Big Data Systems on Future Hardware

Bingsheng He

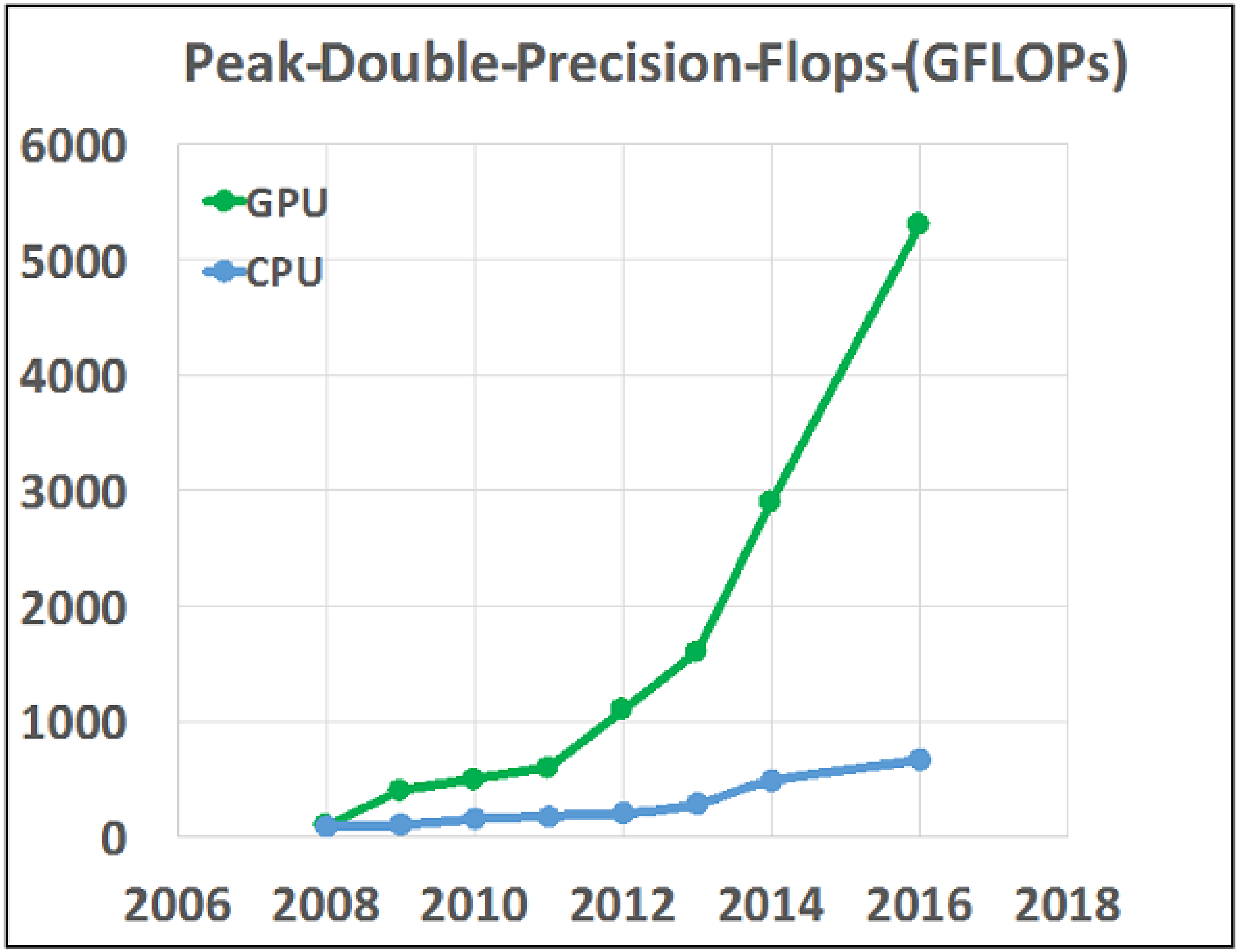
NUS Computing

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

Outline

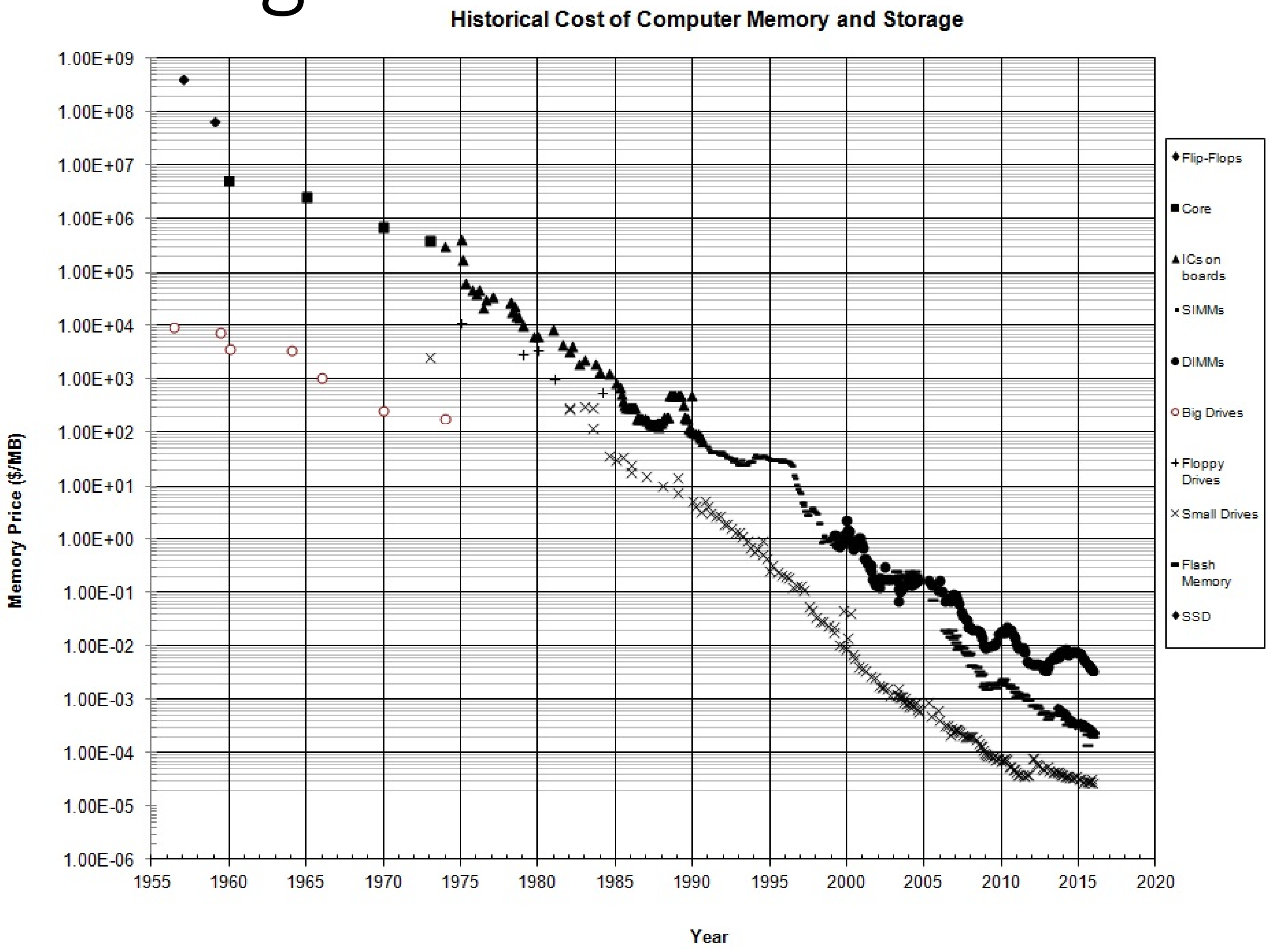
- Why Hardware Matters?
- Challenges & Experiences
- On-going Projects
- Summary

