quantidade de registros que demora um certo período de tempo para retornar um resultado, o desempenho é crucial nesse processo. Por esse motivo as pessoas envolvidas com o banco de dados se preocupam muito com o tempo de resposta das consultas feitas no banco de dados.

Existem muitos métodos para assegurar um melhor tempo de resposta sobretudo em tabelas com grandes quantidades de dados registrados. Como por exemplo selecionar só as colunas essenciais, assim diminuindo os dados devolvidos, outra possibilidade é o uso de índices.

Empregar o uso de índices especialmente nos campos numéricos das tabelas auxilia o gerenciador do banco de dados a encontrar com mais facilidade o dado procurado.

2 Entendendo o conceito de Índices

Índices são considerados como a organização de temas em listagem, assuntos, palavras e tópicos, e também um indicador de sintoma, um fator de referência que serve para comparar e explicar a situação ou condição.

O uso do índice em banco de dados vai promover uma melhoria em seu desempenho. Os principais sinônimos de índice são: lista, tabela, rol, relação, padrão e índex.

Um tipo comum de índice é uma lista ordenada, de determinados valores de uma coluna e tabela com ponteiros, que associam as linhas a cada valor. O índice vai permitir que as informações nas linhas de uma tabela tenham um desempenho rápido na sua localização, utilizado para encontrar registros com um determinado valor da coluna de forma rápida.

Sem uso de índice o MySQL terá de fazer leitura em toda tabela até que encontre os registros essenciais que se busca. Em relação a tabela ser grande será o seu custo também será maior

Se a tabela possuir índice para determinadas colunas, o MySQL terá rapidez em conseguir uma posição para busca entre os arquivos de dados, e se necessário fazer a varredura de todos os registros.

Em uma tabela que apresenta 1000 registros, teremos uma velocidade 100 vezes maior do que teríamos caso fosse feito uma leitura em todos os registros em sequência.

Pode-se entender as chaves-primárias como índices, que permitem buscas de forma rápida e com eficiência baseada em um valor que não tem repetição. Pode-se também usar o mesmo conceito nas colunas que não sejam chaves tendo assim melhora no desempenho de busca usando aquela coluna.

Antes do uso de índice é preciso pensar na possibilidade de usá-lo ou não, quando em uma tabela temos muitos índices e a mesma sofre muitas alterações, remoções e inserções todos os índices deverão ser atualizados de acordo com a nova situação.

Esse processo acaba ficando crítico principalmente em tabelas com grandes quantidades de registros.

3 Vantagens e Desvantagens do uso de índices no banco de dados

3.1 Vantagens

- Melhoria de performance na consulta e acesso: a velocidade de acesso aos registros reduz. Os índices são ligados em forma de árvore binária (existem outros tipos) conserva os registros ordenados, conseguindo assim realizar uma consulta binária de tal modo que encontrar as informações disponíveis em índice dentro de uma escala maior. Aproximadamente 14% mais rápido, o que significa diante da quantidade imensa de dados que temos hoje.

- Busca de dados específicos de forma mais rápida: trabalhando com índice é possível que em apenas uma consulta já baste para obter as informações desejadas não necessitando consultar a tabela de dados.

- Possibilita o acesso aos registros ordenados sem necessidade de realizar a ordenação: em uma busca os dados naturalmente são ordenados, uma vez que sabe-se que vai se usar um acesso aos dados sequencialmente o índice tem muita serventia.

- Facilita na prevenção de informações chaves duplicadas: de maneira que a procura é veloz e fácil localizar chaves duplicadas se torna uma tarefa quase que banal (quase não gasta tempo).

3.2 Desvantagens

- Diminui o desempenho da escrita no banco de dados: sempre que houver necessidade de modificar (inserir, alterar, deletar) uma chave forçará a escrever o índice. Se o registro modificado tem várias chaves (índices) a troca do índice afeta a todos, todavia terão que ser mudados.

- Eleva o gasto de armazenamento do banco de dados: uma tabela extra de índices ocupará mais memória e disco.

- Amplifica a inevitabilidade de manutenção interna do banco de dados: dependendo da implantação é corriqueiro o fato de que chaves sejam deixadas de lado a medida que são alteradas. Adiante também de outros elementos que o DBA deverá se preocupar.

- Arrisca-se a queda na eficiência das consultas: não tem-se uma garantia de que toda e qualquer consulta será mais veloz com a utilização dos índices diante da operação extra de acesso ao índice anteriormente ao acesso aos registros principais, sendo assim há a possibilidade dessa operação anterior ao acesso aos dados somada ao acesso ser maior que o acesso direto.

