Big Data Systems on Future Hardware

Bingsheng He

NUS Computing

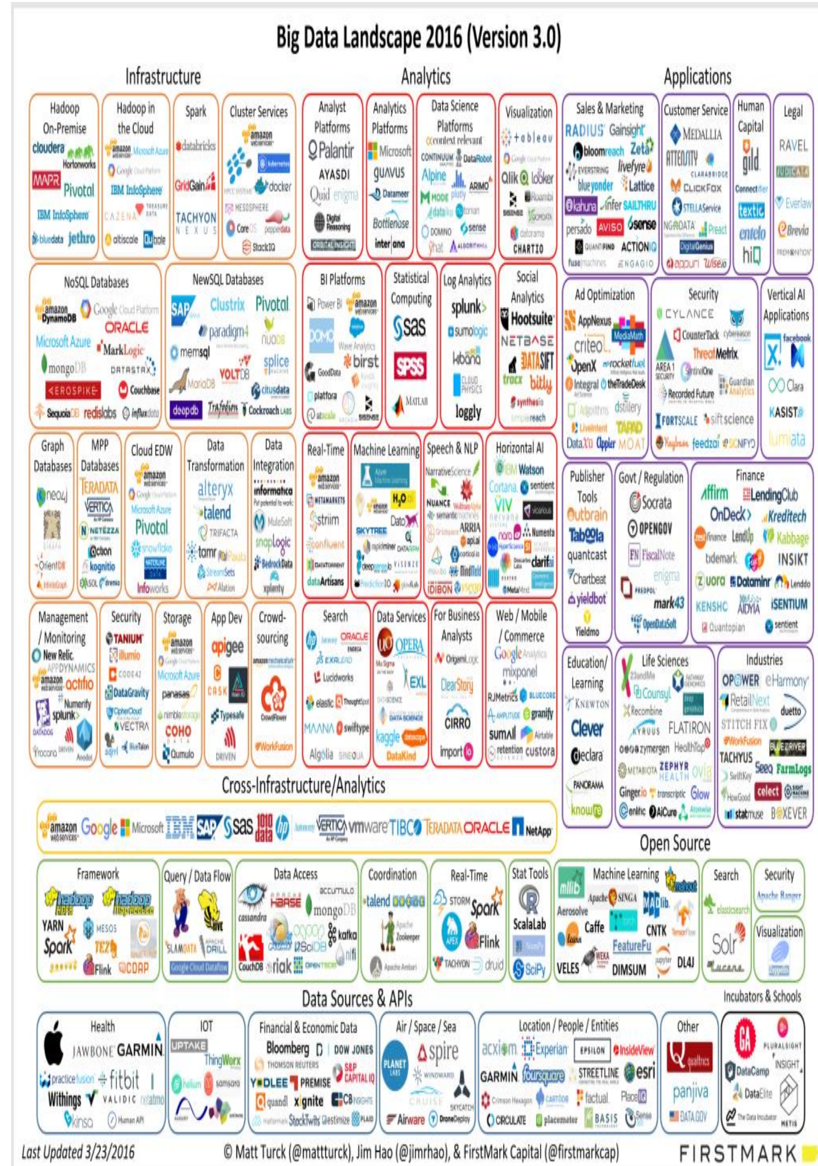
http://www.comp.nus.edu.sg/~hebs/

Outline

- Challenges for Big Data Systems
- Why Hardware Matters?
- Open Challenges
- Summary

3 ANYs in Big Data

- From enterprises to **anyone**
  - Internet of things, mobile, NGS (next gen sequencing)…
- From structured data to **any form**
  - Data warehouse, text, streaming, graphs, JSON …
- From SQL to **any analytics/processing**
  - MapReduce, R, eScience…