Why HW Matters: Processors



Source: https://www.hpcwire.com/2016/08/23/2016-important-year-hpc-two-decades/

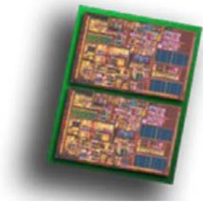
Why HW Matters: Memory and Storage



Source: **hblok.net/storage**

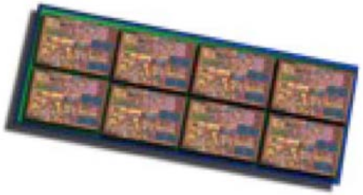
Emerging HPC Hardware: Parallelism and Heterogeneity

- Towards many cores



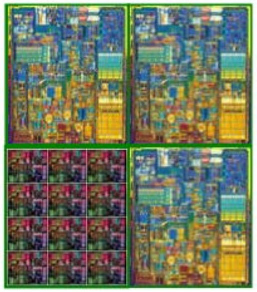
Dual cores

>>



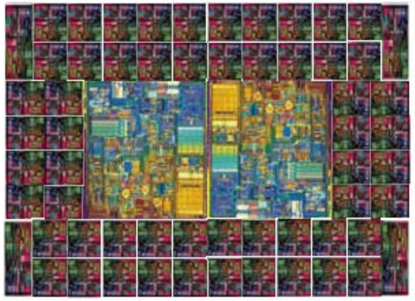
Multi-core array

>>



Scalar plus many cores

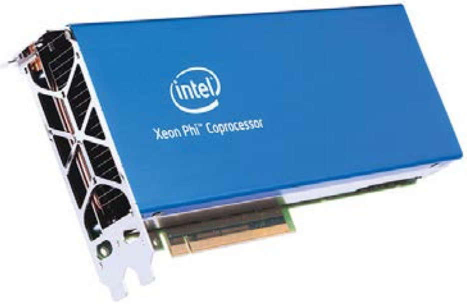
>>



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')

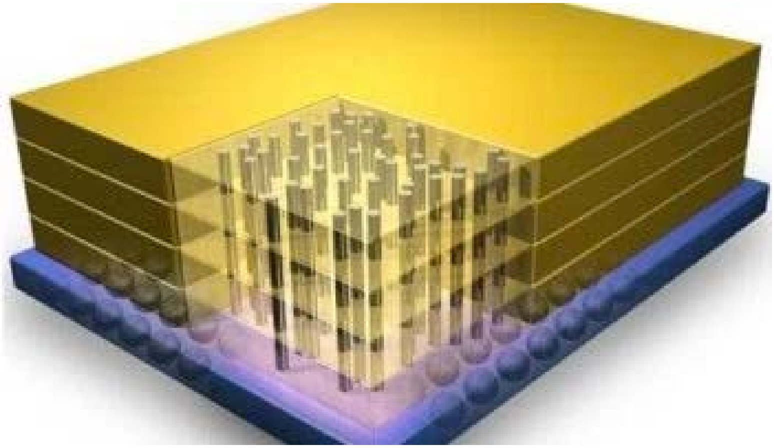
- Towards tightly coupled heterogeneous systems

AMD APU

Intel-Altera Heterogeneous Accelerators

…

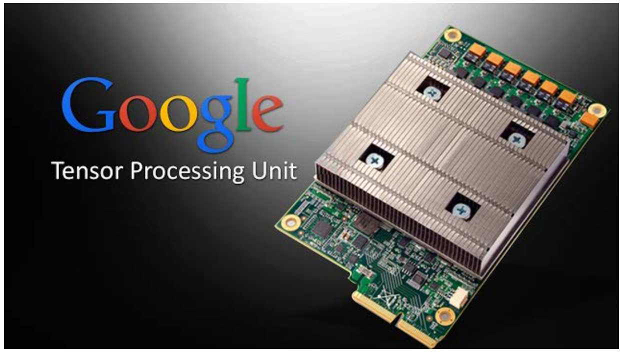
- High bandwidth memory (HBM)

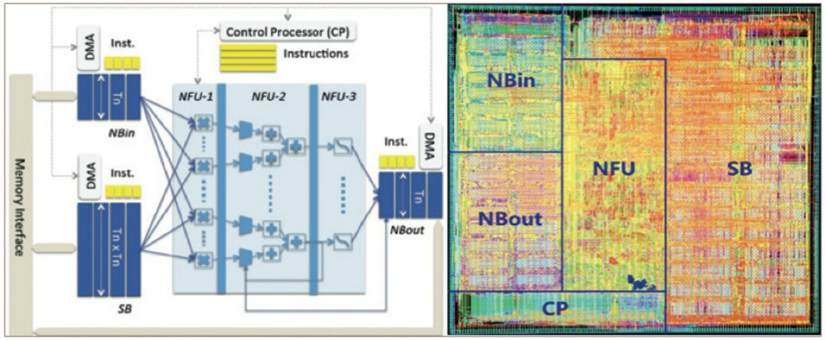


Figures are adopted from AMD, Intel and Altera.

