Systems@ETH zürich

Data Processing with FPGAs on Modern Architectures

Gustavo Alonso
Systems Group
Department of Computer Science
ETH Zurich, Switzerland

The tutorial

- Slides available at:

https://systems.ethz.ch/research/data-processing-on-modern-hardware/hacc/sigmod-23-tutorial--data-processing-on-fpgas-with-modern-archite.html

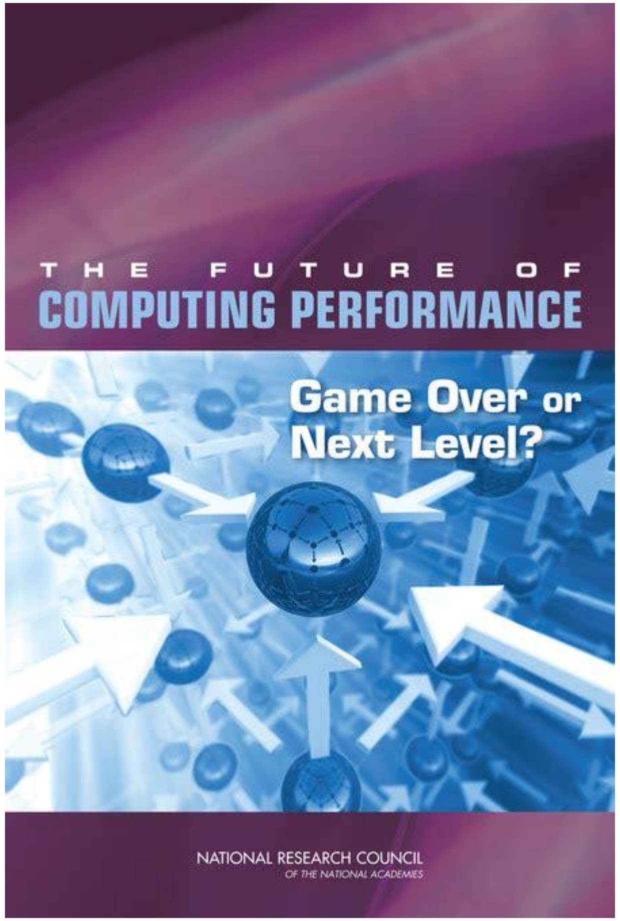
- Another tutorial available with more details on networking using FPGAs
- More coming in the next months

Schedule

- Introduction and Motivation
- Programming FPGAs
- Resources Available
- Use cases
  - Smart Disaggregated Memory
  - Recommendation System
  - Approximated Nearest Neighbor Search

The Hardware Era

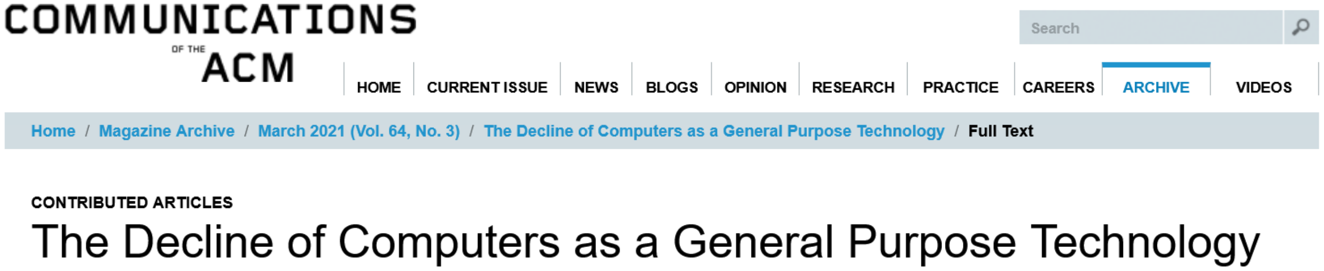
Not a new concept …



- 2011 Report
- Exponential growth for several decades
- Exponential growth no longer possible
- Switch to multicore and parallelism
  - Energy consumption becomes an issue
  - Multicore introduces parallelism that we do not know how to exploit well
- Situation will not change in near future
- Alternative is specialization
- Either somebody comes up with a new great invention or there is a problem

