Otomatisasi Visualisasi Data dan Kueri Database Analitik Berkinerja Tinggi Menggunakan GPU dengan OmniSciDB

Albertus Bayu Aji Priyono, ST., MMSI.

13 Oktober 2020

Otomatisasi Visualisasi data dan Kueri Database Analitik Berkinerja Tinggi menggunakan GPU dengan OmniSciDB

PHD THESIS PROPOSAL

ALBERTUS BAYU AJI PRIYONO

PROGRAM DOKTOR TEKNOLOGI INFORMASI
UNIVERSITAS GUNADARMA
2020

Daftar Isi

Daftar Tabel

Daftar Gambar

Bab 1

Pendahuluan

1.1 Latar Belakang Penelitian

Permintaan untuk pemrosesan berkecepatan tinggi di dunia komputasi yang terus berkembang ini meningkat dari hari ke hari. Aplikasi berbasis web dan sistem yang menggunakan database di *backend* membutuhkan pemrosesan kecepatan tinggi untuk mengakses data atas database.

Pada beberapa tahun terakhir, perkembangan *database* baik dalam sisi kebutuhan terhadap volume data dan bagaimana penggunaan terhadap data tersebut dilakukan sudah sangat berubah. Menurut *micoresolutions* dalam artikelnya terdapat 5 tantangan utama yang dihadapi dalam manajemen *database* saat ini :

1. *Growing complexity in landscape*

   Seiring dengan berkembangnya jenis data dan penggunannya, saat ini terdapat berbagai jenis *database* yang menjadi pilihan seperti *relational database, columnar database, object-oriented database serta database lain yang menggunakan konsep NoSQL.* Ditambah dengan munculnya vendor-vendor yang menyajikan *spin-off* dari masing - masing database tersebut sesuai dengan ide dan visi yang disajikan.

2. *Limits on scalability*

   Semua perangkat lunak memiliki skalabilitas dan batasan terhadap penggunaan sumber daya, termasuk *database server.* Komponen katalog, arsitektur database, dan bahkan sistem operasi serta konfigurasi perangkat keras mempengaruhi terhadap skalabilitas.

3. *Increasing data volumes*

Ketika jumlah data yang dihasilkan dan dikumpulkan meledak, banyak organisasi ataupun perusahaan memerlukan perjuangan lebih untuk mengkuti perkembangan tersebut. Penelitian menunjukkan bahwa saat ini data yang telah dibuat dalam dua tahun terakhir jumlahnya jauh lebih besar dibandingkan jumlah populasi manusia.

4. *Data security*

   Database adalah hasil kerja yang tersembunyi dari banyak sistem TI perusahaan, yang menyimpan data publik serta data pribadi yang penting. Keamanan terhadap data tersebut menjadi prioritas utama khususnya dengan adanya peraturan terhadap keamanan data.

5. *Decentralized data management*

   Meskipun ada manfaat dari pengelolaan data yang terdesentralisasi, namun hal tersebut juga menghadirkan tantangan, seperti bagaimana data akan didistribusikan? Apa metode desentralisasi terbaik yang sesuai dengan kebutuhan? Tantangan utama dalam merancang dan mengelola database terdistribusi berasal dari kurangnya pengetahuan yang terpusat dari keseluruhan database.

Graphic Processing Unit (GPU), telah terbukti menjadi sebuah *co-processor* yang efisien di bidang komputasi konvensional. Akselerasi dramatis telah dicapai dalam operasi database yang berbeda menggunakan GPU yang bukan bagian dari bahasa database konvensional seperti SQL pada umumnya. Pada dasarnya GPU dirancang untuk visualisasi primitif geometris, namun GPU juga dapat digunakan untuk menjalankan operasi database secara efisien dengan menggunakan inherent pipelining dan paralelisme, arsitektur multi-threaded, fungsionalitas pemrosesan secara vektor dari GPU bersama dengan Single Instruction and Multiple Data (SIMD) untuk mengevaluasi semi-linier kueri berdasarkan atribut.

SQL adalah pintu gerbang antara pemrograman dan data di tabel terstruktur. Dengan percepatan kueri SQL, programmer dapat meningkatkan waktu eksekusi dalam menit serta tanpa adanya perubahan dalam kode program.

Pada disertasi ini akan dibangun sebuah rancangan arsitektur database menggunakan OmniSciDB dengan bantuan GPU untuk pemrosesan data dalam kueri [Litwintschik, 2020].