Emerging HPC Hardware: Generalization vs. Specialization

- AI chips
- Graph accelerators
- …



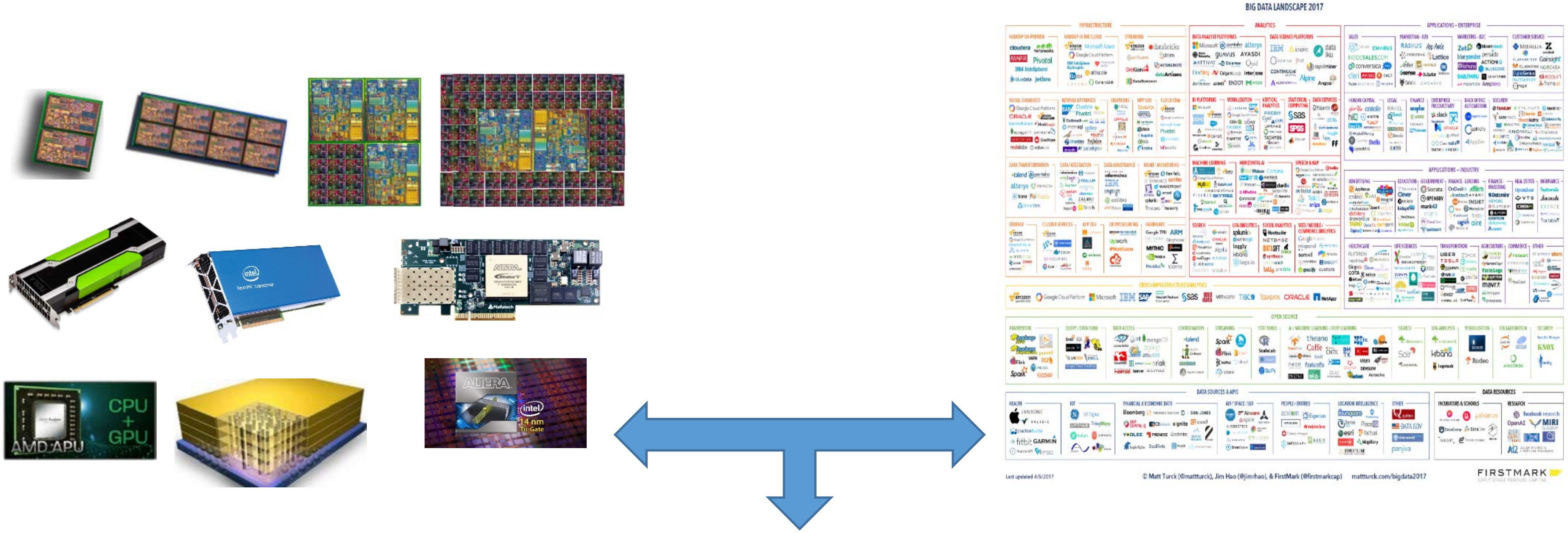


Future Hardware

- 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 (Con’t)



- Architecture-aware optimizations: how to make our software systems fully optimize for the hardware?
- New hardware: how can new applications drive the design of new hardware?
- New systems/applications: can superb hardware power enable new systems/applications?

