Middleware Framework Approach for BigData Analytics Using GPGPU

Ettikan Kandasamy Karuppiah
(ettikan.karuppiah@mimos.my)
GPU Team @MIMOS for GTC2014

R&D Activities carried out at
Nvidia-MIMOS Joint Lab
(First in South East Asia)

AccLib

Innovation for Life™

Outline

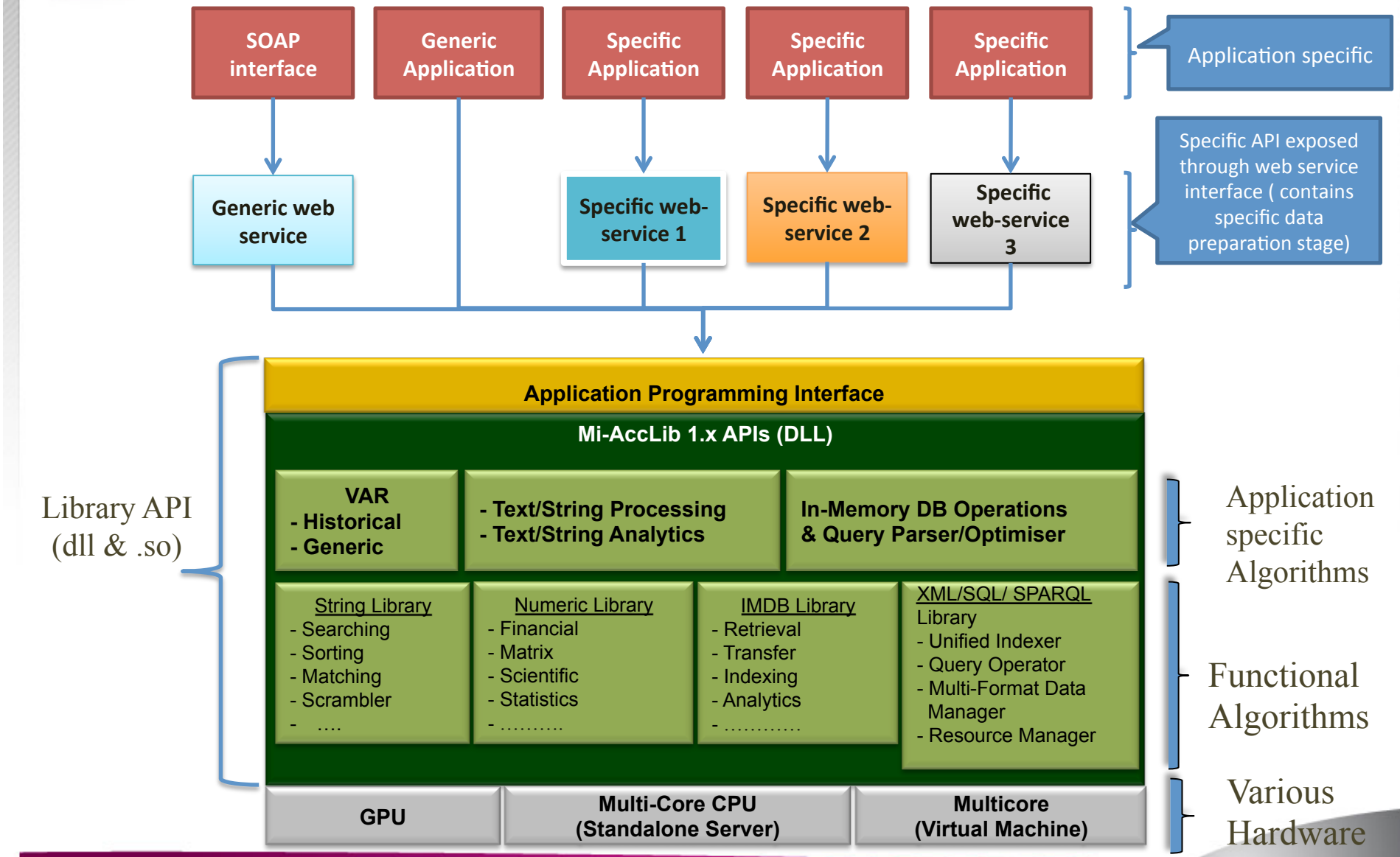
AccLib

- Challenges of GPU Libraries adaptation in BigData Analytics
- Middleware Framework Approach
- Hiding the Computation and Exposing the Presentation
  - Computation by GPGPU
  - Presentation by Visualizer
- Performance Comparison
- Conclusion
- Future Works

Challenges of GPU Libraries adaptation in BigData Analytics

- Most of our library users are non-scientific in nature
- GPGPU is seen as an "Acceleration Co-Processor"
- Expectation is for ease of use from application layer
- Libraries are treated seamlessly with independent function call
  - Preference for simpler programming languages
  - Java, Web service calls using SOAP, RMI, Socket connection
- Hide the algorithmic complexity with simple API parameters
- Structured & Unstructured Data in xxx Gigabytes

