

DAFTAR ISI

Kata Pengantar	v
Ucapan Terima Kasih	vi
Daftar Isi	vii

I. Pengenalan	1
----------------------------	----------

II. Database Design	7
2.1 Bufferpool	7
2.2 Tablespace	8
2.3 Log	10
2.4 Concurrency	10
2.4.1 Fenomena Multi Transaksi	11
2.4.2 Isolation Level.....	12
2.4.3 Locking	13

III. Database Tuning Feature	17
---	-----------

3.1 Configuration Advisor	17
3.2 Design Advisor	25
3.3 Indexing	32
3.4 Runstat / Reorg	37
3.5 Database Partition Feature, Table Partition dan Multi Dimensional Clustering	41
3.5.1.1 Pengenalan Database Partition Feature (DPF)	41
3.5.1.2 Kelebihan Database Partition Feature Feature	41
3.5.1.3 Perencanaan Database Partition Feature.....	45
3.5.2.1 Pengenalan Table Partition	50
3.5.2.2 Kelebihan Table Partition	50
3.5.2.3 Perencanaan Table Partition	51
3.5.3.1 Pengenalan Multi Dimensional Clustering	53
3.5.3.2 Kelebihan Multi Dimensional Clustering	54
3.5.3.3 Perencanaan Multi Dimensional Clustering	55
3.5.4 Implementasi Database Partition.....	59
3.5.5 Implementasi Table Partition dan Multi Dimensional Clustering	65
3.6 Materialized Query Table (MQT)	71
3.7 Row Compression	78
3.8 STMM	85

IV. Database Tuning Parameter	91
4.1 Instance parameter	91
4.2 Database parameter	96
4.3 Registry variables	101
Daftar Link	115
Tentang Penulis	117
Buku Lainnya	118
Daftar Pustaka.....	119

BAB 1

Pengenalan

Performance merupakan sebuah topik penting dalam dunia DBA. Performance database dipengaruhi oleh banyak faktor karena lingkungan database terdiri dari hardware (baik server maupun network), software, dan aplikasinya. Oleh karena itu untuk melakukan tuning database untuk mencapai peak performance bukanlah perkara mudah.

Performance adalah sebuah ukuran dimana sebuah sistem komputer memberikan reaksi atau respon terhadap sebuah workload. Performance sebuah sistem dapat diukur dari beberapa faktor seperti response time, throughput maupun utilisasi resource.

Secara umum, dalam melakukan tuning sistem kita ingin meningkatkan yang namanya cost-benefit ratio. Beberapa hal diantaranya seperti berikut

- Memproses lebih banyak tanpa menambah processing cost
- Mendapatkan respon yang lebih cepat atau throughput yang lebih besar tanpa menambah processing cost
- Mengurangi processing cost tanpa mengurangi service ke user
- Tuning dilakukan tanpa mengganggu user (transparent)

Dalam melakukan tuning, ada beberapa guidelines yang sebaiknya diikuti

- Ingatlah hukum law of diminishing returns, bahwa performance terbesar biasanya didapat dari effort yang pertama kita lakukan

- Lakukan tuning dengan memperhatikan keseluruhan. Jangan mengubah sebuah parameter untuk meningkatkan performance disatu sisi namun mengakibatkan keseluruhan sistem menjadi lambat
- Lakukan perubahan satu per satu sehingga kita mendapatkan kepastian dampak apa yang terjadi.
- Lakukan tuning per level. Level sistem ada beberapa yaitu (hardware, sistem operasi, application server, database manager, SQL statement, application program)
- Jangan terburu- buru untuk menambah resource seperti CPU, memory ataupun disk tanpa mengetahui dimana root cause dari masalah. Misalnya kita menambah CPU dan memory menjadi 2x lipat padahal terjadi bottleneck disisi disk storage, maka sesudah penambahan tetap tidak akan terjadi perubahan.
- Buatlah rencana fallback untuk berjaga- jaga jika terjadi sesuatu yang tidak sesuai harapan.

Sebelum melakukan tuning, sebaiknya kita mencari informasi yang detail mengenai sistem yang ada, database design, beserta informasi complain dari user. Mengenai sistem dan database design kita bisa bertanya kepada sistem administrator atau database administrator yang bertugas. Jika tidak ada, DB2 menyediakan beberapa tools untuk mendapatkan informasi tersebut. Untuk informasi complain dari user, bertanya- tanya terlebih dahulu kepada user sehingga mempermudah kita untuk mengidentifikasi masalah yang terjadi. Sering kali penulis mengalami bahwa sebenarnya masalah performance terjadi untuk beberapa hal yang spesifik namun terkadang user mengatakan bahwa keseluruhan sistem lambat. Berikut beberapa daftar pertanyaan yang bisa digunakan

Berikut beberapa informasi yang cukup penting untuk didapatkan dari level hardware dan sistem operasi

- Apa sistem operasi yang dipakai dan versi berapa?
- Apa CPU yang digunakan dan clockspeednya?
- Berapa banyak core CPU yang digunakan ?
- Berapa besar memory RAM yang tersedia?
- Berapa besar virtual memory yang dibuat?
- Berapa besar memory RAM yang tidak dipakai?
- Apa jenis storage yang digunakan?
- Berapa besar kapasitas disk storage ?
- Berapa besar kapasitas disk storage yang tidak dipakai?