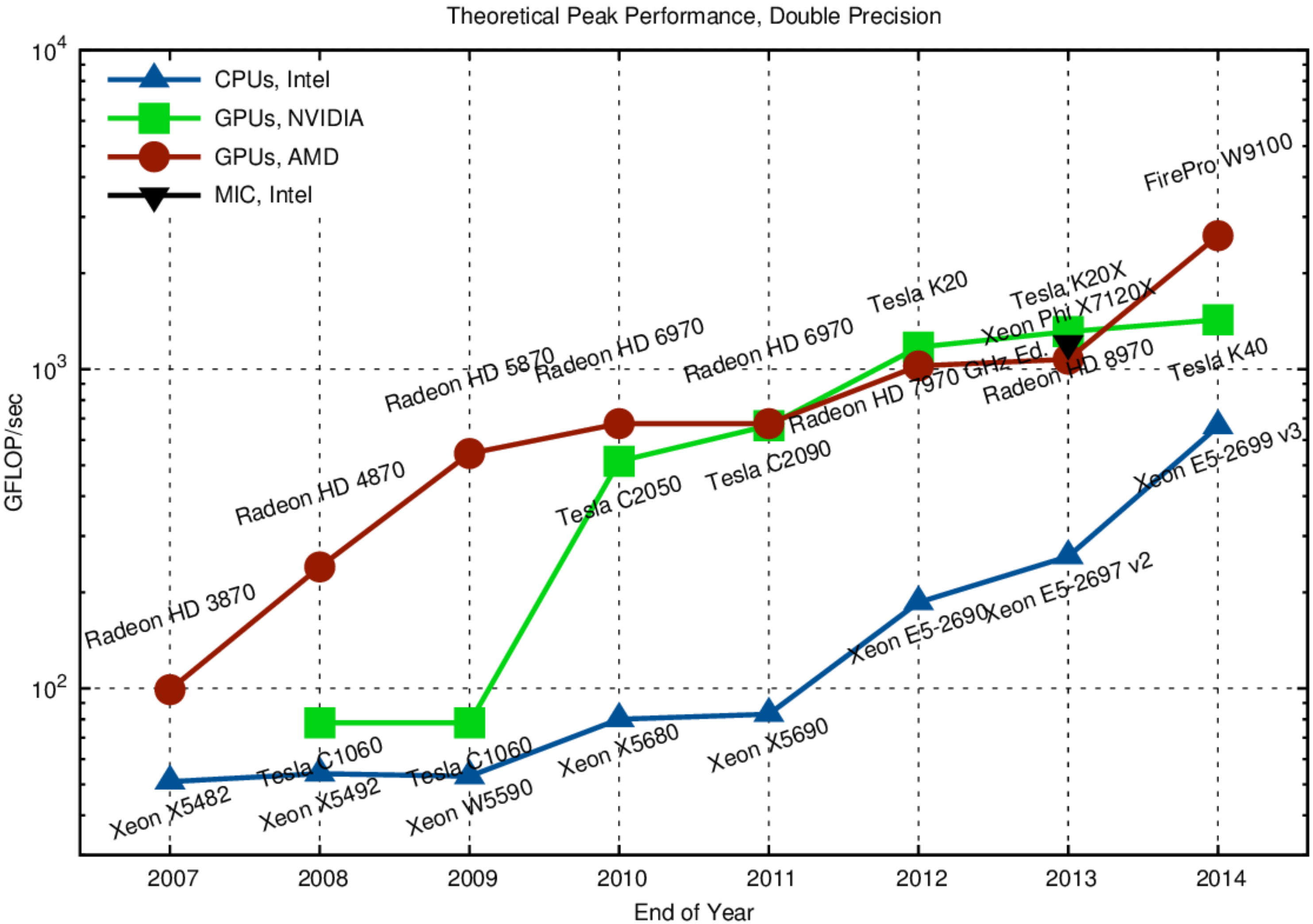


"One size does not fit all"