4 Teste Proposto

A partir de toda a teoria estudada sobre índices, foi proposto um teste de desempenho de dois banco de dados, um com índices e outro sem índices.

Características dos bancos de dados:

- Cada banco é composto por uma tabela chamada: "usuarios".
- Cada uma das tabelas possui dois campos: "id" e "nome".
- O banco de dados 1 possui os campos "id" e "nome" com índices.
- Em ambas as tabelas, 10400000 dados foram inseridos.
- Os dados dos dois bancos são indênticos.

Método utilizado para o teste:

- Foram utilizados 100000 "selects" executados 30 vezes em cada banco de dados.
- Foi medido o tempo de cada uma das execuções desses "selects".
- A partir do tempo foi tirado a média para a comparação do desempenho do banco de daados com índice e do banco de dados sem índice.

5 Códigos Utilizados

- Criando o usuário de teste:

```
create user if not exists 'admin_teste'@'localhost' identified by
    'banco_teste';

grant all on bd_teste1.* to 'admin_teste'@'localhost';
grant all on bd_teste2.* to 'admin_teste'@'localhost';
```

recursos/codigo/criacao_usuario.sql

- Criando o banco de dados 1 com a tabela usuários:

```
CREATE DATABASE bd_teste1;
USE bd_teste1;

CREATE TABLE NOMES(
ID INT NOT NULL AUTO_INCREMENT,
NOME VARCHAR (100) NOT NULL,
PRIMARY KEY(ID)
);
```

recursos/codigo/criacao_bd1.sql

- Criando os índices para os campos "id" e "nome" do banco de dados 1:

```
CREATE INDEX index_id ON bd_teste1.usuarios(id);
CREATE INDEX index_nome ON bd_teste1.usuarios(nome);
```

recursos/codigo/criacao_index.sql

- Criando o banco de dados 2:

```sql
CREATE DATABASE bd_teste2;
USE bd_teste2;

CREATE TABLE NOMES(
ID INT NOT NULL AUTO_INCREMENT,
NOME VARCHAR (100) NOT NULL,
PRIMARY KEY(ID)
);
```

