WHAT’S HOT IN COMPUTER ARCHITECTURE? DATACENTERS (LITERALLY)

Babak Falsafi

PARALLEL SYSTEMS ARCHITECTURE LAB

parsa.epfl.ch

EPFL

OUR DIGITAL UNIVERSE

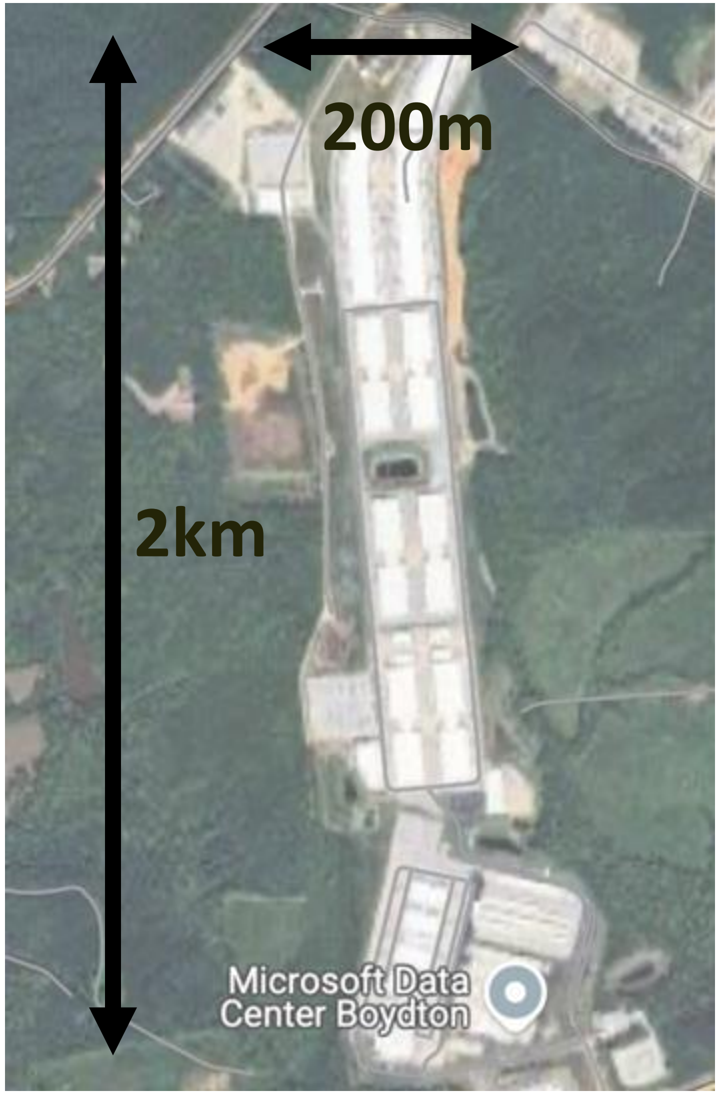
Fueled by:

- Data volume
- Data growth rate
- Monetization of data
- Data’s impact on GDP
- ….now AI

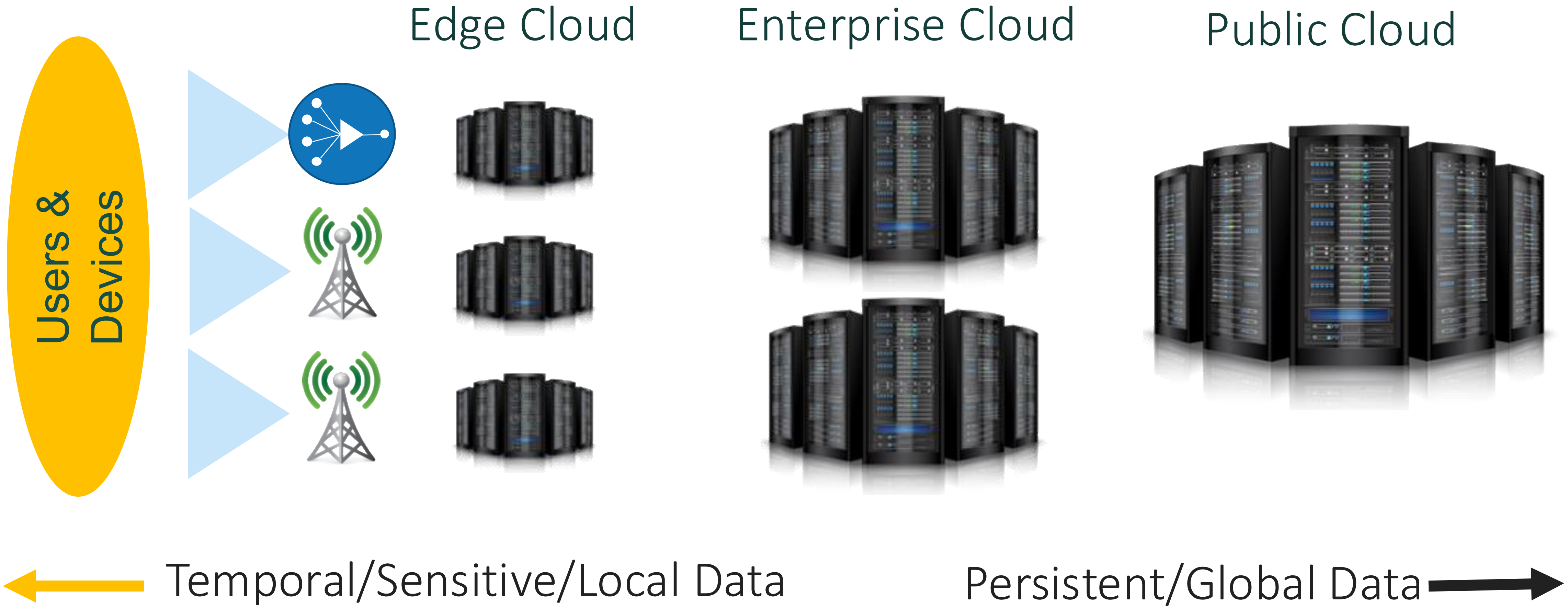
DATACENTERS ARE BACKBONE OF CLOUD

- 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

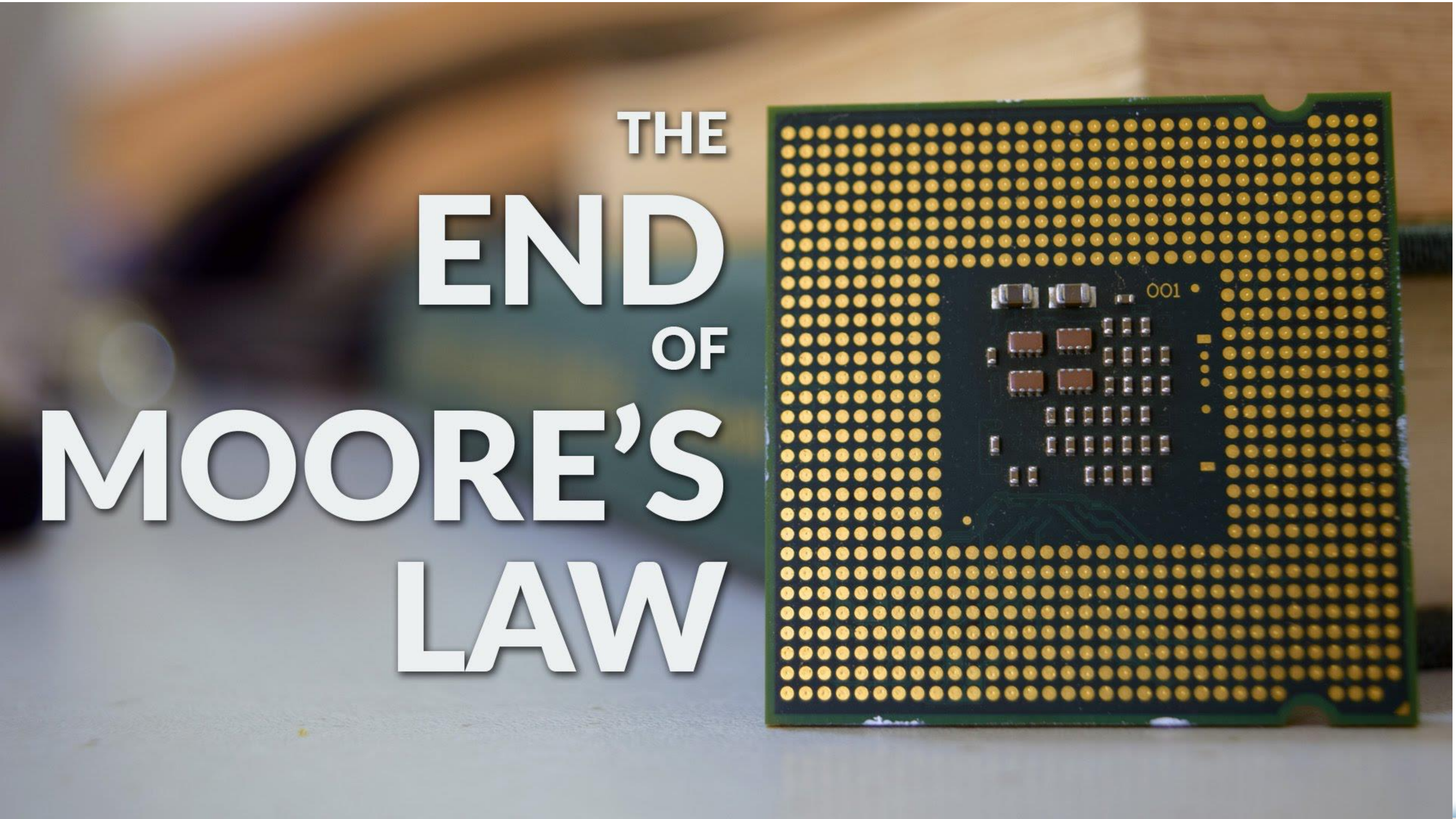
Boydton, VA (300 MW)