Outline

- Why Hardware Matters?
- Challenges & Experiences
- On-going Projects
- Summary

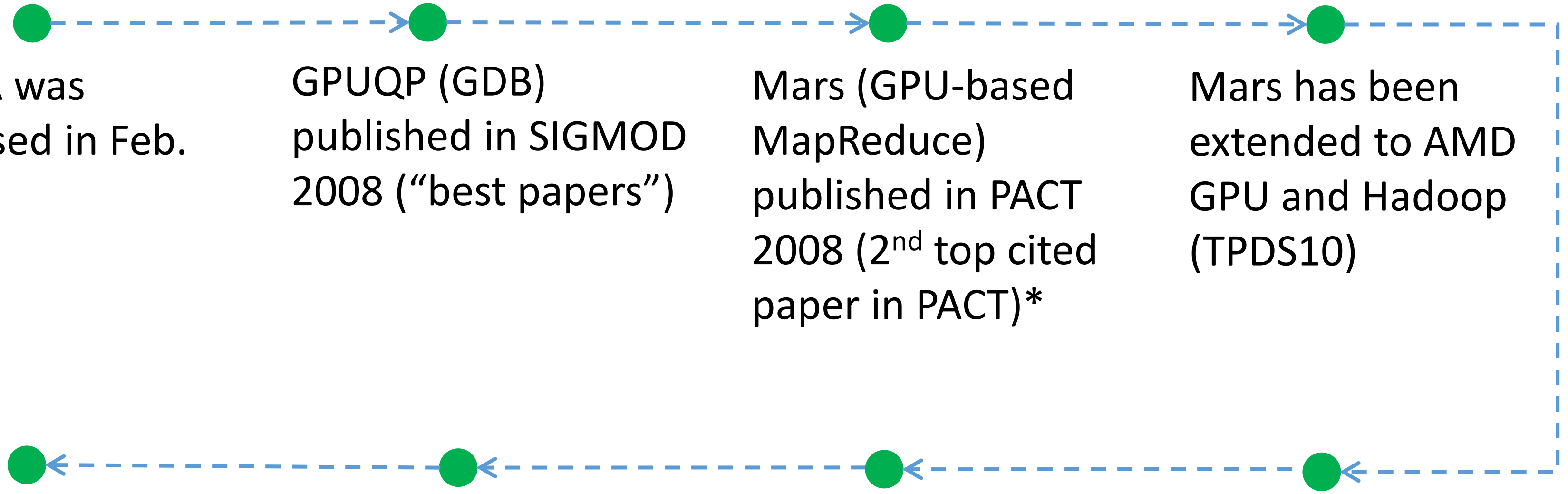
Challenges & Experiences

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

- Parallel algorithmic design
  - Revisit existing parallel algorithm and develop new algorithm.
  - Running a large task in parallel vs. running many small tasks in parallel
- Hardware features
  - GPU: shared memory, cache, constant memory etc.
  - FPGA: local memory, pipeline, pipe etc.
- Architecture-aware software system design can make a big difference.

Our Experiences in GPGPU-based Data Management Systems



CUDA was released in Feb. 2007

GPUQP (GDB) published in SIGMOD 2008 (“best papers”)

Mars (GPU-based MapReduce) published 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,…)

Example speedup over CPU:

- OLAP: 2-7X (vs. one CPU)
- OLTP: 4-10X (vs. one CPU)
- Graph: 5.5X (vs. two CPUs)

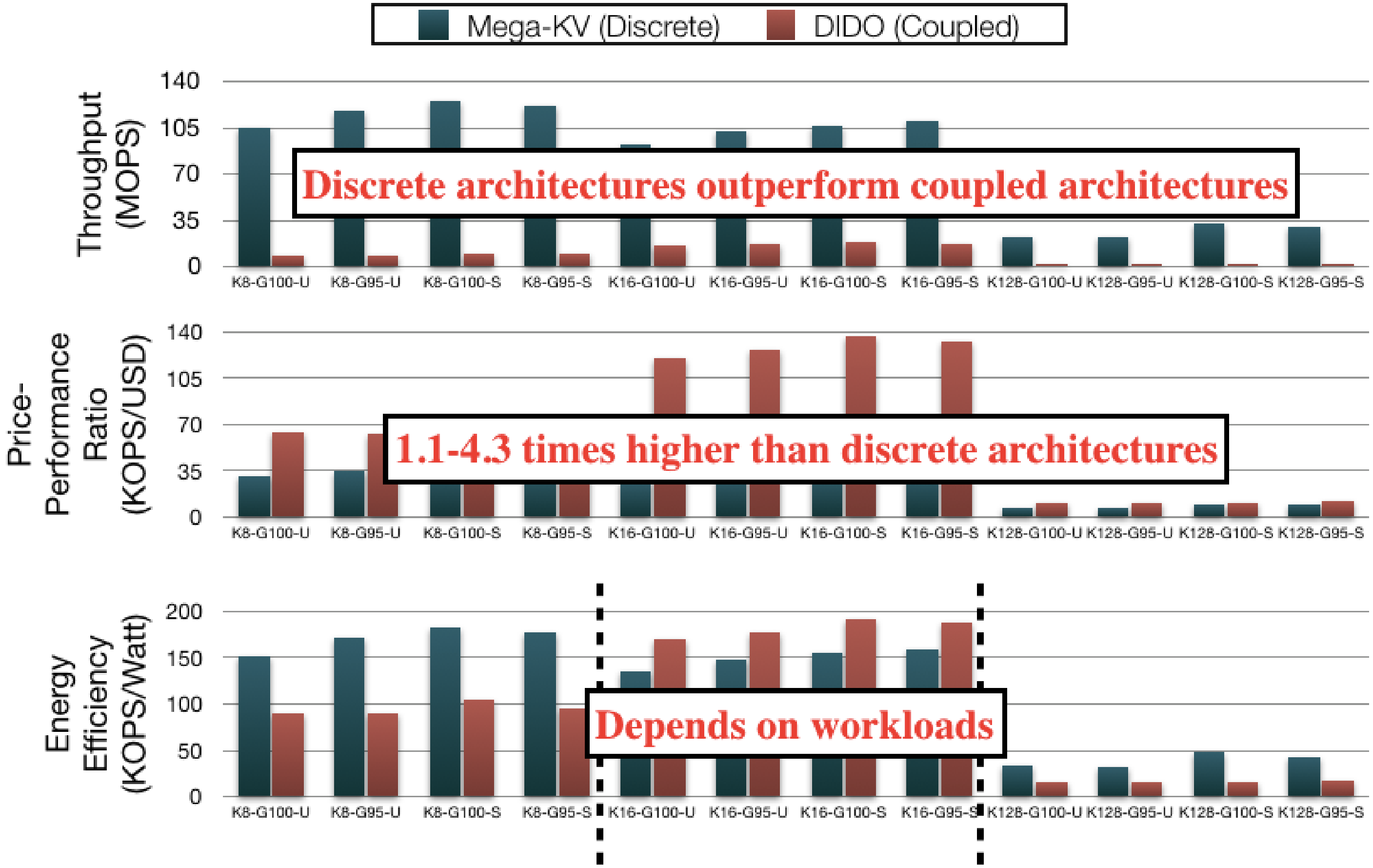
Challenge #2: The leap from prototype to production

- 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?



