Post-Moore Server Architecture
Babak Falsafi
HPC is (mostly) moving to the Cloud

- AI needs data and data is in the cloud
- Cloud investment in (proprietary) AI platforms
- Cloud pushes the cost of computing to the limit
Modern Datacenters: The Backbone of Cloud

- A million home-brewed servers
- Centralized to exploit economies of scale
- Network fabric w/ μ-second connectivity
- At physical limits
- Need sources for
  - Electricity
  - Network
  - Cooling

250MW, Bedford
But, Silicon out of steam!

Silicon efficiency is dead (long live efficient silicon)

Moore’s Law dying
[David Brooks, SIGARCH’18]
But, Silicon out of steam!

Silicon efficiency is dead (long live efficient silicon)  

Moore’s Law dying  
[David Brooks, CAT’18]

Conventional scaling 41%/year  
Recent years 15%/year!  
[David Brooks, Computer Architecture Today]
The future of Digital Platforms: Cross-Stack Optimization

ISA opportunities

- Integration
  - Move less frequently
  - Move less distance

- Specialization
  - Customize work
  - Less work/computation

- Approximation
  - Adjust precision
Decarbonizing datacenters:

- Center at EPFL founded in 2011
- 21 faculty, 100+ researchers

Holistic datacenter design:

- Minimizing electricity in IT services
- Post-Moore server design
- Integrated cooling, renewables
- From algorithms to infrastructure
Outline

- Overview
  - Post-Moore Servers
    - Blades are 80’s Desktops
    - Specialized logic
    - Integrated logic/memory
    - Integrated network
    - ML approximation
  - Summary
Scale-Out Datacenters

Vast data sharded across servers

Memory-resident workloads
- Necessary for performance
- Major TCO burden

Put memory at the center
- Design system around memory
- Optimize for data services

Servers driven by the DRAM market!
Today’s Server Blades

Today’s platforms are PC's of the 80’s
- CPU “owns” and manages DRAM in hardware
- OS moves data back/forth from peripherals (SSD, Net)
- Legacy interfaces connecting the CPU/mem to outside
- Legacy POSIX abstractions

Fragmented logic/memory:
- NIC’s w/ manycores interfacing DRAM
- Flash controllers with cores and DRAM
- Discrete accelerators on PCIe with DRAM
80's Desktop

- 33 MHz 386 CPU, 250ns DRAM
- OS: Windows, Unix BSD (or various flavors)
- Focus: multiprogrammed in-memory compute
Today's Blade: 80's Desktop

- Dual 2GHz CPU’s, 50ns DRAM
- OS: Linux (and various distributions)
Today’s Blade: 80’s Desktop

- Dual 2GHz CPU’s, 50ns DRAM, Linux
- Bottlenecked by legacy interfaces
- Fragmented silicon
Today’s Blade: 80’s Desktop

- Dual 2GHz CPU’s, 50ns DRAM, Linux
- Bottlenecked by legacy interfaces
- Fragmented silicon
The Elephant in the Room
Ideal Blade of Tomorrow

- Intra-node fabric connecting components
- Control path: setup via CPU & OS
- Data path: direct access @ hardware speed
Accelerator on the network:
1. CPU/OS host set up control plane
2. Accelerators directly communicate over network
3. Implement distributed ML service
Outline

- **Overview**

- **Post-Moore servers**
  - Blades are 80's Desktops
  - Specialized logic
  - Integrated logic/memory
  - Integrated network
  - ML approximation

- **Summary**
The Specialization Funnel

General Purpose
- Intel CPU
- Oracle DB
- Linux OS
- Python/C PL
- ...

Specialized
- GPU/Thunder X
- DBToaster
- IX Kernel
- PyTorch

Domain-specific languages to platforms
Novel HW/SW interfaces

ASIC
- Crypto
- Network logic
- Analog NN
Server Benchmarking with CloudSuite 3.0 (cloudsuite.ch)

Data Analytics
- Machine learning
  - Wikipedia
  - Mahout

Graph Analytics
- GraphX
  - Twitter
  - Spark

In-Memory Analytics
- Recommendation System
  - Movielens

Web Search
- Apache Solr & Nutch
  - Solr
  - Nutch

Media Streaming
- Nginx, HTTP Server
  - YouTube
  - Nginx

Web Serving
- Nginx, PHP server
  - Nginx
  - PHP

Data Caching
- Memcached
  - Twitter
  - Memcached

Data Serving
- Cassandra NoSQL
  - Facebook
  - Cassandra

Building block for Google PerfKit, EEMBC Big Data!
Services Stuck in Memory [ASPLOS'12]