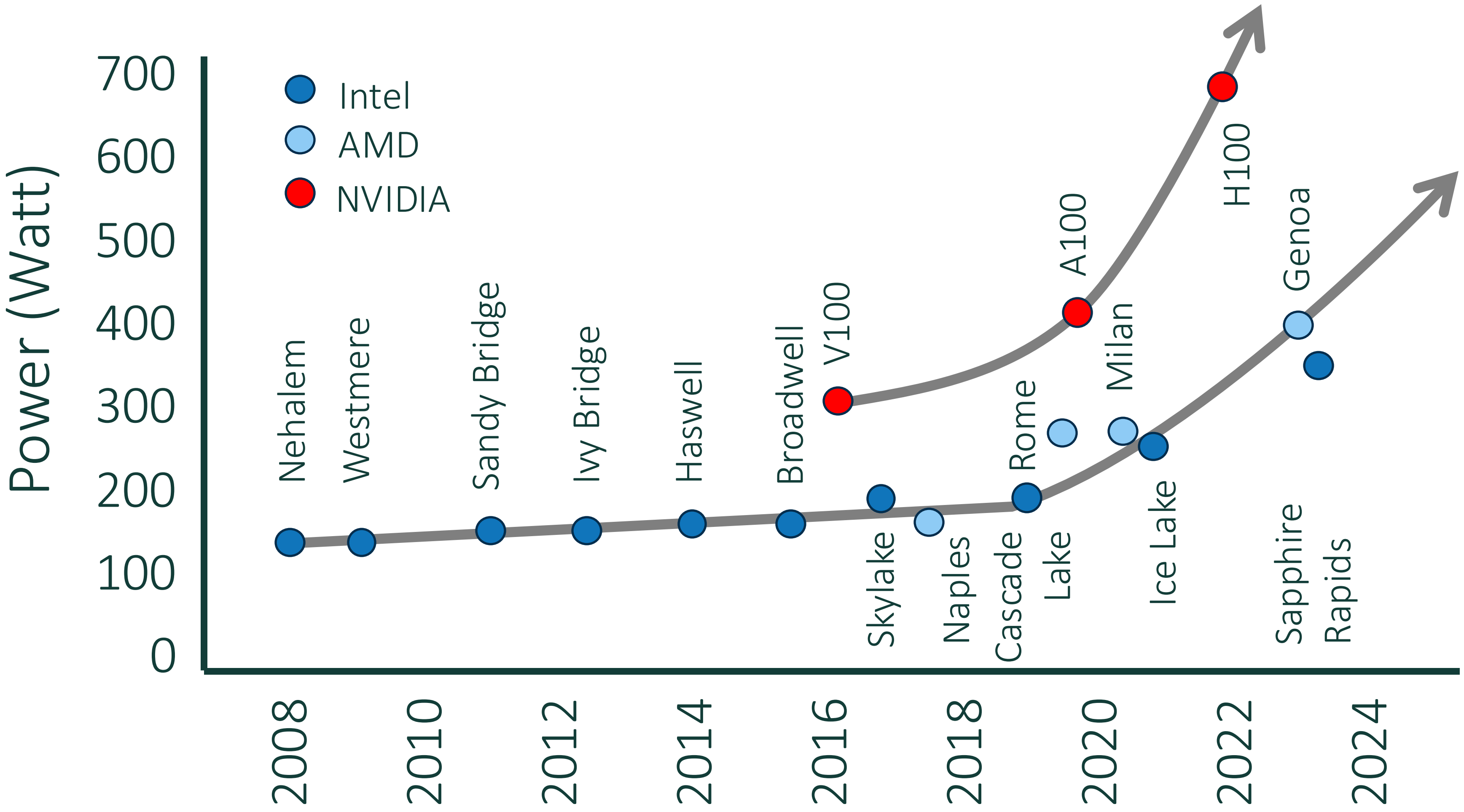
CLOUDS AT VARIOUS SCALES



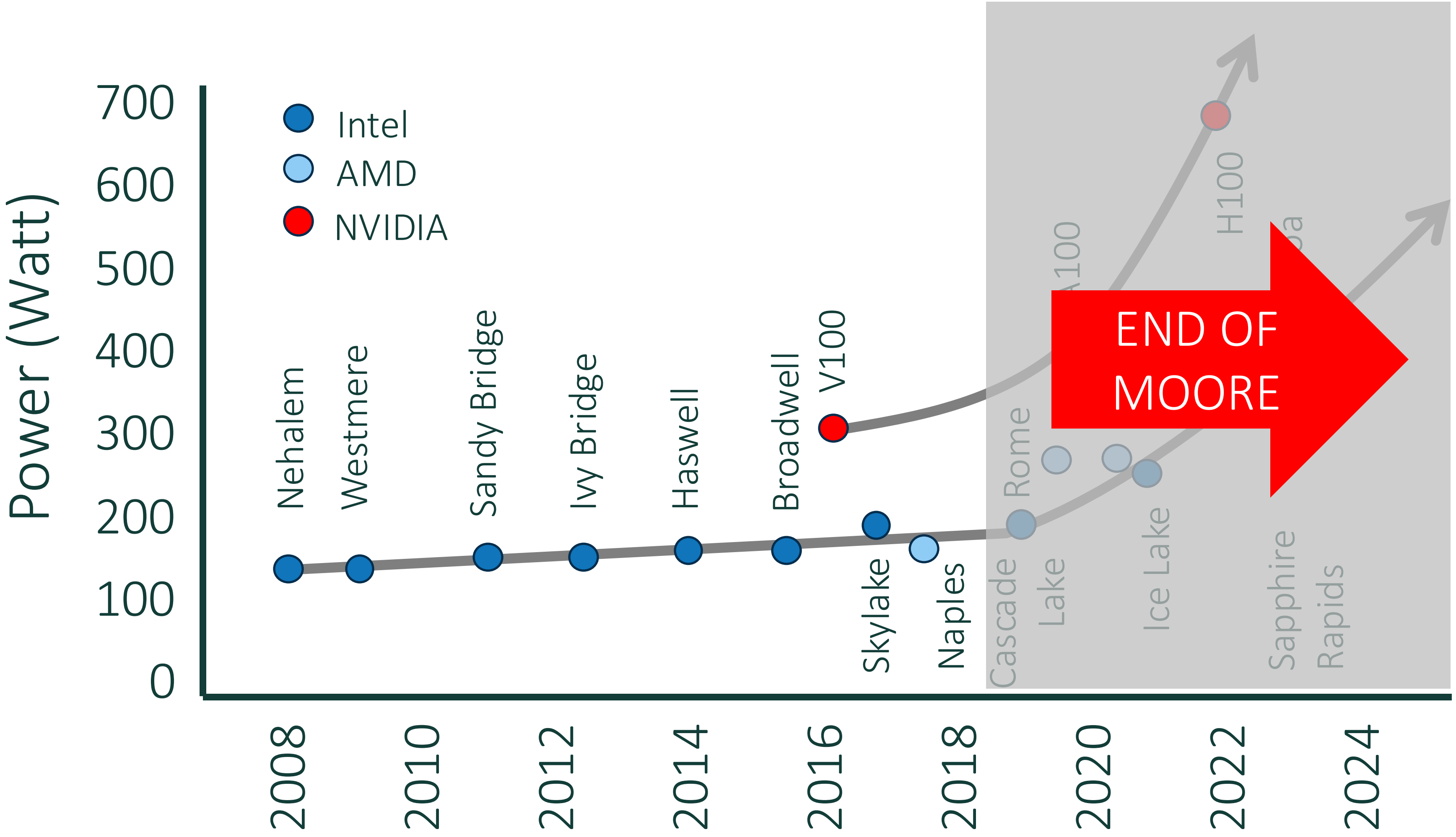
LONG LIVE MOORE'S LAW



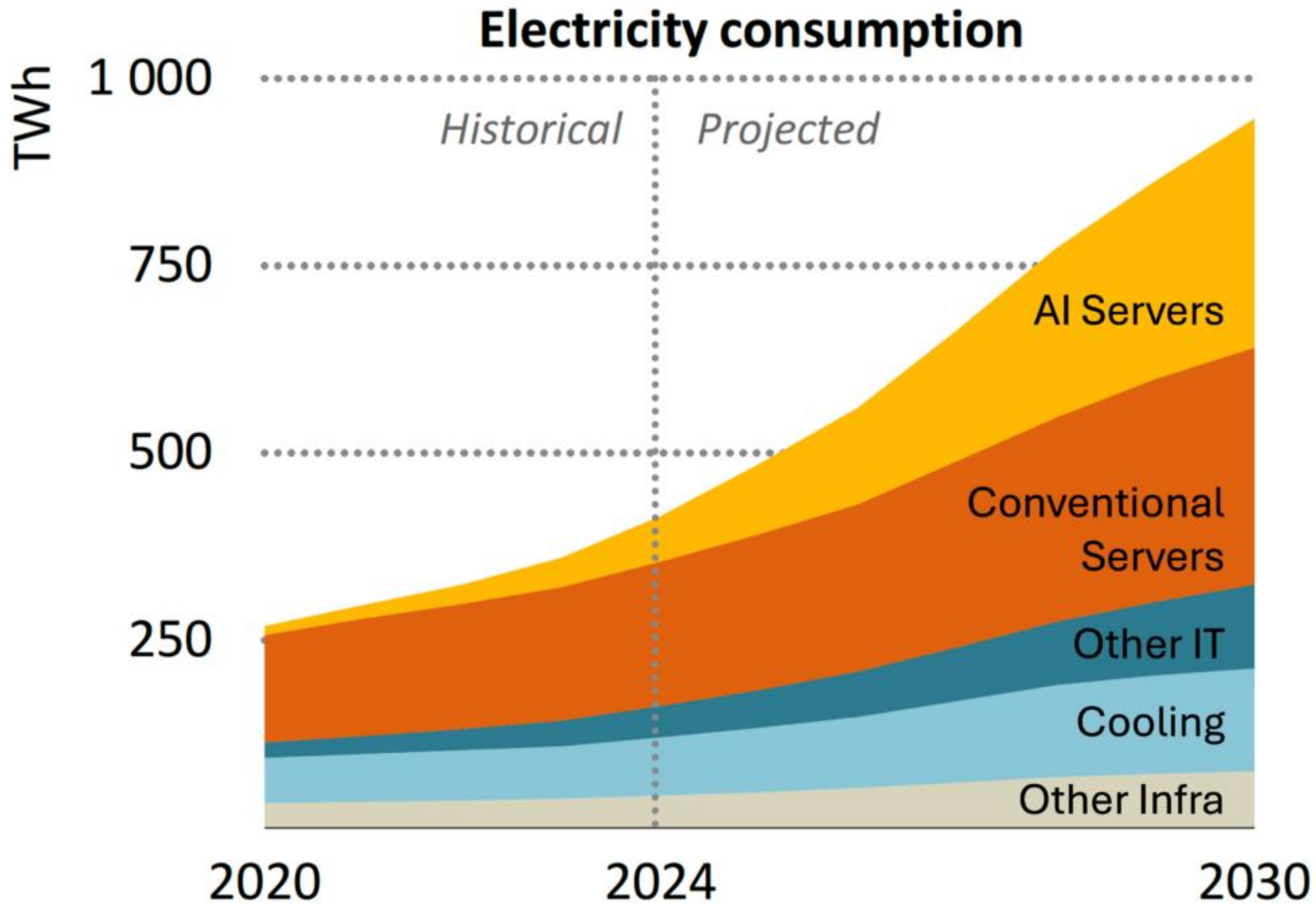
BUILDING BIGGER & FASTER CHIPS



BUILDING BIGGER & FASTER CHIPS



ENERGY GROWTH PROJECTIONS



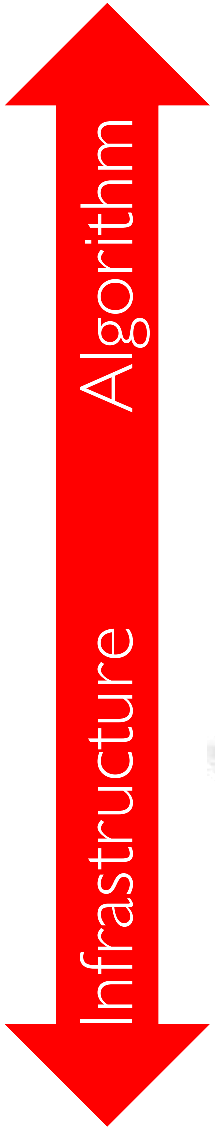
IEA. CC BY 4.0.

POST-MOORE DATACENTERS

Design for “ISA”

- Integration
  - reduce data movement
- Specialization
  - cut resources to analyze data
- Approximation
  - compress data & computation

From algorithms to infrastructure



EcoCloud

CENTER AT EPFL SINCE 2011

Mission

- Sustainable computing
- Best practices, metrics & methodologies

Impact

- Server-grade ARM CPU
- Cloud-native network/database stacks