General purpose computing

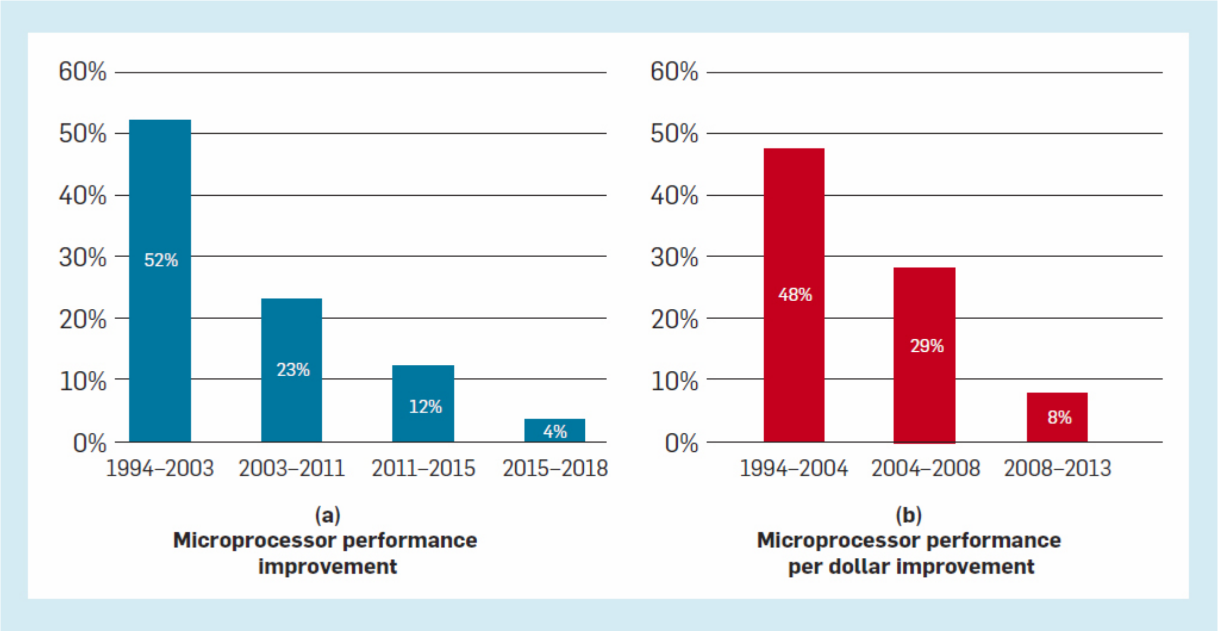
Slow improvements lead to specialization



Home / Magazine Archive / March 2021 (Vol. 64, No. 3) / The Decline of Computers as a General Purpose Technology / Full Text