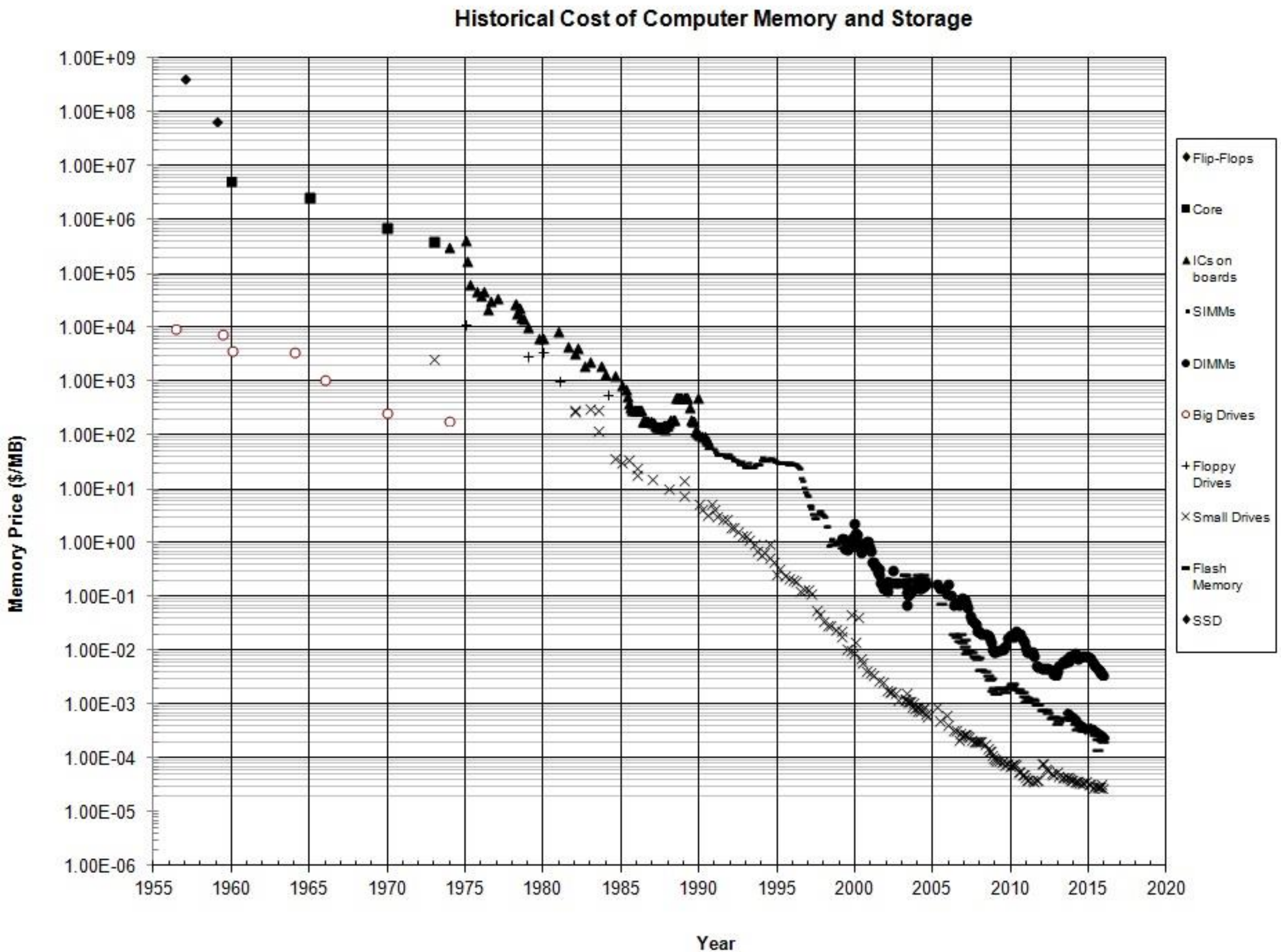
Outline

- Challenges for Big Data Systems
- Why Hardware Matters?
- Open Challenges
- Summary

Why HW Matters: Processors

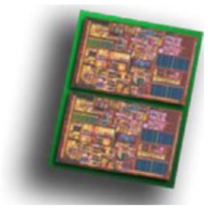


Why HW Matters: Memory and Storage



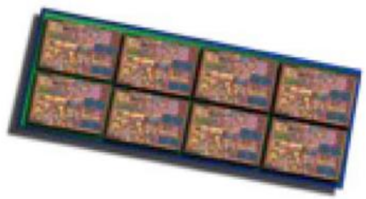
Emerging HPC Hardware: Parallelism and Heterogeneity

- Towards many cores



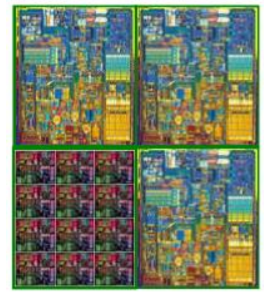
Dual cores

>>



Multi-core array

>>



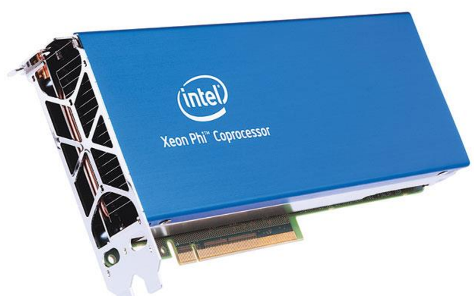
Scalar plus many cores

>>

Many-core array

- From CPU to accelerators (co-processors)

GPU



Xeon Phi

FPGA

Figures are adopted from Intel, NVIDIA and Altera.

Emerging HPC Hardware: Parallelism and Heterogeneity (Cont')

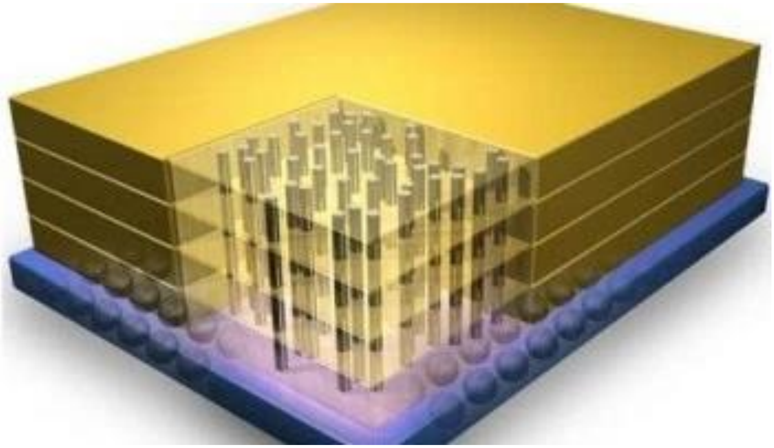
- Towards tightly coupled heterogeneous systems

AMD APU

Intel-Altera Heterogeneous Accelerators

…

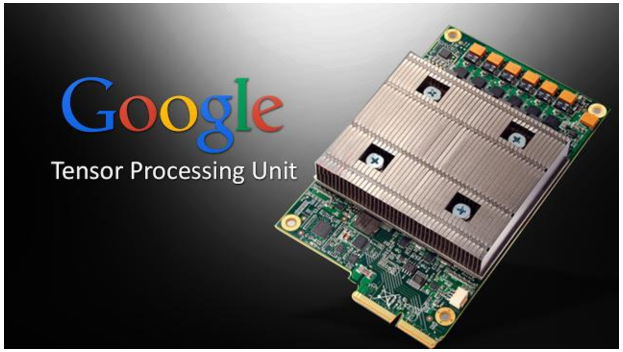
- High bandwidth memory (HBM)