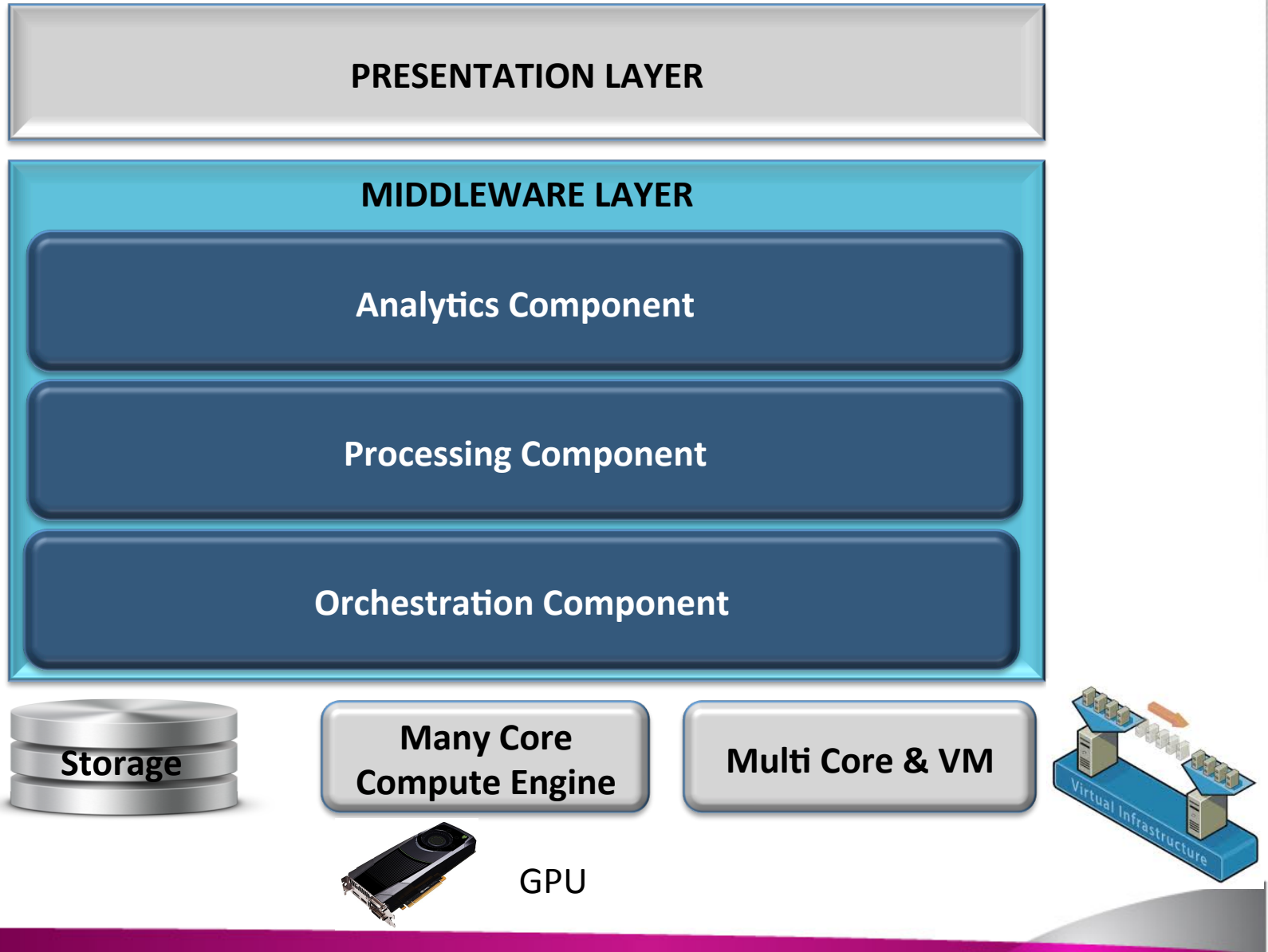
Overall Architecture



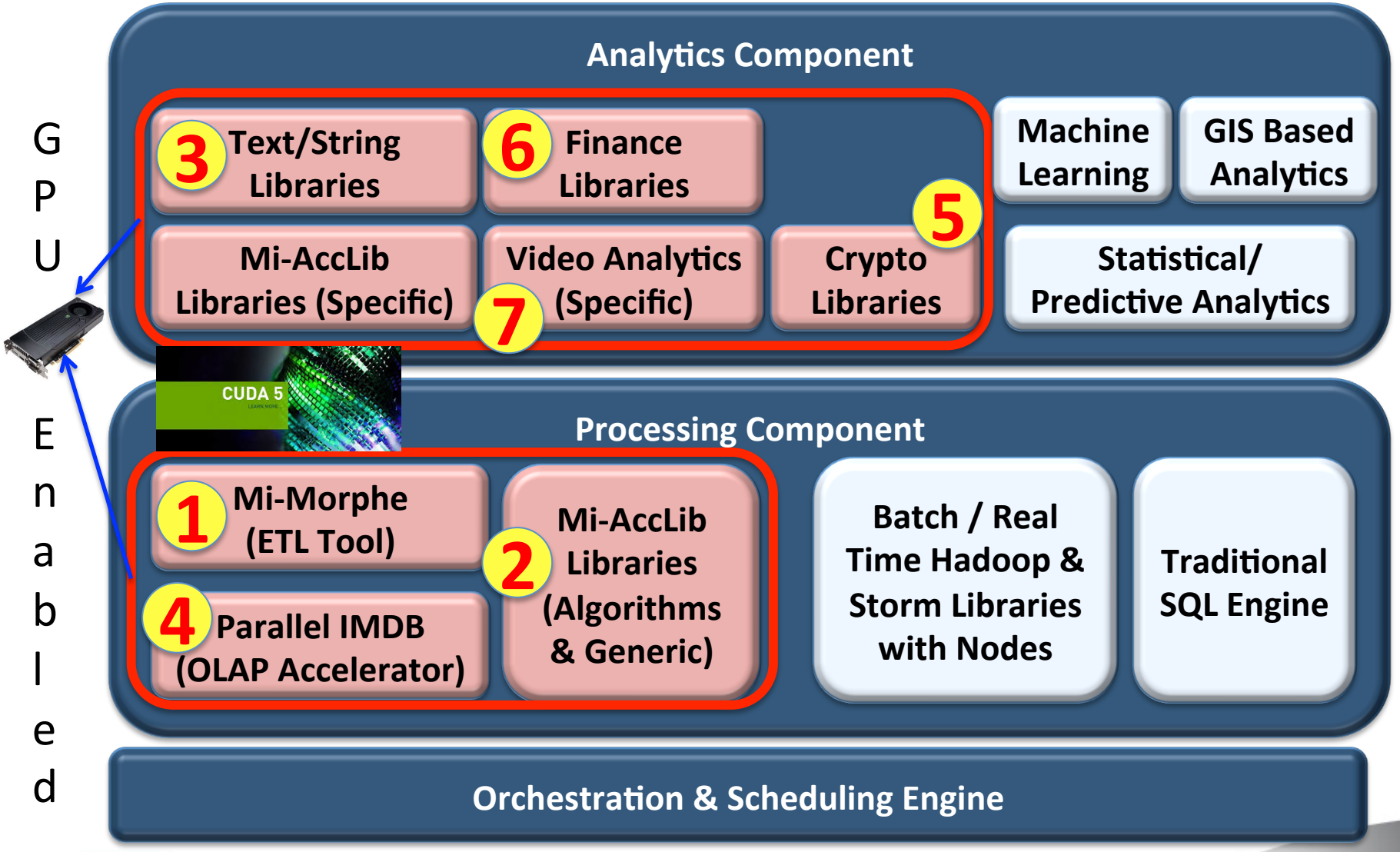
Outline

- Challenges of GPU Libraries adaptation in BigData Analytics
- Middleware Framework Approach
- Hiding the Computation and Exposing the Presentation
  - Computation by GPGPU
  - Presentation by Visualizer
- Performance Comparison
- Conclusion
- Future Works

MIMOS Middleware Framework - Layers



MIMOS Middleware Layer



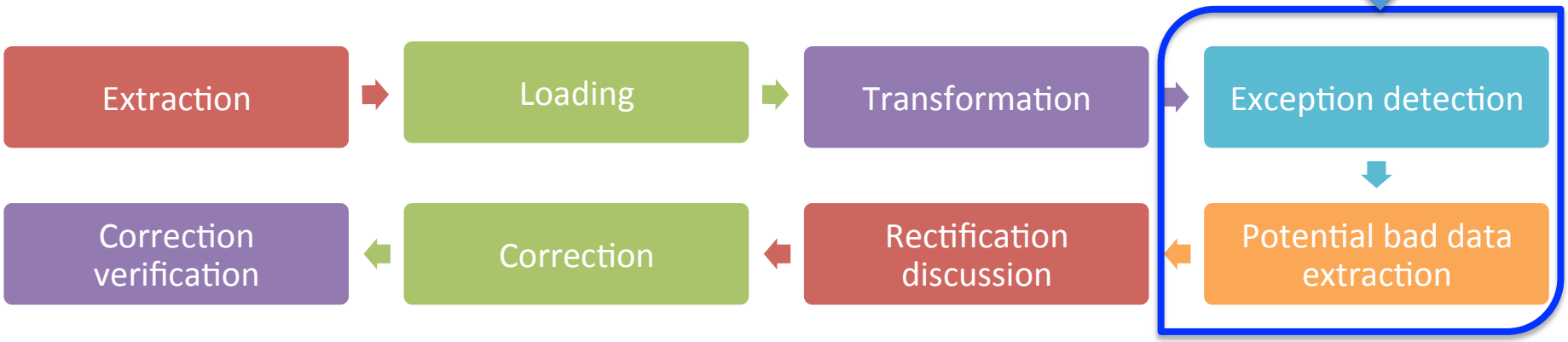
Outline

- Challenges of GPU Libraries adaptation in BigData Analytics
- Middleware Framework Approach
- Hiding the Computation and Exposing the Presentation
  - Computation by GPGPU
  - Presentation by Visualizer
- Performance Comparison
- Conclusion
- Future Works

1 Use Case: Data Cleansing

- Data Detection and Rectification
- **7711 cleansing rules**
- Key 8 steps:

Using MiAccLib



AccLib

Data Cleansing Challanges

M-MORPHE

1

- 31 systems from heterogeneous environment:
  - Environment: UNISYS, AS400, Windows, Linux.
  - Data source: DMS1100, DB2, Informix, MS SQL, MySQL, Excel, MS Access, Foxpro and flat files.

- Big Data with Big Computation:
  - **319 source data**
  - Involves ~**1 billions records**, e.g.:
    - 15 millions employee with 150 millions of monthly contribution
    - 880,000 employer with 65 millions of monthly contributions
    - Match against reference/valid data with 15 million records

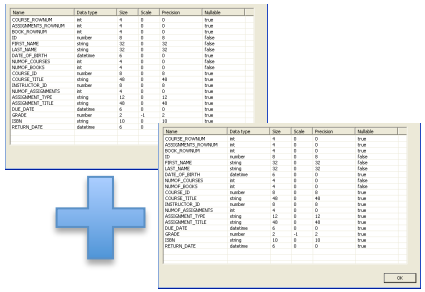
AccLib

Data Cleansing

i-MORPHE

1

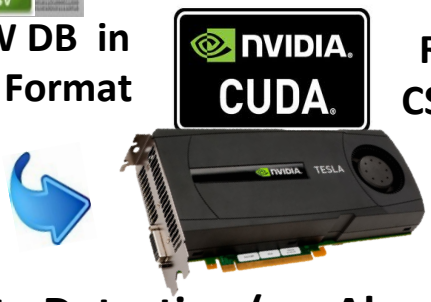
Source Database



Accelerated Duplicate Detection

~15 Millions VS ~15 Millions search

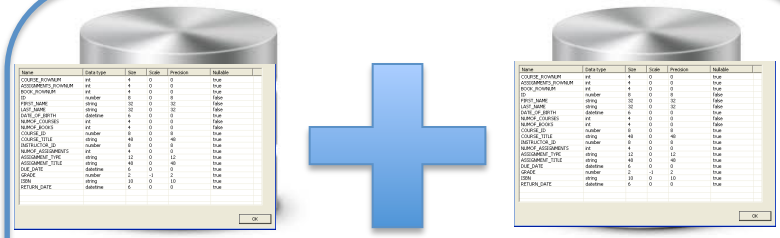
RAW DB in CSV Format



Results in CSV Format

Data Detection (e.g Algorithm used):

1. **Exact match**
2. **Edit distance**
3. **Numeric Distance**
4. **Date Distance**