- Liquid-cooling from chip to rack

IBM

HUAWEI

Hewlett Packard Enterprise

intel

IMO Ventures

GEMATEG

infomaniak

ORACLE

EPFL

ecocloud.ch

METRICS & METHODLOGIES

DC INFRASTRUCTURE EFFICIENCY

- electricity w/ renewables, cooling, heat recycling

IT INFRASTRUCTURE EFFICIENCY

+ compute, storage, network and workloads

DC CARBON FOOTPRINT

+ emissions from input electricity sources

DATA CENTER

EFFICIENCY

Scan the code to find about our label

EPFL EcoCloud Hewlett Packard Enterprise asut SWISS DATACENTER ASSOCIATION energieschweiz HOCHSCHULE LUZERN

OUTLINE

- ~~Overview~~
- Post-Moore servers
  - Today’s servers
  - ISA opportunities
- Wrapping up

SCALE-OUT DATACETNERS

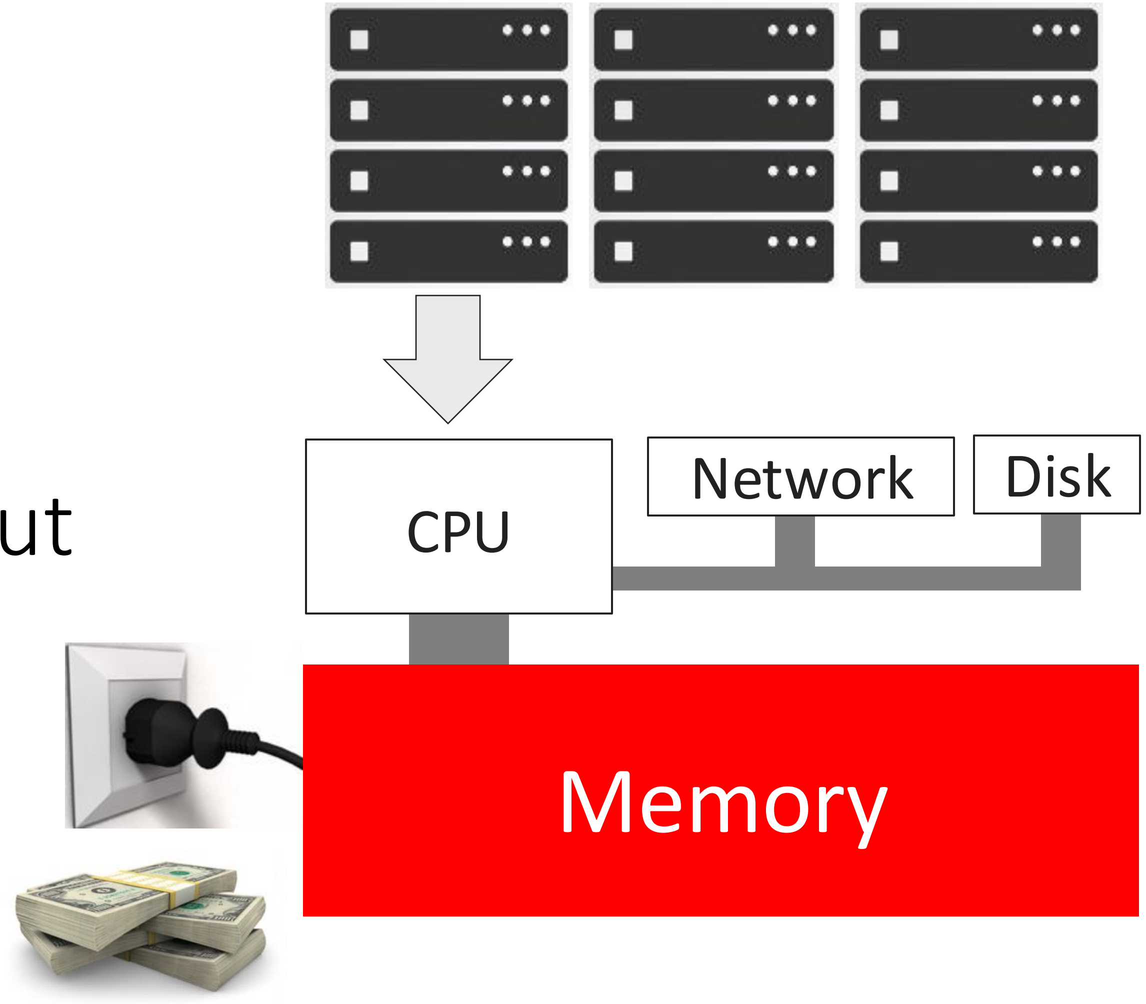
Cost is the primary metric (~50%)

