Data Processing in Modern Hardware

Gustavo Alonso

Systems Group

Department of Computer Science

ETH Zurich, Switzerland

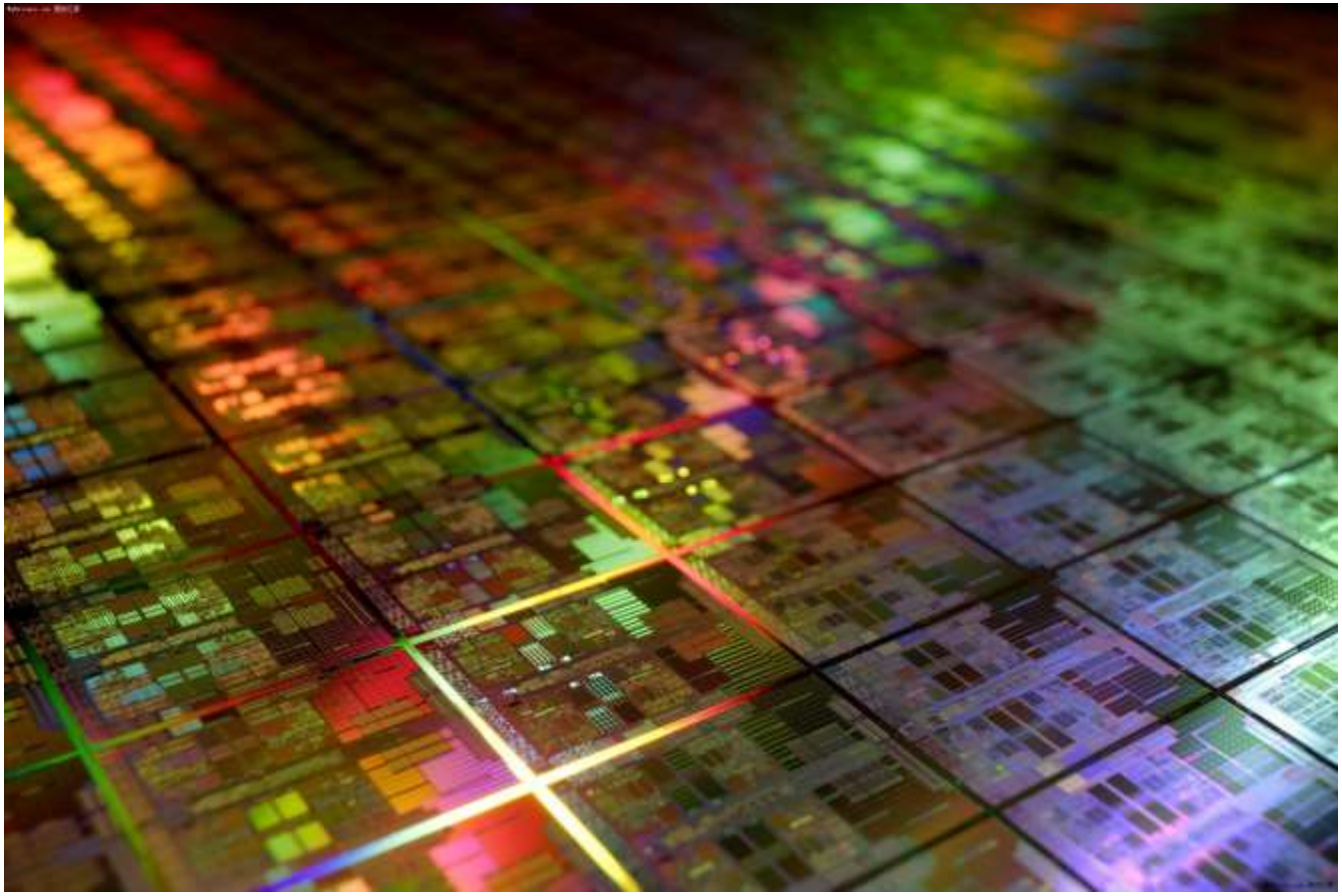
EDBT/ICDT 2016

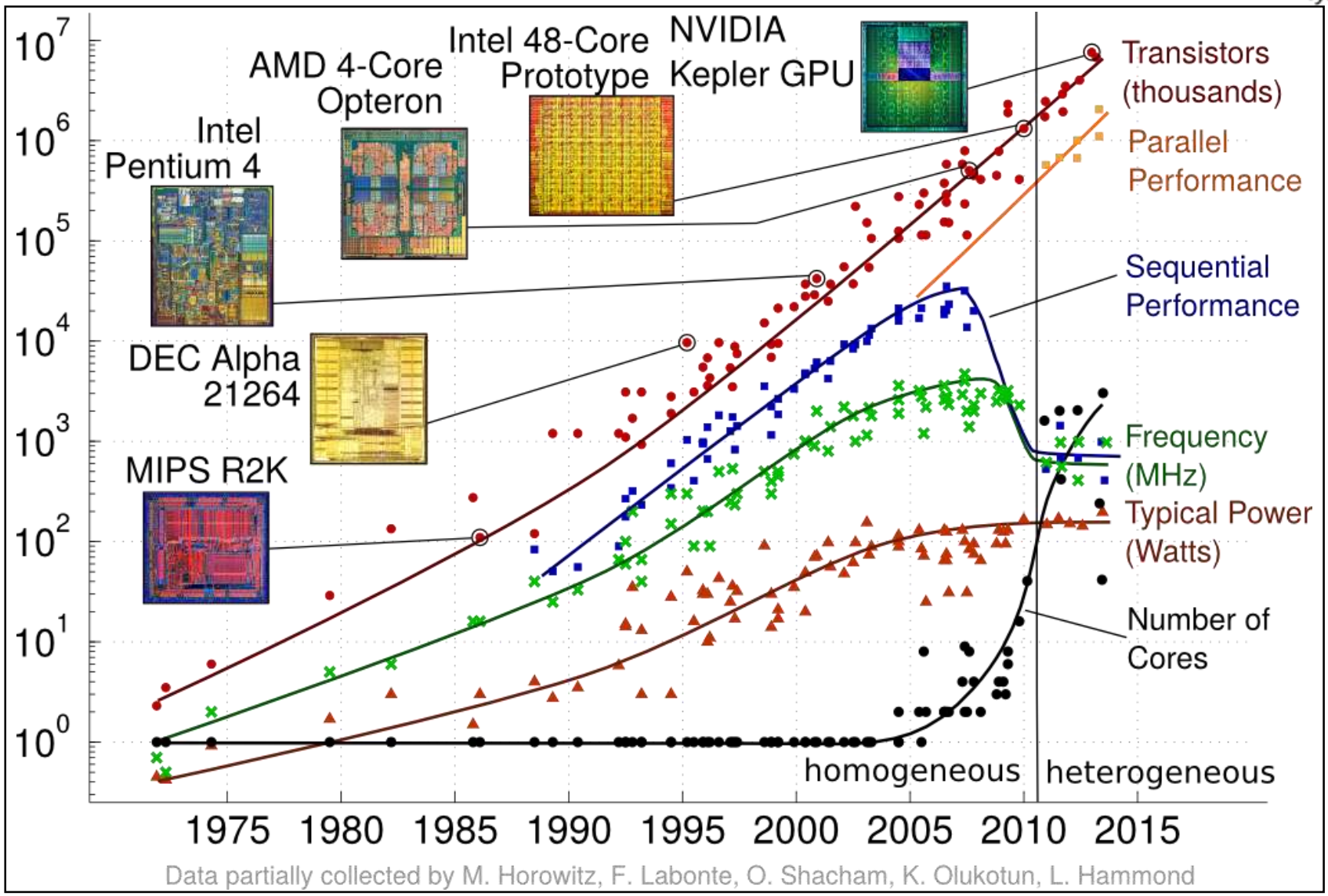
ETH Systems Group

www.systems.ethz.ch

Systems@ETH Zürich

Take home message





Slide courtesy of Torsten Hoefler (Systems Group, ETH Zürich)

In a nutshell …

Hardware going crazy

More transistors no longer means faster machines but more specialized

Big data is the killer app

Specialized hardware to support data processing