Berikut beberapa informasi yang cukup penting untuk didapatkan dari level database

- Apa ada penambahan user di sistem ?
- Apa ada penambahan jumlah data ?
- Apa ada terjadi masalah locking ?
- Apakah statistic yang ada up to date ?
- Bagaimana status tablespace dan containernya ?
- Bagaimana penggunaan bufferpool ? Berapa hit rasionya ?
- Apakah terjadi sort overflow ?
- Apakah terjadi perubahan di sistem, misalnya perubahan network, pergantian spesifikasi server dsb ?

Berikut beberapa informasi yang cukup penting untuk didapatkan dari user

- Kapan masalahnya terjadi, apakah selalu terjadi atau baru saja terjadi ?
- Apakah user lain mengalami hal yang sama atau hanya beberapa user saja ?
- Apakah user yang mengalami hal yang sama berada dalam jaringan yang sama ? Apakah user yang tidak mengalami berada di jaringan lain ?
- Apakah masalahnya hanya terjadi di operasi tertentu di aplikasi ? Misalnya lambat ketika mengakses database payroll saja, namun ketika mengakses database HR tidak ada masalah
- Apakah masalahnya selalu berlangsung setiap hari atau hanya jam tertentu ?

Berikut merupakan faktor yang mempengaruhi DB2 performance

- **desain aplikasi**

Penggunaan algoritma yang berbelit-belit, query yang tidak efisien, atau stored procedure yang kurang tepat dapat mengurangi kinerja database.

- **desain sistem dan database**

Desain seperti storage layout, peletakan index dan log di disk yang salah, menggunakan index yang kurang tepat, tidak digunakannya fitur-fitur database tuning beberapa hal yang cukup berpengaruh dalam menurunkan kinerja database.

- **CPU dan memory**

Besarnya CPU dan memory sangat berpengaruh dalam kinerja. Namun dalam beberapa kasus, CPU dan memory yang lebih besar namun desain sistem, database dan aplikasi yang tidak baik akan

lebih buruk daripada CPU dan memory yang lebih kecil namun menggunakan desain yang baik dan benar.

- **disk**

Dalam kebanyakan kasus, orang lebih terfokus pada CPU dan memory yang besar. Namun sering kali terjadi ketimpangan antara CPU dan memory dengan kinerja disk, sehingga terjadi IO bottleneck. Bisa dianalogikan bahwa sistemnya memiliki 8 jalur tol, namun hanya memiliki 1 gerbang tol, sehingga tetap saja kinerjanya adalah 1 mobil per jalur.

- **network**

Mirip dengan faktor disk. Dalam beberapa kasus yang pernah penulis temukan, faktor network seperti bandwidth, sering terjadinya RTO (Request Time Out), beban berlebihan di jaringan dapat menyebabkan terjadi bottleneck kembali.

Dilihat dari pertanyaan- pertanyaan di atas serta faktor- faktor yang mempengaruhi, topik mengenai performance tuning memang mencakup bidang yang luas. Oleh karena itu, dalam buku ini hanya akan dibahas cara-cara untuk melakukan performance tuning dari sisi DB2.

Perlu diketahui performance tuning dari sisi DB2 akan memiliki batasan. Jika setelah melakukan tuning dan memang sudah mencapai kinerja maksimum, maka focus tuning bisa dialihkan ke faktor lain, seperti penambahan resource ataupun bandwidth dari jaringan.

BAB 2

Database Design

Salah satu faktor yang berpengaruh sangat besar terhadap kinerja sebuah database adalah database design. Saya pernah mengalami di sebuah project yang menggunakan sebuah server yang kencang dengan memory mencapai ratusan GB, namun karena database design yang kacau, belum dilakukannya konfigurasi parameter dengan benar, dan tidak digunakan fitur seperti Materialized Query Table (MQT), Table Partition dsb maka kinerja sistemnya berjalan lambat.

Oleh karena itu, sebelum masuk ke bab untuk melakukan tuning, alangkah baiknya jika kita membahas terlebih dahulu mengenai database design.

2.1 Bufferpool

Bufferpool merupakan sebuah objek database yang berkaitan cukup erat dengan kinerja. Objek ini berhubungan dengan sebuah database dan dapat digunakan oleh lebih dari satu tablespace, namun satu tablespace hanya dapat menggunakan satu bufferpool. Besarnya tablespace pagesize harus sama dengan bufferpool pagesize.