Discrete architectures outperform coupled architectures

1.1-4.3 times higher than discrete architectures

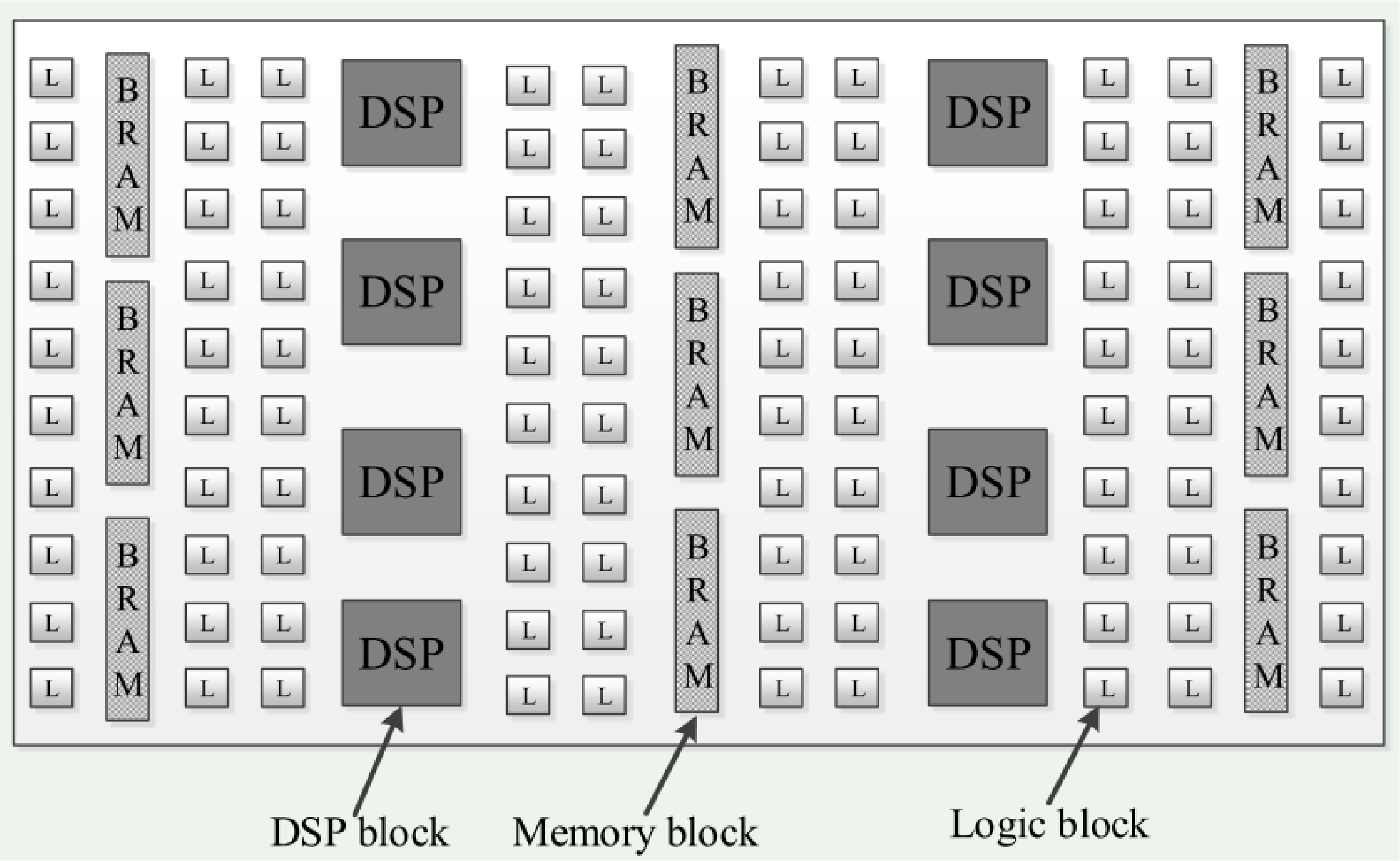
Depends on workloads

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

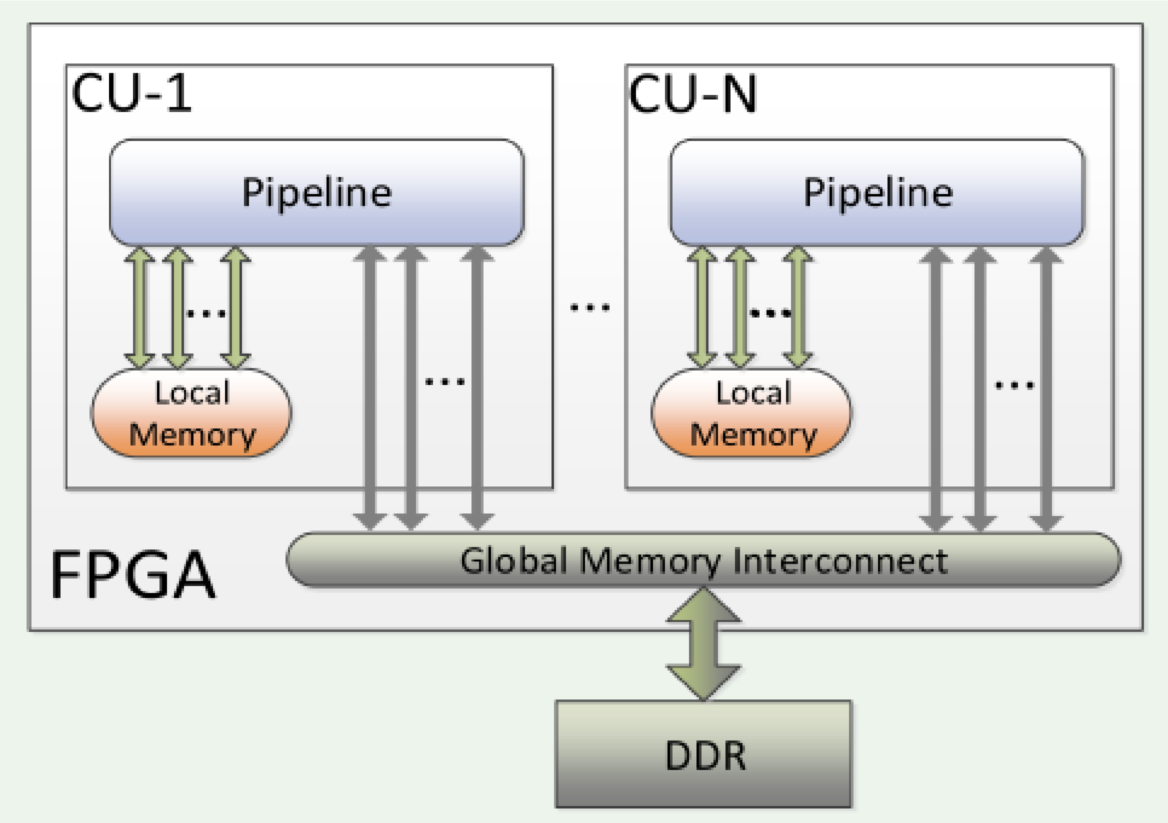
- Our legacy software systems are monsters
  - National labs have MPI programs of millions of code lines.
  - Google's Internet services spans some 2 billion lines of code.
  - Microsoft's Windows operating system has around 50 million lines.
  - Other younger ones: Hadoop 2millions, Spark 0.9 million, 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)