We must get into this game

Future prog t allow the
programmer a high-level,
target-indepen nize the target-
depender available
par ce.

Bill Dally, NVIDIA Chief Scientist

(Keynote at HiPEAC'15)

Hardware makes life difficult

Joins in main memory, multicore

Kim et al. PVLDB'09

- Hash joins faster than sort merge joins
- Will change when SIMD wide enough
- Showed tuning to multicore, SIMD

Blanas et al. SIGMOD'11

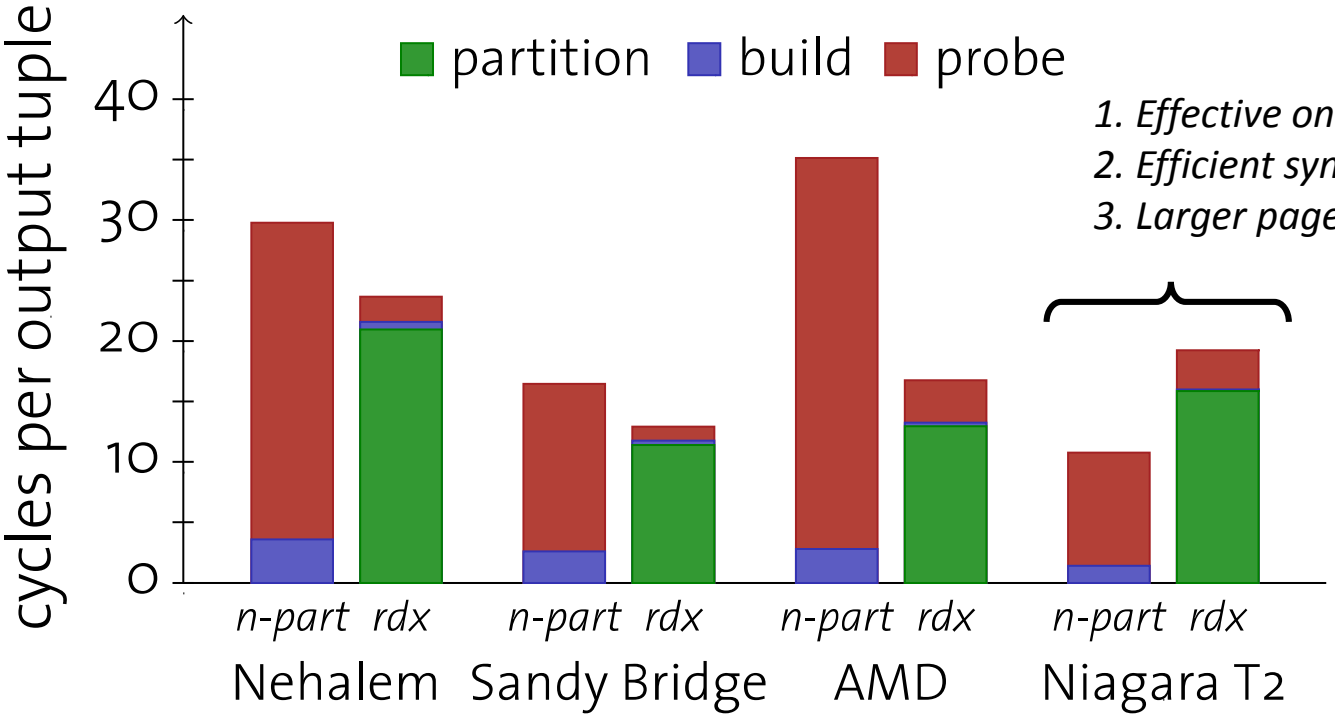
- No need for tuning a has join
- No need for careful partitioning
- Hardware hides complexity