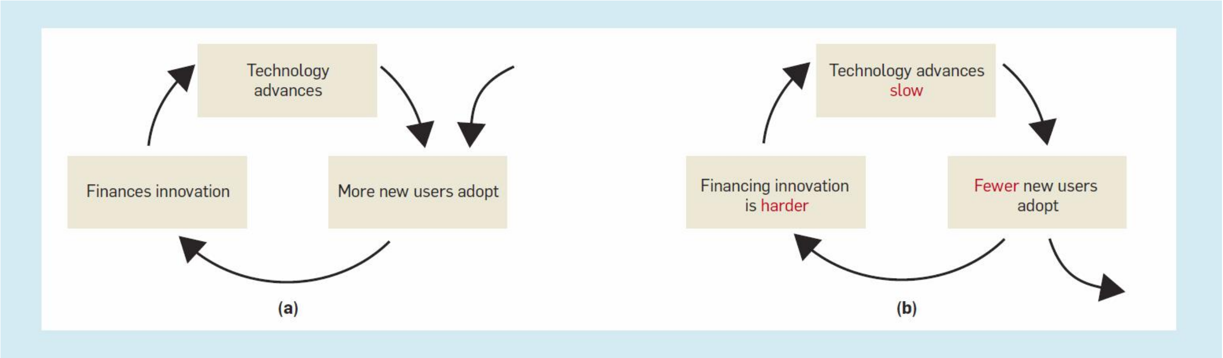
CONTRIBUTED ARTICLES

The Decline of Computers as a General Purpose Technology



(a)
Microprocessor performance improvement

(b)
Microprocessor performance per dollar improvement

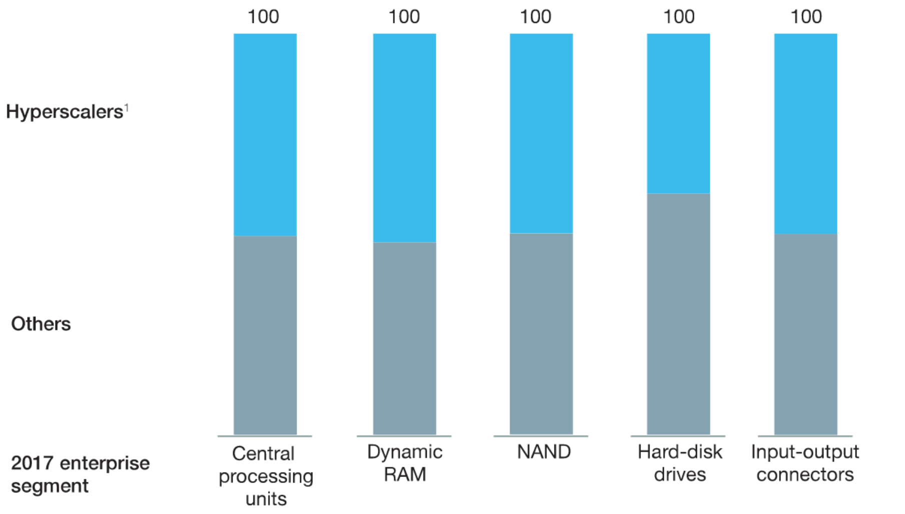


Driving specialization

- The cloud is the big game changer:
  - New business model
  - Economies of scale
  - Very large workloads
- Every hyper scaler is its own "Killer App"
  - The scale makes many things feasible
  - The gains have a very large multiplier

Hyperscalers, commanding a growing share of the market, are emerging as significant customers for many components.

2017 share of hyperscalers in component markets, market estimates, %



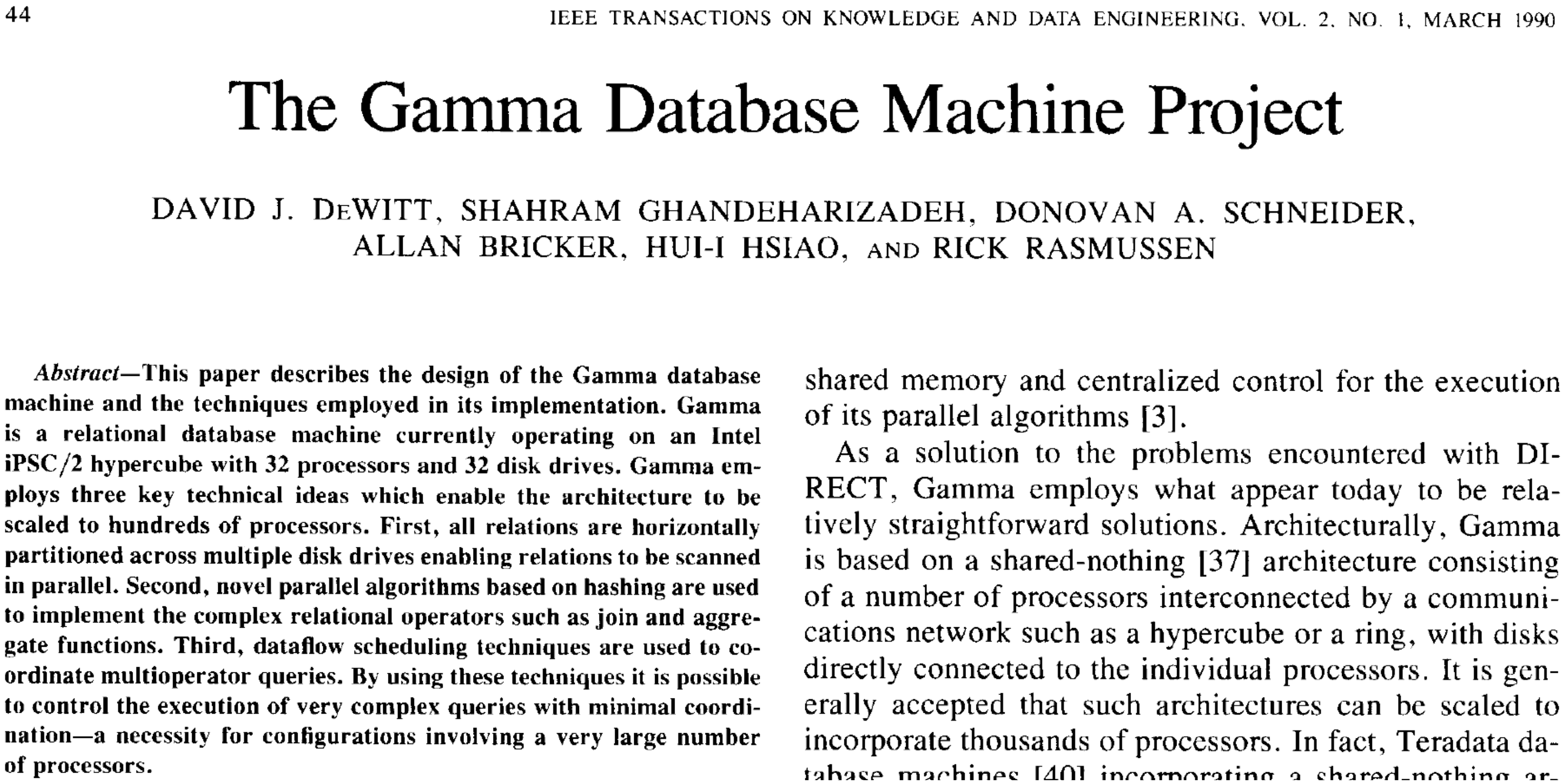
[1] Includes Alibaba, Alphabet, Amazon, Baidu, Facebook, Microsoft, and Tencent.