Source Database

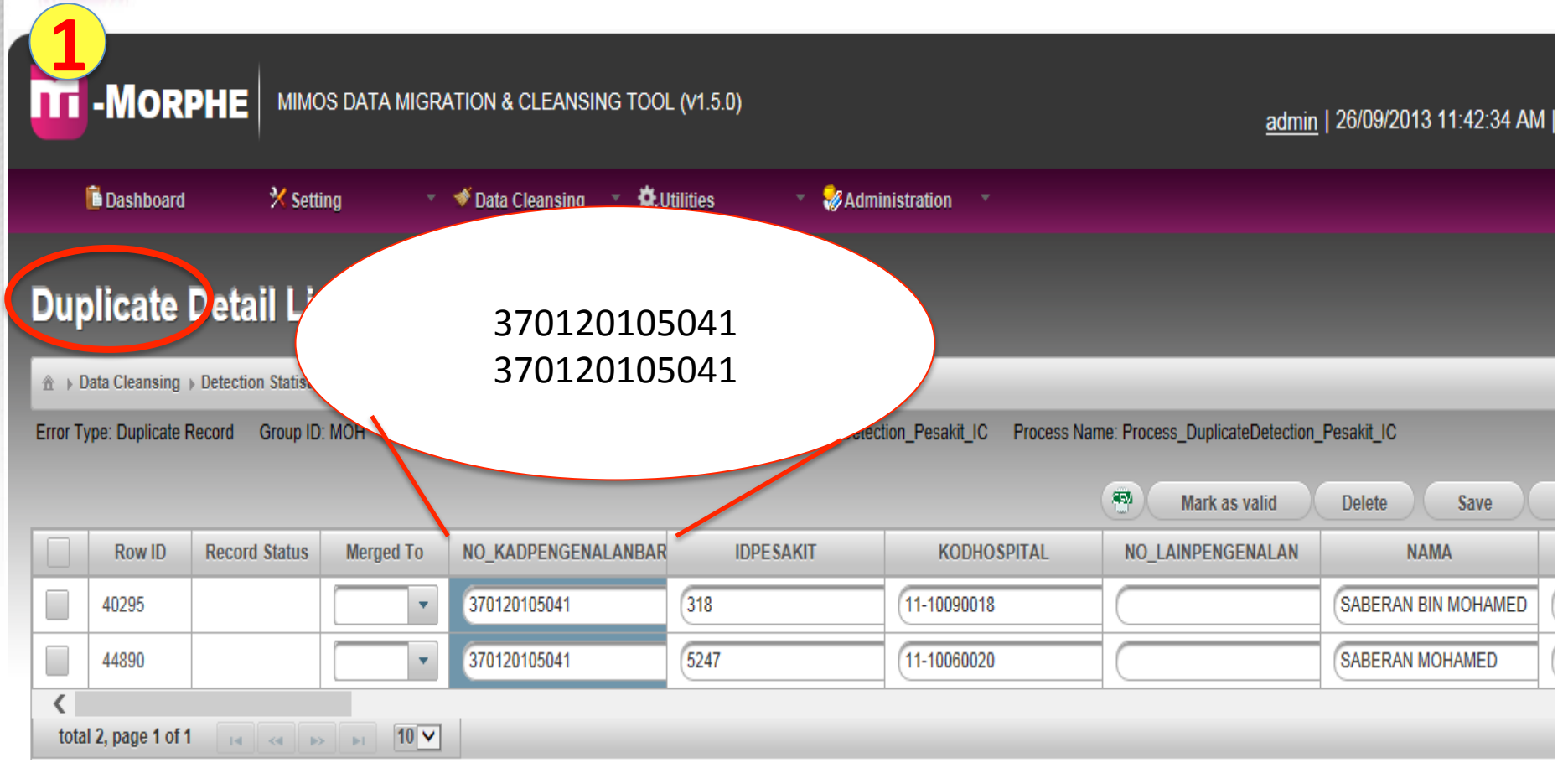
Reference Database

Accelerated Data Validation with Reference Data

AccLib

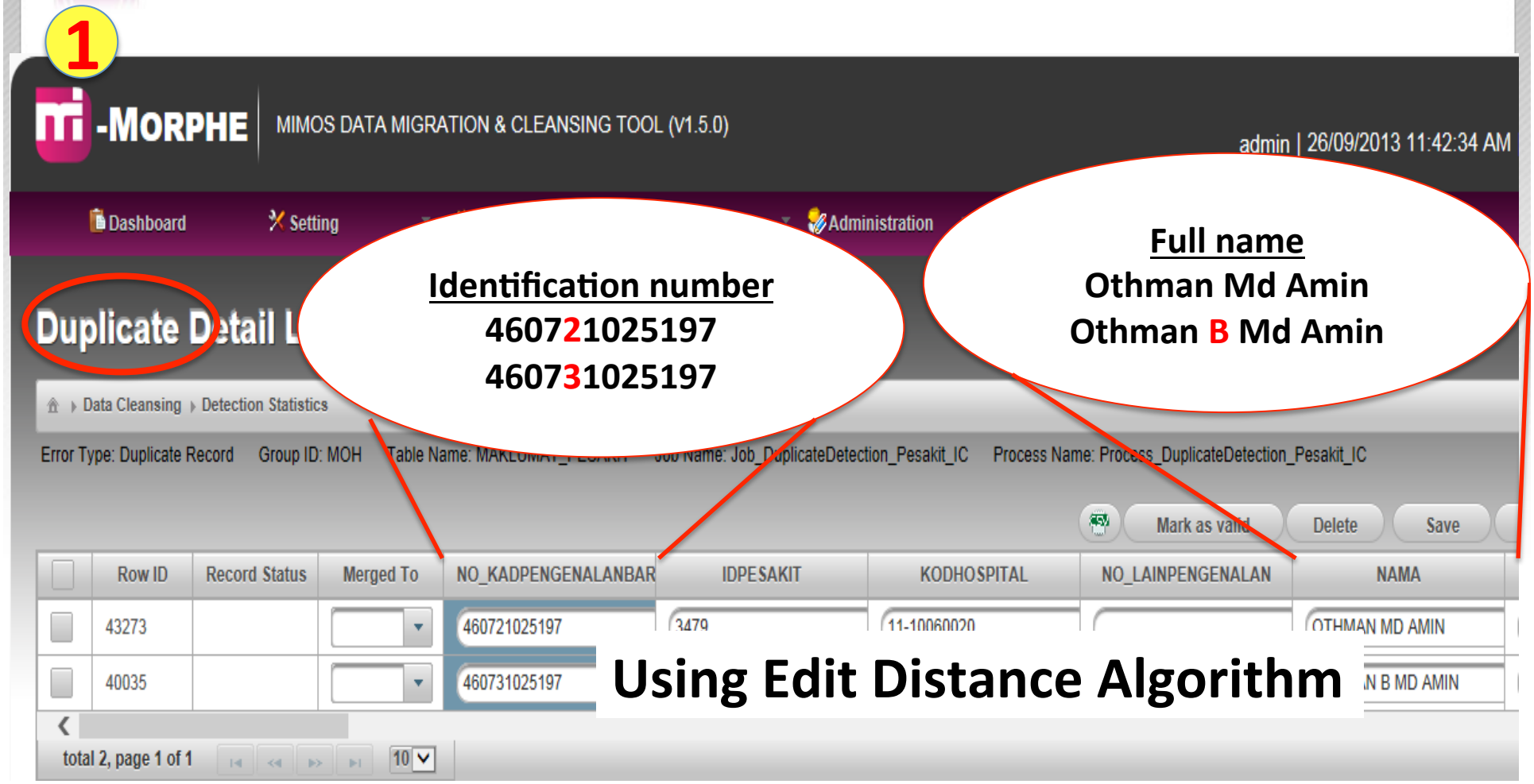
Data Duplication e.g (1)

Mi-Morphe



Mi-Morphe – Data Cleansing Tool, accelerated by GPU for performance and cost efficiency

Data Duplication e.g (2)

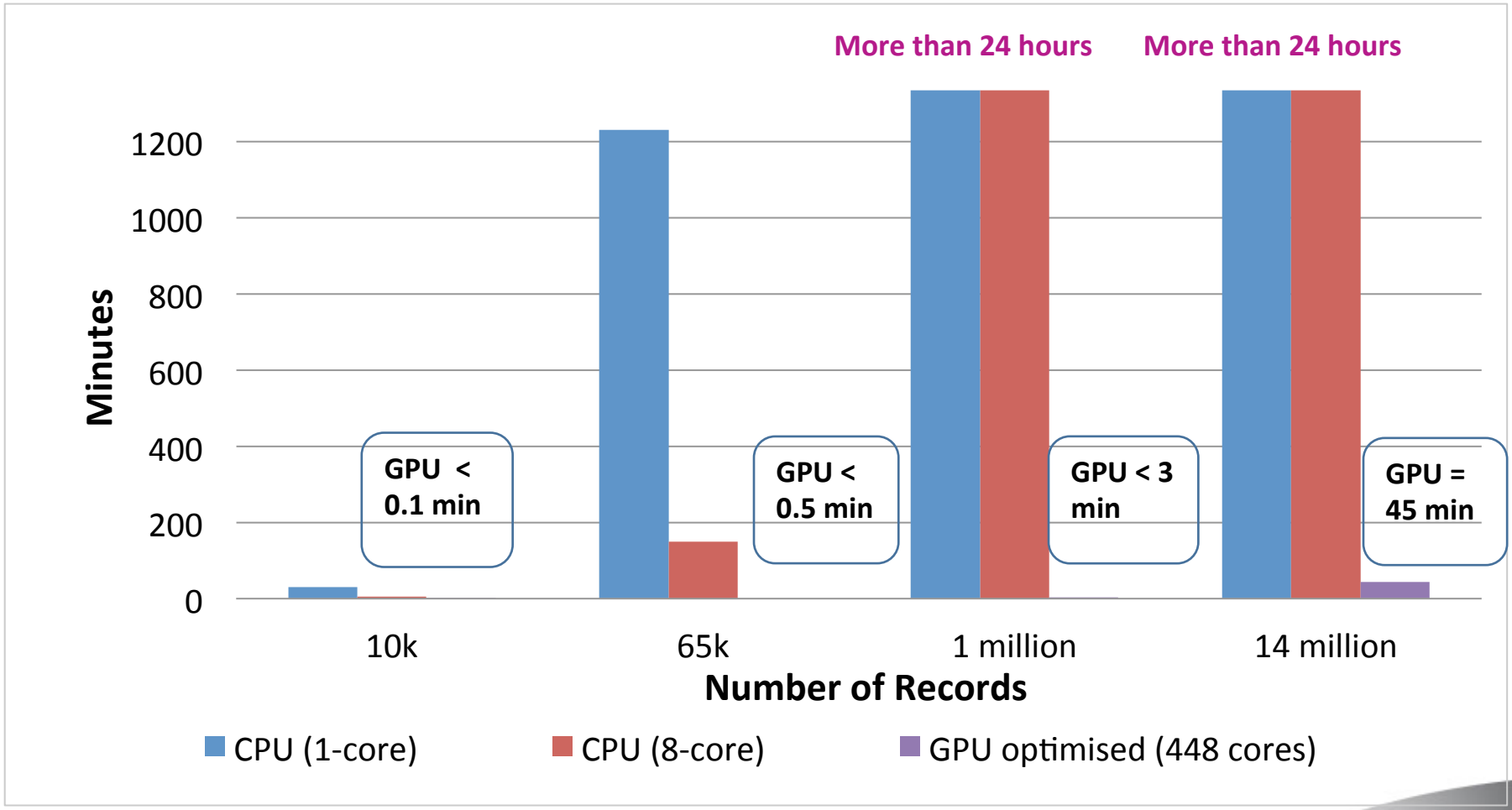


AccLib

Data Cleansing Performance

2

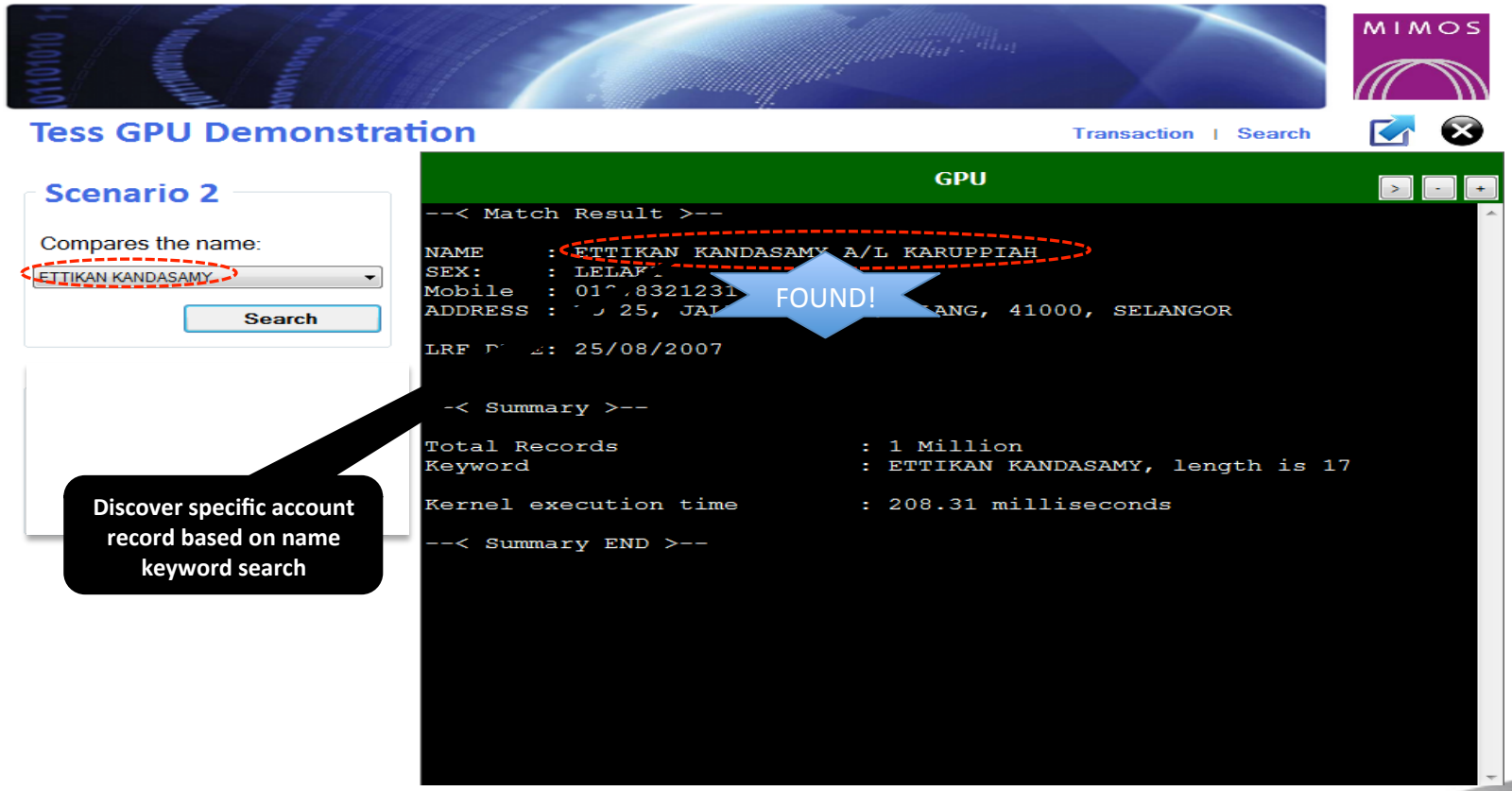
Performance Comparison Using Edit Distance Algorithm