Albutiu et al. PVLDB'12

- Sort merge join better already
- No need to use SIMD

Join with small build table

Workload A: 16M ⋈ 256M, 16-byte tuples
(256MiB ⋈ 4096MiB)

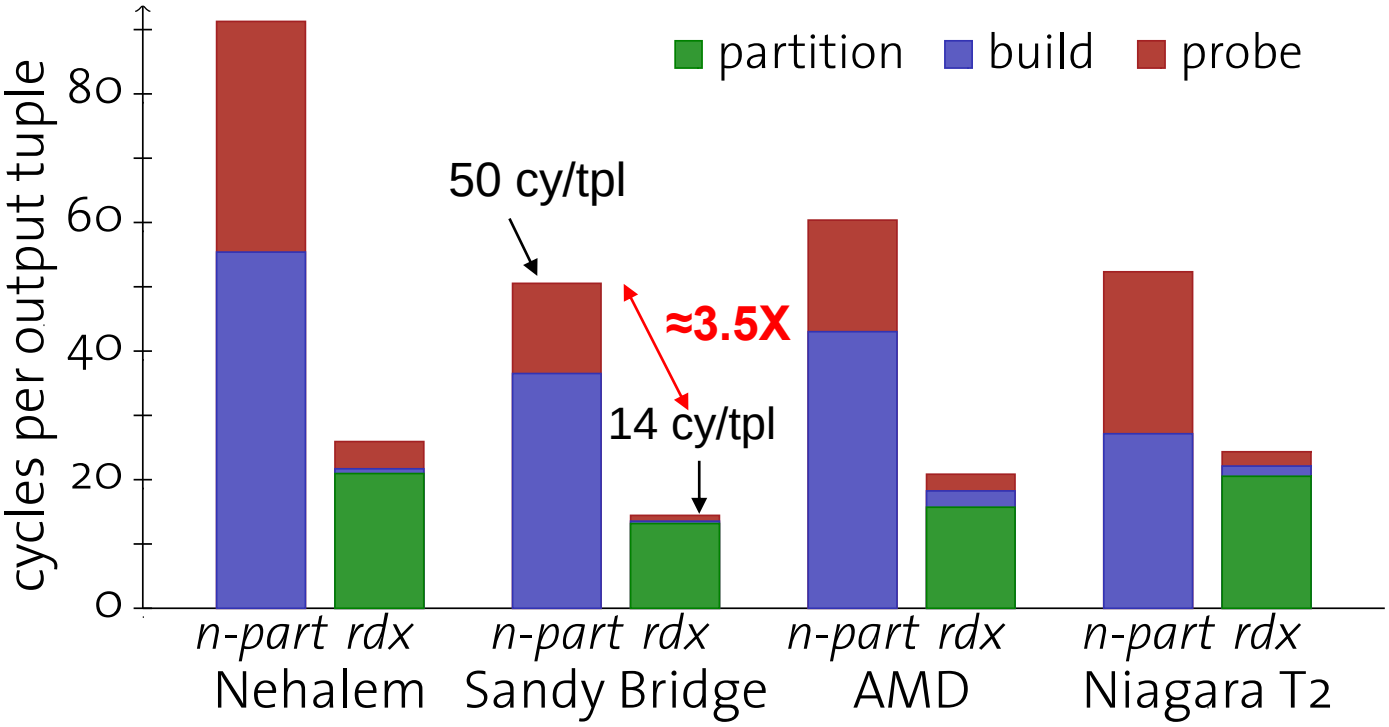


1. *Effective on-chip threading*
2. *Efficient sync. primitives (ldstub)*
3. *Larger page size*

Balkesen et al., ICDE13, PVLDB14

Join with large build table

Workload B: Equal-sized tables,
(977MiB ⋈ 977MiB, 8-byte tuples)



Balkesen et al., ICDE13, PVLDB14

What does it mean?

Hardware characteristics define the performance of the algorithm

Many additional results with faster algorithms: all through more tuning to the hardware

Underlying hardware affects performance in may ways

- Difficult for the optimizer
- Difficult for the database designer
- Quickly changing platforms

Algorithm performance affected by many factors

- Who knows about the hardware?
- Who should made the decision?

Spatial Joins in Main Memory: Implementation Matters!

Darius Šidlauskas
Aarhus University
dariuss@cs.au.dk

Christian S. Jensen
Aalborg University
csj@cs.aau.dk

"This study demonstrates that in main memory, where no time-consuming I/O can mask variations in implementation, implementation details are very important;

… … the implementations of the data structures and algorithms are more important for the performance than the data structures and algorithms themselves."