McKinsey&Company

https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-high-tech-suppliers-are-responding-to-the-hyperscaler-opportunity

HW Specialization for databases

Have we been here before?



44 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 2, NO. 1, MARCH 1990

The Gamma Database Machine Project

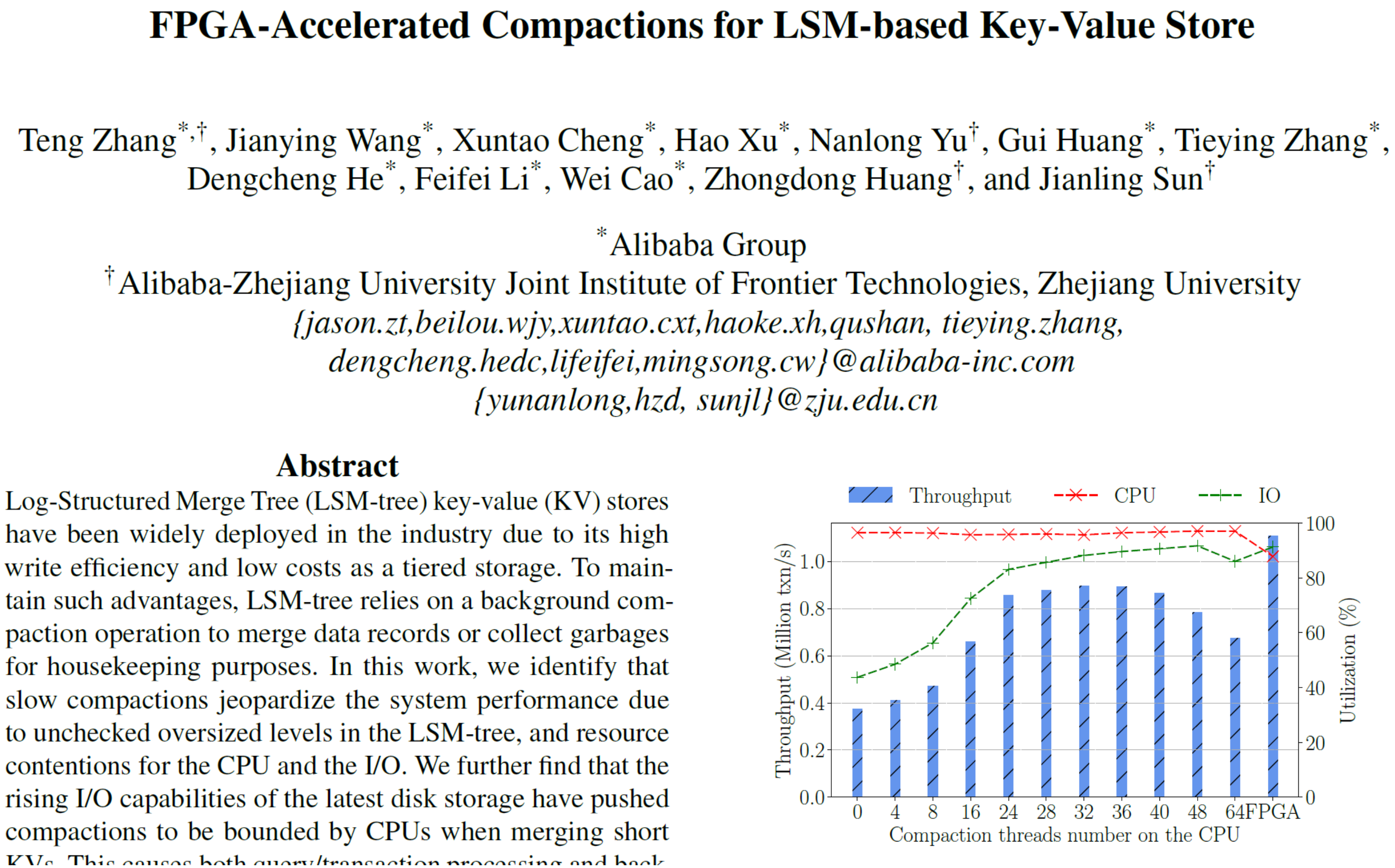
DAVID J. DEWITT, SHAHRAM GHANDEHARIZADEH, DONOVAN A. SCHNEIDER,
ALLAN BRICKER, HUI-I HSIAO, AND RICK RASMUSSEN

