Integration, Specialization and Approximation
the “ISA” of Post-Moore Servers

Babak Falsafi
Datacenter Growth

Market Growth 2018-2023
[Technavio, IDC]

- Data → fuel for digital economy
- Exponential demand for digital services
- Many apps (e.g., AI) with higher exponential demand
DATA CENTERS ARE BACKBONE OF CLOUD

- 100s of 1000 of commodity or home-brewed servers
- Centralized to exploit economies of scale
- Network fabric w/ μ-second connectivity
- Often limited by
  - Electricity
  - Network
  - Cooling

350MW, Bedford
CLOUDS AT VARIOUS SCALES

Users & Devices

Temporal/Sensitive/Local Data

Edge Cloud

IXP Cloud

Public Cloud

.Persistent/Global Data

[src: Peterson, et. al.]
End of Moore's Law (of Silicon)
- Five decades of doubling density
- Recent slowdown in density
- Chip density limited by physics

Growth means building more
- 41%/year → 28x in ten years
- At 15%/year → 7x more DCs
Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by Karen Hao

Jun 6, 2019
Design for “ISA”

- **Integration**
  - Move data less frequently
  - Move data less distance

- **Specialization**
  - Customize resources
  - Less work/computation

- **Approximation**
  - Adjust precision
INTEGRATED COOLING [Thome, Atienza]

3D server chip
Two-phase liquid cooling
- Uniform higher thermals
- Higher heat removal
- Localized cooling
SPECIALIZED NETWORKS [Bugnion]

- Data plane principles: zero-copy, run-to-completion, coherence free
- Protected operating system with clean-slate API
- Accelerates object sharing in datacenters
- IX Kernel → best paper at OSDI’14
- Follow-on work → SIGOPS’21 dissertation award

3.6x throughput with <50% latency @ 99th percentile
SPECIALIZED DATABASES [Koch]

TOASTER

Compilation of offline analytics into online/incremental engines
Aggressive code specialization

Low-latency stream processing
Up to 6 OOM faster than commercial systems

dbtoaster.org
QUANTIFYING EFFICIENCY/EMISSIONS BEYOND "PUE" (sdea.ch)

DC INFRASTRUCTURE EFFICIENCY (PUE+)
- electrical, cooling and heat recycling components

IT INFRASTRUCTURE EFFICIENCY
+ compute, storage, network and workloads

DC CARBON FOOTPRINT
+ emissions from input electricity sources
OUTLINE

- Overview
- Post-Moore Server Architecture
  - 80’s Desktops
  - Specialized CPUs
  - Integrated logic/memory
  - Integrated networks
  - Approximating AI
- Summary
Cost is the primary metric
Online services hosted in memory
Divide data up across servers
Design server for low cost, scale out
Memory most precious silicon
Today’s platforms are PC’s of the 80’s

- CPU “owns” and manages memory
- OS moves data back/forth from peripherals
- Legacy interfaces connecting the CPU/mem to outside
- Legacy POSIX abstractions

Fragmented logic/memory:

- Manycore network cards w/ own memory
- Flash controllers with embedded cores and memory
- Discrete accelerators with own memory
80'S DESKTOP

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

HW Zone

OS Zone

CPU

Chipset

DRAM

NIC

I/O Bus
TODAY’S SERVER: 80’S DESKTOP

- Dual 2GHz CPU’s, 50ns DRAM
- OS: Linux (and various distributions)
TODAY'S SERVER: 80'S DESKTOP

- Dual 2GHz CPU's, 50ns DRAM, Linux
- Bottlenecked by legacy interfaces
- Fragmented silicon
TODAY’S SERVER: 80’S DESKTOP

- Dual 2GHz CPU’s, 50ns DRAM, Linux
- Bottlenecked by CPU, OS & legacy interfaces
- Fragmented silicon
IDEAL POST-MOORE SERVER

Control/Data planes

- Think of the server as a network
- Control plane: set up via CPU & OS
- Data plane: protected access to memory
- Eliminates silicon fragmentation
OUTLINE

- Overview
- Post-Moore servers
  - 80's Desktops
    - Specialized CPUs
    - Integrated logic/memory
    - Integrated networks
    - Approximating AI
- Summary
THE SPECIALIZATION FUNNEL

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

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

ASIC
- Crypto
- Network logic
- Analog NN

Domain-specific languages to platforms
New interfaces (i.e., IRs, hardware abstractions)
THE LIMITS OF CPUS

CPUs follow the von Neumann machine organization
- Machine instructions fetched from memory
- Operands fetched/written to memory
- Referred to as von Neumann bottleneck

Only 6% power in Pentium 4 spent in arithmetic (ALU)

[src: Chen, et. al., IEEE Transactions, 2006]
Three classes of workloads in datacenters

- First-party workloads (e.g., search, retail, media)
  1. Data management
  2. Analytics
    - Multi-tier to microservices
- Third-party workloads (cloud)
  3. Containerized
    - Emerging serverless
CloudSuite (cloudsuite.ch, 4.0 coming)

- Data Analytics
  - Machine learning
- Graph Analytics
  - GraphX
- In-Memory Analytics
  - Recommendation System
- Web Search
  - Apache Solr & Nutch
- Media Streaming
  - Nginx, HTTP Server
- Web Serving
  - Nginx, PHP server
- Data Caching
  - Memcached
- Data Serving
  - Cassandra NoSQL
- Dynamic Content
- OpenSource Web
  - Performance