(Sidlauskas & Jensen, PVLDB'14,

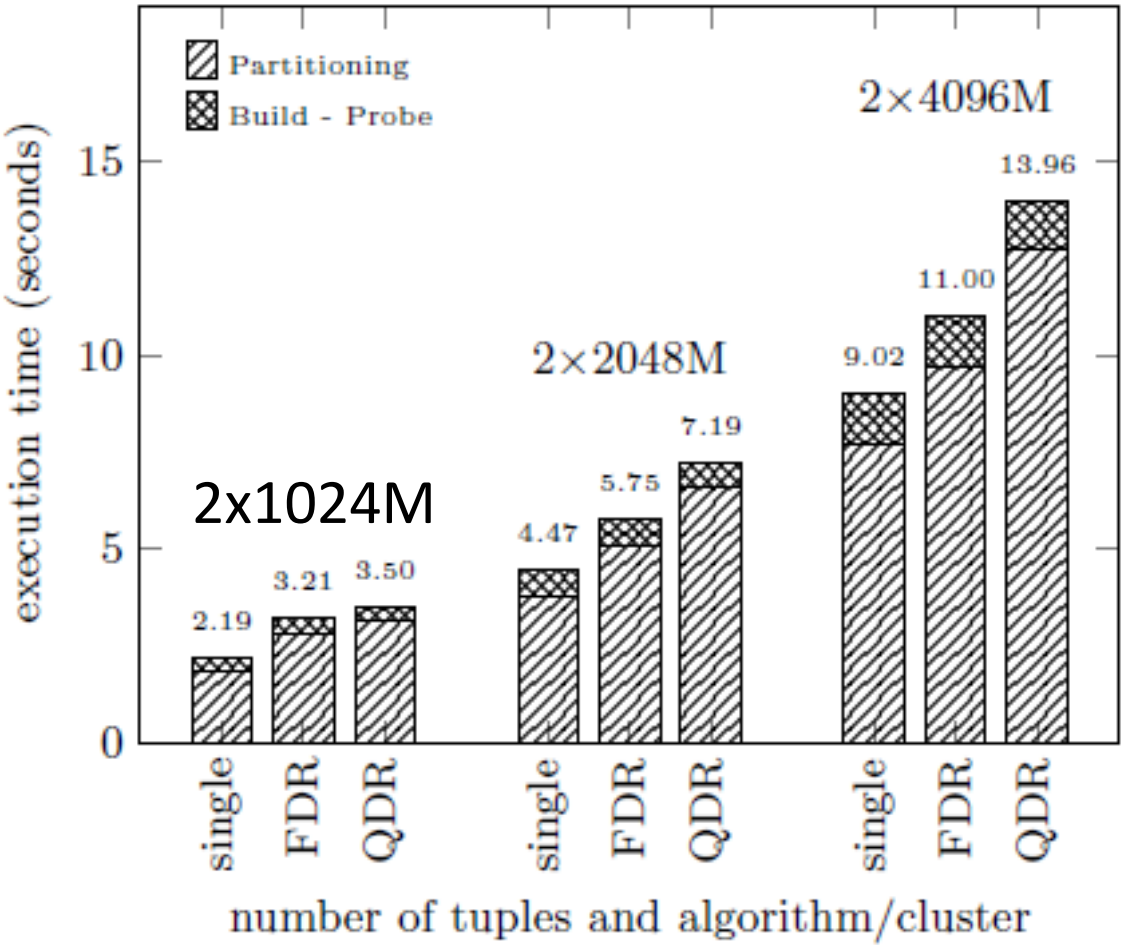
commenting on Sowell et al., PVLDB'13)

Hardware makes life easier

Thesis 1: The computer you know no longer exists

(at least, the interesting ones)

Multicore vs cluster



Barthels et al., SIGMOD'15

Nobody ever got fired for using Hadoop on a Cluster

A. Rowstron, D. Narayanan, A. Donnely, G. O'Shea, A. Douglas

HotCDP 2012, Bern, Switzerland

Analysis of MapReduce workloads:

- Microsoft: median job size < 14 GB
- Yahoo: median job size < 12.5 GB
- Facebook: 90% of jobs less than 100 GB