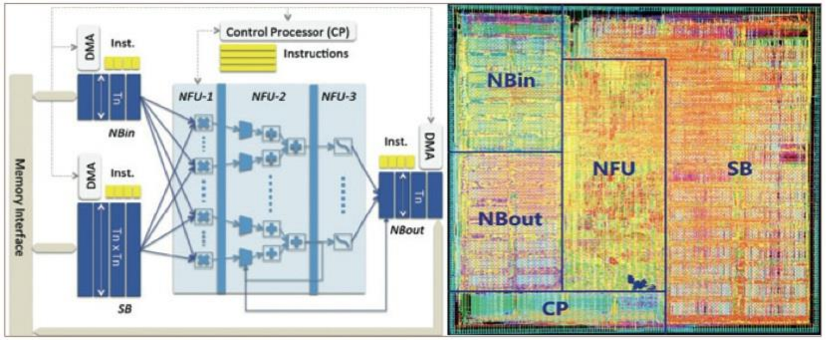


Figures are adopted from AMD, Intel and Altera.

What About Other Options?

- CGRA
- ASIC
  - AI chips
- ASIP





Future Hardware

- Processors
  - 1, 000 cores
  - Heterogeneous/specialized hardware (FPGA/ASIC)
- **Disk is dead, NVRAM is disk, DRAM is cache, Locality is still the King.**
  - NVRAM/3D stacking
  - "Tape is Dead, Disk is Tape, Flash is Disk, RAM Locality is King" by Jim Gray
- Cluster as a personal supercomputer
  - Fast and cheaper interconnects
  (e.g., Infiniband)

When Big Data Meets Emerging Hardware

Emerging hardware architectures

+

(Emerging) data-intensive applications

→

System issues:

- Performance
- Programmability
- Energy consumption
- User interfaces
- …

Our solution:

Hardware-software codesign

Outline

- Challenges for Big Data Systems
- Why Hardware Matters?
- Open Challenges
- Summary

Open Challenges

- Challenge #1: HW is simply one side of the coin; SW is the other side.
- Challenge #2: The leap from prototype to production.
- Challenge #3: Millions of lines of legacy code.
- Challenge #4: Generalization vs. Specialization
- Challenge #5: Many (supposedly great) HW architectures did not survive.

Challenge #1: HW is simply one side of the coin; SW is the other side.

- Terasort sorting 100TB data [RasmussenNSDI2011]
  - Platform 1: vanilla Hadoop, 2100 nodes, 12 cores per node, 64 Gb per node and 134 Tb memory.

  4300 seconds

  - Platform 2: Tritonsort (optimized for HW), 52 nodes, 8 cores per node, 24 Gb, 416 cores and 1.2 Tb memory.

  8300 seconds

  - Platform 1 uses 40x hardware only to achieve 2x speedup over Platform 2.
- SW needs to be hardware conscious.

Our Experiences in GPGPU-based Data Management Systems

CUDA was released in Feb. 2007

GPUQP (GDB) accepted in SIGMOD 2008 ("best papers")

Mars (GPU-based MapReduce) accepted in PACT 2008 (2nd top cited paper in PACT)*

Mars has been extended to AMD GPU and Hadoop (TPDS10)

GDB supports compressed column-based processing (VLDB10)

Transaction executions on GDB (VLDB11)

Medusa: GPU-based graph processing (TPDS13/14, VLDB13 best demo, CloudCom13)

OmniDB: relational database on coupled CPU/GPU architectures (VLDB'13/14/15, SIGMOD 16, VLDB'13 demo,…)

- http://arnetminer.org/conference/pact-124.html
- Thanks to my advisor, colleagues, and students.