Fast Name Search - Growing List

3

Compares the name from the blacklisted names against all daily transactions, and MiAccLib will detect and alert the users.

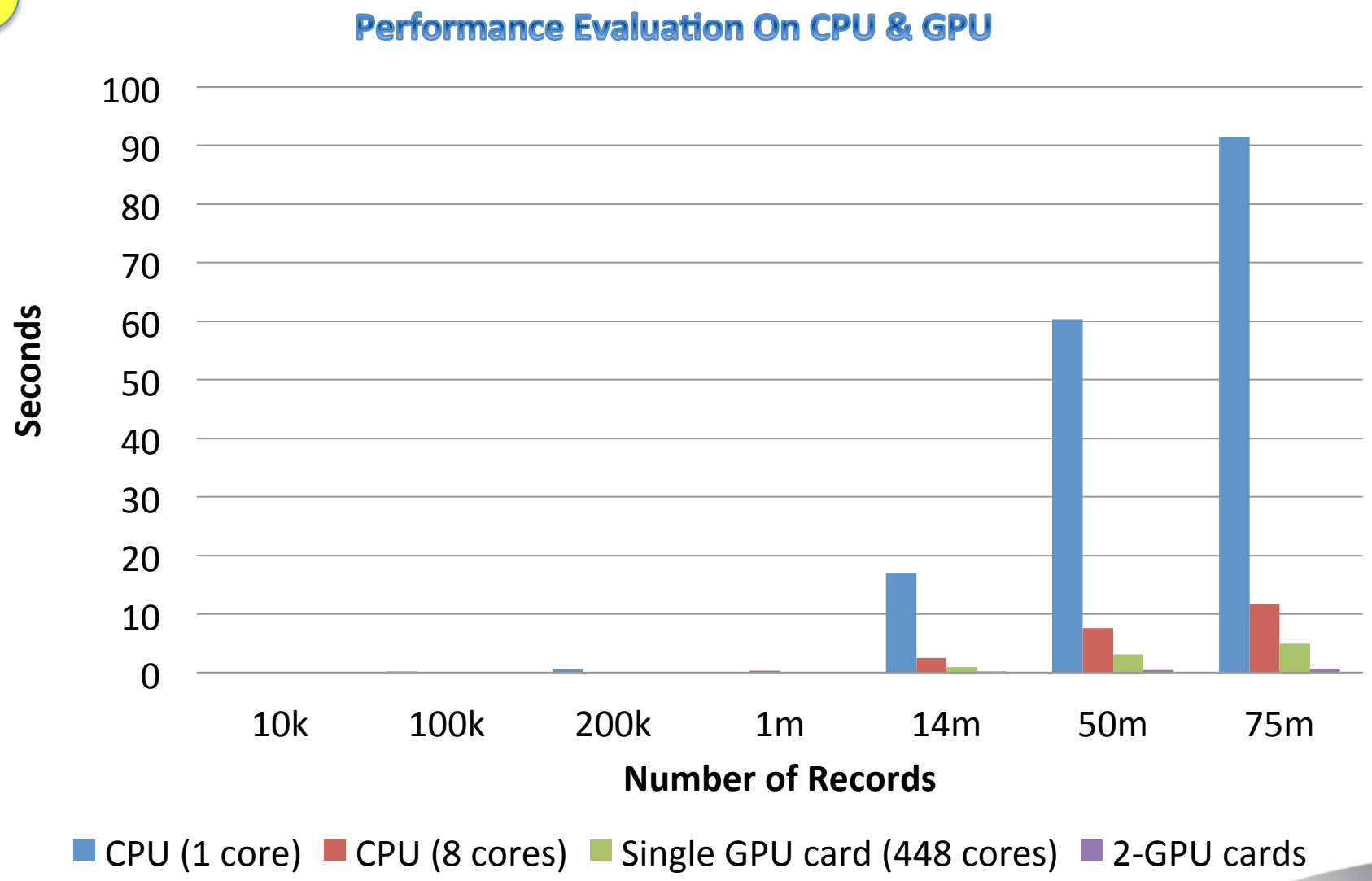


Example of Text/String Analytics

AccLib

3

Performance of String Matching using Edit Distance Algorithm



AccLib

BI Use Case

bis

4

Dashboard

PRESENTATION LAYER

MIDDLEWARE LAYER

Analytics Component

Parallel DB (Galactica)

Processing Component

Orchestration Engine

Data Mart

Data Mart

Data Mart

Data Warehouse

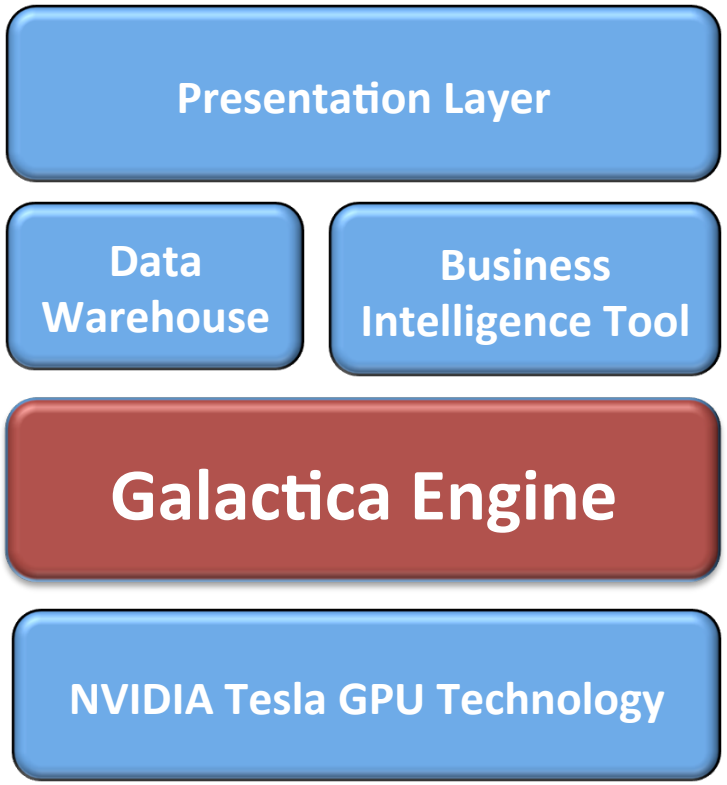
AccLib

Parallel Database

bis

4

- Orchestration of Heterogeneous Hardware Components
  - CPU with GPU
- Massively data parallel processing
  - Analytical algorithms
- Easy to access parallel engine
  - SQL style accessing



CATEGORY: BIG DATA ANALYTICS & DATA ALGORITHMS - BD03
POSTER P4149 | CONTACT NAME Keh Kok Yong: kk.yong@mimos.my
GPU TECHNOLOGY CONFERENCE

Galactica - Accelerated Queries Processing

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

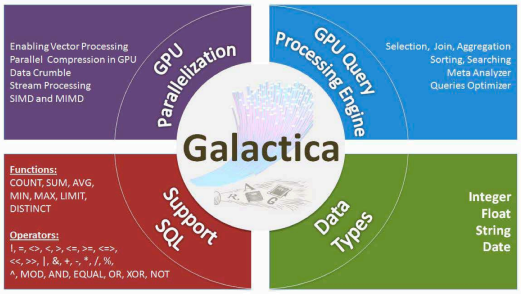
AccLib

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.

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

Evaluation 1: Vs PostgreSQL

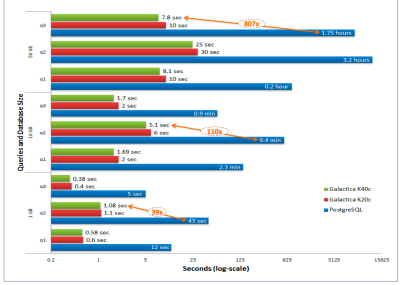
Test Data

- 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

RESULT 1



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

Evaluation 2: Vs Hadoop

Test Data

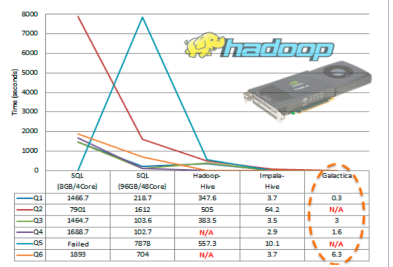
- Using Health Care Database with 32GB data size
- Performing 7 different sets of query
- Comparing Galactica against PostgreSQL 9.3, Hadoop Hive and Impala Hive

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

RESULT 2



	SQL (8GB/4Core)	SQL (96GB/48Core)	Hadoop-Hive	Impala-Hive	Galactica
Q1	1466.7	218.7	347.6	3.7	0.3
Q2	7901	1612	505	64.2	N/A
Q3	1464.7	103.6	383.5	3.5	3
Q4	1688.7	102.7	N/A	2.9	1.6
Q5	Failed	7878	557.3	10.1	N/A
Q6	1893	704	N/A	3.7	6.3

N/A – Not Available – Query not supported.