Fit in main memory

One server more efficient than a cluster

Adding memory to a big server better than using a cluster

Data movement is bad

It costs energy

It takes time

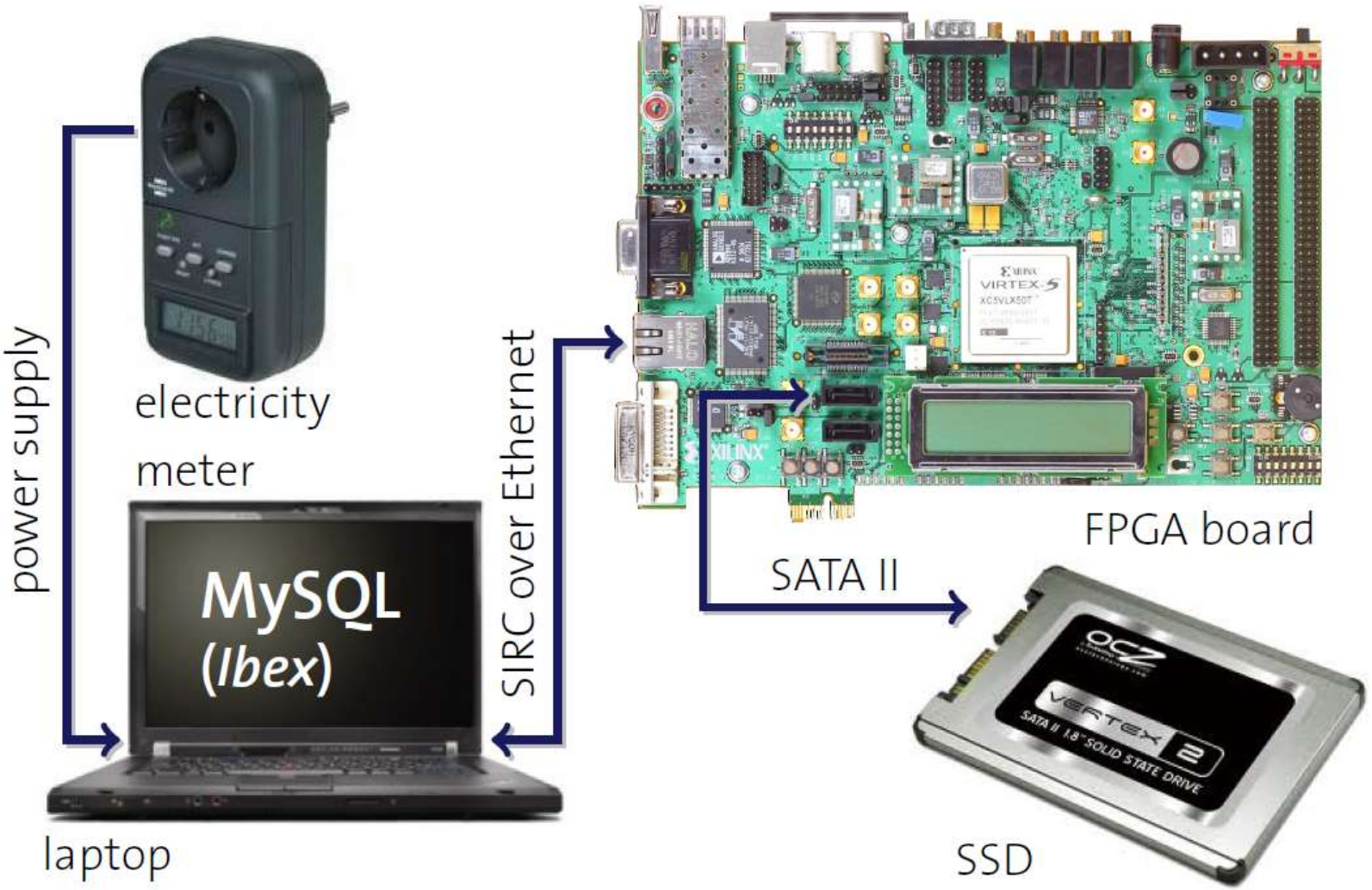
Bandwidth bottlenecks (I/O, network, cache hierarchies, etc.)

Solution:

use hardware to process data

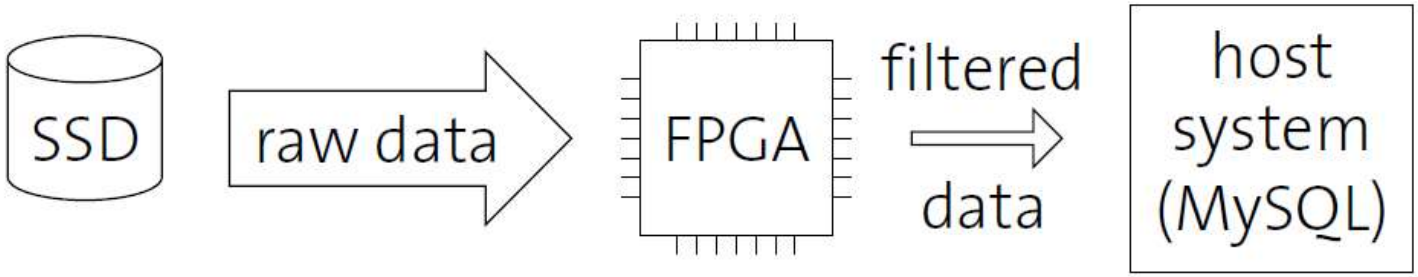
in place or as it flows

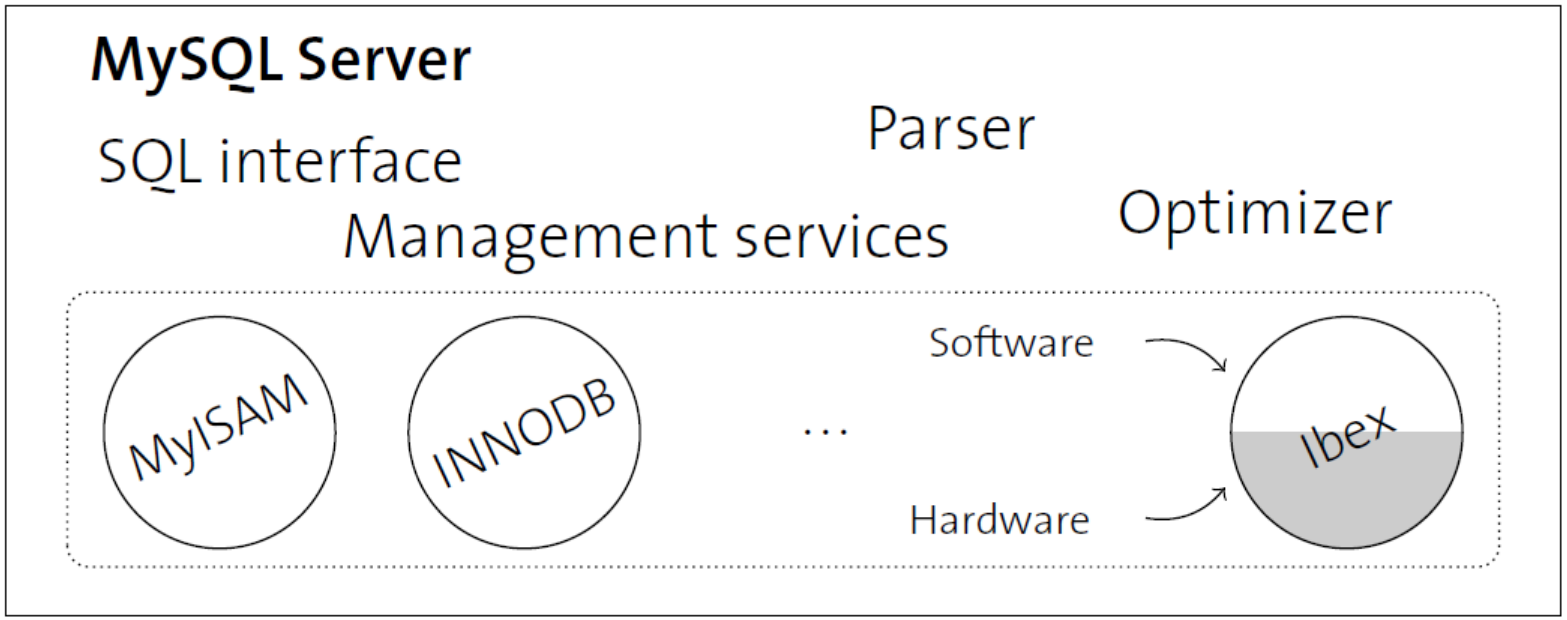
Thesis 2: Process Data as it flows



IBEX (Woods, VLDB'14; Istvan, SIGMOD'14)

A processor on the data path





Not a one trick pony:

- Skyline (FCCM'13)
- Complex Event Detection (PVLDB'11)
- Histogram calculation (SIGMOD'14)
- Aggregation (PVLD'14)
- Simple statistics (PVLDB'09)

Sounds good?

Imagine the same at all levels:

- Smart storage
- On the network switch (SDN like)
- On the network card (smart NIC)
- On the PCI express bus
- On the memory bus (active memory)

Every element in the system

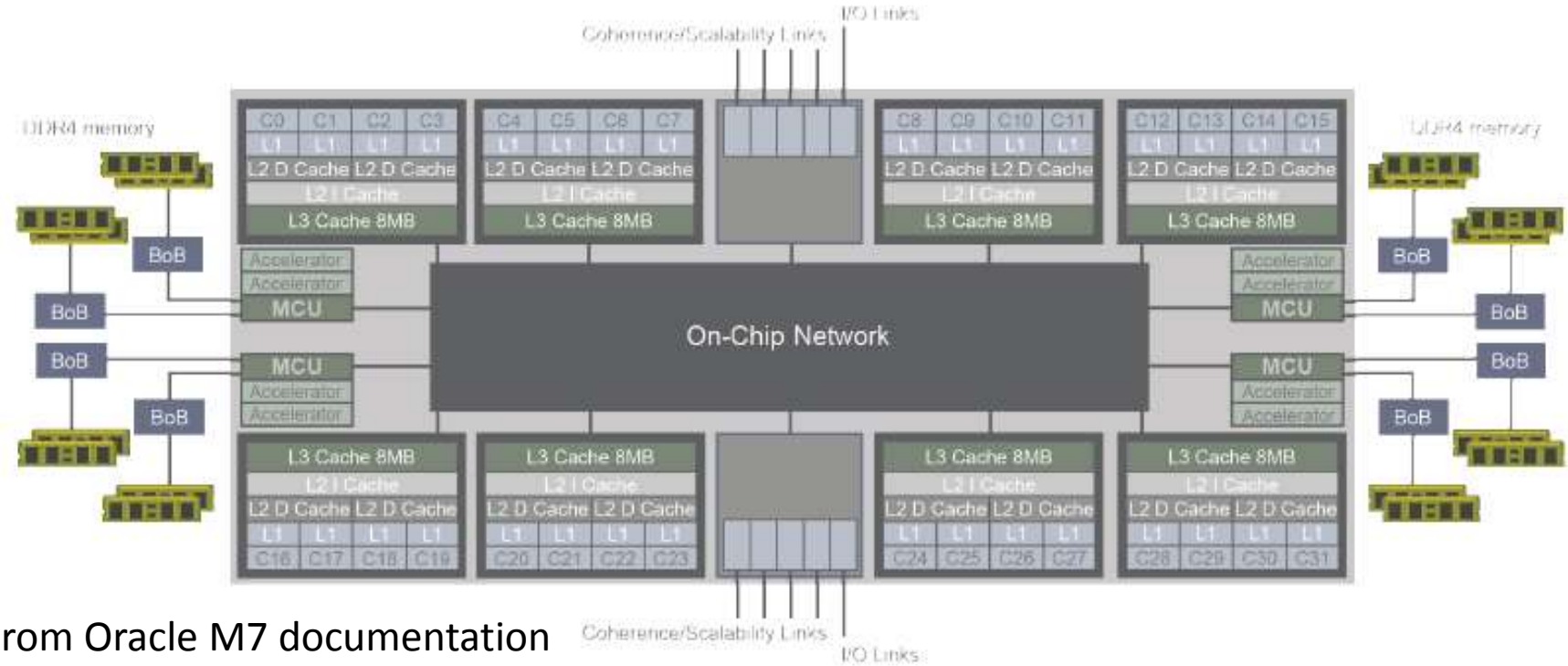
(a computer rack)