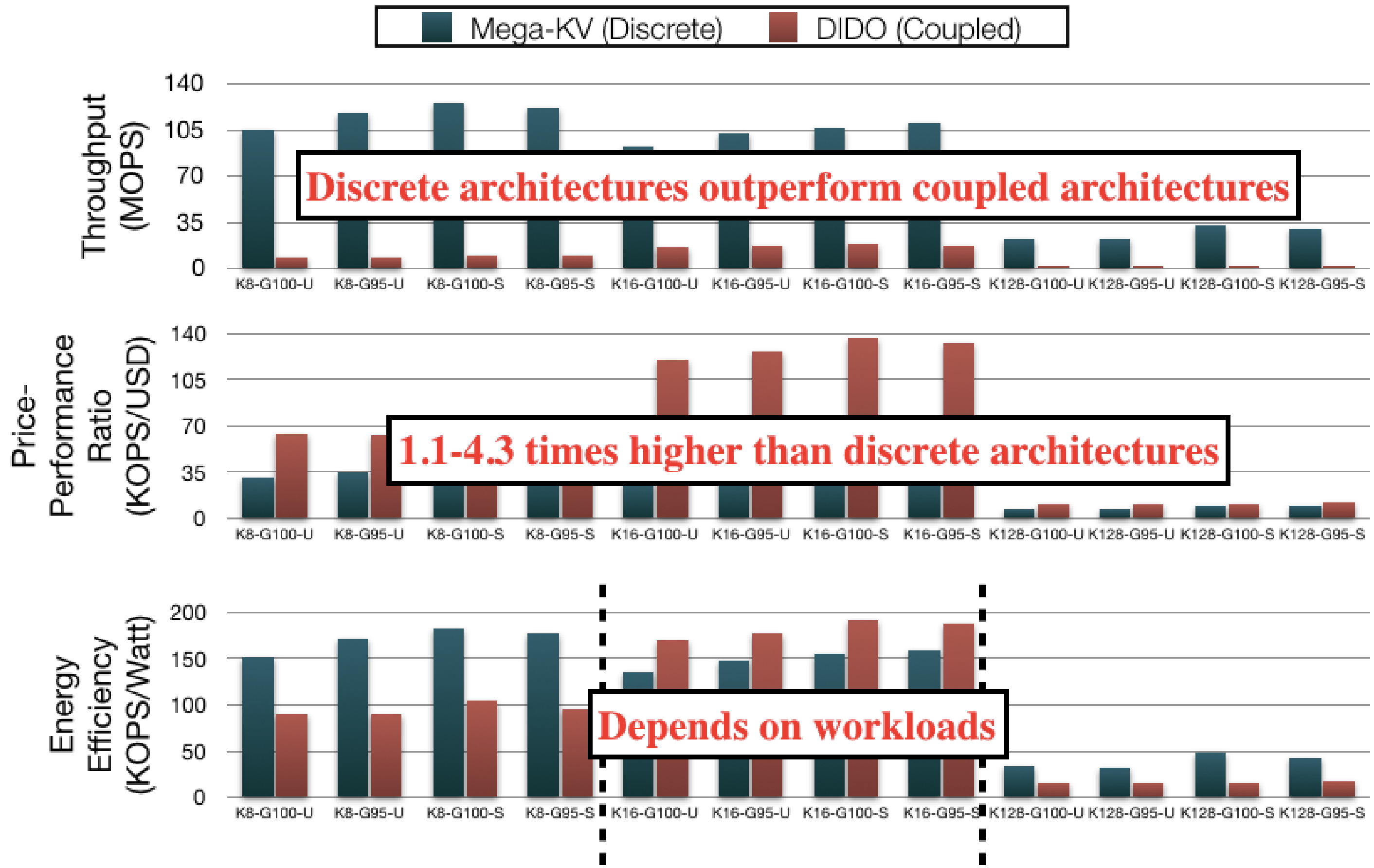
Challenge #2: The leap from prototype to production

- Many research papers demonstrate that the speedup of GPU/FPGA can reach 10-100X.
- Still, GPU/FPGA has rather limited adoptions in production environments (although increasingly more).
- Why?
  - Hardware: power supply budget, space, costs,…
  - Software: software maintenance, reliability, manpower expertise, sharing, virtualization ….
  - Workload: deep learning, database …
- Besides algorithmic innovation, various system aspects have to be addressed.

Experiences in Addressing Practical Issues of GPU Computing

- PCI-e bus via data compressions
  - Database compressions on column-based databases [VLDB2010]
- Concurrent kernel executions
  - Resource complementary kernel co-scheduling [TPDS 2014]
- GPU virtualizations
  - Gaming virtualization [USENIX ATC 2016]
- Minimizing data flow overhead among processors
  - Pipeline execution [SIGMOD 2016]

Performance, or Perf per $, or Perf per Joule?

