PRAKTIKUM DATABASE MANAJEMENT SYSTEM

TUNING DATABASE SYSTEM

Dosen Pengampu :

Arief Ichwani S.Kom, M.Cs

Disusun oleh :

Afdi Fauzul Bahar

NIM. 14117149

MBD – RD

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI PRODUKSI INDUSTRI DAN INFORMASI

INSTITUT TEKNOLOGI SUMATERA

TAHUN 2019

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABLE

BAB I
PENDAHULUAN

1.1 Tuning: Indexing

Index tuning merupakan bagian optimasi basis data dengan memilih dan membuat indeks. Tujuan indeks tuning yaitu untuk mengurangi waktu eksekusi kueri / pencarian. Saat basis data dibuat tanpa menggunakan index, maka kinerja server basis data dapat menurun secara drastis. Index tuning melibatkan kueri berdasarkan indeks. indeks dibuat secara otomatis secara *realtime* / saat itu juga. Tidak diperlukan tindakan lebih oleh pengguna basis data untuk penyetelan indeks.

Indeks dalam database serupa dengan indeks dalam buku. Di suatu buku, satu indeks memungkinkan anda untuk menemukan informasi dengan cepat tanpa membaca seluruh buku. Di suatu database, indeks memungkinkan program database menemukan data di suatu tabel tanpa menelusuri seluruh tabel. Satu indeks di suatu buku adalah daftar kata-kata dengan angka-angka halaman berisi masing-masing kata. Satu indeks di suatu database adalah daftar data tertentu dari tabel dengan lokasi penyimpanan baris dalam tabel berisi masing-masing nilai. indeks dapat diciptakan dimanapun suatu kolom atau suatu kombinasi dari kolom di suatu tabel dan diterapkan dalam wujud B-trees. Satu indeks berisi masukan dengan satu atau lebih kolom (kunci pencarian) dari masing-masing baris di suatu tabel. B-tree disortir di kunci pencarian, dan dapat dicari secara efisien di setiap subset yang terdepan dari kunci pencarian. Sebagai contoh, satu indeks di kolom A, B, C dapat dicari secara efisien di A, A, B, dan A, B, C.Sebagian besar buku berisi satu indeks dari kata-kata umum, nama, tempat, dan seterusnya. Database berisi indeks yang individu untuk jenis atau kolom yang terpilih dari data: ini serupa dengan sebuah buku yang berisi indeks untuk nama dari orang dan indeks lain untuk tempat. Ketika anda membuat suatu database dan menadaptkan kinerja, anda perlu membuat indeks untuk kolom yang digunakan di dalam query untuk menemukan dataDalam contoh database, tabel karyawan mempunyai satu indeks di kolom emp_id. Ilustrasi yang berikut menunjukkan bagaimana indeks menyimpan masing-masing nilai emp_id dan poin-poin ke baris Ketika aplikasi database melaksanakan suatu statemen untuk menemukan data dalam tabel karyawan berdasar suatu nilai emp_id yang ditetapkan, mengenali indeks untuk kolom emp_id dan menggunakan indeks itu untuk menemukan data. Jika indeks tidak ada, melaksanakan suatu permulaan scan tabel yang penuh pada awal tabel dan melangkah melalui masing-masing baris, mencari-cari nilai temp_id yang ditetapkan.

1.2 Tuning: Setting Configuration DBMS

MySQL adalah salah satu DBMS yang paling populer dan paling banyak digunakan untuk menyimpan data-data baik itu untuk aplikasi desktop, mobile, maupun web. Untuk mempercepat kinerja dari database engine, maka terdapat konfigurasi yang dapat dilakukan yaitu :

1. InnoDB_buffer_pool_size
   Tips ini berlaku untuk database mysql yang menggunakan engine innodb, jadi jika server anda menggunakan engine InnoDB maka optimasi dasar yang harus anda lakukan adalah mengatur innodb_buffer_pool_size, kenapa? Karena pada dasarnya pemrosesan yang dilakukan oleh mysql baik itu insert delete update select itu banyak dilakukan oleh cpu sehingga apabila kita memproses data yang besar maka cpu akan mengalami kenaikan resource yang cukup tinggi dan akhirnya akan terjadi freeze atau crash. Oleh karena itu kita mengatur innodb_buffer_pool_size untuk membantu meningkatkan performa pemrosesan data utamanya di proses insert. Karena innodb_buffer_pool_size ini memanfaatkan ram yang pemrosesannya jauh lebih cepat daripada hardisk dan resourcenya

bisa kita sesuaikan, saran saya adalah maksimal 80% dari ukuran RAM, contoh RAM 8GB maka innodb_buffer_pool_size adalah 6GB. Jadi nantinya proses seperti insert pemrosesannya akan dilakukan oleh RAM. Ada 2 cara mengatur innodb_buffer_pool_size.