Cache overprovisioned

Instruction supply bottlenecked
Scale-Out Processors (SOP)

- General-purpose CPU
- Logic 60% of silicon
- 6x bigger cores

- 3-way OoO ARM
- 85% logic, 7x more cores
- Faster instruction supply
Commercialized SOP

Thunder X1

- Blueprinted @ EPFL
- Designed to serve data
- Optimized instruction supply
- Trade off SRAM for cores
- Runs stock software
- CloudSuite 10x faster than Xeon
Walkers: DB Accelerators [MICRO'13]

- Traverse data structures (e.g., hash table, B-tree)
- Parallelize pointer chains
- Decouple hash&walk, overlap multiple walks

15x better perf/Watt over Xeon
Walkers in Software [VLDB’16]

Use insights to help Xeon
- Decouple hash & walk(s) in software
- Schedule off-chip pointer access with co-routines

2.3x speedup on Xeon
- Unclogs dependences in microarchitecture
- Maximizes memory level parallelism
- DSL w/ co-routines
- To be integrated in SAP HANA [VLDB’18]
Integrated Logic/Memory

DRAM stack w/ nearby logic
- Minimize data movement
- Massive internal bandwidth

Opportunities for algorithm/hardware co-design
Cost of moving data

Data access much more expensive than arithmetic operation
DRAM BW bottleneck

100's of GB/s internally

Internal DRAM BW presents big opportunity

24 GB/s off-chip BW per controller
Near-Memory Processing (NMP)

3D logic/DRAM stack (or interposer)
- Exposes internal BW to processing elements
- But constrains logic layer’s area/power envelope

Exploit the bandwidth without data movement
NMP Commandments
[IEEE Micro issue on Big Data’16]

Not (CPU) business as usual
1. DRAM favors streaming over random access
2. DRAM favors parallelism over arithmetic speed
3. NMP DRAM must maintain CPU memory semantics

Co-design algorithm/HW for NMP
Why not random access?

Internally DRAM is a block device
- Activating a 1KB row
- High latency & energy per row
- Exploit row locality for efficiency

Example:
- For DRAM with 128 GB/s internal bandwidth
- Optimal (parallel) random access only captures ~8 GB/s
- Requires 5x more power

Use algorithms that favor sequential access
The Mondrian Data Engine [ISCA’17]

SIMD cores + data streaming
- Saturates b/w with parallel SIMD streams
- 1024-bit SIMD @ 1 GHz
- No caches

Runs Spark Analytic Ops
50x over Xeon

Algorithm/hardware co-design maximize near-memory performance
Case Study: Join on Mondrian

Revisiting Sort join:
- Sort join ($O(n \log n)$) vs. Hash Join ($O(n)$)
- Sort tables and then merge join
- Sequential vs. random access

Perform way more work
But, finish faster and use less power!

Trade off algorithm complexity for sequential memory accesses
Performance

- Algorithm alone gets \(\sim 10x\) [ASBD'15]
- Algorithm/hardware co-design gets 50x
TB-Scale Hierarchies

Yesterday

Faster

Bigger

Moving Forward

Regs
Caches (SRAM)
Main memory (DRAM)
SSD
Hard disk

Regs
Caches (SRAM)
3D Caches (DRAM)
Main memory (DRAM)
Storage-Class Memory
SSD
Hard disk
Capacity/Miss Rate 101

- HOT
  - 32 GB

- COLD
  - 1 TB
Emerging Hierarchies

- Cache
- CPU
- Serial links
- On-package DRAM
- DDR bus
- Off-package DRAM
- TB (cold data)
- MB (instructions)
- GB (hot data)
- GB (hot data)
- HOT
- COLD
Broken Legacy Abstractions: Address Translation

Probing a hash table (32GB)

Page size
- 4KB
- 2MB

16K (2MB) entries

ASCYLIB on Broadwell (1.5K-entry TLB)
TB-Scale Address Translation

- Namespace in cache hierarchy fragmented into pages
- TLBs of 1000s of entries replicated per core
- Fragmentation hurts both lookup & access control
TB-Scale Translation