Online services hosted in memory

Divide data up across servers

Design server for low cost, scale out

☞ Memory most precious silicon



CPU

Network

Disk

Memory

TODAY’S SERVERS

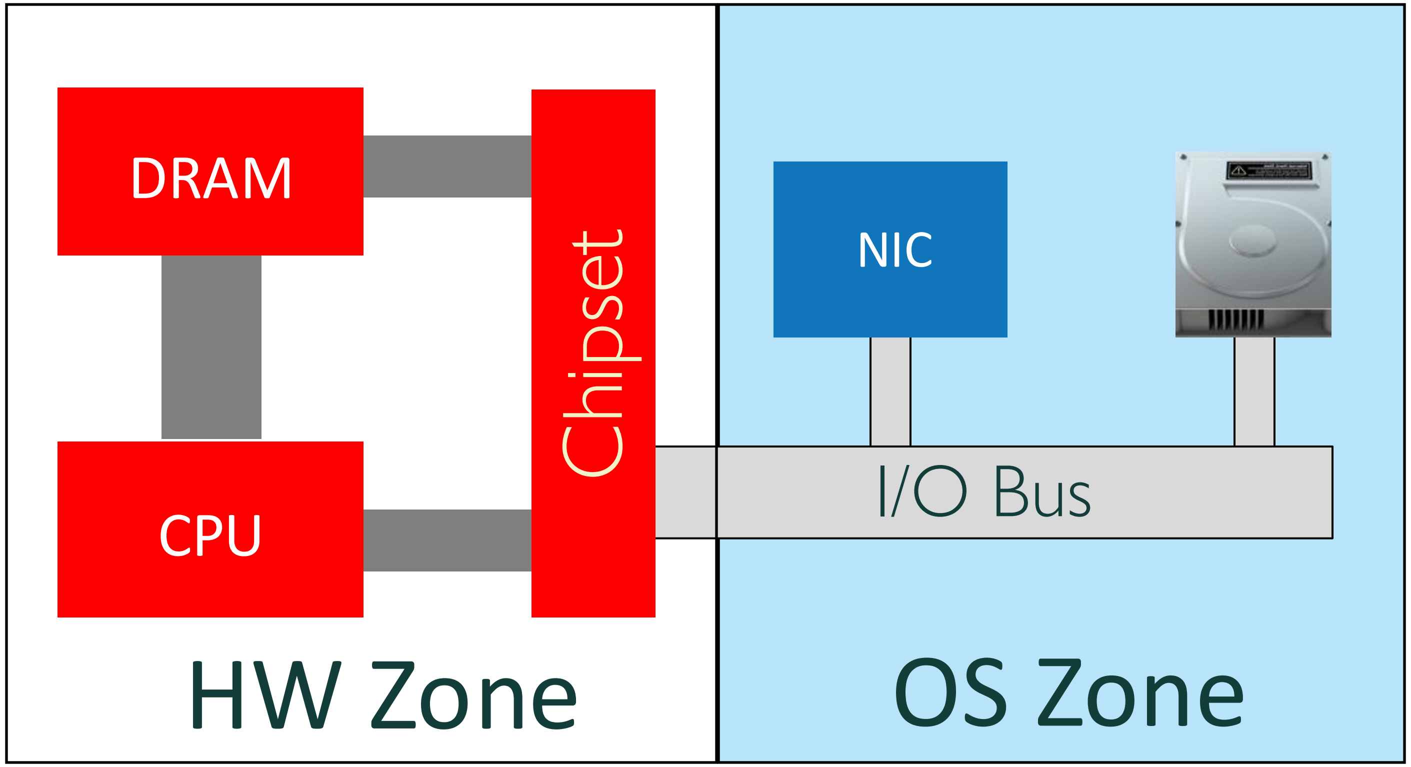
Today’s platforms are PCs of the 80’s

- 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:

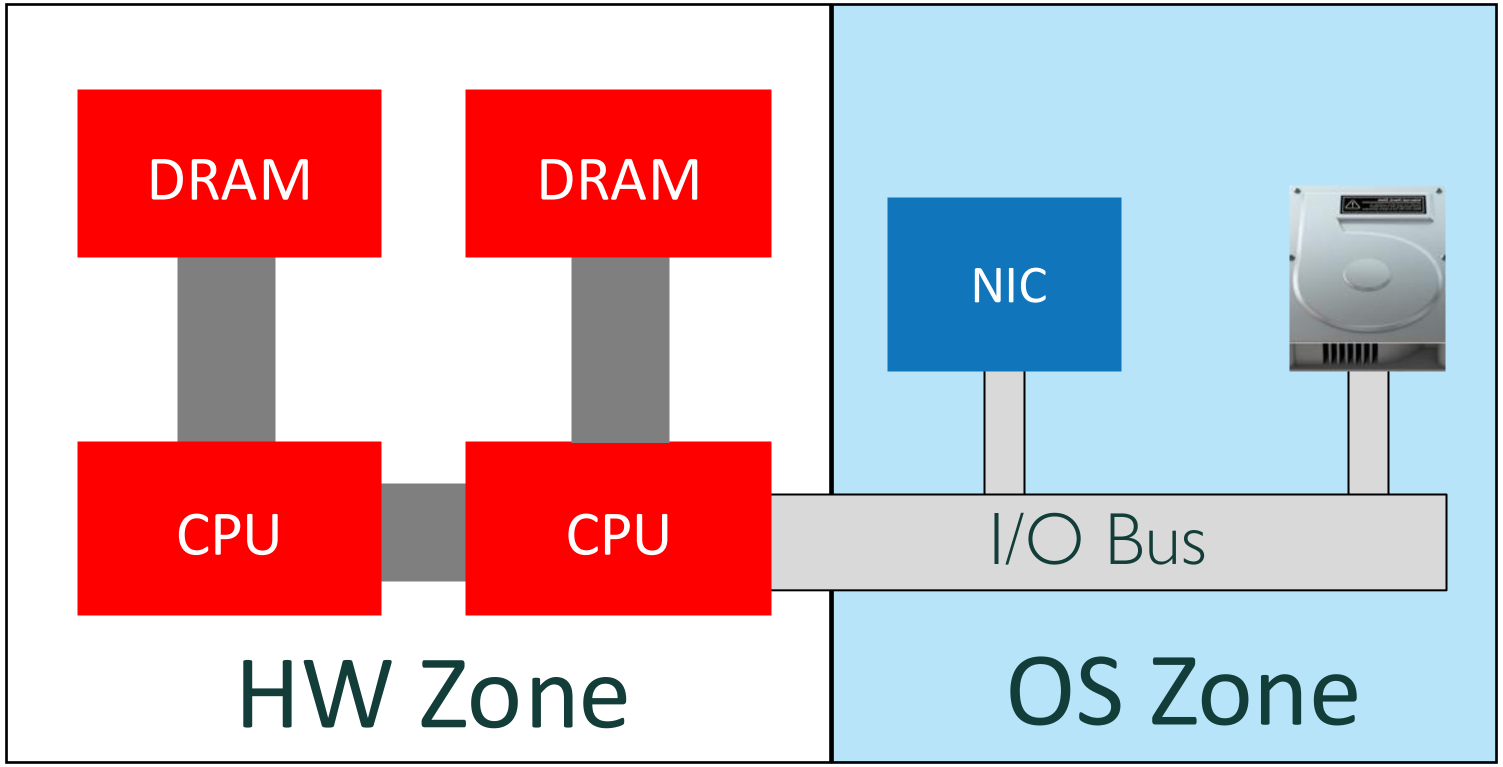
- Manycore network cards w/ own memory
- Flash controllers with embedded cores and memory
- Discrete accelerators with own memory

80S' DESKTOP



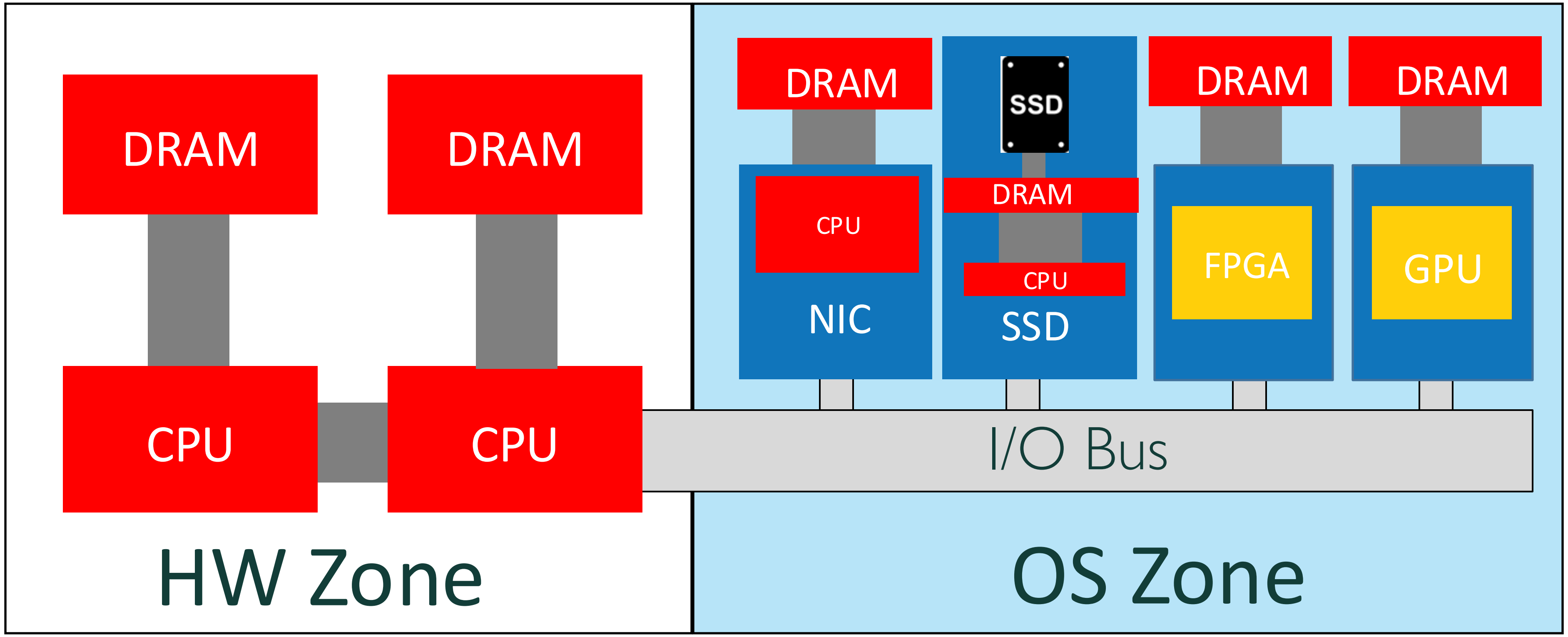
- 33 MHz 386 CPU, 250ns DRAM
- OS: Windows, Unix BSD (or various flavors)
- Focus: mlutiprogrammed in-memory compute