recursos/codigo/criacao_bd2.sql

- Inserindo os dados no banco com java:

```java
package inserir_banco;

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Inserir_banco {

    public static void main(String[] args) throws SQLException, ClassNotFoundException {
        Class.forName("com.mysql.jdbc.Driver");

        // TODO code application logic here
        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/bd_teste1", "root", "251180");

        PreparedStatement pstmt = con.prepareStatement("insert into usuarios (nome) values(?)");

        String nomes[] = {"Wilson", "Cleuza", "Weslley", "Renata", "Lourdes", "William", "Welton", "Willy", "Wallace", "Wermeson", "Warley", "Warlen", "Kenia", "Cleusa", "Maria", "Simone", "Marilia", "Allan", "Eliany", "Emilly", "Karla", "Clara", "Julia", "Lara", "Adrey", "jonatas", "Gilvan", "Jose", "Eduardo", "Abrao", "Absalao", "Abssilao", "Acacio", "Acilino", "Acilio", "Acursio", "Adail", "Adalberto", "Adalsindo", "Adalsino", "Adamantino", "Adamastor", "Adao", "Adauto", "Adauto", "Adelindo","Adelmiro", "Adelmo", "Ademar", "Ademir", "Adeodato", "Aderico", "Aderio", "Aderito", "Barac", "Barao", "Barbaro", "Barcino", "Barnabe", "Bartolomeu", "Bartolomeu", "Basilio", "Balbino", "Baldemar", "Baldomero", "Balduino", "Baltasar", "Baptista", "Bassarme", "Bastiao", "Carlo", "Carlos", "Carmerio", "Carmim", "Carsta", "Casimiro", "Cassiano", "Caetano", "Caio", "Caleb", "Calisto", "Calvino", "Camilo", "Candido", "Canto", "Cassio", "Davi", "David", "Davide", "Decimo", "Decio", "Deivid", "Dejalme", "Delcio", "Delfino", "Delio", "Delmano", "Delmar", "Delmiro", "Demetrio", "Dacio", "Dagmar", "Damas", "Damasceno", "Damiao", "Daniel", "Danilo", "Dante", "Darcio", "Dario", "Dario", "Dener", "Denil", "Denis", "Deodato", "Deolindo", "Dercio", "Deusdedito", "Dhruva", "Diamantino", "Didaco", "Diego", "Edmundo", "Edmur", "Edo", "
```

```
Eduardo", "Eduartino", "Eduino", "Edvaldo", "Edvino", "Egas",
"Egidio", "Egil", "Eladio", "Eleazar", "Eleuterio", "Dario", "
Dario", "Dener", "Denil", "Denis", "Deodato", "Deolindo", "
Dercio", "Deusdedito", "Dhruva", "Diamantino", "Didaco", "
Diego", "Edmundo", "Edmur", "Edo", "Eduardo", "Eduartino", "
Eduino", "Edvaldo", "Edvino", "Egas", "Egidio", "Egil", "
Eladio", "Eleazar", "Eleuterio", "Elgar", "Eli", "Eberardo", "
Eda", "Eder", "Edgar", "Fabiano", "Fabiao", "Fabio", "Fabricio
", "Falcao", "Falco", "Faustino", "Fausto", "Feliciano", "
Felicio", "Felicissimo", "Felisberto", "Felismino", "Felix", "
Feliz", "Ferdinando", "Fernandino", "Fernando", "Fernao", "
Gabinio", "Gabino", "Gabriel", "Galiano", "Galileu", "Gamaliel
", "Garcia", "Garibaldo", "Gascao", "Gaspar", "Gastao", "
Gaudencio", "Gavio", "Gedeao", "Haraldo", "Haroldo", "Hazael",
 "Heber", "Heitor", "Heldemaro", "Helder", "Heldo", "Heleno",
"Iag", "Iago", "Ian", "Ianis", "Iberico", "icaro", "Idalecio",
 "Idalio"};
24      String sobreNomes[] = {"OLiveira", "Coelho", "Silva", "
Rosa", "Pereira", "Godofredo", "Jesus", "Almeida", "Mendoca",
"Delfim", "Delfino", "Delio", "Delmano", "Delmar", "Delmiro",
"Demetrio", "Dacio", "Dagmar", "Damas", "Damasceno", "Damiao",
 "Daniel", "Danilo", "Dante", "Darcio", "Dario", "Dario", "
Dener", "Denil", "Denis", "Deodato", "Deolindo", "Dercio", "
Deusdedito", "Dhruva", "Diamantino", "Didaco", "Diego", "
Edmundo", "Edmur", "Edo", "Eduardo", "Eduartino", "Eduino", "
Edvaldo", "Edvino", "Egas", "Egidio", "Egil", "Eladio", "
Eleazar", "Eleuterio", "Elgar", "Eli", "Eberardo", "Eda", "
Eder", "Edgar", "Fabiano", "Fabiao", "Fabio", "Fabricio", "
Falcao", "Falco", "Faustino", "Fausto", "Feliciano", "Felicio"
, "Felicissimo", "Felisberto", "Felismino", "Felix", "Feliz",
"Ferdinando", "Fernandino", "Fernando", "Fernao", "Gabinio", "
Gabino", "Gabriel", "Galiano", "Galileu", "Gamaliel", "Garcia"
, "Garibaldo", "Gascao", "Gaspar", "Gastao", "Gaudencio", "
Gavio", "Gedeao", "Haraldo", "Haroldo", "Hazael", "Heber", "
Heitor", "Heldemaro", "Helder", "Heldo", "Heleno", "Iag", "
Iago", "Ian", "Ianis", "Iberico", "icaro", "Idalecio", "Idalio
", "Abrao", "Absalao", "Abssilao", "Acacio", "Acilino", "
Acilio", "Acursio", "Adail", "Adalberto", "Adalsindo", "
Adalsino", "Adamantino", "Adamastor", "Adao", "Adauto", "
Adauto", "Adelindo","Adelmiro", "Adelmo", "Ademar", "Ademir",
"Adeodato", "Aderico", "Aderio", "Aderito", "Barac", "Barao",
"Barbaro", "Barcino", "Barnabe", "Bartolomeu", "Bartolomeu", "
Basilio", "Balbino", "Baldemar", "Baldomero", "Balduino", "
Baltasar", "Baptista", "Bassarme", "Bastiao", "Carlo", "Carlos
", "Carmerio", "Carmim", "Carsta", "Casimiro", "Cassiano", "
Caetano", "Caio", "Caleb", "Calisto", "Calvino", "Camilo", "
Candido", "Canto", "Cassio", "Davi", "David", "Davide", "
Decimo", "Decio", "Deivid", "Dejalme", "Delcio", "Zacarias", "
Zaido", "Zaido", "Zairo", "Zaqueu", "Zadeao", "Zadoque", "
Zafriel", "Zalman", "Zarco", "Zared", "Zarao", "Zebadiah"};
25
26      for (int i = 0; i < nomes.length; i++){
27          for (int j = 0; j < sobreNomes.length; j++){
28              for (int k = 0; k < sobreNomes.length; k++){
29                  String nome = nomes[i] + " " + sobreNomes[j] +
 " " + sobreNomes[k];
30
31                  pstmt.setString(1, nome);
32                  pstmt.executeUpdate();
33              }
```

```
34                  }
35              }
36
37              con.close();
38
39          }
40
41  }
```

recursos/codigo/inserir_banco.java

(É um script que leva muito tempo para ser executado.)