Pertama, dengan menggunakan perintah sql. Perintahnya SET GLOBAL innodb_buffer_pool_size = 6G. Ini jika kalian tidak ingin melakukan restart pada sql. Kedua, dengan menambahkan variabel innodb_buffer_pool_size = 6G di bawah konfigurasi-konfigurasi mysql yang sudah ada, kemudian restart service mysql anda. Silahkan gunakan cara manapun yang menurut anda lebih mudah, tapi ingat ukuran innodb_buffer_pool_size jangan lebih dari 80% agar sisanya dapat digunakan untuk sistem lain dan tidak menimbulkan crash.

2. innodb_flush_log_at_trx_commit
   Melakukan konfigurasi pada innodb_flush_log_at_trx_commit sangat berpengaruh terhadap performa dari mysql, kenapa? karena pada dasarnya proses innodb adalah melakukan transaksi dan menyimpan catatan transaksinya. konfigurasi innodb_flush_log_at_trx_commit ini berpengaruh pada keamanan data, ada 3 pilihan setting yaitu 0, 1, 2. Saran saya adalah set variablenya ke 0 atau 2 karena disini proses flush log akan dilakukan lebih cepat, sedangkan pilihan 1 proses flush akan memastikan keamanan data dalam proses transaksi. Namun bukan berarti pilihan 0 atau 2 tidak aman, memang ada resiko tapi itu sangat kecil sehingga lebih baik memilih pilihan 0 atau 2, saya sendiri menerapkan pilihan 2 karena cepat dan lebih aman daripada pilihan 0. Anda dapat merubah nilai variable dari innodb_flush_log_at_trx_commit seperti cara di tips no 2.
3. Skip Name Resolve
   Pencarian DNS (DNS Lookup) untuk host MYSQL bisa dikatakan hampir tidak perlu, karena hanya menambahkan rountrip tambahan untuk permintaan yang harus diselesaikan. Ini merupakan proses yang banyak berjalan dan cukup membuat delay karena sever harus menyelesaikan pencarian DNS tersebut. Oleh karena itu lebih baik server tidak perlu melakukan pencarian DNS. Caranya adalah menggunakan perintah skip-name-resolve yang bisa dimasukkan pada file konfigurasi mysql.
4. Buat Slow Query Log
   Membuat (mengaktifkan) log untuk menyimpan query yang berjalan lambat sangat bermanfaat untuk melakukan evaluasi query yang berjalan lambat dan membebani kinerja server. Untuk mengaktifkan slow query log, perlu ditambahkan konfigurasi berikut pada
5. file konfigurasi mysql:

   slow_query_log = 1

   long_query_time = 3

   log_output = TABLE

   Slow_query_log = 1 adalah perintah untuk mengaktifkan logging pada query. Kemudian long_query_time = 3 adalah batas waktu query dijalankan apabila melebhi 3 detik maka masuk ke dalam slow query log, nilainya dapat diubah sesuai kebutuhan. Untuk log_output = TABLE adalah menentukan data query disimpan di dalam table slow_log yang ada pada database mysql ini saya sarankan karena lebih mudah daripada output dengan file.

BAB II
HASIL DAN PEMBAHASAN

2.1 Tuning: Indexing

Melakukan indexing pada tabel student pada setiap Data set dan melakukan indexing pada table section pada setiap Data set dengan *command*

create index student_pk on student(id);

create index section_pk on student(sec_id);

Seperti pada **Gambar 1**, **Gambar 2**, dan **Gambar 3**.

```
MariaDB [(none)]> use dbms_one;
Database changed
MariaDB [dbms_one]> create index student_pk on student(id);
Query OK, 0 rows affected (0.385 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [dbms_one]> create index section_pk on section(sec_id);
Query OK, 0 rows affected (1.099 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Gambar 1. Indexing dataset 1

```
MariaDB [dbms_one]> use dbms_two;
Database changed
MariaDB [dbms_two]> create index student_pk on student(id);
Query OK, 0 rows affected (0.510 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [dbms_two]> create index section_pk on section(sec_id);
Query OK, 0 rows affected (0.474 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Gambar 2. Indexing dataset 2

```
MariaDB [dbms_two]> use dbms_three;
Database changed
MariaDB [dbms_three]> create index student_pk on student(id);
Query OK, 0 rows affected (0.373 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [dbms_three]> create index section_pk on section(sec_id);
Query OK, 0 rows affected (0.675 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Gambar 3. Indexing dataset 3

2.2 Tuning: Setting Configuration DBMS

Melakukan perubahan pada file my.ini pada direktori C:\xampp\mysql\bin Dengan merubah innodb_buffer_pool_size = 16M menjadi innodb_buffer_pool_size = 1G dan innodb_log_file_size = 5M menjadi innodb_log_file_size = 250M. Adapun konfigurasi yang telah diubah apat dilihat pada **Gambar 4.**