Secara default, ketika database dibuat maka sebuah bufferpool juga akan dibuat dengan nama IBMDEFAULTBP. Bufferpool ini digunakan secara shared oleh semua tablespace. Seiring berjalannya waktu, bufferpool dapat ditambah. Besar bufferpool yang sesuai berguna untuk meningkatkan kinerja karena dapat mengurangi

proses I/O. Bufferpool yang besar juga akan mempengaruhi optimasi query karena dapat dilakukan di memory.

Pada saat bufferpool dibuat, jika tidak didefinisikan, maka besarnya bufferpool akan mengikuti nilai dari parameter BUFFPAGE. Namun jika dibutuhkan kita dapat mengatur besarnya bufferpool dengan menggunakan option SIZE saat membuat bufferpool.

Berikut merupakan beberapa hal yang harus dipertimbangkan dalam membuat bufferpool

Page size merupakan sebuah parameter yang harus dipertimbangkan dengan baik. Hal ini dikarenakan page size berpengaruh terhadap besar maksimal tablespace (besar page size sebuah tablespace harus sama dengan besar page size bufferpool yang digunakan) . Page size juga mempengaruhi performance. Ada beberapa pertimbangan mengenai berapa page size yang harus digunakan (lihat di bab 2.2 mengenai tablespace)

Bufferpool size juga berpengaruh terhadap kinerja. Walaupun ada banyak cara untuk melakukan tuning, memperbesar bufferpool (cukup sampai sesuai kebutuhan) bisa dikatakan cara tercepat untuk meningkatkan kinerja.

Blocked size memungkinkan kita untuk mengatur besarnya bufferpool yang ingin digunakan untuk block-based prefetching. Block-based I/O akan meningkatkan efisiensi dari prefetching dengan cara menyimpan secara berurutan di dalam memory.

2.2 Tablespace

Jika bufferpool berhubungan dengan memory, maka tablespace berhubungan erat dengan storage. Penggunaan design tablespace akan sangat berpengaruh terhadap proses I/O yang akan terjadi.

Berikut beberapa jenis tablespace yang ada.

- **regular tablespace**

Tablespace ini menyimpan data dan index. Secara default juga akan dibentuk ketika database dibuat pertama kali dengan nama USERSPACE1. Berbeda dengan catalog tablespace, regular tablespace dapat berjumlah lebih dari satu.

- **large tablespace**

Tablespace ini berguna untuk menyimpan data long atau LOB dan harus berada dalam Database Manage Space (DMS). Tablespace ini bersifat optional, dan jika tidak ada long tablespace, maka LOB akan disimpan di regular tablespace.

- **system temporary tablespace**

Tablespace ini berguna untuk menyimpan temporary data internal ketika terjadi operasi SQL seperti sorting, reorg, membuat index, dan join table. Secara default akan dibentuk dengan nama TEMPSPACE1

- **user temporary tablespace**

Tablespace ini digunakan untuk menyimpan declared global temporary table. Tablespace ini bersifat optional, namun harus dibuat jika ingin membuat temporary table

Berikut merupakan beberapa hal yang harus dipertimbangkan dalam membuat tablespace

- **Page size**

Page size berpengaruh terhadap kapasitas maksimum dari sebuah tablespace, kinerja dan storage. (Lebih detailnya dapat dibaca di buku "IBM Database DB2 Intermediate bab 1.3)

- **Extent size**

Extent size menentukan berapa banyak page yang akan ditulis ke dalam container sebelum pindah ke container lainnya. Parameter ini

hanya berfungsi jika sebuah tablespace disimpan dalam beberapa container.

- **Prefetch size**

Prefetch size menentukan jumlah page yang dibaca dari tablespace ketika data prefetching dijalankan. DB2 akan menentukan apakah prefetching akan digunakan atau tidak ketika sebuah query dijalankan.

- **Overhead rate**

Angka ini berhubungan dengan aktifitas I/O controller, disk seek time dan rotational latency.

- **Transfer rate**

Transfer rate adalah seberapa lama waktu yang dibutuhkan untuk membaca sebuah page ke dalam memory.

2.3 Log

Kecepatan sebuah log ditulis ke dalam disk dapat mempengaruhi kinerja sistem. Oleh karena itu berikut beberapa hal yang harus diperhatikan dalam pengaturan log.

- Disable autocommit. Jika terjadi commit setiap kali terjadi transaksi maka akan menambah waktu selesainya transaksi
- Jangan mencatat log untuk large object (CLOB, BLOB) kecuali memang dibutuhkan.
- Gunakan dedicated disk untuk menyimpan log.
- Gunakan disk dengan write cache yang kencang
- Naikan log buffer menjadi 256 pages atau lebih besar.