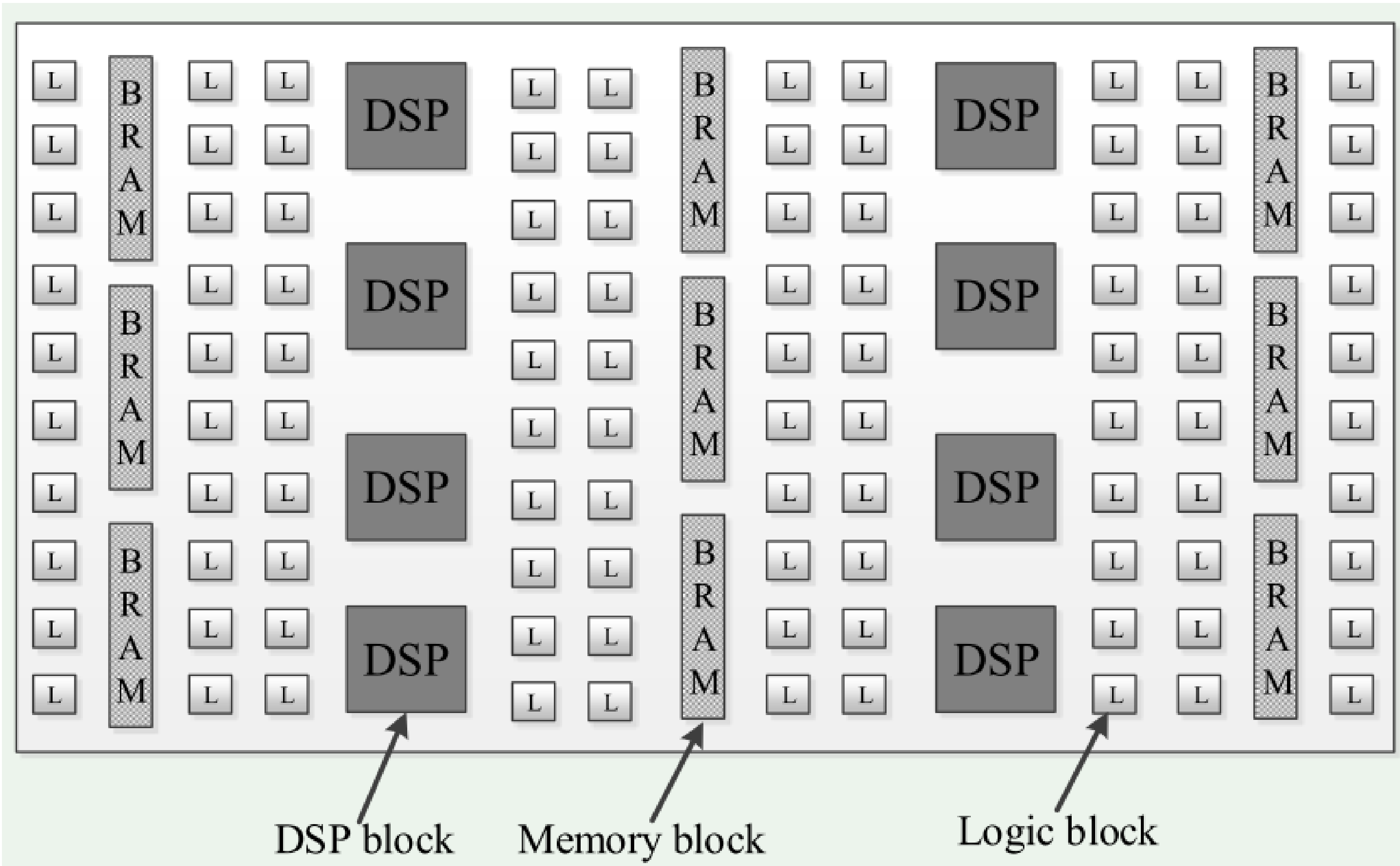


Kai Zhang^, Jiayu Hu, Bingsheng He, Bei Hua. DIDO: Dynamic Pipelines for In-Memory Key-Value Stores on Coupled CPU-GPU Architectures. ICDE 2017.

Challenge #3: Millions of lines of legacy code

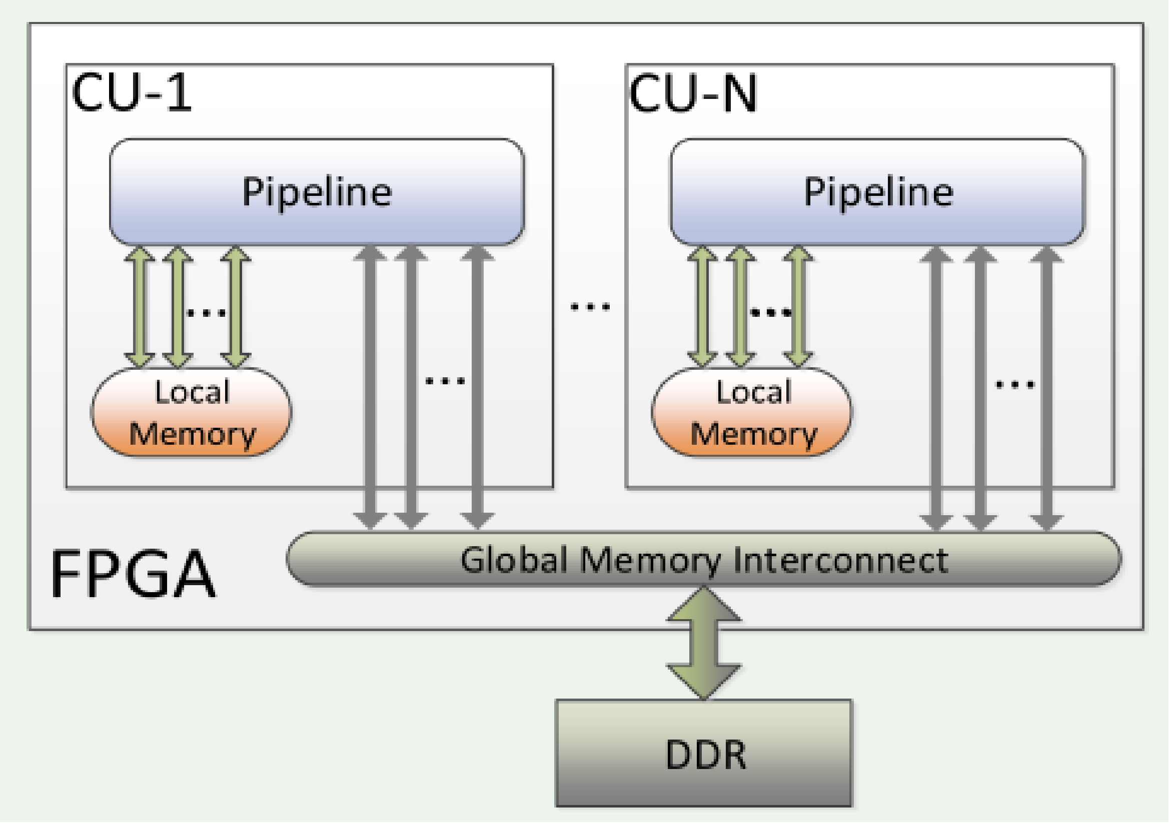
- Our legacy software systems are monsters
  - National labs have MPI programs of millions of code lines.
  - Google's Internet services spans some 2 billion lines of code.
  - Microsoft's Windows operating system has around 50 million lines.
  - Other younger ones: Hadoop 2millions, Spark 0.9 million, MySQL 2.7 millions…
- The reality is, "write once, reuse till many many times".
- The research on automatic parallel optimizer is dead.
- (Semi-)Automated tools are needed to resolve the pain points.