1.2 Pertanyaan Penelitian

Penelitian ini diharapkan dapa menjawab beberapa pertanyaan:

- Bagaimana menggunakan GPU sebagai *co-processor* dalam kueri SQL?
- Bagaimana merancang arsitektur database menggunakan OmniSciDB dengan memanfaatkan GPU?
- Bagaimana performa yang dihasilkan dan kemungkinan skalabilitasnya?
- Bagaimana menghasilkan otomatisasi visualisasi data berdasarkan data yang diolah secara *real-time*?

1.3 Batasan Masalah

Dalam penelitian ini membahas tentang penggunaan GPU sebagai *co-processor* pada OmniSciDB dan visualisasi data yang terotomatisasi.

1.4 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penyusunan disertasi ini adalah:

- Tinjauan pustaka dalam penerapan GPU pada kueri database;
- Pengumpulan sample data
- Pengembangan arsitektur yang cocok untuk database berbasis kaan GPU;
- Publikasi baik nasional maupun internasional

1.5 Kontribusi dari Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi, sebagai berikut:

- Tersedianya suatu model high performance architecture database dengan memanfaatkan GPU
- Menciptakan visualisasi data secara otomatis untuk kepentingan dalam data analitik

Bab 2

Telaah Pustaka

2.1 Definisi Database

Menurut Connoly dan Begg (2015), Database merupakan kumpulan data beserta deskripsi dari data itu sendiri yang terhubung secara logikal, yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

Sebuah database dapat berupa satu atau lebih tempat penampungan data yang dapat digunakan secara bersamaan dari berbagai departemen atau pengguna. Semua data terintegrasi dengan jumlah duplikasi yang minimal, daripada file-file yang tidak saling berhubungan dengan data yang redundan. Sebuah database tidak lagi menjadi milik satu departemen, namun satu perusahaan dimana setiap departemen saling terintegrasi satu sama lain, begitu juga juga dengan database di dalamnya. Database tidak hanya memuat data operasional suatu perusahaan, tapi juga deskripsi dari data-data tersebut [Chaudhri, 2020].

Sebagai tambahan, Silberschatz, Korth, dan Sudarshan (2006) mengungkapkan bahwa database merupakan sekumpulan data yang memuat informasi yang relevan bagi sebuah perusahaan. Tujuan pemanfaatan database adalah sebagai berikut:

- Kecepatan dan kemudahan (Speed)

  Pemanfaatan database memungkinkan untuk dapat menyimpan, memanipulasi, dan/atau menampilkan kembali data dengan cepat dan mudah.

- Efisiensi ruang penyimpanan (Space)

  Mengeksekusi sejumlah code, atau dengan membuat relasi antar kelompok data yang saling berhubungan untuk menghindari redundansi (pengulangan) data.

- Keakuratan (Accuracy)

  Menentukan ketidak-akuratannya penyimpanan data dengan penerapan code atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data, domain data, keunikan data, dll dalam database.

- Ketersediaan (Availability)

  Seiring dengan bertambahnya waktu, ruang penyimpanan data pun juga akan semakin bertambah. Database dapat menghapus atau memindahkan data yang sudah jarang atau bahkan yang tidak pernah digunakan lagi ke media penyimpanan.

- Kelengkapan (Completeness)

  Penambahan struktur database untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang.

- Keamanan (Security)

  Menentukan siapa saja yang dapat mengakses database dengan menggunakan sistem security.

- Kebersamaan dan pemakaian

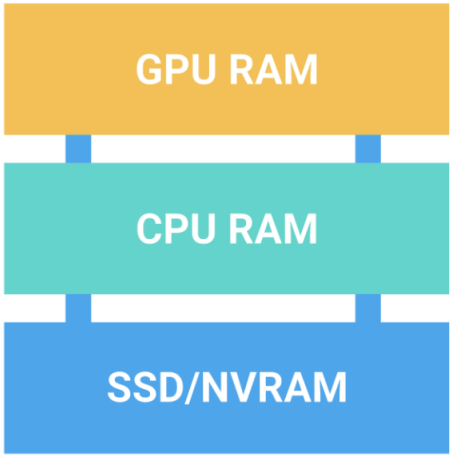
  Menghindari inkonsistensi data dalam lingkungan multi-user, karena pemakai database tidak terbatas hanya pada satu pemakaian dalam satu waktu saja.