Supports x86, ARM64, RISC-V
SERVICES STUCK IN MEMORY [ASPLOS'12]

Cache overprovisioned

Instruction supply bottlenecked
SCALE-OUT PROCESSOR (SOP)

- General-purpose CPU
- Logic 60% of silicon
- 6x bigger cores
- 3-way OoO ARM
- 85% logic, 7x more cores
- Faster instruction supply
Thunder X

- Based on SOP blueprint
- Designed to serve data
- 7x more core than cache
- Optimizes instruction supply
- Ran stock software
- 10x throughput over Xeon
CHASING POINTERS W/ WALKERS
[MICRO’13]

- Traverse data structures (e.g., hash table, B-tree)
- Parallelize pointer chains
- Overlap pointer access across chains

15x better performance/Watt over Xeon
WALKERS IN SOFTWARE [VLDB’16]

Use insights to help CPUs
- 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
- Integrated in SAP HANA [VLDB’18]
POST-MOORE VIRTUAL MEMORY [ISCA’21]  

- Keeps POSIX (VMA) interface to apps
  - Linux, MacOS/iOS, Android
- Eliminates page-based translation in $$
  - Unclogs virtual memory for security, virtualization, accelerators

Higher overhead

Lower overhead

VM Overhead (%)

Cache Hierarchy ($$) Capacity

midgard.epfl.ch
OUTLINE

- Overview
- Post-Moore servers
  - 80's Desktops
  - Specialized CPUs
    - Integrated logic/memory
    - Integrated networks
    - Approximating AI
- Summary
INTEGRATED LOGIC/MEMORY

Memory chip stack w/ nearby logic

- Minimize data movement
- Massive internal bandwidth

Opportunities for algorithm/hardware co-design

[source: AMD]
COST OF MOVING DATA

Data access much more expensive than arithmetic operation
MEMORY B/W BOTTLENECK

100's of GB/s internally

10's of GB/s off-chip BW

Internal DRAM BW presents big opportunity
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 streaming access
CASE STUDY: DB JOIN ON MONDRIAN

Revisiting Sort join:
- Sort join (O(nlogn)) vs. Hash Join (O(n))
- Sort tables and then merge join
- Streaming vs. random access

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

Trade off algorithm complexity for sequential memory accesses
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
OUTLINE

- Overview
- Post-Moore servers
  - 80’s Desktops
  - Specialized CPUs
  - Integrated logic/memory
- Integrated networks
- Approximating AI
- 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
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

300ns round-trip latency to remote memory
- Wire time and protocol stacks have shrunk
- RPC dominates CPU cycles in μServices
- E.g., data transformation @ ~2.4Gbps w/ Thrift on Xeon
RPC processing at line rate:

- A "schema" (not instructions) interface to an RPC core
- Implements load balancing/affinity scheduling for μServices
OUTLINE

- Overview
- Post-Moore servers
  - 80’s Desktops
  - Specialized CPUs
  - Integrated logic/memory
  - Integrated networks
- Approximating AI
- Summary
COST OF LOGIC VS. MEMORY

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8b Add</td>
<td>0.03</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>16b Add</td>
<td>0.05</td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>32b Add</td>
<td>0.1</td>
<td></td>
<td>137</td>
</tr>
<tr>
<td>16b FP Add</td>
<td>0.4</td>
<td></td>
<td>1360</td>
</tr>
<tr>
<td>32b FP Add</td>
<td>0.9</td>
<td></td>
<td>4184</td>
</tr>
<tr>
<td>8b Mult</td>
<td>0.2</td>
<td></td>
<td>282</td>
</tr>
<tr>
<td>32b Mult</td>
<td>3.1</td>
<td></td>
<td>3495</td>
</tr>
<tr>
<td>16b FP Mult</td>
<td>1.1</td>
<td></td>
<td>1640</td>
</tr>
<tr>
<td>32b FP Mult</td>
<td>3.7</td>
<td></td>
<td>7700</td>
</tr>
<tr>
<td>32b SRAM Read (8kb)</td>
<td>5.0</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>32b DRAM Read</td>
<td>640</td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>

[src: Gholami, et. al.]
DNN PLATFORM DIVERGENCE

Inference platforms:
• Tight latency constraints
• Ubiquitous deployment
• Relies on fixed-point arithmetic

Training platforms:
• Throughput optimized
• Server deployment
• Requires floating-point arithmetic
FLOATING VS. FIXED POINT

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

- **Fixed point**
  - Mantissa
  - Narrow representable range
  - Values range pre-determined
HYBRID BLOCK FLOATING POINT (HBFP) [NeurIPS’18]

1. Block floating point (BFP): one exponent/tensor
   - Low magnitude variation in tensor products
   - > 90% of all arithmetic operations

2. FP32 for all activations
   - High magnitude variation in gradient updates

Co-Located Training & Inference (ColTraIn)
- One accelerator for training and inference
- Eliminates quantization
- Enables online learning

∀ Block of Mantissas
∀ Exponent

Representable Range

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

parsa.epfl.ch/coltrain
Resnet-50 on ImageNet

<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 for CNN, LSTM, BERT
SUMMARY

Trends:
- Demand is growing faster than Moore
- Moore's law is slowing down

Post-Moore servers:
- Revisit legacy abstractions, SW/HW interfaces
- Holistic algorithm/SW/HW co-design
- Division of control vs. data plane

Integration + Specialization + Approximation
THANKYOU!

For more information, please visit us at parsa.epfl.ch