“Architectural Evolution” of FPGA (Field Programmable Gate Arrays)



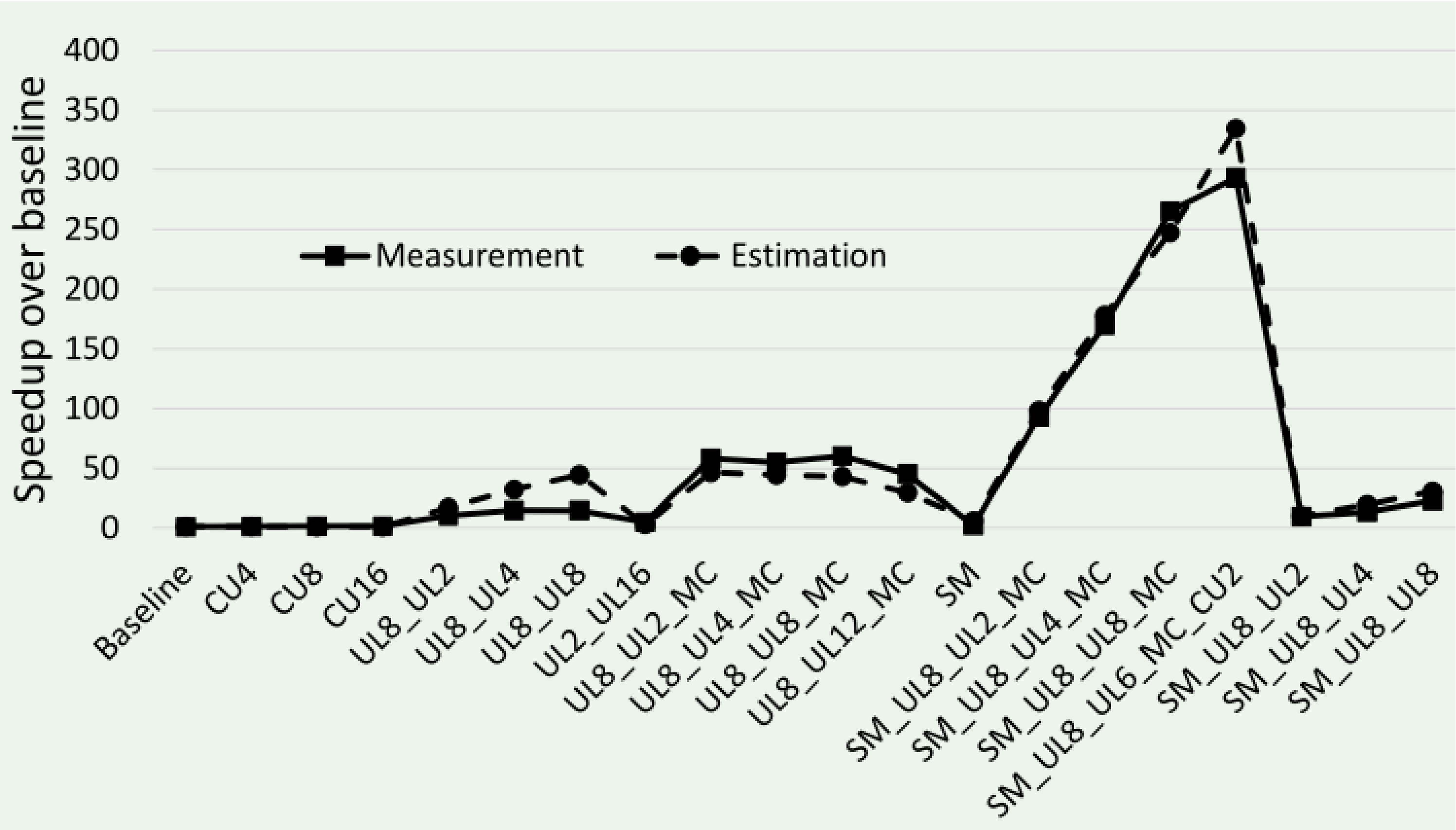
- Hardware centric
- Users need to program with low-level hardware description languages. ☹

"Architectural Evolution" of FPGAs: From OpenCL's Perspective



- Software centric → FPGA is viewed as a parallel architecture.
- Users can program with OpenCL. ☺