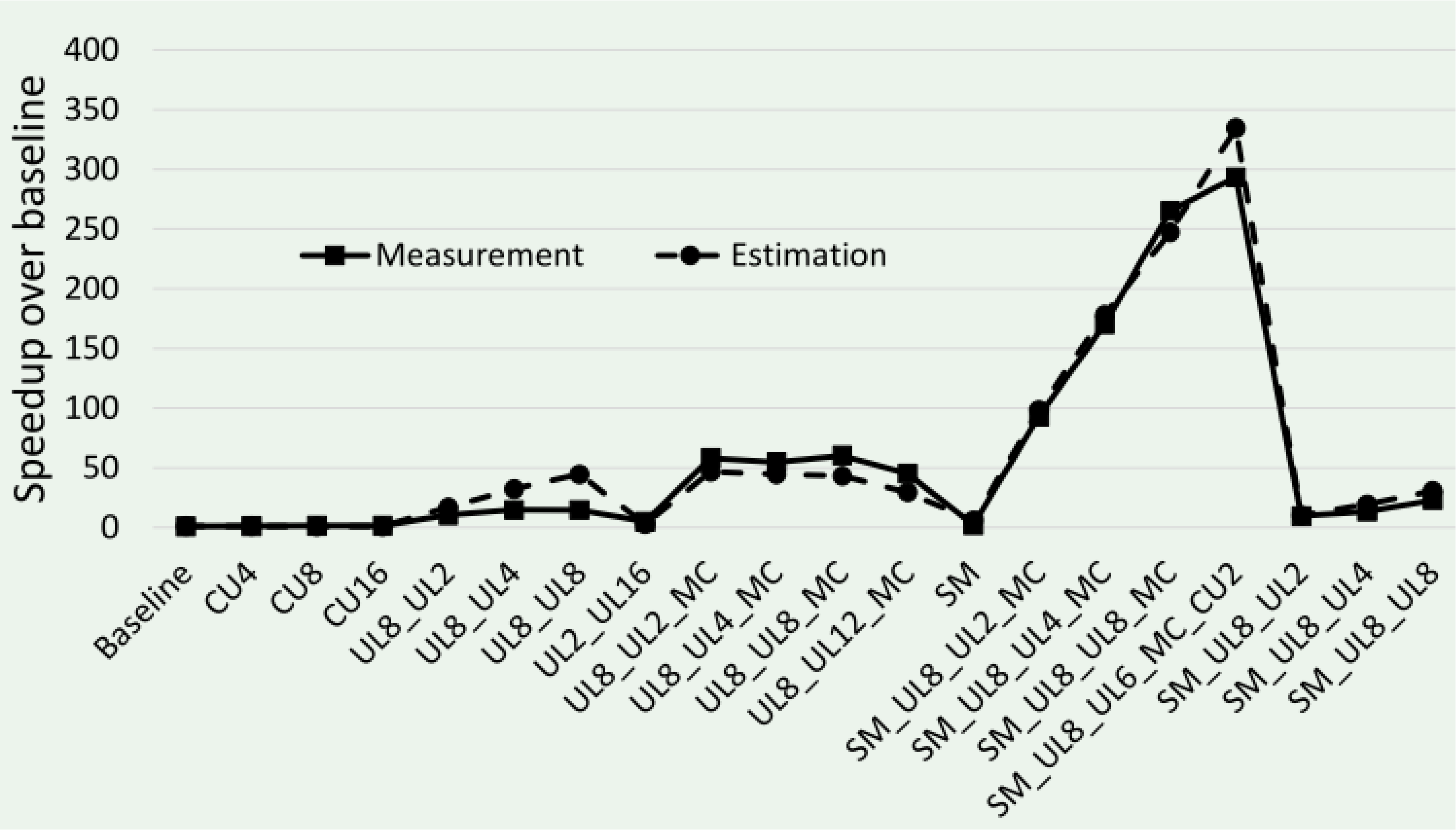
- Hardware centric
- Users need to program with low-level hardware description languages. ☹

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



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

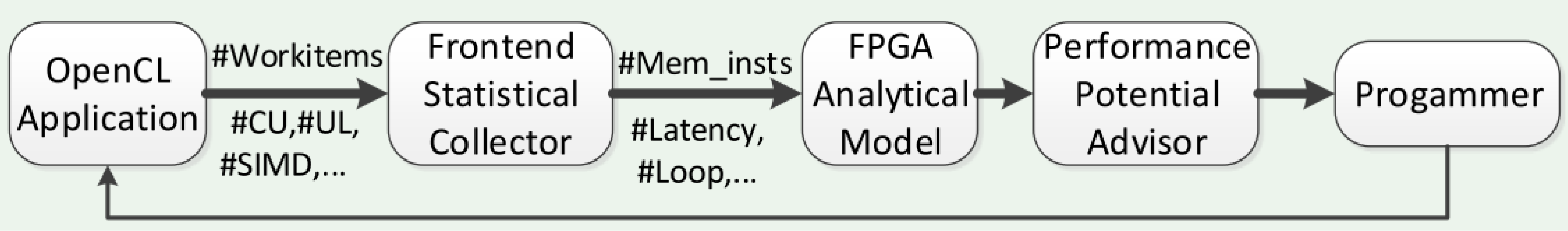
Our Experiences on FPGA



- 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.