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
: Chua Meng Wei
: Dr. Ettikan Kandasamy Karuppiah
Company : MIMOS Berhad (Malaysia)
Accelerative Technology Lab - ATL
Email : kk.yong@mimos.my
Website : http://gpu.mimos.my

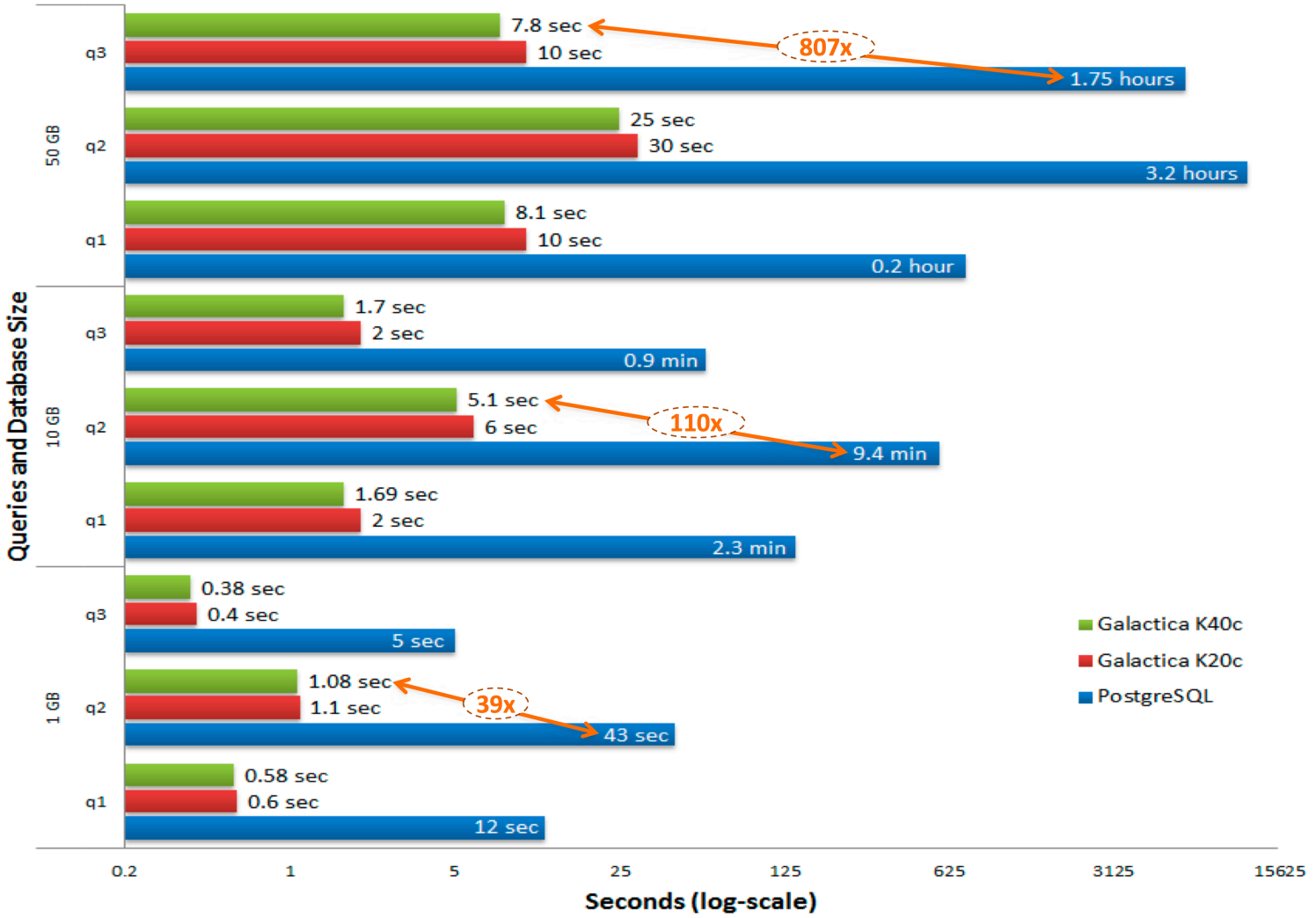
MIMOS Innovation for Life

1. P. K. Chong, Ettikan K. Karuppiah and K.K. Yong, "A Multi-GPU Framework for In-Memory Text Data Analytics", The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA), Barcelona, Spain, Mar. 2013.
2. Ettikan K. K., K.K. Yong and Keeratpal Singh., "A Middleware Framework for Programmable Multi-GPU based BigData Applications", Chapter of GPU Computing and Applications, Springer. (Expected publishing June 2014)
3. K.K. Yong and Ettikan K. K., "Hash Match on GPU", Open Systems (ICOS), 2013 IEEE Conference on , Dec. 2013
4. W.M. Ong, V. M. Baskaran, P.K. Chong, K.K. Yong and Ettikan K. K., "A Parallel Bloom Filter String Searching Algorithm on a Many-core Processor", Open Systems (ICOS), 2013 IEEE Conference on , Dec. 2013

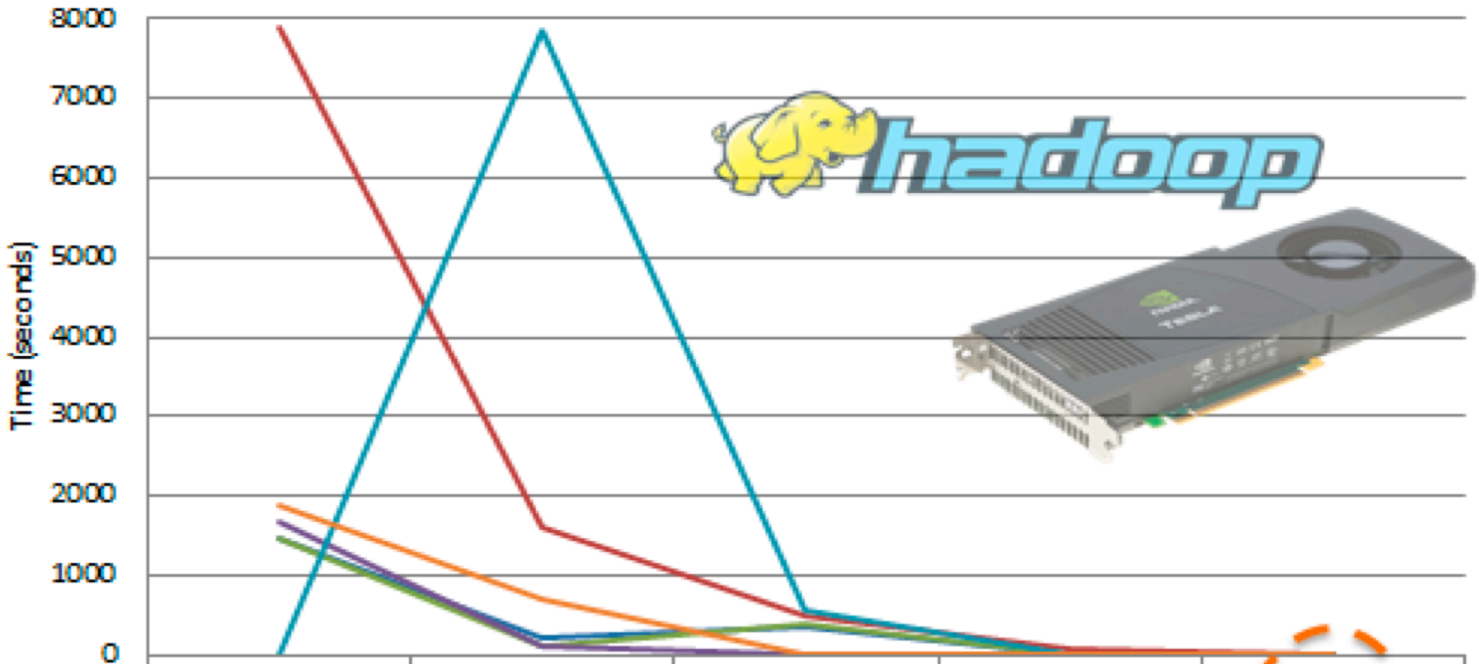
MIMOS

4

RESULT 1



RESULT 2



	SQL (8GB/4Core)	SQL (96GB/48Core)	Hadoop-Hive	Impala-Hive	Galactica
Q1	1466.7	218.7	347.6	3.7	0.3
Q2	7901	1612	505	64.2	N/A
Q3	1464.7	103.6	383.5	3.5	3
Q4	1688.7	102.7	N/A	2.9	1.6
Q5	Failed	7878	557.3	10.1	N/A
Q6	1893	704	N/A	3.7	6.3

N/A – Not Available – Query not supported.