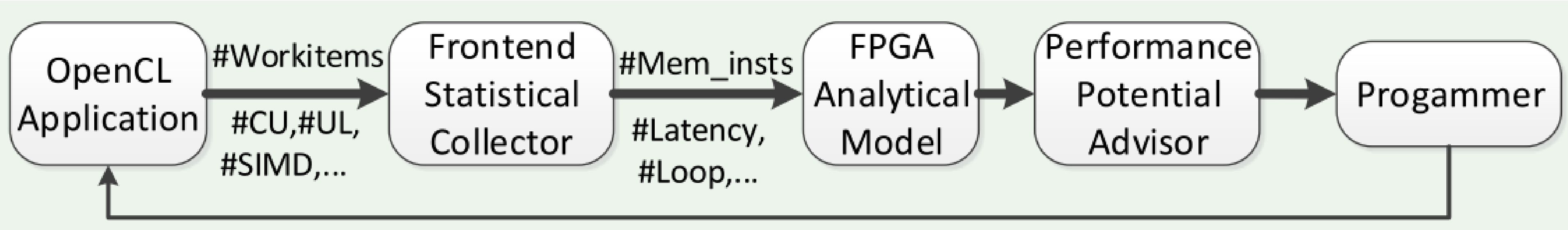
Our Experiences on FPGA



- The example: K-Means
- Applying different combinations of optimization leads to huge performance differences → Tools for optimizations are needed.

Our Solution: Static and Dynamic Program Analysis

- We propose a performance analysis framework to assist programmers to optimize the OpenCL program on FPGA
  - Static statistical collection on the corresponding LLVM IR code.
  - Dynamic profiling of the OpenCL application execution.
  - FPGA analytical model predicts the performance of OpenCL application.
  - The performance advisor digests the model information and provides the four potential metrics to understand the performance bottleneck.



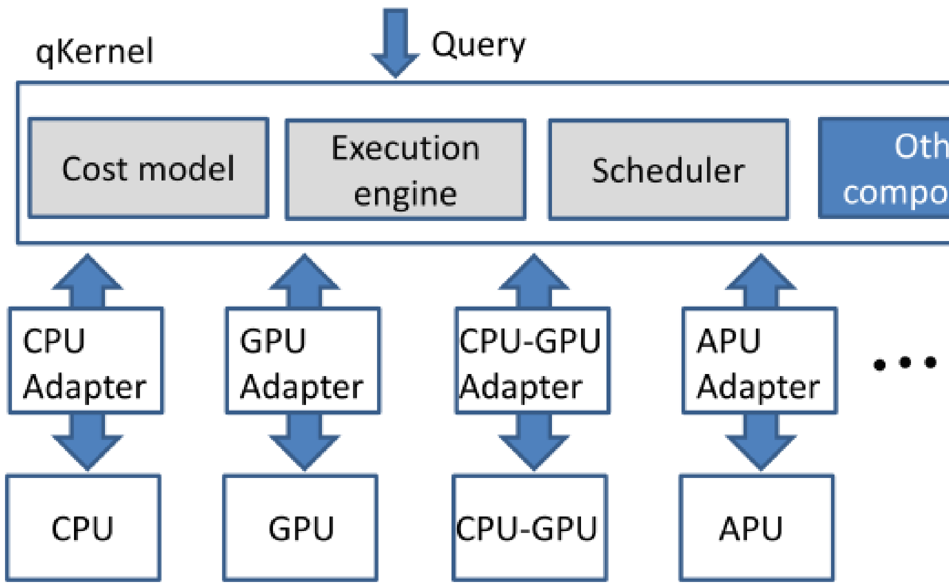
Details in "Zeke Wang^, Bingsheng He, Wei Zhang, Shunning Jiang. A Performance Analysis Framework for Optimizing OpenCL Applications on FPGAs. HPCA 2016"

Challenge #4: Generalization vs. Specialization

- Specialized hardware
  - “SQL in Silicon” (in Oracle SPARC M7 processor)
  - Google TPU for deep learning
- Specialized software
  - System call overhead for memcached
  - Layers of abstractions in OS (e.g., for NVRAM)
- A compromise is possible (but difficult to find the optimal cut between generalization and specialization).

SW Portability vs. Specialization

- OmniDB: General Engine Design + Adapter to Specific Architecture



Shuhao Zhang*, Jiong He*, Bingsheng He, Mian Lu. OmniDB: Towards Portable and Efficient Query Processing on Parallel CPU/GPU Architectures. International Conference on Very Large Data Bases (VLDB) 2013. (also published in Proceedings of the VLDB Endowment, Volume 6 Issue 10, August 2013, pages = {1—4}, system demonstration).

Challenge #5: Many (supposedly great) HW architectures did not survive.

- Database machines
  - Even top database researchers have paid tremendous efforts into this “vain” project.
- Cell processor (Playstation 3 processor)
  - Hard to program the master-slave architecture
- Intel Itanium processor
  - VLIW-> Compiler fails to exploit the required parallelism
- How to predict?

Outline

- Challenges for Big Data Systems
- Why Hardware Matters?
- Open Challenges
- Summary

Summary

- Data management systems on emerging hardware continue to be a challenging and exciting research area.
- Our experiences demonstrate the system insights as well as open challenges of building big data systems on future architectures.
- Hardware and software co-design might be the key for the success of this battle.

Thank you!

More about Xtra Computing Group:
http://www.comp.nus.edu.sg/~hebs/