Abstract—**This paper describes the design of the Gamma database machine and the techniques employed in its implementation. Gamma is a relational database machine currently operating on an Intel iPSC/2 hypercube with 32 processors and 32 disk drives. Gamma employs three key technical ideas which enable the architecture to be scaled to hundreds of processors. First, all relations are horizontally partitioned across multiple disk drives enabling relations to be scanned in parallel. Second, novel parallel algorithms based on hashing are used to implement the complex relational operators such as join and aggregate functions. Third, dataflow scheduling techniques are used to coordinate multioperator queries. By using these techniques it is possible to control the execution of very complex queries with minimal coordination—a necessity for configurations involving a very large number of processors.**

shared memory and centralized control for the execution of its parallel algorithms [3].

As a solution to the problems encountered with DIRECT, Gamma employs what appear today to be relatively straightforward solutions. Architecturally, Gamma is based on a shared-nothing [37] architecture consisting of a number of processors interconnected by a communications network such as a hypercube or a ring, with disks directly connected to the individual processors. It is generally accepted that such architectures can be scaled to incorporate thousands of processors. In fact, Teradata database machines [40] incorporating a shared-nothing ar-

X-Engine (Alibaba)



FPGA-Accelerated Compactions for LSM-based Key-Value Store

Teng Zhang[*,†], Jianying Wang[*], Xuntao Cheng[*], Hao Xu[*], Nanlong Yu[†], Gui Huang[*], Tieying Zhang[*],
Dengcheng He[*], Feifei Li[*], Wei Cao[*], Zhongdong Huang[†], and Jianling Sun[†]

[*]Alibaba Group
[†]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Zhejiang University
{jason.zt,beilou.wjy,xuntao.cxt,haoke.xh,qushan, tieying.zhang,
dengcheng.hedc,lifeifei,mingsong.cw}@alibaba-inc.com
{yunanlong,hzd, sunjl}@zju.edu.cn

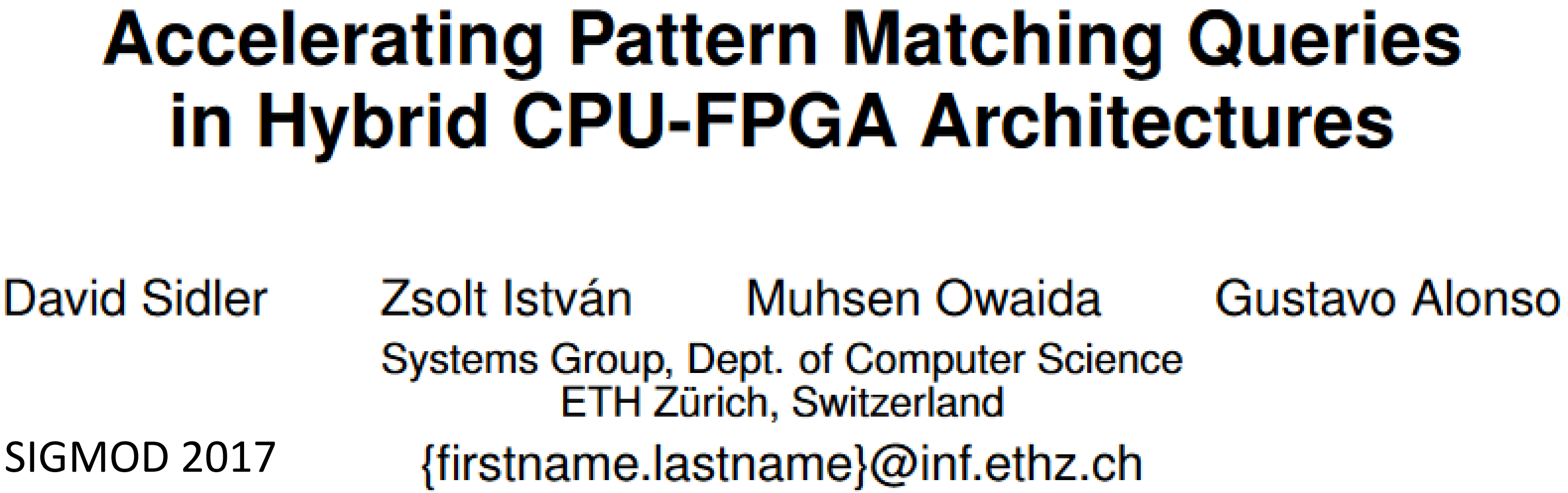
Abstract

Log-Structured Merge Tree (LSM-tree) key-value (KV) stores have been widely deployed in the industry due to its high write efficiency and low costs as a tiered storage. To maintain such advantages, LSM-tree relies on a background compaction operation to merge data records or collect garbages for housekeeping purposes. In this work, we identify that slow compactions jeopardize the system performance due to unchecked oversized levels in the LSM-tree, and resource contentions for the CPU and the I/O. We further find that the rising I/O capabilities of the latest disk storage have pushed compactions to be bounded by CPUs when merging short KVs. This causes both query/transaction processing and back