2.2 OmniSciDB

OmniSciDB adalah dasar dari platform OmniSci. OmniSciDB berbasis SQL, relasional, berbentuk kolom dan secara khusus dikembangkan untuk memanfaatkan paralelisme besar dari perangkat keras CPU dan GPU modern. OmniSciDB dapat melakukan kueri hingga miliaran baris dalam milidetik, dan mampu melakukan kecepatan penyerapan yang belum pernah terjadi sebelumnya, menjadikannya mesin SQL yang ideal untuk era data berkecepatan tinggi yang besar.

OmniSciDB optimizes the memory and compute layers to deliver unprecedented performance. OmniSciDB was designed to keep hot data in GPU memory for the fastest access possible. Other GPU database systems have taken the approach of storing the data in CPU memory, only moving it to

GPU at query time, trading the gains they receive from GPU parallelism with transfer overheads over the PCIe bus.



Gambar 2.1: Advanced Memory Management OmniSciDB

OmniSciDB secara native mendukung SQL standar dan mengembalikan hasil kueri ratusan kali lebih cepat daripada platform database analitik khusus CPU. Analis dan ilmuwan data masih dapat mengandalkan pengetahuan SQL mereka yang ada, membuat kueri data menggunakan SQL standar industri.

OmniSci dapat beroperasi sebagai mesin SQL mandiri menggunakan alat baris perintah mapdql, atau editor SQL yang merupakan bagian dari antarmuka analisis visual OmniSci Immerse. Hasil kueri OmniSci dapat dikeluarkan ke OmniSci Immerse atau ke perangkat lunak pihak ketiga seperti Birst, Power BI, Qlik atau Tableau, melalui berbagai konektor.

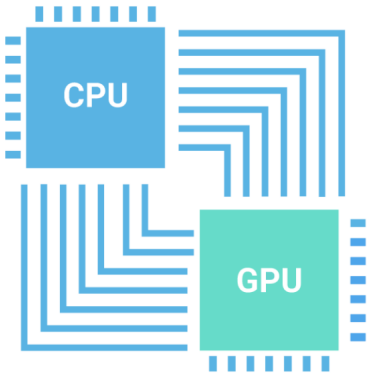
Komponen utama keunggulan inovasi OmniSciDB adalah kerangka kerja kompilasi JIT (Just-In-Time) yang dibangun di LLVM (Mesin Virtual Tingkat Rendah). Dengan membuat kode yang dikompilasi sebelumnya untuk kueri, OmniSci menghindari banyak bandwidth memori dan inefisiensi ruang cache dari pendekatan mesin virtual atau transpiler tradisional.

Dengan menggunakan LLVM, waktu kompilasi jauh lebih cepat — biasanya di bawah 30 milidetik untuk kueri SQL yang benar-benar baru. Selain itu, sistem dapat menyimpan cache versi template dari rencana kueri yang dikompilasi untuk digunakan kembali. Ini penting dalam situasi di mana pengguna memanfaatkan OmniSci Immerse untuk menyaring miliaran baris melalui beberapa visualisasi yang berkorelasi.

Mesin SQL OmniSciDB dapat menyimpan dan meminta data menggunakan jenis Open Geospatial Consortium (OGC) asli, termasuk POINT, LINESTRING, POLYGON, dan MULTIPOLYGON. Dengan dukungan tipe geografis

asli, analis dapat mengkueri data geo dalam skala besar menggunakan fungsi geospasial khusus yang jumlahnya terus bertambah. Ini membuka berbagai kasus penggunaan baru bagi analis geospasial, yang dapat menggunakan kekuatan penuh perangkat keras CPU dan GPU modern untuk menghitung jarak antara dua titik dan persimpangan antar objek dengan cepat dan interaktif. Sekarang analis dapat menemukan semua titik yang termasuk dalam tapak bangunan atau mencari persimpangan di antara mereka.

Komponen kunci dari keunggulan performa mesin SQL OmniSci adalah eksekusi kueri hibrid, atau paralel. Kode paralel memungkinkan prosesor untuk menghitung beberapa item data secara bersamaan. Ini diperlukan untuk mencapai kinerja optimal pada GPU, yang berisi ribuan unit eksekusi.



