Galactica - Accelerated Queries Processing

Malaysian Institute of Microelectronics System (MIMOS Berhad), Malaysia.

Motivation

Traditional databases aren't equipped to efficiently compute massive volume of data, and there are related costs for storing, moving, computing and developing a fast database operation system. These areas observe quick growth of large data which needs either timely analytics or batched processing. Thus, the challenge is to analyze and mine these big data in order to effectively exploit the information to improve efficiency and quality of service for consumers and producers alike. With size, it comes performance issues.

Galactica is an emerging GPU database engine that accelerating analytical computation with parallelizing queries processing and exploiting NVIDIA high performance Tesla GPUs.

Evaluation 1: Vs PostgreSQL

Test Data

- Using a standard and popular TPC-H benchmark
- Performing 3 different sets of query in 1GB, 10GB and 50 GB sizes.
- Comparing Galactica against one of the standard and powerful open source object-relational database system (PostgreSQL 9.3)

Environment

- Dual Intel(R) Xeon(R) CPU X5680 @ 3.33GHz with 6 cores processors
- 22 GB RAM, WD HDD 1TB
- NVIDIA Tesla K20c and K40c
- Microsoft Windows 7 (64 Bits)
- CUDA 5.5 and 6.0

Evaluation 2: Vs Hadoop

Test Data

- Using Health Care Database with 32GB data size
- Performing 7 different sets of query
- Comparing Galactica against PostgreSQL 9.3, Hadoop Hive and Impala Hive

Environment

Hadoop: HP DL380p G8 Server with 48 cores and 96GB RAM, 7 VM (12 cores for master node and 6 cores each for worker nodes)

Galactica: Intel(R) Xeon(R) E5630 @ 2.53GHz with Quad Core Processor, NVIDIA Tesla K20c, 12 GB RAM, WD HDD 1 TB, Windows Server 2008 R2 Enterprise 64-bit

Introduction

1. Preprocessor
- Profiling hardware and data
- Crumbling the input data into segments
- Processing data into a disk storage

2. GPU Query Processing Engine
- Providing the analysis and optimization for the input multi queries
- Parsing, transforming and executing the generated queries into a parallel instruction sets

3. Load Balance Controller
- Managing the streams which is required APIs based on the instruction sets for parallel execution

4. Daemon Server
- Communicating Galactica to the outside world application through network
- Queuing and managing queries execution

LESSONS LEARNED

- Galactica shows the speedup of over two orders of magnitude over the same operation that had done in PostgreSQL on a multicore machine
- Galactica gained better speedup by having larger data for parallel processing
- Galactica performs on commodity data operation, also, it is an effective co-processor for performing database and query operations.
- Low cost GPU workstation has a competitive result against high end server with a distributed Hadoop system
- Parallel processing massive volume of string objects are given challenges

Future Work

- Advancing the parallel compression engine by optimizing data transfer time
- Supporting more standard SQL queries
- Exploring distributed queries execution across GPU cluster

ACKNOWLEDGEMENT

This research was done under Joint Lab of "NVIDIA - HP - MIMOS GPU R&D and Solution Center". This is the first GPU solution center in South East Asia established in October 2012. Funding for the work came from MOSTI, Malaysia. The authors would like to thank Prof. Simon See and Pradeep Gupta from NVIDIA for the supports.

CONTACT INFORMATION

Author : Yong Keh Kok
Email : kk.yong@mimos.my
Website : http://gpu.mimos.my

Company : MIMOS Berhad (Malaysia)
Accelerative Technology Lab - ATL