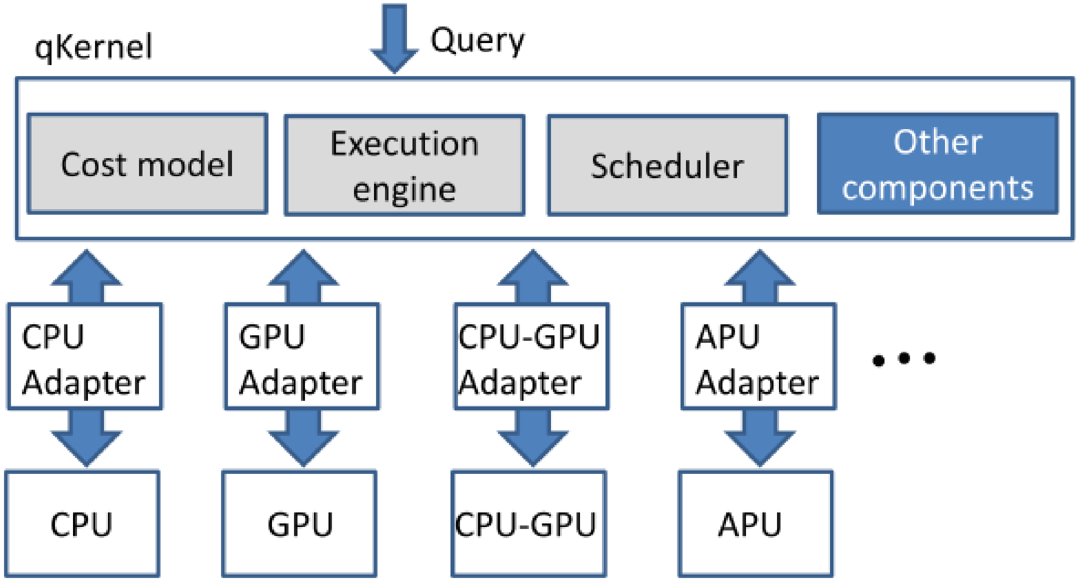
Details in “Zeke Wang^, Bingsheng He, Wei Zhang, Shunning Jiang. A Performance Analysis Framework for Optimizing OpenCL Applications on FPGAs. HPCA 2016”

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).

Outline

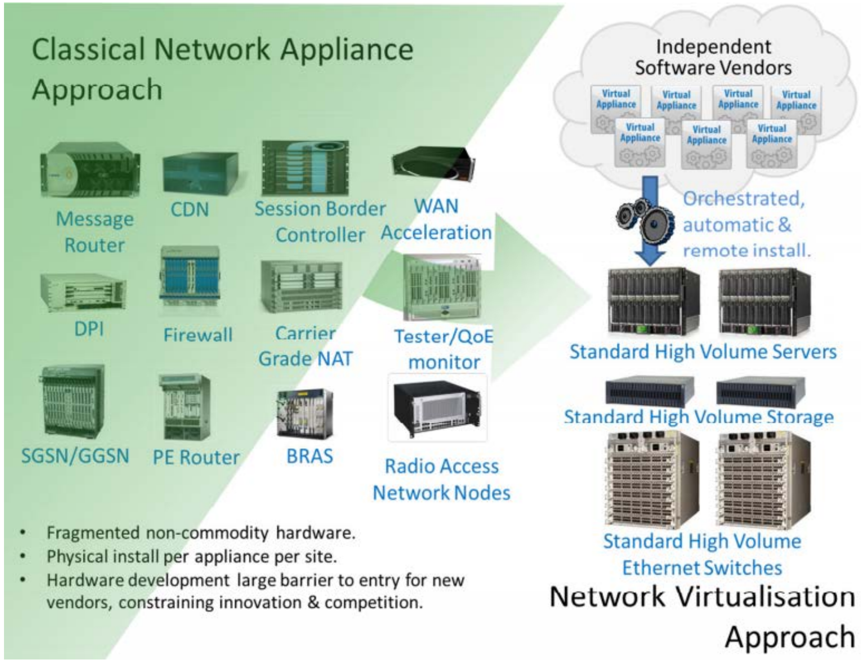
- Why Hardware Matters?
- Challenges & Experiences
- On-going Projects
- Summary

On-going Projects

- High performance graph databases
  - Goal: a graph database that can outperform Neo4j by 10+times.
- Network Function Virtualization (NFV)
  - Goal: a cost-effective solution to replace specialized network devices.

Network Function Virtualization

- Basic trend: supporting network functions shifting from specialized hardware to software.
- Huawei predicts, in 3-5 years, CPU + GPU will satisfy the computational requirement for telecom equipment.
- Challenges:
  - Massive network flows
  - Low latency
  - Flexible deployment