TODAY'S SERVER: 80S' DESKTOP



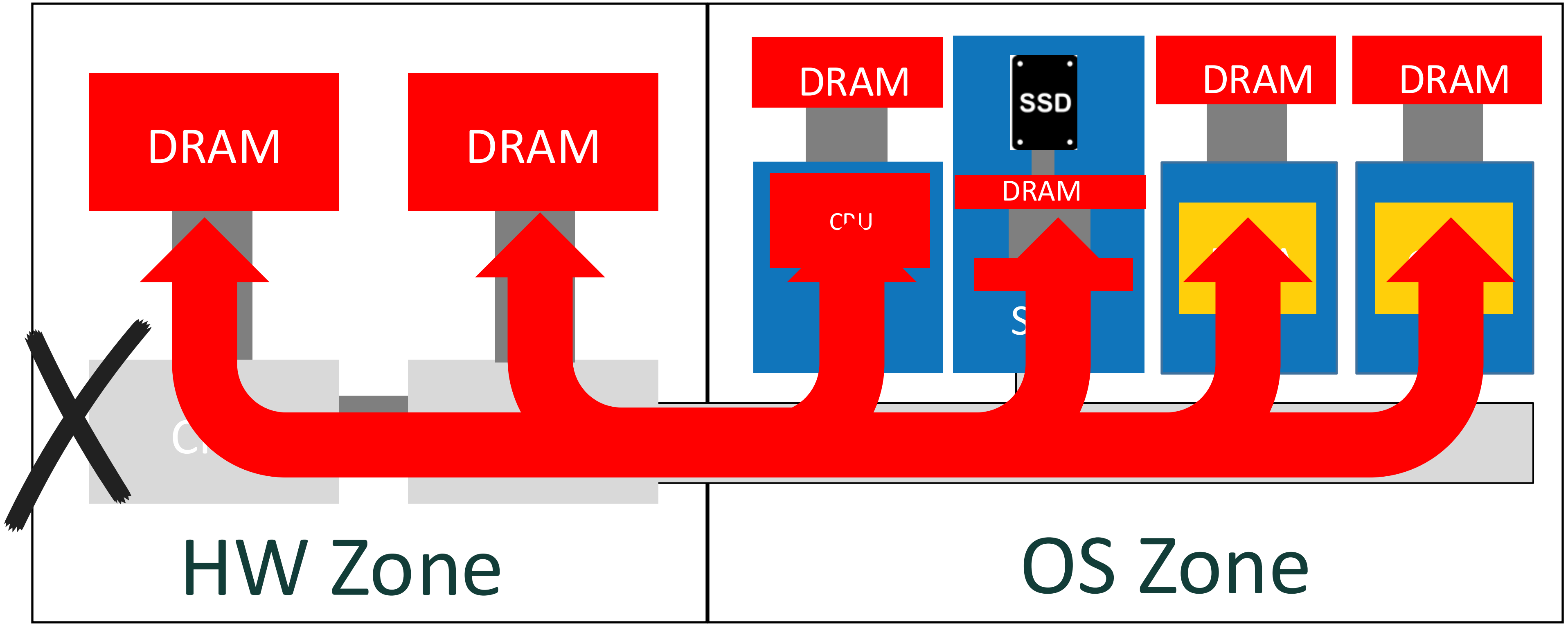
- Dual 2GHz CPU's, 50ns DRAM
- OS: Linux (and various distributions)

TODAY'S SERVER: 80S' DESKTOP



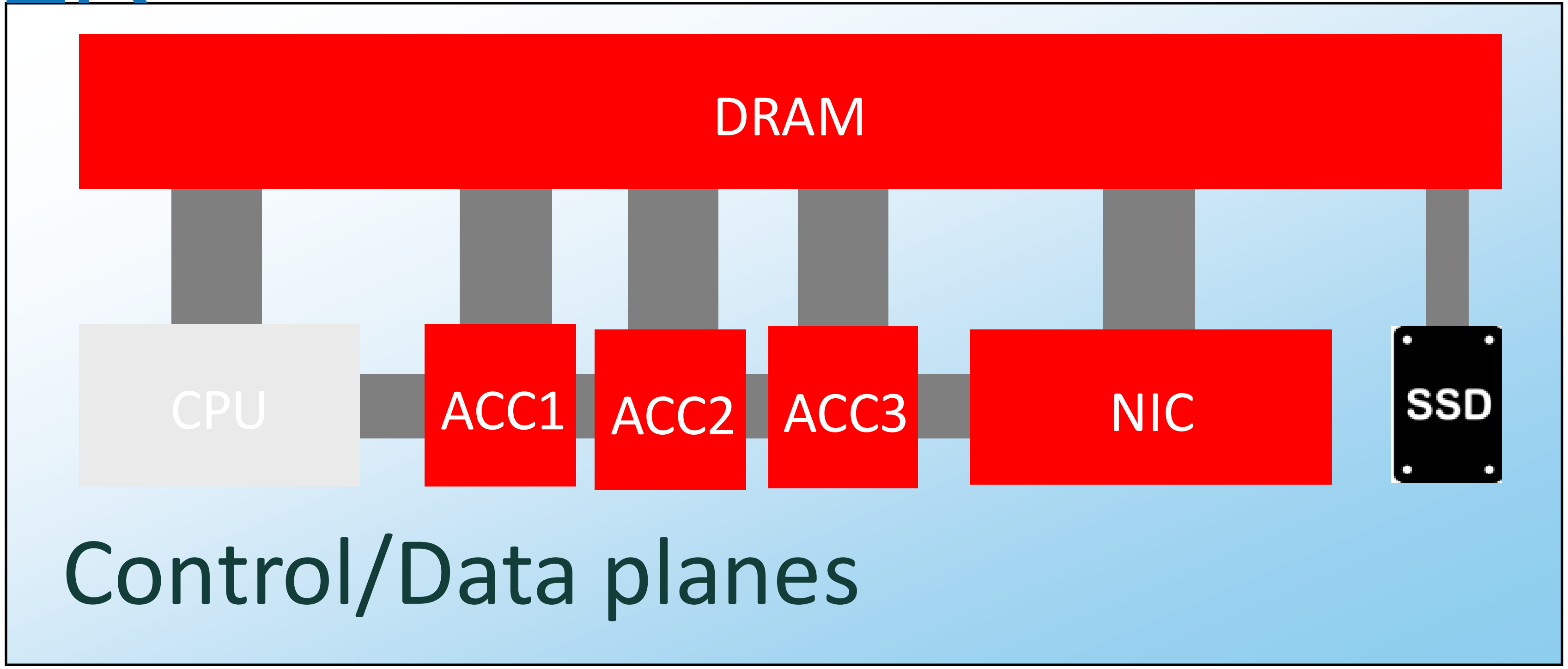
- Dual 2GHz CPU's, 50ns DRAM, Linux
- Bottlenecked by legacy interfaces
- Fragmented silicon

TODAY'S SERVER: 80S' DESKTOP



- Dual 2GHz CPU's, 50ns DRAM, Linux
- Bottlenecked by CPU, OS & legacy interfaces
- Fragmented silicon

POST-MOORE INTEGRATED SERVER

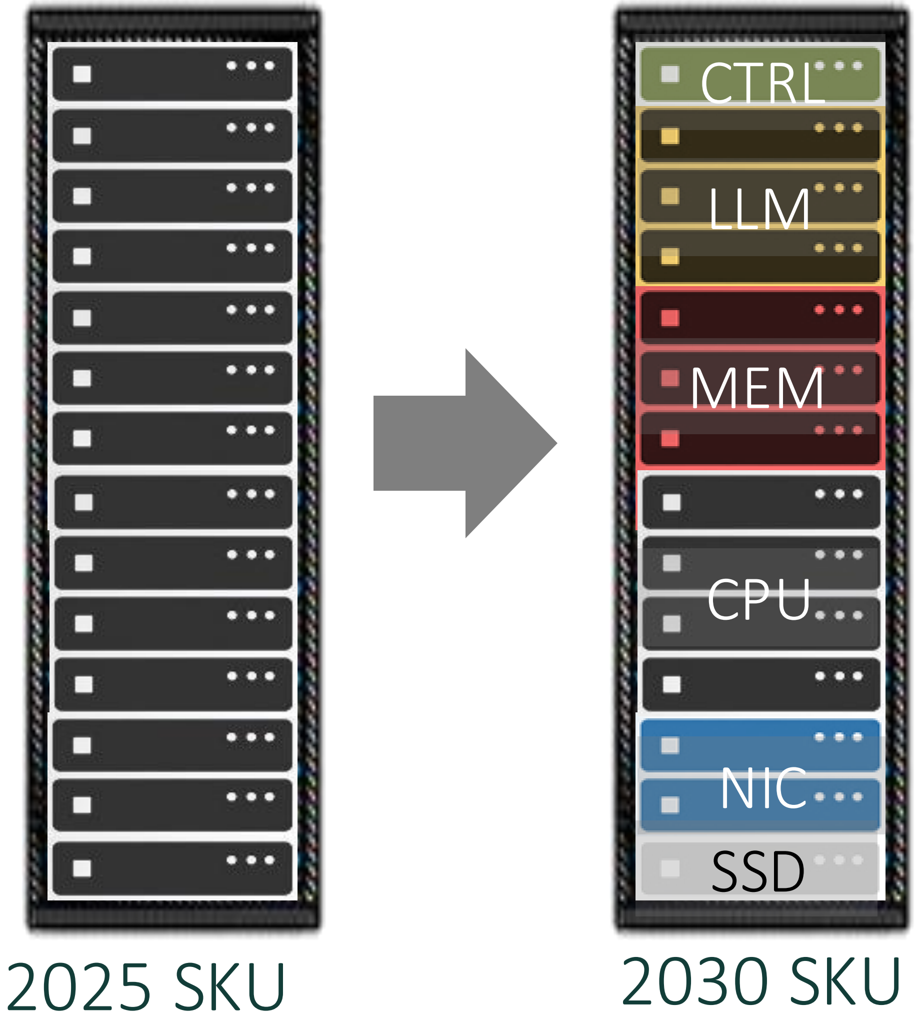


Control/Data planes

- Think of the server as a network
  - Control plane: set up via CPU & OS
  - Data plane: protected access to memory
- Use chiplets to disaggregate at node level

POST-MOORE RACK

- Fabrics at 100s Gb/s/lane
  - E.g., Nvlink, NeuronLink, UB
  - Hardware-terminated protocols
- Disaggregated rack-level hardware
  - Reduces fragmented silicon
- OS services in nanoseconds
  - Return of single-address space OS