```
innodb_buffer_pool_size = 1G
## Set .._log_file_size to 25 % of buffer p
innodb_log_file_size = 250M
innodb_log_buffer_size = 8M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
```

Gambar 4. Konfiguras my.ini dbms

Dan konfigurasi setiap database dengan perintah:

set global query_cache_size = 268435456;

set global query_cache_type = 1;

set global query_cache_limit = 1048576;

(**Gambar 5, Gambar 6, Gambar 7**)

```
MariaDB [(none)]> use dbms_one;
Database changed
MariaDB [dbms_one]> set global query_cache_size = 268435456;
Query OK, 0 rows affected (0.001 sec)

MariaDB [dbms_one]> set global query_cache_type = 1;
Query OK, 0 rows affected (0.005 sec)

MariaDB [dbms_one]> set global query_cache_limit = 1048576;
Query OK, 0 rows affected (0.000 sec)
```

Gambar 5. Set Konfigurasi DBMS pada dataset 1

```
MariaDB [dbms_one]> use dbms_two;
Database changed
MariaDB [dbms_two]> set global query_cache_size = 268435456;
Query OK, 0 rows affected (0.005 sec)

MariaDB [dbms_two]> set global query_cache_type = 1;
Query OK, 0 rows affected (0.000 sec)

MariaDB [dbms_two]> set global query_cache_limit = 1048576;
Query OK, 0 rows affected (0.000 sec)
```

Gambar 6. Set Konfigurasi DBMS pada dataset 2

```
MariaDB [dbms_two]> use dbms_three;
Database changed
MariaDB [dbms_three]> set global query_cache_size = 268435456;
Query OK, 0 rows affected (0.006 sec)

MariaDB [dbms_three]> set global query_cache_type = 1;
Query OK, 0 rows affected (0.000 sec)

MariaDB [dbms_three]> set global query_cache_limit = 1048576;
Query OK, 0 rows affected (0.000 sec)
```

Gambar 7. Set Konfigurasi DBMS pada dataset 3

2.3 Hasil Tuning

Query pada tabel sesuai petunjuk pengerjaan :

1. SELECT * FROM student
2. SELECT * FROM student WHERE tot_cred > 30;
3. SELECT `name`, dept_name FROM student WHERE tot_cred > 30;
4. SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
5. SELECT student.`name`,student.dept_name,takes.sec_id AS pengambilan,takes.semester,section.room_number,section.building,course.course_id,course.dept_name FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id JOIN course ON section.course_id = course.course_id

Dengan data sebelum dilakukannya tuning ditunjukkan pada **Tabel 1** selanjutnya melakukan tuning dan hasil dari tuning tersebut ditujukan pada **Table 2.**

Table 1. Waktu eksekusi sebelum tuning

Data	Waktu Sebelum Tunning (ms)				
	Query 1	Query 2	Query 3	Query 4	Query 5
1	0.00044380	0.00041460	0.00038590	0.00184840	0.00237200
2	0.00052060	0.00048710	0.00044780	0.00352350	0.00352350
3	0.00077130	0.00081460	0.00067920	0.19959490	0.06971610

Table 2. Waktu eksekusi setelah tuning

Data	Waktu Sesudah Tuning (ms)				
	Query 1	**Query 2**	**Query 3**	**Query 4**	**Query 5**
1	0.00013030	0.00010340	0.00011860	0.00014400	0.00023570
2	0.00011050	0.00012370	0.00010310	0.00017800	0.00016900
3	0.00014000	0.00011270	0.00014600	0.00045960	0.00030480

BAB III
KESIMPULAN

Dari percobaan yang telah dilakukan dapat dilihat bahwa proses tuning bertujuan untuk memaksimalkan kinerja DBMS, proses tuning dapat dilakukan pada berbagai aspek seperti query optimization, hardware tuning, relational schema dan lain-lain, namun dalam percobaan ini digunakan tuning menggunakan indexing dan dbms configuration. Pada tahap tuning pertama, menggunakan indexing, terlihta jelas perbedaan running timenya, dikarenakan proses tuning dengan indexing akan membuat proses pencarian yang tadinya full scan table, menjadi partial scan, sehingga data yang akan dicari akan langsung tertuju pada index yang ditetapkan. Lalu selanjutnya, pada tahap tuning selanjutnya, yaitu konfigurasi engine innodb cukup efektif untuk membantu mempercepat running time, karena konfigurasi dbms akan memaksimalkan kinerja hardware yang digunakan oleh dbms, sehingga lebif efisien dalam penggunaan sumber daya.

DAFTAR PUSTAKA

1. MYSQLLAB, "InnoDB," 22 12 2019. [Online]. Available: http://www.mysqlab.net/knowledge/kb/detail/topic/innodb/id/6553.

2. NixCP, "Skip Name Resolve," 22 12 2019. [Online]. Available: https://nixcp.com/skip-name-resolve/.

3. "What is MySQL Query Caching?," 22 12 2019. [Online]. Available: https://www.interserver.net/tips/kb/mysql-query-caching/.