AccLib

Data Visualization

mi bis

4

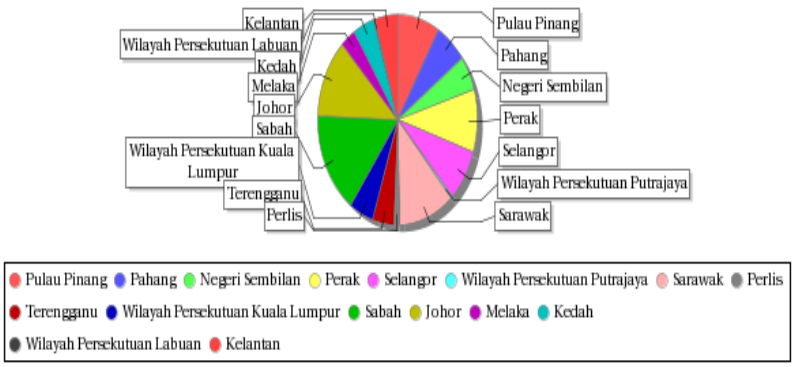
bis

User

Document browser | DOC_COMP_HDW_SMALL

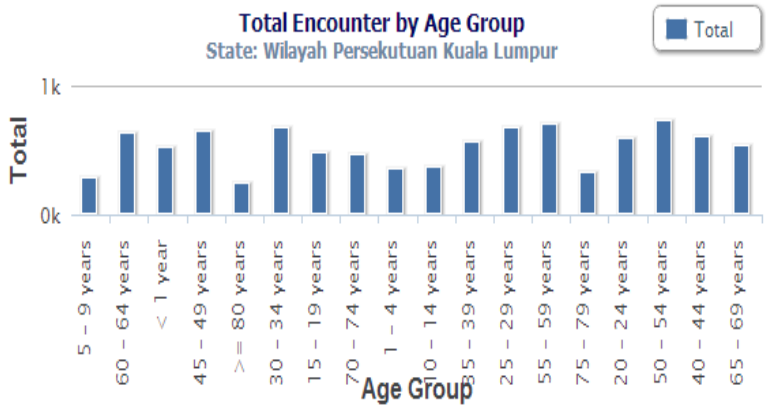
File | Info | Shortcuts

Total Encounter Of State



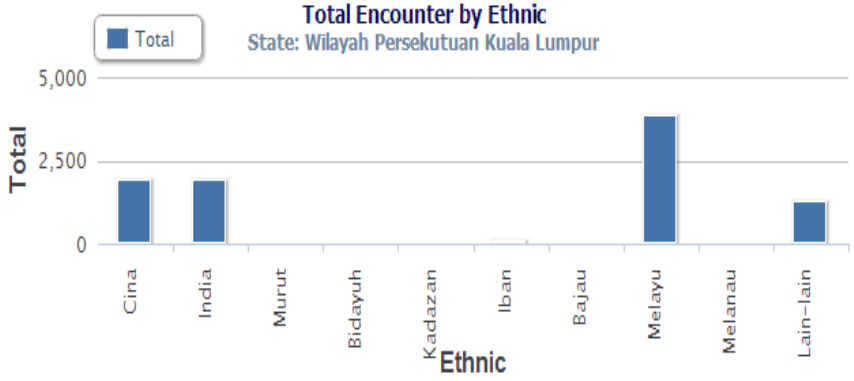
Total Encounter by Age Group

State: Wilayah Persekutuan Kuala Lumpur



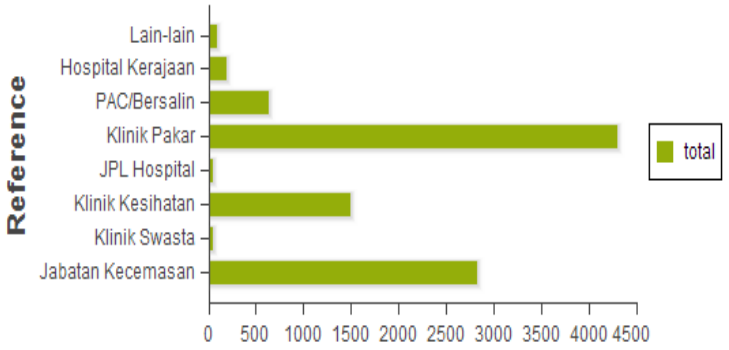
Total Encounter by Ethnic

State: Wilayah Persekutuan Kuala Lumpur



Total Encounter By Reference

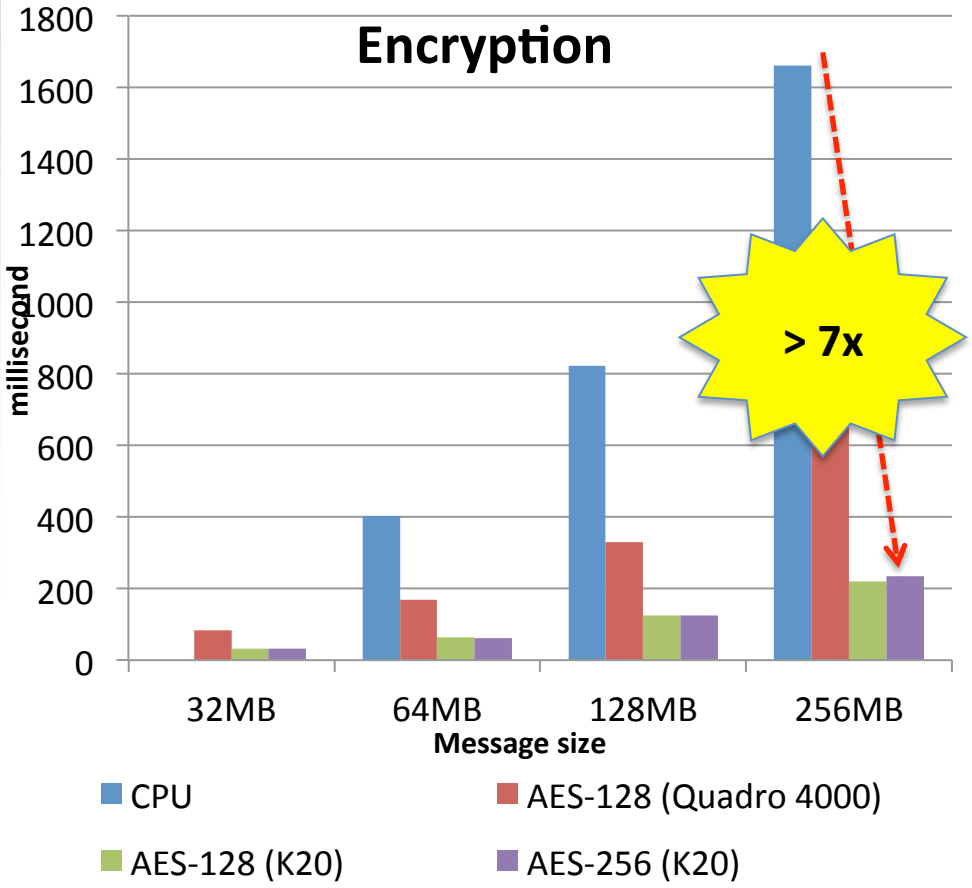
State : Wilayah Persekutuan Kuala Lumpur

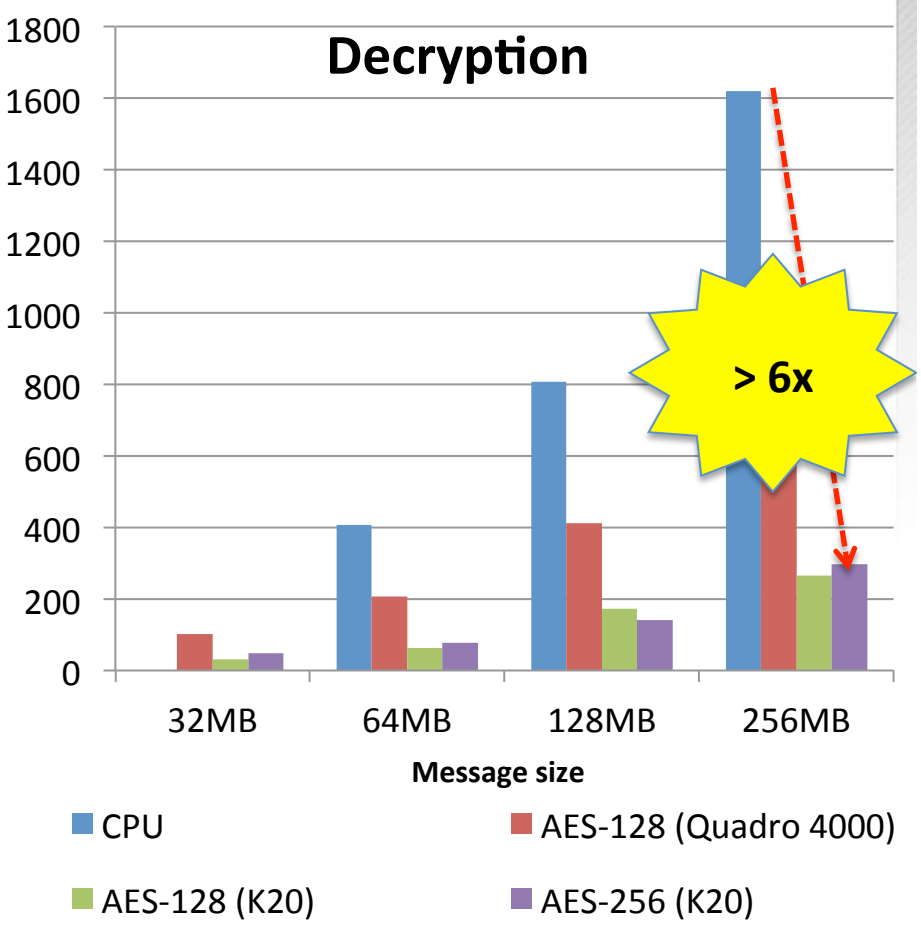


AccLib

Scrambling (Database Encrypt & Decrypt)

5 Advanced Encryption Standard(AES) 128-Bit and 256-Bit Performance Evaluation on K20 GPU card





* ECB = electronic code book;

Video Analytics Implementation in GPU

Intelligent Surveillance Platform

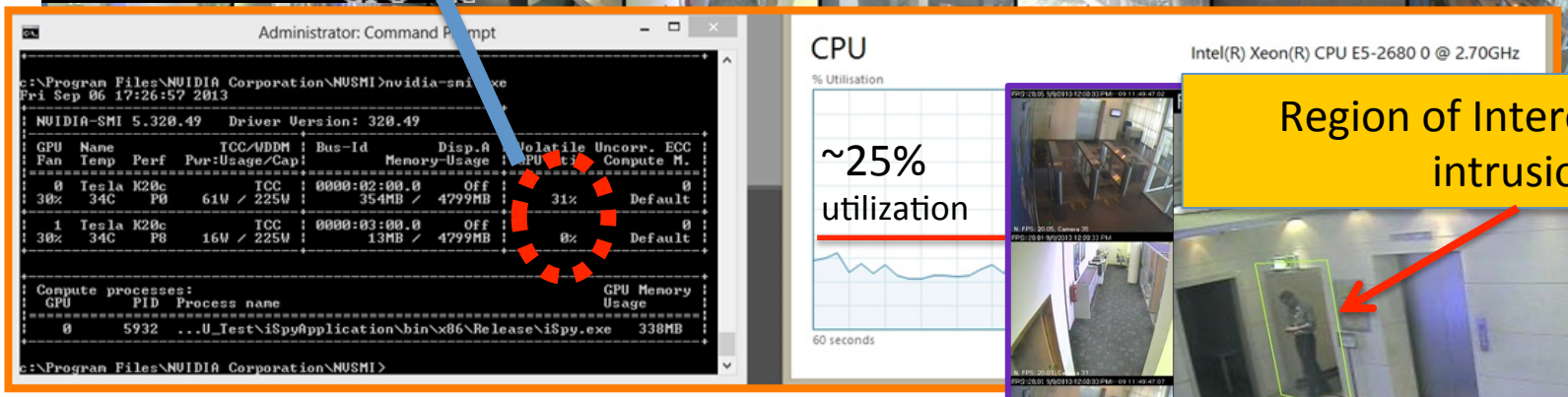
Video Analytics

- Intrusion detection
- Loitering Detection
- Slip and Fall Detection
- Unattended Object Detection
- Object Removal Detection

Video Analytics Implementation in GPU

7

35 cameras implementation



Region of Interest during intrusion

Resource Utilization for MiAccVaLib (Single & Multiple Camera)

7

Process Frame Size	CPU Utilization	Processing VA Time per frame in GPU	GPU Utilization*	GPU Memory Utilization	No. of Cameras
512	1.6 - 1.8%	~4.7ms	6 - 7%	113MB	1 camera

System	CPU Utilization	Processing VA Time per frame	GPU Utilization	GPU Memory Utilization	Total time processing time per frame	No. of Cameras
CPU + GPU/ per card	~35%	3.1ms	40%	7.3%	12ms	~35 camera

More room for computation

Transfer Rate CPU to GPU =
5.53GB/s
(for 256KB it takes about 45.5ms)

Transfer Rate GPU to CPU =
5.01GB/s
(for 64KB it takes about 13ms)

Transfer throughput is similar when using frame size 512x512

*GPU Utilization is dependent on the image scene.