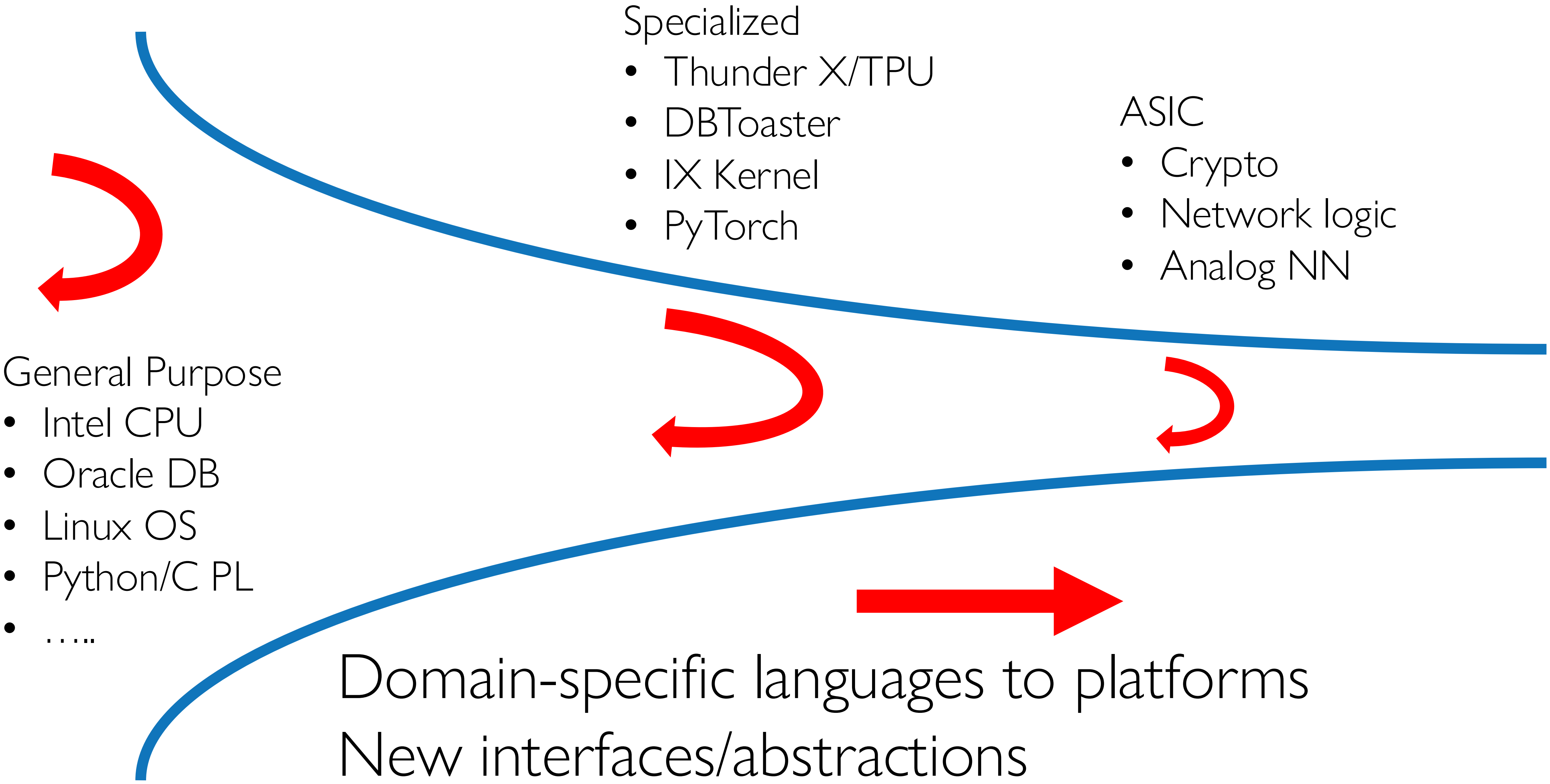


OUTLINE

PARALLEL SYSTEMS ARCHITECTURE LAB

- ~~Overview~~
- Post-Moore servers
  - ~~Today's servers~~
  - ISA opportunities
    - CPU
    - Network
    - Memory
- Wrapping up

THE SPECIALIZATION FUNNEL



Specialized

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

ASIC

- Crypto
- Network logic
- Analog NN

General Purpose

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

Domain-specific languages to platforms

New interfaces/abstractions

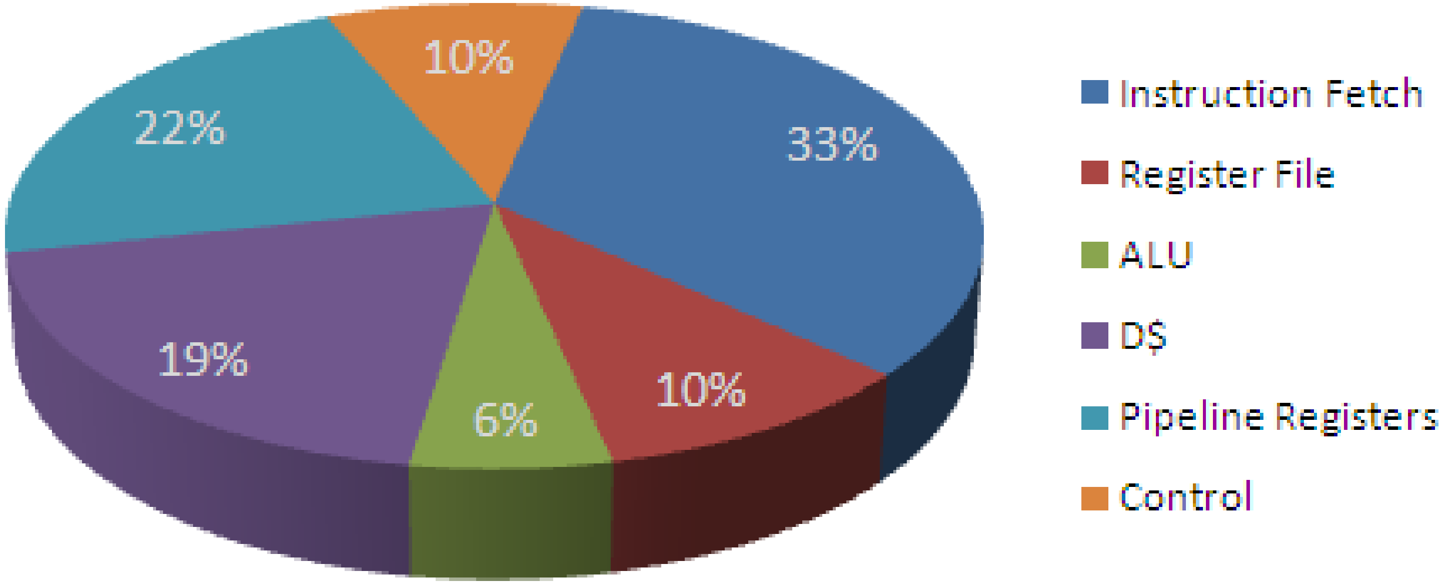
THE LIMITS OF CPUS

PARALLEL SYSTEMS ARCHITECTURE LAB

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]

CPU SPECLIAZATION FOR WORKLOADS

- First-party workloads (e.g., search, retail, media)
  - Data management
  - Analytics
  - Monoliths to microservices

- Third-party workloads (cloud)
  - Containerized
  - Serverless

CloudSuite0 release @ cloudsuite.ch)