- Código que executa os selects no banco de dados 1, que possui índices:

```
1  package inserir_banco;
2
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5  import java.sql.SQLException;
6  import java.sql.Statement;
7  import java.util.Date;
8
9  public class AnaliseTempo1 {
10     public static void main(String[] args) throws SQLException,
       ClassNotFoundException {
11         Class.forName("com.mysql.jdbc.Driver");
12         String nomeBuscado = "Antonio Moreira Almeida";
13
14         Connection con = DriverManager.getConnection(
15             "jdbc:mysql://localhost:3306/bd_teste1", "root", "
       251180");
16
17         Statement stmt = con.createStatement();
18
19         long tempoInicial = new Date().getTime();
20
21         for (int i = 0; i < 100000; i++){
22             String sqlConsulta = "SELECT * FROM bd_teste1.usuarios
       "
23                     + " where nome = '" + nomeBuscado +" ' ";
24
25             stmt.executeQuery(sqlConsulta);
26         }
27
28         long tempoFinal = new Date().getTime();
29
30         con.close();
31
32         long duracao = tempoFinal - tempoInicial;
33         System.out.println("* " + duracao + " *");
34     }
35
36 }
```

recursos/codigo/analise_tempo1.java

- Código que executa os selects no banco de dados 2:

```
1  package inserir_banco;
2
```

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;

public class AnaliseTempo2 {
    public static void main(String[] args) throws SQLException,
    ClassNotFoundException {
        Class.forName("com.mysql.jdbc.Driver");
        String nomeBuscado = "Antonio Moreira Almeida";

        Connection con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/bd_teste2", "root", "
    251180");

        Statement stmt = con.createStatement();

        long tempoInicial = new Date().getTime();

        for (int i = 0; i < 100000; i++){
            String sqlConsulta = "SELECT * FROM bd_teste2.usuarios
    "
                    + " where nome = '" + nomeBuscado +" ' ";

            stmt.executeQuery(sqlConsulta);
        }

        long tempoFinal = new Date().getTime();
        con.close();
        long duracao = tempoFinal - tempoInicial;
        System.out.println("* " + duracao + " *");
      }

}
```

recursos/codigo/analise_tempo2.java

6 Resultados

- Média de tempo de consulta sem índices:
- Média de tempo de consulta com índices:

7 Conclusão

Numa sociedade moderna onde a quantidade de dados aumenta significativamente a cada dia, é de mera importância o rápido acesso e consulta a eles.

A utilização de índices e técnicas de otimização promete uma boa performance no acesso e nas consultas ao banco de dados. Seria um devaneio achar que os índices são a solução para todos os dilemas, sendo assim deve-se utilizá-los quando realmente houver necessidade.

Cabe frisar que há outros tipos de índices (além da árvore binária que é o mais comum) adequados para cada circunstância, assim como o hash no qual a chave é definida por um índice posicional (comum em memória) ou índices invertidos que usam indexação.

8 Referências Bibliográficas

Índices MySQL: Otimização de consultas. Disponível em: www.linhadecodigo.com.br/artigo/3620/indices-mysql-otimizacao-de-consultas.aspx. (*Acesso em 11/2017*)

Entendendo e usando índices - Parte 1. Disponível em: www.devmedia.com.br/entendendo-e-usando-indices-parte-1/6567. (*Acesso em 11/2017*)

Uso de índice em bando de dados. Disponível em: https://www.portaleducacao.com.br/conteudo/artigos/idiomas/uso-de-indices-em-banco-de-dados/46287. (*Acesso em 11/2017*)

Significado de Índice. Disponível em: https://www.significados.com.br/indice/. (*Acesso em 11/2017*)

Quais as vantagens e desvantagens do uso de índices em base de dados?. Disponível em: pt.stackoverflow.com/questions/35088/quais-as-vantagens-e-desvantagens%c3\%adndices-em-base-de-dados. (*Acesso em 11/2017*)