GPU VA Library

Surveillance Server

IP Network

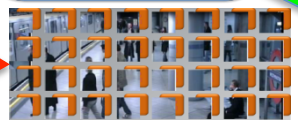
CPU + 

Camera N MJPEG Decoding

...

Camera 2

Camera 1 MJPEG Decoding



Video Analytics Processing

TESTED 100 % ACCURATE !!!!!
With offline datasets

ALERT !!!

Client PC

misp

Video Analytics Processing

CPU

AccLib

R

Parallelization of the VA algorithms

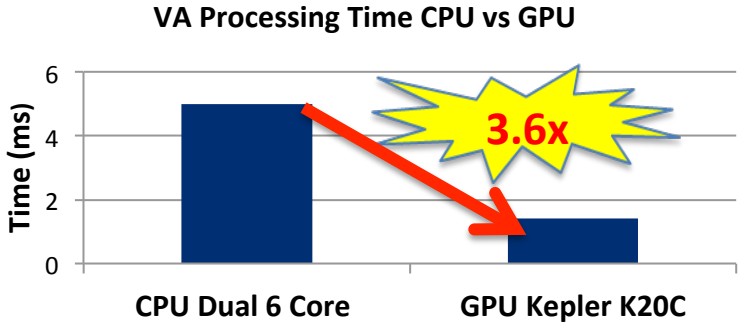
- Previous data dependency
- Efficient memory management.
- Algorithm Decomposition

CPU + GPU

D

Background Subtraction	AccBackgroundSubtractionFrameDiff AccCompMotion, AccUpdateBackground AccCompShadow, AccRGB2HSV
Morphing Process	AccMorphFilterVariable
CCL	AccConnectComponentLabel
Region Analyzer	AccExtractPropertiesCentroid, AccExtractPropertiesSize, AccExtractPropertiesBB, AccExtractPropertiesHWRatio, AccExtractPropertiesOrientation, AccExtractPropertiesHProject, AccExtractPropertiesSkew, AccRegionLabelUpdate, AccCompOverlap, AccPropUpdate, AccCombineBlob
Filters	AccFlickerFilter, AccRegionFilter
Detection	AccVAParallelIntrusionDetection

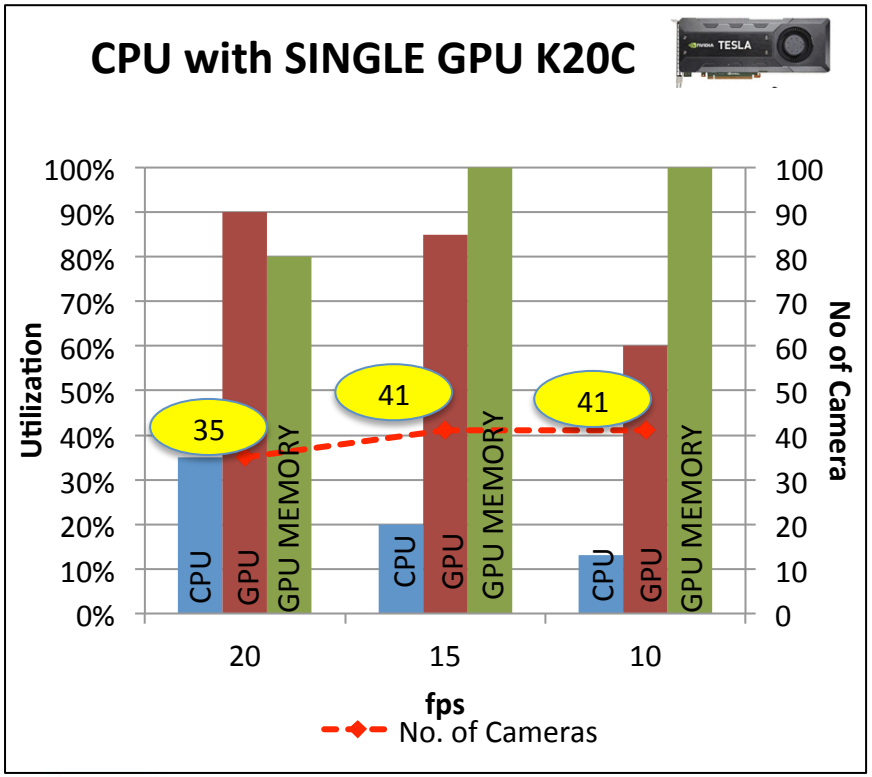
GPU VA System Results

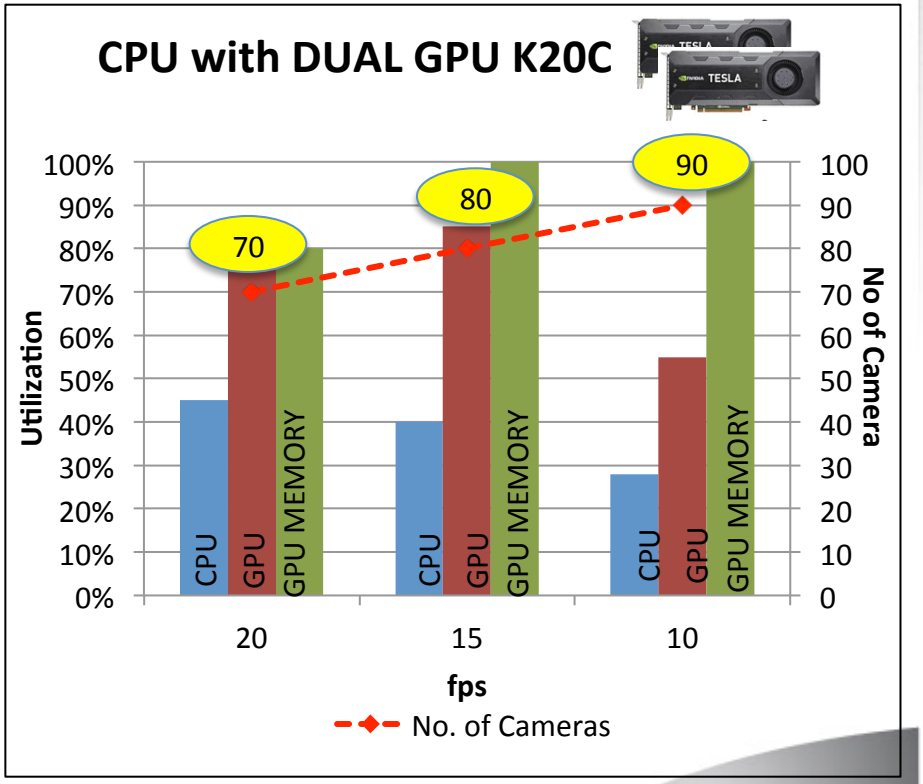


MJPEG & MPEG Supported

Tasks	CPU + GPU	* CPU Utilization
Network Stream In	CPU	10%
Decompression	CPU	5%
Video Analytics	GPU	35%
Streaming Out & Display	CPU	50%