PARSA PARALLEL SYSTEMS ARCHITECTURE LAB

Data Analytics
Machine learning

Graph Analytics
GraphX

In-Memory Analytics
Recommendation System

Web Search
Apache Solr & Nutch

Media Streaming
Nginx, HTTP Server

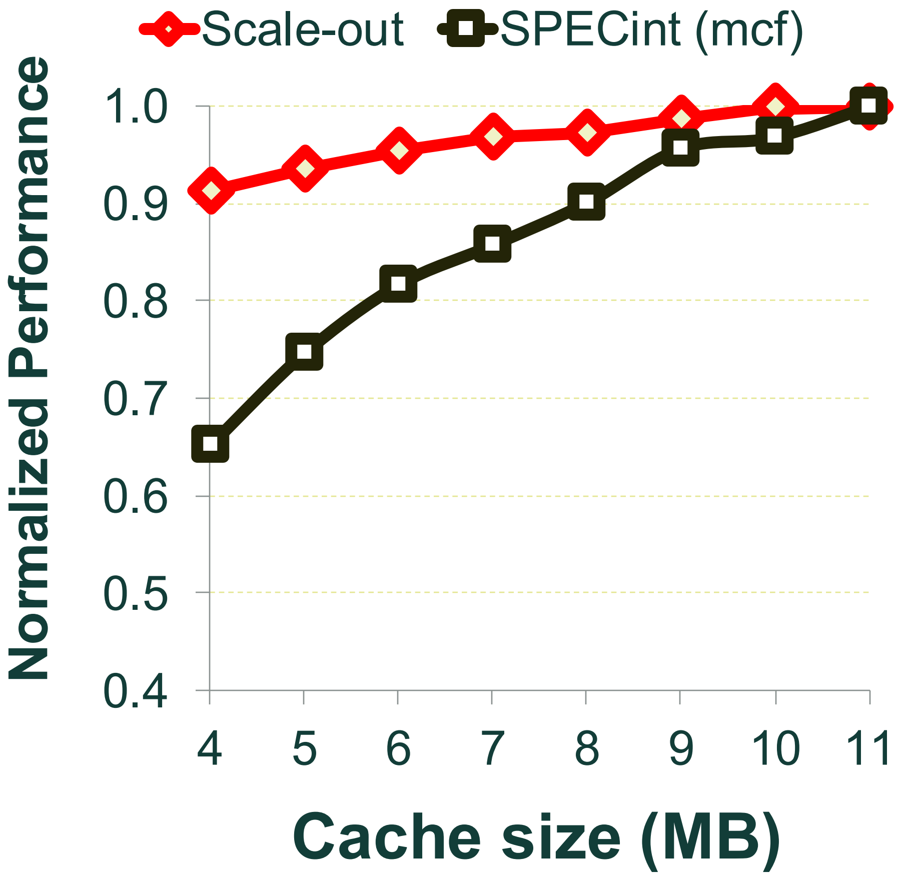
Web Serving
Nginx, PHP server

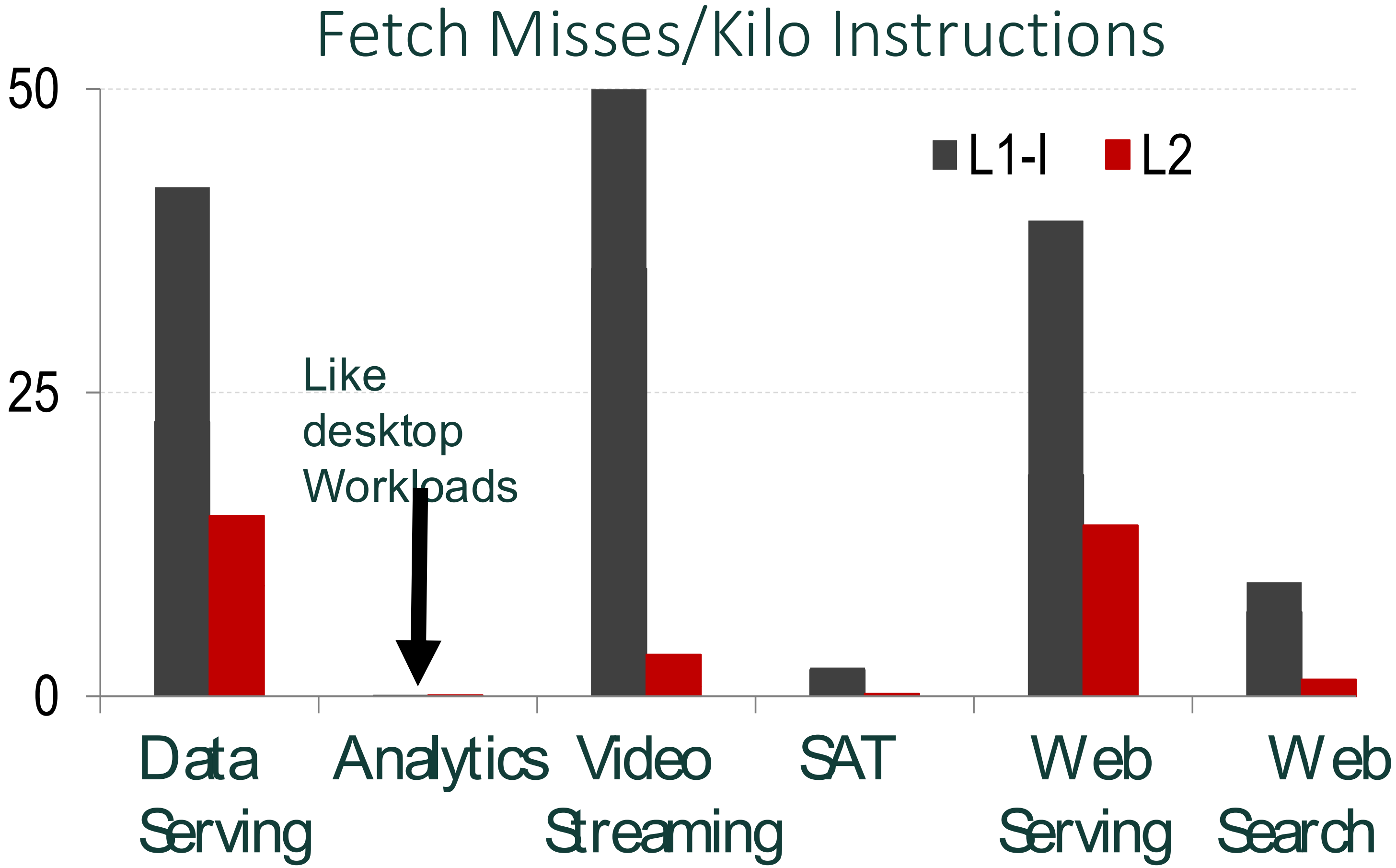
Data Caching
Memcached

Data Serving
Cassandra NoSQL

Supports x86, ARM64, RISC-V (coming)

SERVICES STUCK IN MEMORY [ASPLOS'12]

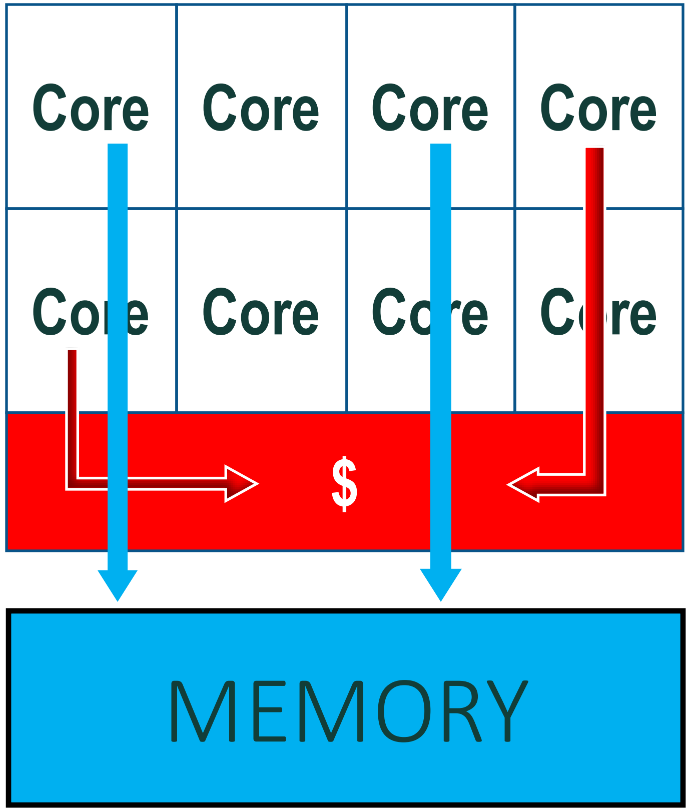


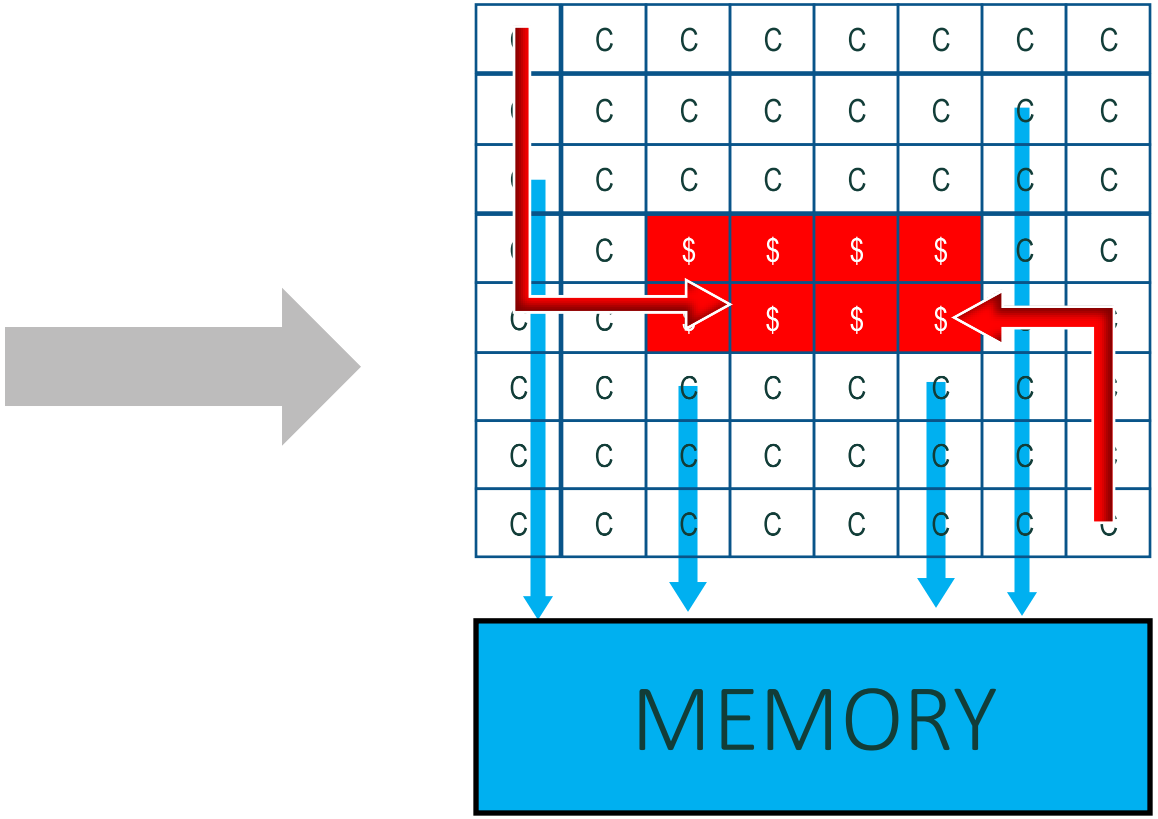


Cache overprovisioned

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

CLOUD-NATIVE CPU 1.0 [c.a. 2014]