Coalescing entries:
- ✗ Small factor (e.g., 2x) in improvement

Partitioned NUMA [Picorel’18]:
- ✓ Linux support for both data placement & page walks
- ✗ Small factor in improvement

Segmentation/Direct Access [Haria’18]:
- ✗ Segment fragmentation
- ✗ Software exposure

Virtual hierarchies:
- ✓ Push translations off the critical path
- ✗ Software exposure (synonyms)
Outline

- Overview
- Post-Moore servers
  - Blades are 80's Desktops
  - Specialized logic
  - Integrated logic/memory
- Integrated network
- ML approximation
- Summary
Networks

Network stack bottleneck:
- B/W growing faster than silicon
- Emerging µServices + serverless
- RPC, orchestration, ….

Key challenges:
- New abstractions
- Co-design of network stacks
Memory is the most precious silicon

- Pool memory over the network
  - Load balancing shared object store [Novakovic’14]
  - Shared swap space [Gu’17]
- Offload pooling logic
  - Reduce effective access time
  - Intelligent management of pool resources & capacity
- Minimize fragmentation
soNUMA:

- Socket-integrated network interface
- Protected global memory read/write + synch
- Fine-grain (~64B) & bulk objects (~1MB)
- Remote memory ~ 2x local memory latency
- Extensions for messaging & RPC
RPC Accelerators: Dispatch
[ASPLOS’19, ISCA’20]

Socket-integrated NICs
Can reduce RPC queuing:
- Load imbalance among cores
- DRAM B/W interference w/ spilling

1. Single-queue dispatch w/ monitors
2. HW terminated protocol in LLC
Orchestration (including transformation) is a bottleneck
Transformation is 10x-100x slower than network line rate
Hardware/software co-designed transformer @ line rate
Outline

- Overview
- Post-Moore servers
  - Blades are 80's Desktops
  - Specialized logic
  - Integrated logic/memory
  - Integrated network
- ML approximation
- Summary
Arithmetic in Deep Learning (Microsoft Brainwave)

FPGA Performance vs. Data Type

- Stratix 10 280 @ 500MHz

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Tera ops/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit int</td>
<td>12</td>
</tr>
<tr>
<td>8-bit int</td>
<td>31</td>
</tr>
<tr>
<td>ms-fp9</td>
<td>65</td>
</tr>
<tr>
<td>ms-fp8</td>
<td>90</td>
</tr>
</tbody>
</table>

7.5X
Floating vs. Fixed Point: Representable Range

- **Floating point**
  - Mantissa + exponent
  - Wide representable range
  - Value has independent range

- **Fixed point**
  - Mantissa
  - Narrow representable range
  - Values range pre-determined

Wide representable range

Narrow representable range

\[
\log_2(\text{magnitude})
\]

Pct. of values
Floating vs. Fixed Point: Area and Power

- **Floating point**
  - Mantissa + exponent
  - Complex exponent management

- **Fixed point**
  - Mantissa
  - No exponent management

**ALU Hardware**
Hybrid BFP-FP (HBFP) [NeurIPS’18]

Block floating point (BFP) shares exponents in blocks
- Used in DSP’s to reduce silicon footprint
- > 90% of arithmetic with one exponent/tensor

Use FP32 for all activations and other arithmetic

Co-Located Training & Inference (ColTraIn)
- Can piggyback training on an inference accelerator
- Uses fixed point logic (HBFP) for both training & inference
- Optimizes for data movement beyond logic
HBFP (Block FP) vs. FP32

Resnet-50 on ImageNet (BERT numbers coming soon)

<table>
<thead>
<tr>
<th>Config</th>
<th>Top-1 Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBFP8</td>
<td>23.88</td>
</tr>
<tr>
<td>HBFP12</td>
<td>23.58</td>
</tr>
<tr>
<td>FP32</td>
<td>23.64</td>
</tr>
</tbody>
</table>

FP32 performance with 8-bit logic
Summary

Trends:
- Demand is growing faster than Moore
- Moore's law is slowing down
- Memory is a growing fraction of TCO

Post-Moore servers:
- Revisit legacy abstractions
- Holistic Hardware/OS Co-design
- CPUs, accelerators, network, storage, system

Integration + Specialization + Approximation
Thank You!

For more information please visit us at ecocloud.ch