* Reference to 10fps



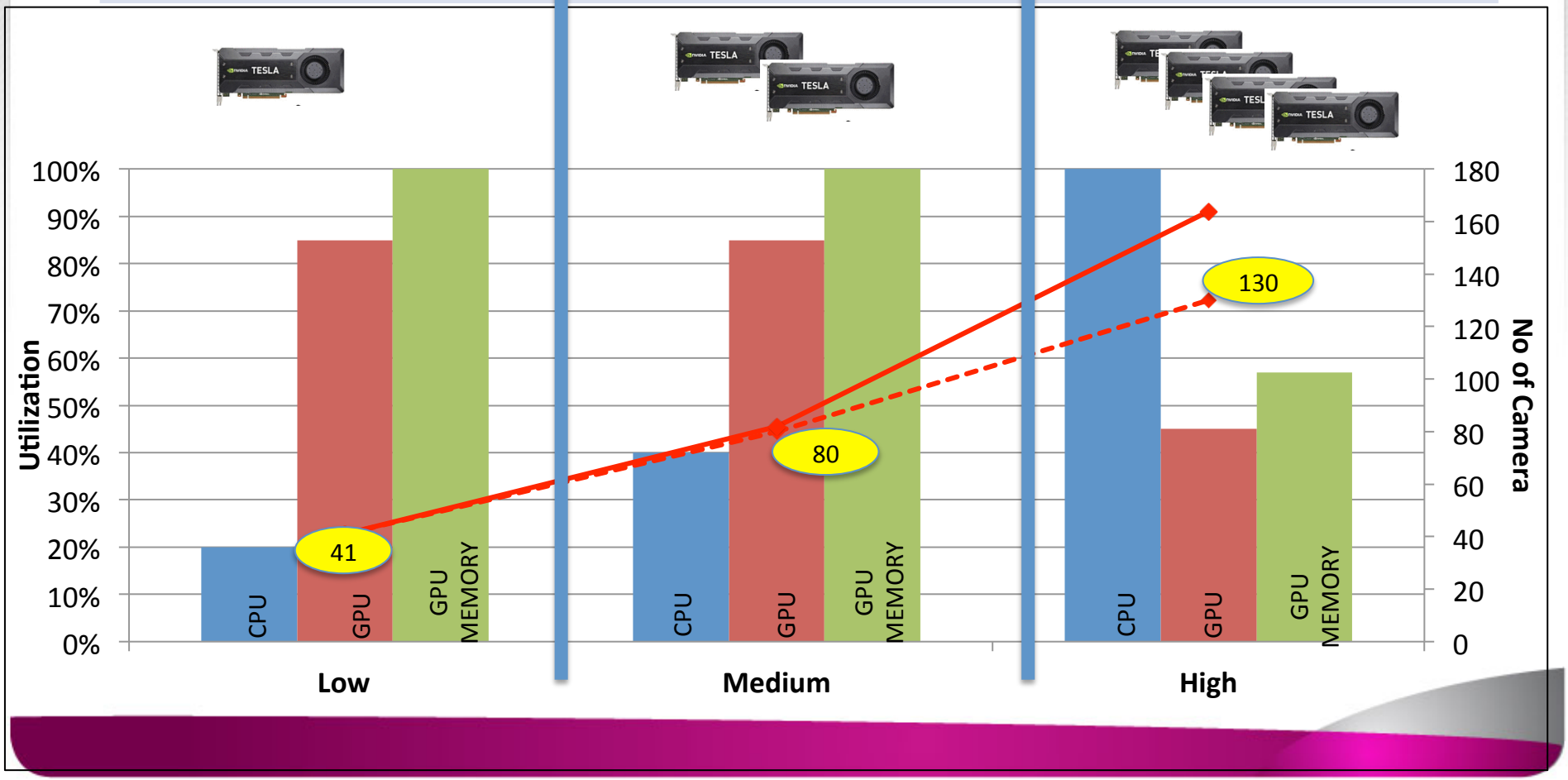


* Data taken on system server CPU - Dual 8 cores

Proposed System Configuration

Estimated Data

Low Spec (DONE)	Medium Spec (DONE)	High Spec (IN PROGRESS)
CPU - Dual 8 core FR - 15fps	CPU - Dual 8 core FR - 15fps	CPU - Dual 8 core FR - 15fps

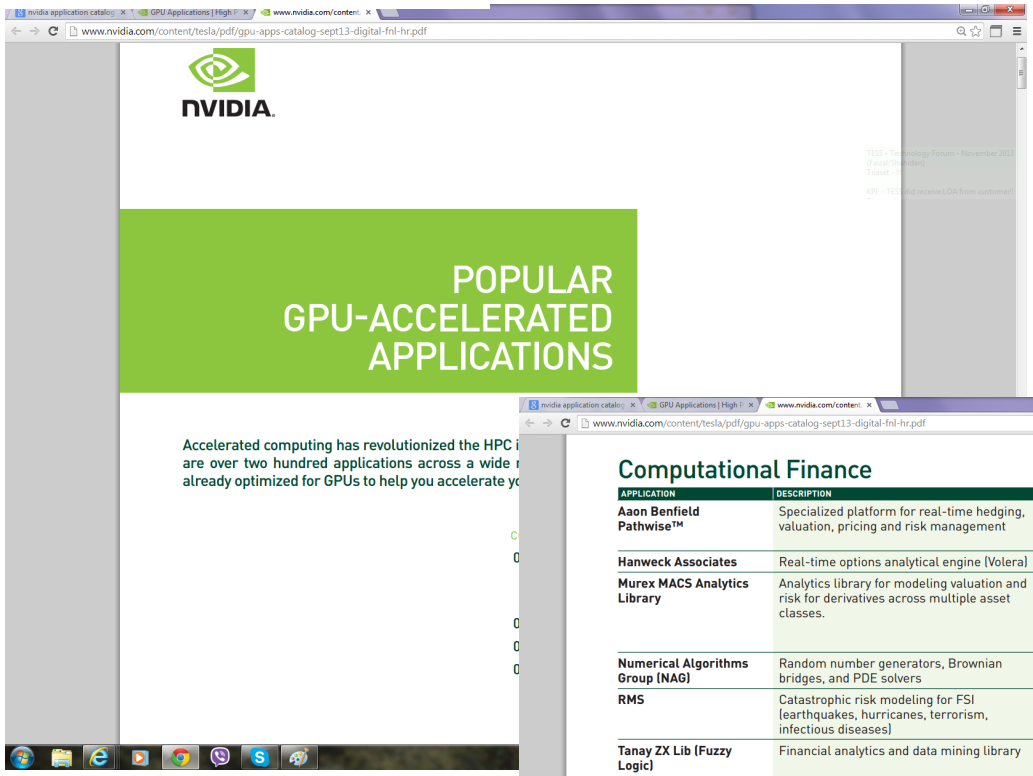


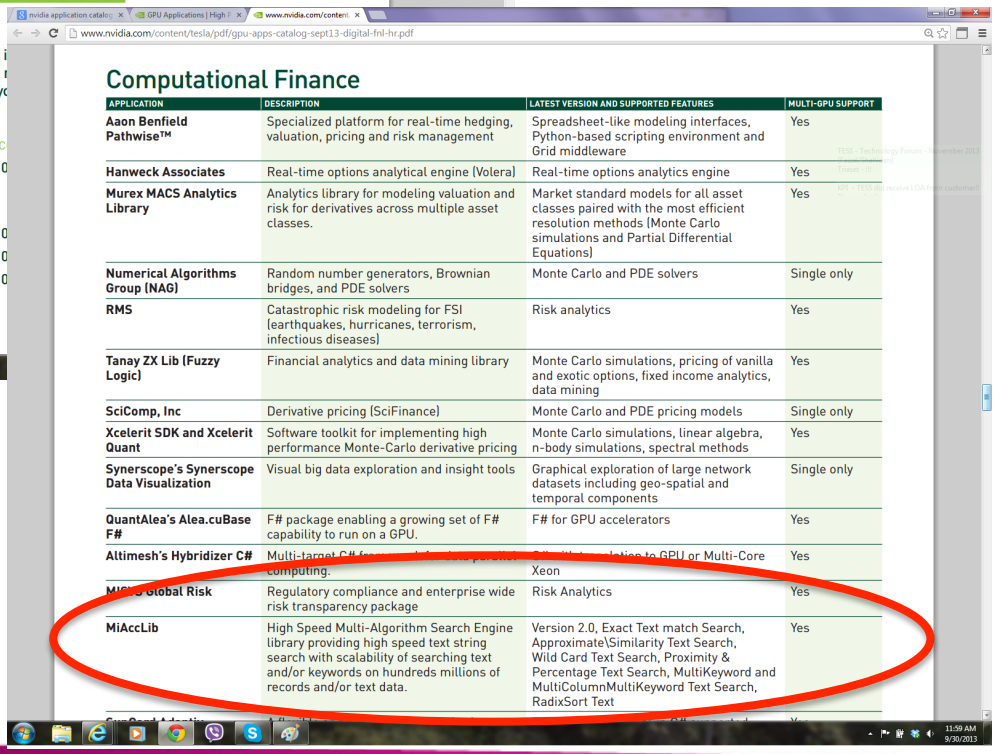
System Savings & Roadmap

No.	Items	Setup Cost (RM '000)		Annual Maintenance (RM '000)	
		CPU 50 nodes	CPU + GPU 10 nodes	CPU 50 nodes	CPU + GPU 10 nodes
1	Space Rental	0	0	136.5	52.92
2	Infrastructure	250.5	190	15	13
3	Utilities	0	0	469.2	192
4	IT Equipment	700	780		
	Total Cost	950.5	970	620.7	257.92
	Savings		-19.5		362.78
	% Savings		-2		58

> RM300k = >50%
per annum

Mimos Accelerator Libraries





Computational Finance

APPLICATION	DESCRIPTION	LATEST VERSION AND SUPPORTED FEATURES	MULTI-GPU SUPPORT
Aaon Benfield Pathwise™	Specialized platform for real-time hedging, valuation, pricing and risk management	Spreadsheet-like modeling interfaces, Python-based scripting environment and Grid middleware	Yes
Hanweck Associates	Real-time options analytical engine (Volera)	Real-time options analytics engine	Yes
Murex MACS Analytics Library	Analytics library for modeling valuation and risk for derivatives across multiple asset classes.	Market standard models for all asset classes paired with the most efficient resolution methods (Monte Carlo simulations and Partial Differential Equations)	Yes
Numerical Algorithms Group (NAG)	Random number generators, Brownian bridges, and PDE solvers	Monte Carlo and PDE solvers	Single only
RMS	Catastrophic risk modeling for FSI (earthquakes, hurricanes, terrorism, infectious diseases)	Risk analytics	Yes
Tanay ZX Lib (Fuzzy Logic)	Financial analytics and data mining library	Monte Carlo simulations, pricing of vanilla and exotic options, fixed income analytics, data mining	Yes
SciComp, Inc	Derivative pricing (SciFinance)	Monte Carlo and PDE pricing models	Single only
Xcelerit SDK and Xcelerit Quant	Software toolkit for implementing high performance Monte-Carlo derivative pricing	Monte Carlo simulations, linear algebra, n-body simulations, spectral methods	Yes
Synerscope's Synerscope Data Visualization	Visual big data exploration and insight tools	Graphical exploration of large network datasets including geo-spatial and temporal components	Single only
QuantAlea's Alea.cuBase F#	F# package enabling a growing set of F# capability to run on a GPU.	F# for GPU accelerators	Yes
Altimesh's Hybridizer C#	Multi-target C#	Xeon	Yes
Global Risk	Regulatory compliance and enterprise wide risk transparency package	Risk Analytics	Yes
MiAccLib	High Speed Multi-Algorithm Search Engine library providing high speed text string search with scalability of searching text and/or keywords on hundreds millions of records and/or text data.	Version 2.0, Exact Text match Search, Approximate\Similarity Text Search, Wild Card Text Search, Proximity & Percentage Text Search, MultiKeyword and MultiColumnMultiKeyword Text Search, RadixSort Text	Yes

Release 1.0
(new functionalities
under development & verification for
MiAccLib2.0)

Outline

- Challenges of GPU Libraries adaptation in BigData Analytics
- Middleware Framework Approach
- Hiding the Computation and Exposing the Presentation
  - Presentation by GPGPU
  - Computation by Visualizer
- Performance Comparison
- Conclusion
- Future Works

Conclusion

- GPU Libraries needs to be written with understanding of users' specific needs
- Users prefer to treat GPU as an hardware accelerator, especially in non-scientific community
- Performance gain by using GPU certainly applauded
- Challenge is to write generic libraries, solving many specific problems!

Future Work

- MiAccLib will be continuously enhanced, supported and released in coming version, ver2.0.

Back-up