Gambar 2.2: Hybrid Execution

Mengoptimalkan eksekusi hybrid juga diterjemahkan dengan baik ke CPU, yang semakin memiliki unit eksekusi "lebar" yang mampu memproses beberapa item data sekaligus. OmniSciDB memparalelkan komputasi di beberapa GPU dan CPU, dan bahkan meningkatkan kinerja kueri pada sistem khusus CPU.

Konfigurasi skala keluar OmniSci memungkinkan kueri tunggal menjangkau lebih dari satu host fisik saat data terlalu besar untuk muat pada satu mesin. Di seluruh node, OmniSci menggunakan arsitektur shared-nothing antara GPU. Saat kueri diluncurkan, setiap GPU memproses sepotong data secara independen dari GPU lain. Meskipun beberapa GPU berada dalam satu mesin, data disebarkan dari CPU ke beberapa GPU dan kemudian dikumpulkan kembali bersama ke dalam CPU.

Arsitektur terdistribusi juga menyediakan waktu muat data yang lebih cepat. Waktu impor dipercepat secara linier dengan jumlah node karena pemuatan dapat dilakukan secara bersamaan di beberapa node. Membaca dari disk

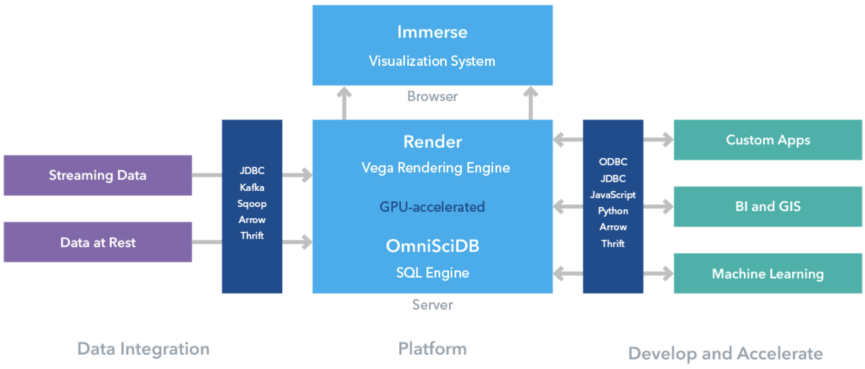
juga mendapat manfaat dari akselerasi serupa dalam konfigurasi scale-out.

Bab 3

Metode Penelitian

3.1 Arsitektur Platform

Platform OmniSci dirancang untuk mengatasi keterbatasan skalabilitas dan kinerja alat analitik lama yang dihadapkan pada atribut skala, kecepatan, dan lokasi dari kumpulan data besar saat ini. Alat-alat itu runtuh, menjadi terlalu lambat dan terlalu intensif perangkat keras untuk menjadi efektif dalam analitik data besar. OmniSci adalah teknologi terobosan, yang berasal dari MIT, yang dirancang untuk memanfaatkan pemrosesan paralel besar-besaran dari GPU bersama dengan komputasi CPU tradisional, untuk kinerja luar biasa dalam skala besar.



Gambar 3.1: Accelerated Analytics Platform

3.2 Visualisasi Data

Dengan memanfaatkan kemampuan GPU dalam OmniSciDB, tahap selanjutnya adalah membuat sebuah automated data visualization yang dapat menarik

data secara real-time dari OmniSciDB. Visualisasi data ini dapat menampilkan data secara crosstab atau cross filtering dalam bentuk geospasial atau pun dalam bentuk diagram.

Gambar 3.2: Interactive Visual Analytics

Bab 4

Jadwal Penelitian

Rencana Penelitian

Penelitian tersebut rencananya akan selesai dalam kurun waktu 2 tahun. Detail tabel waktu penelitian dapat dilihat pada tabel 4.1

Tabel 4.1: Research Time Table

	1^{st} Year												2^{nd} Year											
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Introduction	x	x																						
Literature Review	x	x	x	x	x	x																		
Methodology						x	x	x	x															
Data Gathering								x	x	x	x													
Data Analysis											x	x	x	x	x									
Design Architecture												x	x	x	x	x	x							
Discussion																	x	x	x	x	x			
Summary and Conclusion																						x	x	x

Daftar Pustaka

[Chaudhri, 2020] Chaudhri, A. (2020). A gentle introduction to omnisci.

[Litwintschik, 2020] Litwintschik, M. (2020). 1.1 billion taxi rides using omniscidb and a macbook pro.