18th USENIX Conference on File and Storage Technologies (FAST'20)

Lesson Learned

- Database engines are bad at operations that have now become very important
- This is not because they are bad at it but because the underlying hardware (CPU) makes it expensive
- Other devices enable these operations, let's embrace them …



Accelerating Pattern Matching Queries in Hybrid CPU-FPGA Architectures

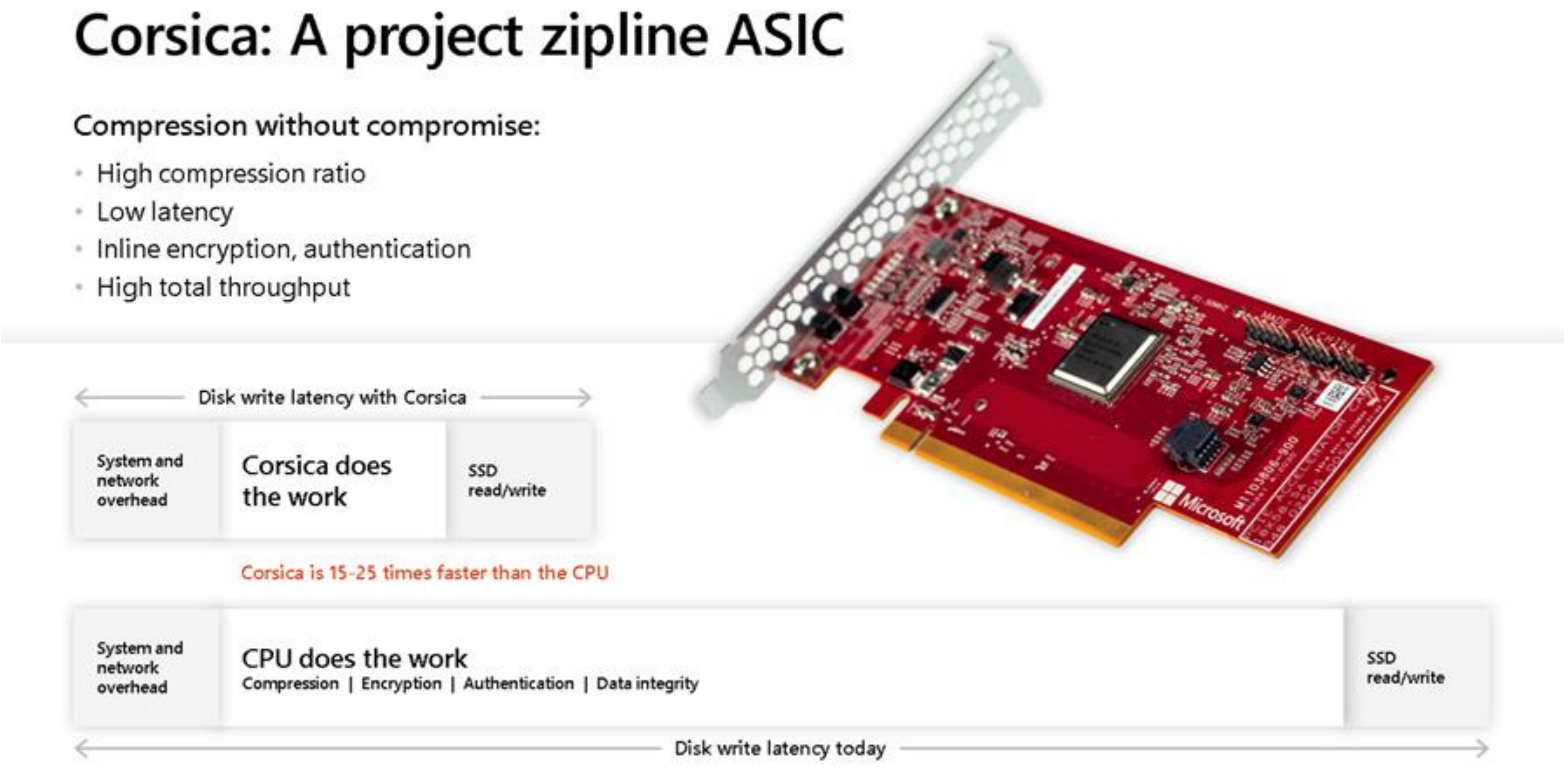
David Sidler Zsolt István Muhsen Owaida Gustavo Alonso
Systems Group, Dept. of Computer Science
ETH Zürich, Switzerland
{firstname.lastname}@inf.ethz.ch