will be a processing component

Concrete examples

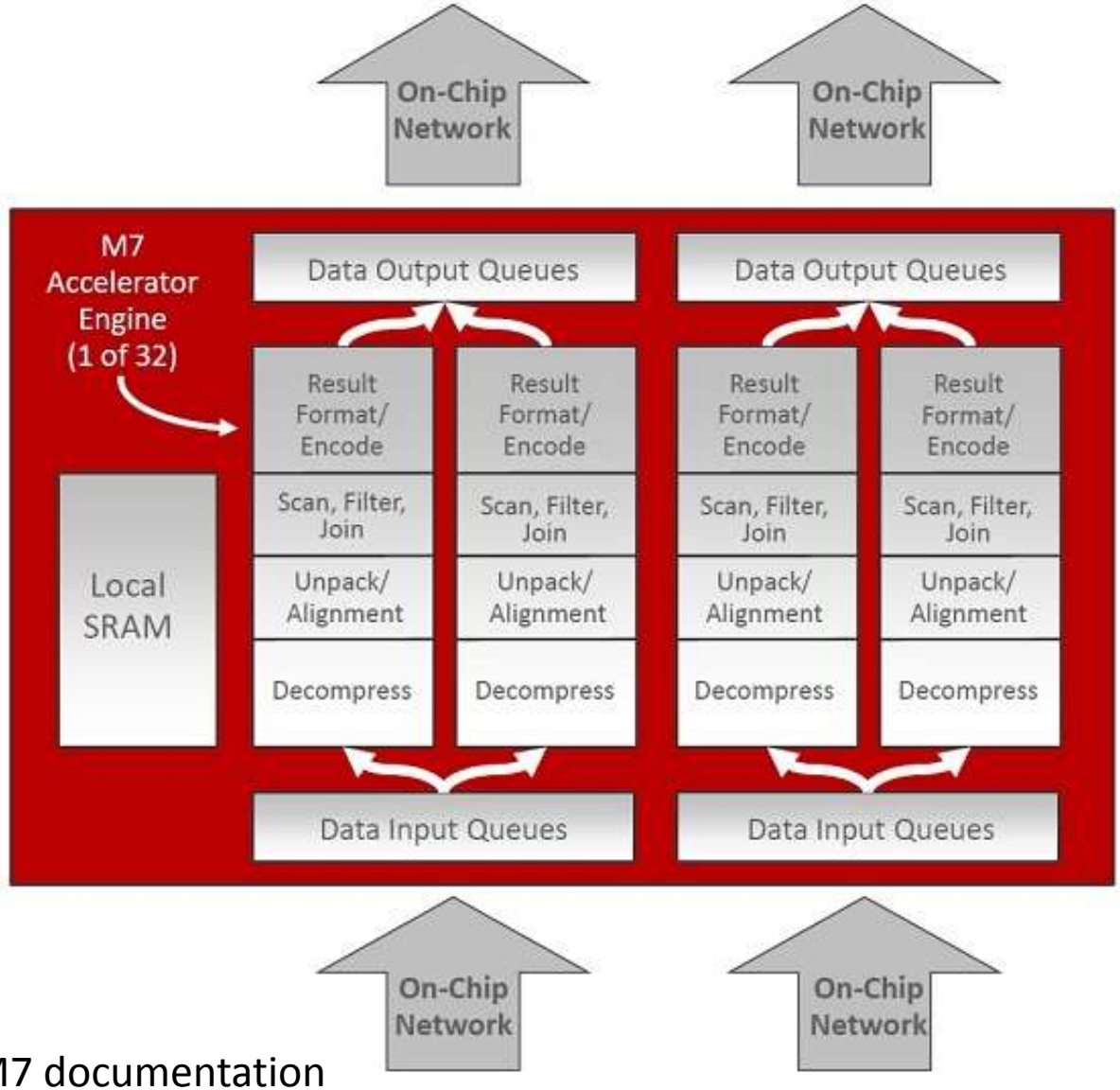
Oracle's SPARC M7 processor: "SQL in silicon" accelerators processing streams of data from memory:

Decompress, Scan, Select, Translate



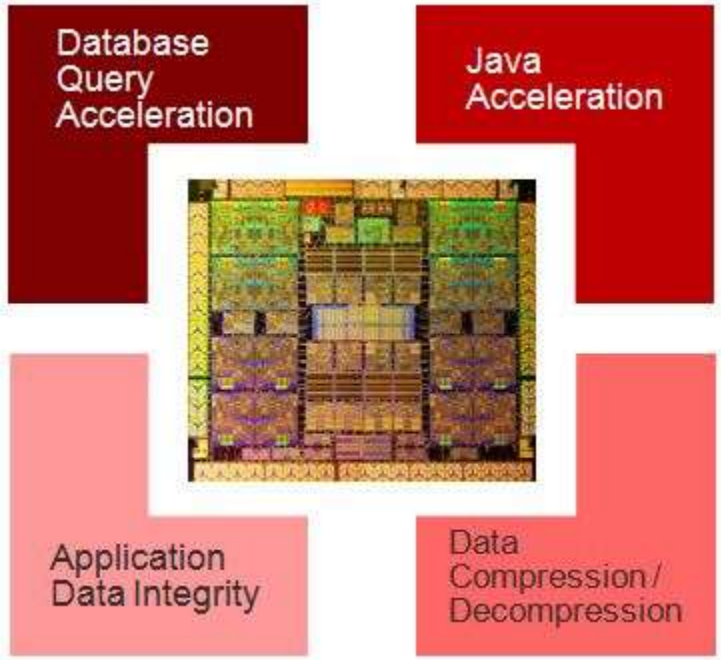
From Oracle M7 documentation

Accelerators to come



From Oracle M7 documentation

Thesis 3: Not everything that processes is a CPU



From Oracle M7 documentation

A processor far, far away …

A co-processor (GPU, Xeon Phi, FPGA) on a PCI bus works only when:

- Load is computationally bound
- Load remains computationally bound on the co-processor
- The data movement cost is less than the performance gain in the co-processor vs the CPU

Relational operators are often memory bound …

But you do not need a processor

Hardware speed through specialized instructions in the processor:

Oracle's SPARC M7: cryptographic functions in hardware

Intel's SGX: encryption decryption of in memory data in hardware to sandbox applications

"Meet the Walkers": specialized hardware for hashing in database operations (Kocberber et al., MICRO'13)

An application-specific instruction set for accelerating set-oriented database primitives. (Arnold et al. SIGMOD'14)

Memory not just memory

- “Active” memory : Parameterizable memory access to retrieve in one go data that is not contiguous (row store, column store)

MICRO 2015

Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-unit Strided Accesses

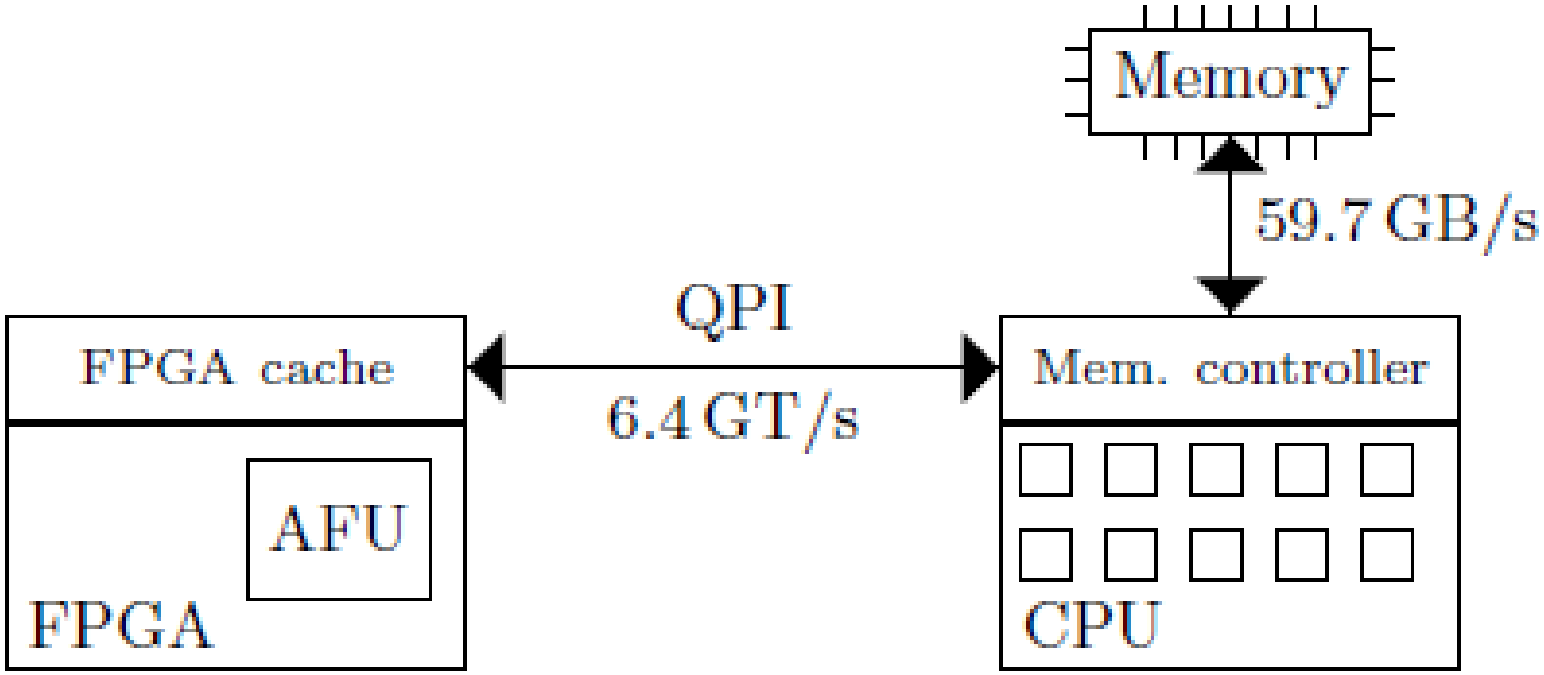
Vivek Seshadri, Thomas Mullins, Amirali Boroumand,
Onur Mutlu, Phillip B. Gibbons*, Michael A. Kozuch†, Todd C. Mowry
Carnegie Mellon University †Intel Labs

Abstract

Many data structures (e.g., matrices) are typically ac-

Categories and Subject Descriptors
B.3.1 [Memory Structures]: Semiconductor Memories

Heterogeneous multicore



INTEL HARP Program

DISCLAIMER: this is pre-production hardware and software, and may not reflect the performance of production or future systems.

User Defined … Hardware

User Defined Functions extend the functionality of a database

Imagine the same but in hardware (extending performance or functionality)

Perform operations on relational data that no database has ever done before: Skylines, Monte-Carlo, pattern matching, clustering, complex text search, advanced statistics, learning, …

Istvan, Siedler, et al. FCCM'16

Many more examples

- Oracle RAPID: High scale parallel processor for Oracle Exadata
- Microsoft Catapult: FPGA acceleration for search tasks (page rank like algorithms)
- HP “The Machine”
- Microsoft Cypherbase: FPGA for encrypted database processing in the cloud
- Intelligent storage systems: NetApp, Oracle, …

Systems@ETH Zürich

This is the end …

The agenda ahead of us

- Very interesting times
  - Many opportunities driven by hardware
  - Plenty of use cases justifying specialization
- Many challenges
  - Hardware changes affect the whole stack
  - How to program heterogeneous architectures
- We need to look at what is happening out there
  - Architecture
  - Data centers
  - Economic pressures and models