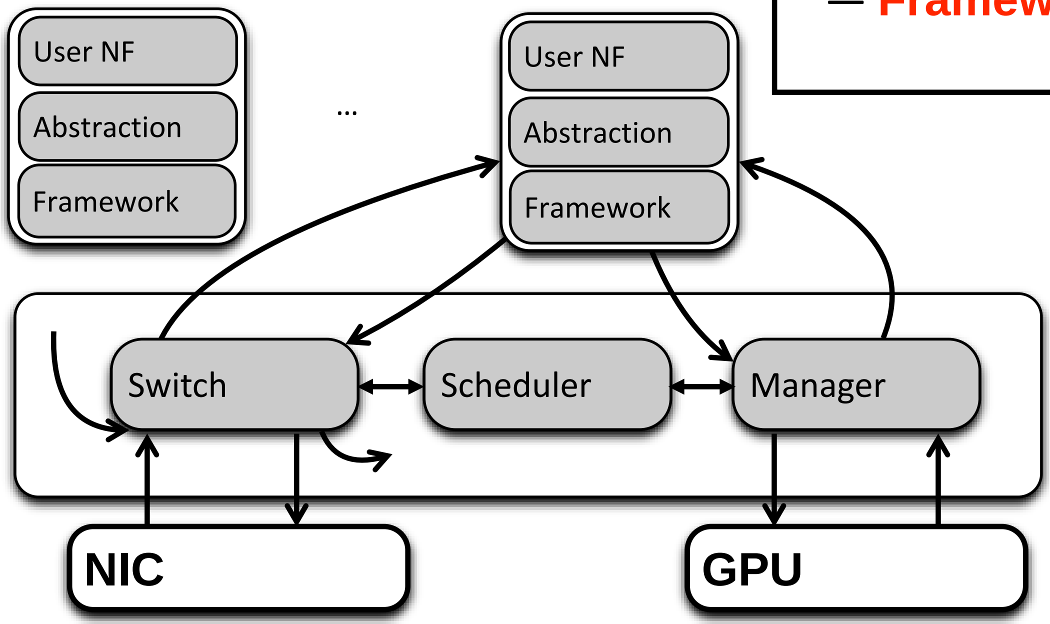

Classical Network Appliance Approach
Message Router
CDN
Session Border Controller
WAN Acceleration
DPI
Firewall
Carrier Grade NAT
Tester/QoE monitor
SGSN/GGSN
PE Router
BRAS
Radio Access Network Nodes
Independent Software Vendors
Virtual Appliance
Orchestrated, automatic & remote install.
Standard High Volume Servers
Standard High Volume Storage
Standard High Volume Ethernet Switches
Network Virtualisation Approach

- Fragmented non-commodity hardware.
- Physical install per appliance per site.
- Hardware development large barrier to entry for new vendors, constraining innovation & competition.

GPU-Accelerated NFVs?

- GPU advantage over specialized hardware
  - Comparable speed
  - Low cost
- However,
  - Low GPU Utilization (Each VM has no enough traffic to fully utilize GPUs)→ GPU virtualization is still an open research problem.
  - Unpredictable Latency
  - Lots of development effort

Poseidon: Use Remote API to Virtualize GPUs in NFV



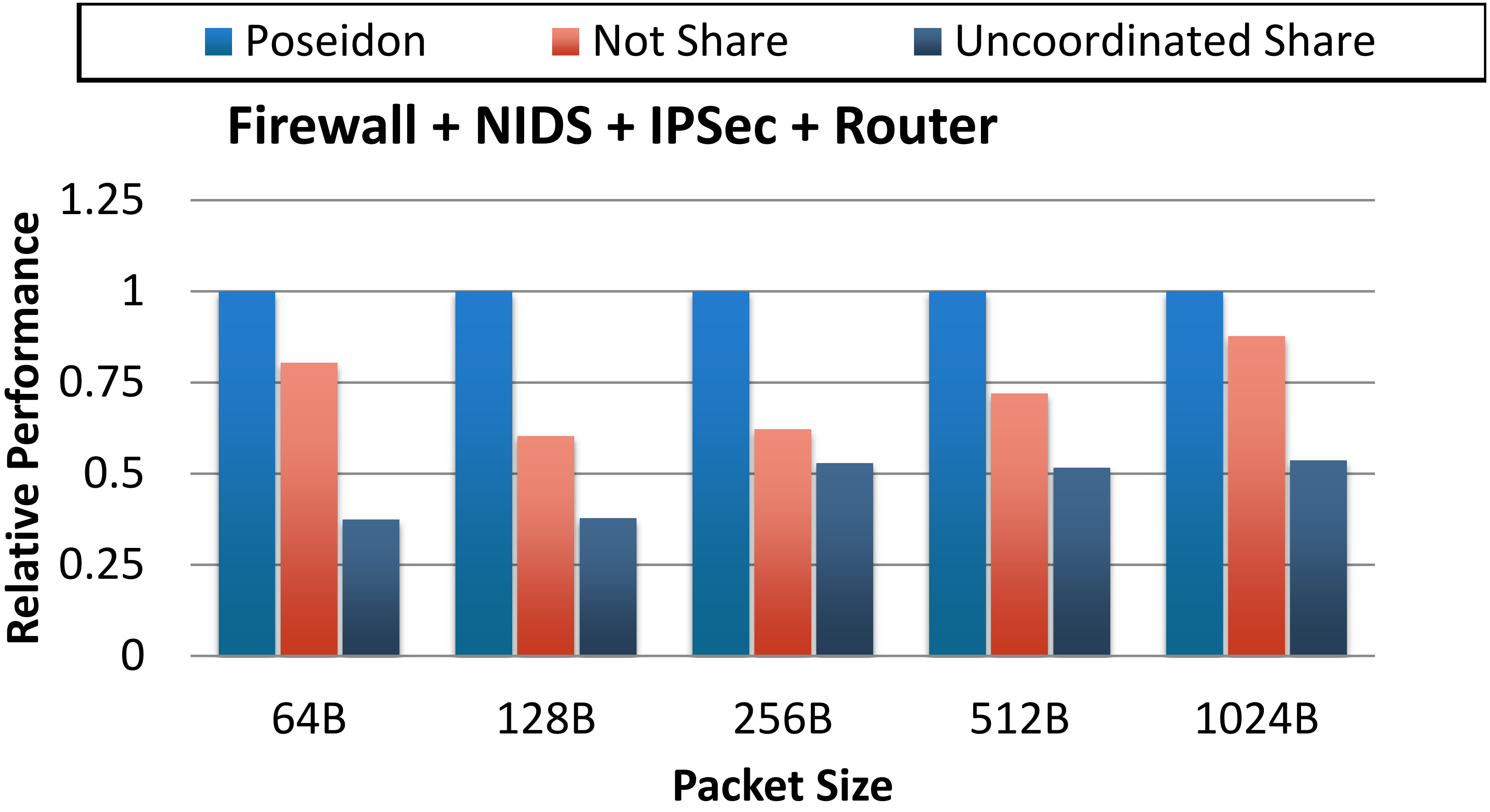
NF:

- **Abstraction**: Reduces development effort
- **Framework**: Handle network I/O, and communicate with the Core

Core:

- **Switch**: Handle network I/O
- **Scheduler**: Schedule GPU for execution
- **Manager**: Launch GPU kernels

Preliminary Results



Our system can significantly improve the performance of GPU sharing.

Outline

- Why Hardware Matters?
- Challenges & Experiences
- On-going Projects
- Summary

Summary

- (Database) 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/