SIGMOD 2017

Lesson Learned

- Even modern designs of database engines consume too many resources for operations internal to the engine
- Some of these operations are better done somewhere else than in a CPU
- Accelerators are available in data centers and the cloud, let’s embrace them …

Data Compression (Microsoft Zipline/Corsica)



Corsica: A project zipline ASIC

Compression without compromise:

- High compression ratio
- Low latency
- Inline encryption, authentication
- High total throughput

Disk write latency with Corsica

System and network overhead | Corsica does the work | SSD read/write

Corsica is 15-25 times faster than the CPU

System and network overhead | CPU does the work Compression | Encryption | Authentication | Data integrity | SSD read/write

Disk write latency today

https://azure.microsoft.com/en-us/blog/improved-cloud-service-performance-through-asic-acceleration/

Lesson Learned

Everything that is demanding enough and common enough will move to dedicated accelerators

Performance is not the only story …

- These deployments have one goal:
  - Free up the CPU for other tasks!

- In traditional database engines and data processing systems, the CPU does everything
  - No longer efficient!

First baby steps …

- Software evolves slower than hardware

- Current deployments focus on elements that can be migrated to hardware without major changes to the engines

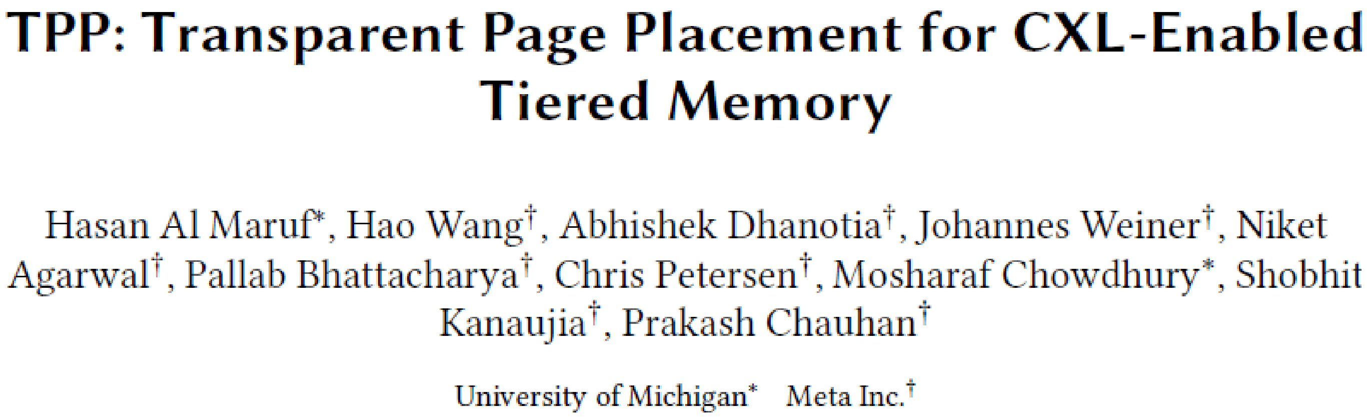
- But once the hardware is available …
  - Engines will be developed for that hardware
  - Pressure will increase to take advantage of heterogeneous architectures
  - That will be the next wave of truly cloud native database engines

Emerging themes

- Reduced CPU utilization
- Accelerate common operations
- Accelerate the infrastructure supporting the system
- Processing data on the fly
- Near data processing (memory, storage, …)
- On demand servers and functionality
- …

