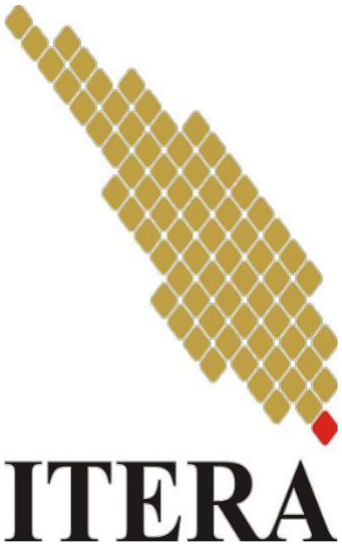
TUGAS BESAR
MANAJEMEN BASIS DATA
Tuning Indexing Dan Configuration DBMS



Disusun Oleh :

M. Anas Nasrulloh (14117028)

Teknik Informatika

Institut Teknologi Sumatera

2019

DAFTAR ISI

BAB 1 TEORI DASAR

Tunning Indexing

Tunning indexing adalah bagian penyetelan basis data untuk memilih dan membuat indeks. Tujuan Tunning indexing adalah untuk mengurangi waktu pemrosesan kueri. Potensi penggunaan indeks di lingkungan dinamis dengan beberapa permintaan ad-hoc sebelumnya adalah tugas yang sulit. Penyetelan indeks melibatkan kueri berdasarkan indeks dan indeks dibuat secara otomatis saat itu juga. Tidak diperlukan tindakan eksplisit oleh pengguna basis data untuk penyetelan indeks. Meningkatkan kinerja kueri dan basis data, dapat menggunakan langkah-langkah sebagai berikut:

1. Menggunakan pengoptimal kueri untuk melakukan analisis kueri sehubungan dengan beban kerja dan berdasarkan pengetahuan ini, ia merekomendasikan penggunaan indeks terbaik.
2. Perubahan dalam penggunaan indeks, distribusi permintaan dan kinerjanya dianalisis untuk memeriksa efeknya.
3. Menggunakan Wisaya Penyetelan Indeks. SQL profiler digunakan untuk menangkap jejak aktivitas, untuk mengoptimalkan kinerja. Jejak dapat diperpanjang untuk jangka waktu tertentu dengan tujuan menangkap berbagai aktivitas.

Tunning Setting Configuration DBMS

Sistem Manajemen Basis Data (DBMS) adalah komponen paling penting dari aplikasi intensif data apa pun. Mereka dapat menangani sejumlah besar data dan beban kerja yang kompleks. Tetapi mereka sulit untuk dikelola karena mereka memiliki ratusan "tombol" konfigurasi yang mengontrol faktorfaktor seperti jumlah memori yang digunakan untuk cache dan seberapa sering menulis data ke penyimpanan. Untuk melakukan performance tuning pada PostgreSQL dengan skala bisnis dengan konfigurasi manajemen database digunakan database administrator yang akan menganalisis parameter konfigurasi database PostgreSQL dan merekomendasikan konfigurasi optimal sesuai dengan workload Anda.

BAB II PERCOBAAN TUNING INDEXING

2.1 Sebelum Tunning Indeing

Query

- SELECT * FROM student
- SELECT * FROM student WHERE tot_cred > 30;
- SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
- 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

1. Percobaan 1
   Pada percobaan pertama menggunakan advisor = 100, student = 100, section = 200,takes = 200. Dapat dilihat pada tabel berikut :

Query	Waktu Sebelum Tuning (msec)
1	96
2	82
3	83
4	89
5	75

2. Percobaan 2
   Pada percobaan kedua menggunakan advisor = 200, student = 200, section = 400,takes = 400. Dapat dilihat pada tabel berikut :

Query	Waktu Sebelum Tuning (msec)
1	108
2	91
3	73
4	148
5	126

3. Percobaan 3
   Pada percobaan kedua menggunakan advisor = 500, student = 500, section = 1000,takes = 1000. Dapat dilihat pada tabel berikut :

Query	Waktu Sebelum Tuning (msec)
1	117
2	119
3	121
4	210
5	119

2.2 Sesudah Tunning Indexing

Query

- SELECT * FROM student
- SELECT * FROM student WHERE tot_cred > 30;
- SELECT * FROM takes JOIN student ON takes.ID = student.ID JOIN section ON takes.course_id = section.course_id
- 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

1. Percobaan 1
   Pada percobaan pertama menggunakan advisor = 100, student = 100, section = 200,takes = 200. Dapat dilihat pada tabel berikut :

Query	Waktu Sesudah Tuning (msec)
1	71
2	77
3	68
4	104
5	96

2. Percobaan 2
   Pada percobaan kedua menggunakan advisor = 200, student = 200, section = 400,takes = 400. Dapat dilihat pada tabel berikut :

Query	Waktu Sesudah Tuning (msec)
1	76
2	88
3	70
4	148
5	110

3. Percobaan 3
   Pada percobaan kedua menggunakan advisor = 500, student = 500, section = 1000,takes = 1000. Dapat dilihat pada tabel berikut :

Query	Waktu Sesudah Tuning (msec)
1	113
2	106
3	99
4	158
5	117

BAB III HASIL DAN PEMBAHASAN

3.1 Hasil

Perbandingan waktu hasil percobaan antara sebelum tunning dengan sesudah tunning.

Percobaan	Query	Waktu Sebelum Tuning(msec)	Waktu Sesudah Tuning(msec)
1	1	96	71
	2	82	77
	3	83	68
	4	89	104
	5	75	96
2	1	108	76
	2	91	88
	3	73	70
	4	148	148
	5	126	110
3	1	117	113
	2	119	106
	3	121	99
	4	210	158
	5	119	117

3.2 Pembahasan

Berdasarkan data pada tabel hasil diatas, dapat dilihat bahwa pada Percobaan 1 (advisor = 100, student = 100, section = 200, takes = 200) menghasilkan waktu eksekusi yang beragam. Data yang dieksekusi menggunakan query 4 dan 5 membutuhkan waktu eksekusi yang lebih. Hal ini terjadi bergantung pada cara pengecekan setiap query yang berbeda-beda. Untuk itu diperlukan tuning agar dapat mempercepat waktu pengeksekusian data. Setelah menggunakan tunning, waktu yang didapatkan untuk melakukan ekseskusi data dapat menjadi lebih kecil dari sebelum dilakukannya tuning.

Dari penjelasan tersebut dapat disimpulkan bahwa melakukan eksekusi data dengan menggunakan teknik tuning dapat memperkecil waktu eksekusi data dan sesuai dengan kebutuhan pengguna.

DAFTAR PUSTAKA

https://thesolidsnake.wordpress.com/2014/12/28/belajar-menyetel-database-mysql-server/

https://www.careerride.com/DB-efficient-transactions.aspx

https://aws.amazon.com/blogs/machine-learning/tuning-your-dbms-automatically-with-machine-learning/

https://www.i-3.co.id/2016/10/07/index-pada-database/