Ignore hardware developments at your own peril

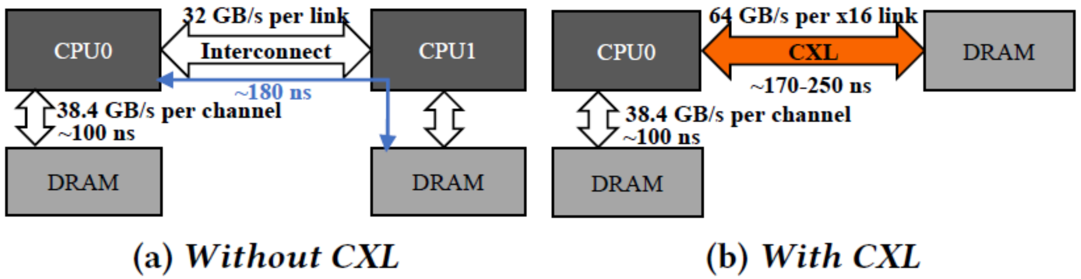
The future of accelerators



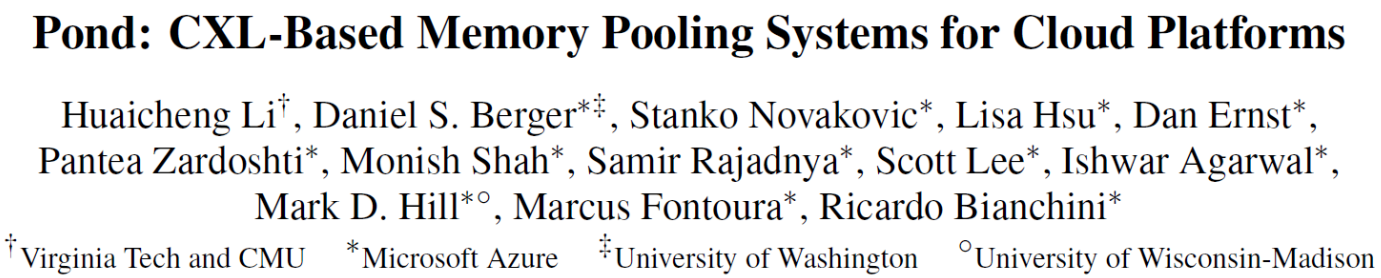
TPP: Transparent Page Placement for CXL-Enabled Tiered Memory

Hasan Al Maruf[*], Hao Wang[†], Abhishek Dhanotia[†], Johannes Weiner[†], Niket Agarwal[†], Pallab Bhattacharya[†], Chris Petersen[†], Mosharaf Chowdhury[*], Shobhit Kanaujia[†], Prakash Chauhan[†]

University of Michigan[*] Meta Inc.[†]



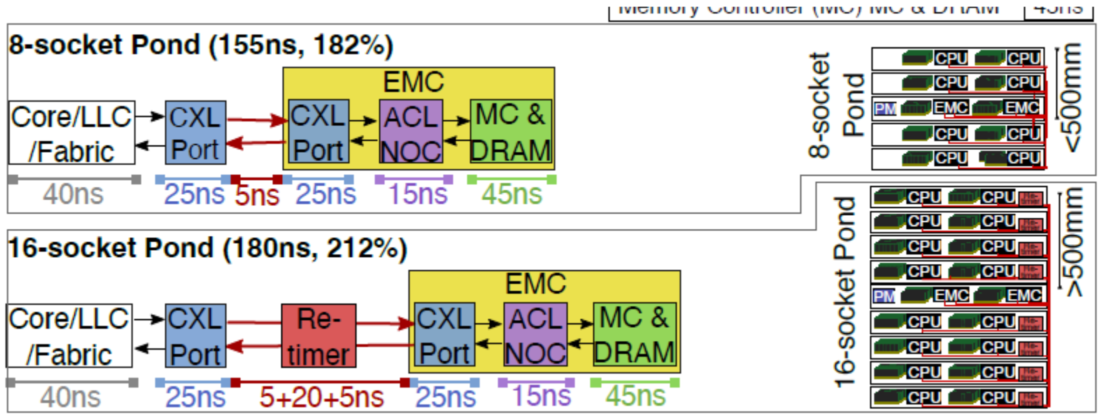
(a) *Without CXL* (b) *With CXL*



Pond: CXL-Based Memory Pooling Systems for Cloud Platforms

Huaicheng Li[†], Daniel S. Berger[*‡], Stanko Novakovic[*], Lisa Hsu[*], Dan Ernst[*], Pantea Zardoshti[*], Monish Shah[*], Samir Rajadnya[*], Scott Lee[*], Ishwar Agarwal[*], Mark D. Hill[*○], Marcus Fontoura[*], Ricardo Bianchini[*]

[†]Virginia Tech and CMU [*]Microsoft Azure [‡]University of Washington [○]University of Wisconsin-Madison



Disaggregated memory

- CXL memory will not be just memory
- It will be a module with a controller/processor that runs the protocol and manages the memory
- The controller is a great point to add near-data processing capabilities

The tutorial in context

FPGAs in context

- We do not sell or market FPGAs
- FPGAs are the only way to explore:
  - New architectural designs (even to the CPU design level, e.g., RISC-V)
  - New computer architectures (near-memory processing, smart storage, smart NICs, accelerators, etc.)
  - Processing of data streams at line rate
- Are FPGAs difficult?
  - No, this is systems level programming, no less involved than writing your own database engine, operating system, etc.
  - Yes, the tools are not what we are used to in the software world (by a long margin)

Goals

- Show the potential of hardware acceleration
- Introduction to FPGA programming
- Demonstrating resources available to researchers
- Several use cases as examples of what can be done

- Overall: encourage the community to explore this opportunity of achieving higher efficiency in data processing in a context where the new hardware is going to be available, even if for other reasons.