CAVIUM

Case for Workload Optimized Processors For Next Generation Data Center & Cloud

Gopal Hegde

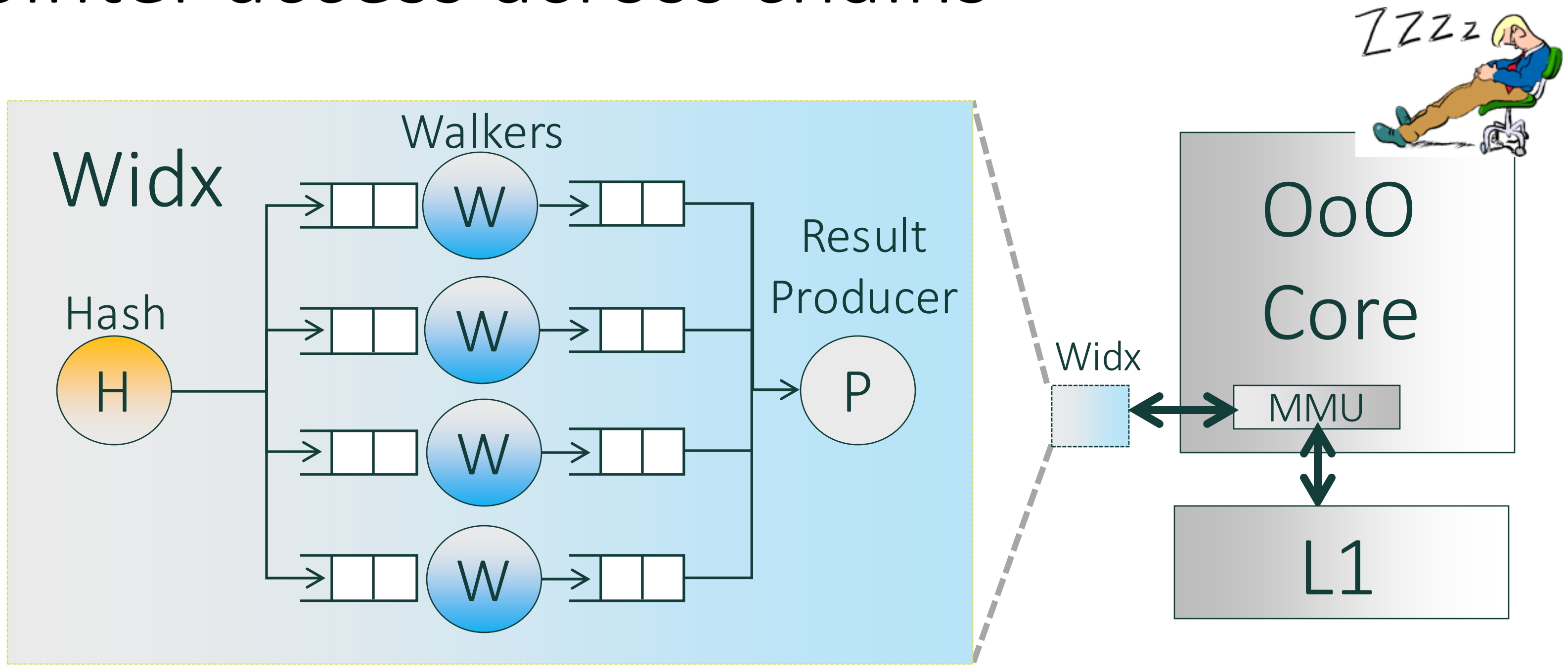
VP/GM, Data Center Processing Group

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

- 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]

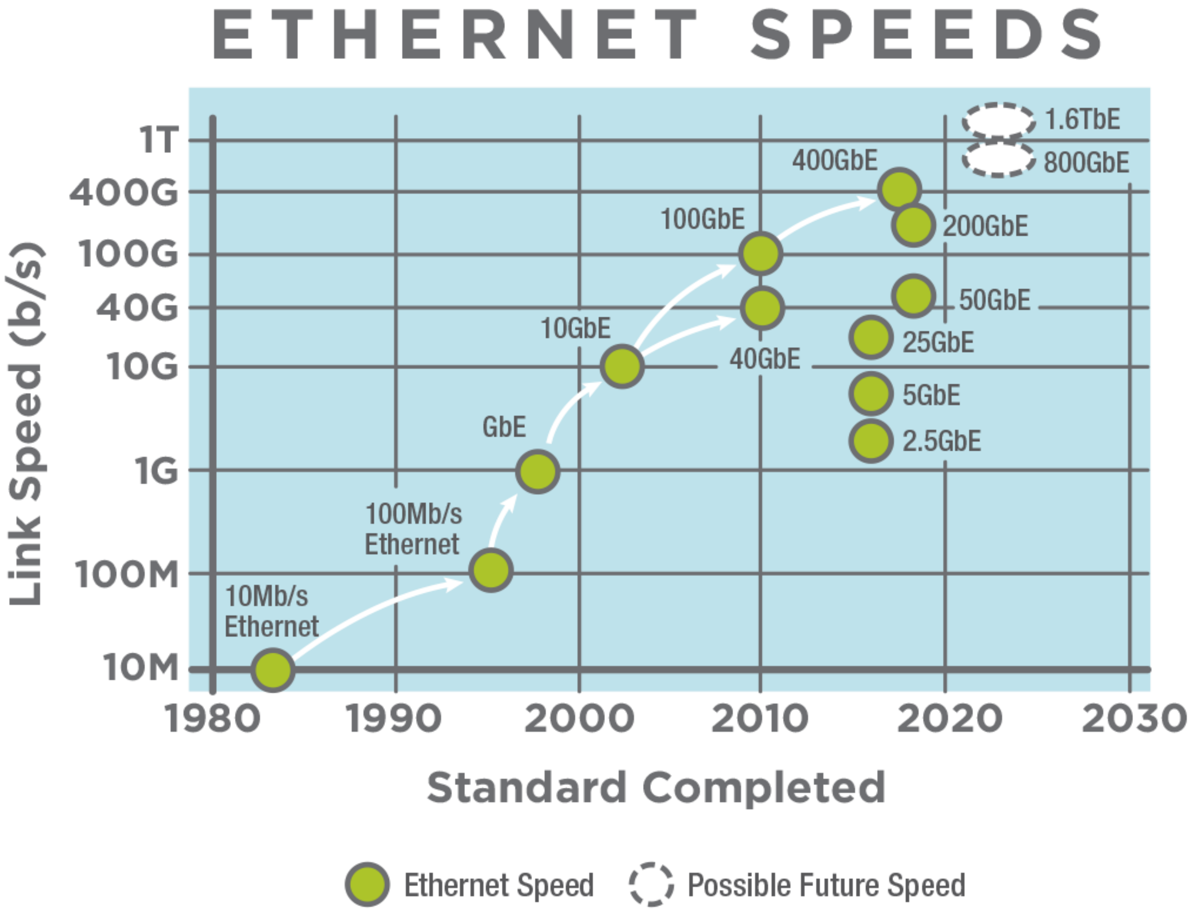
NETWORKS

Network stack bottleneck:

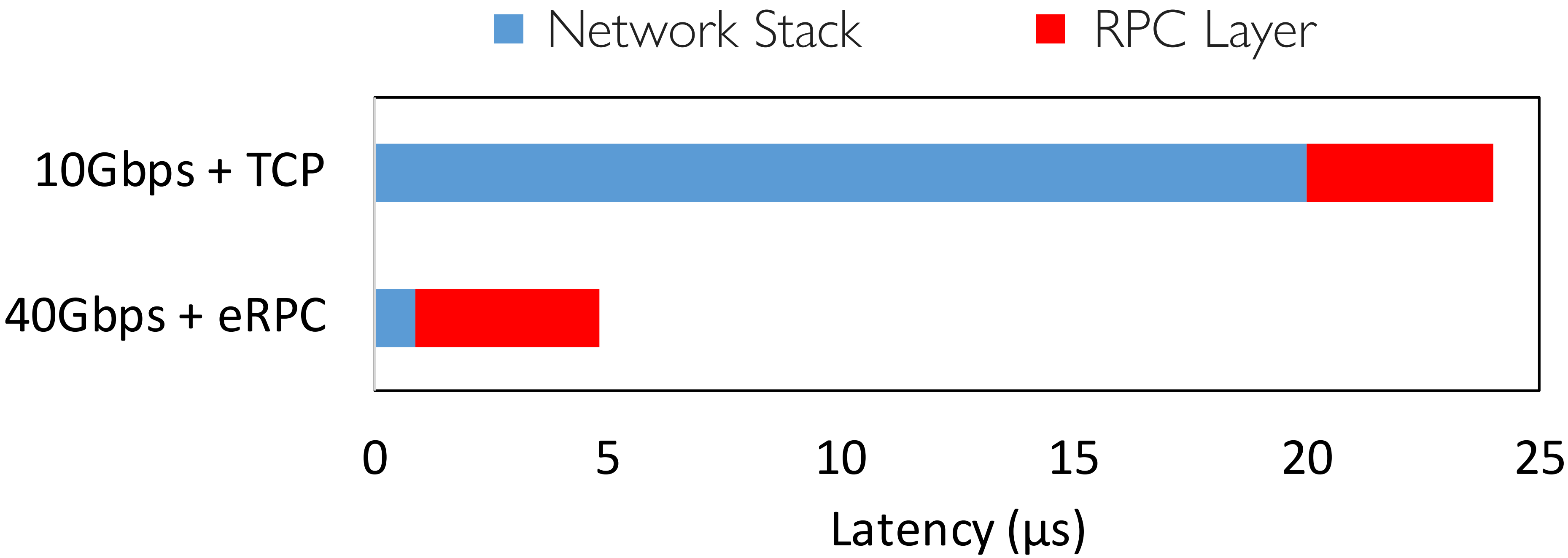
- B/W growing faster than silicon
- Emerging µServices + serverless
- RPC, orchestration, ….

Key challenges:

- New abstractions
- Co-design of network stacks



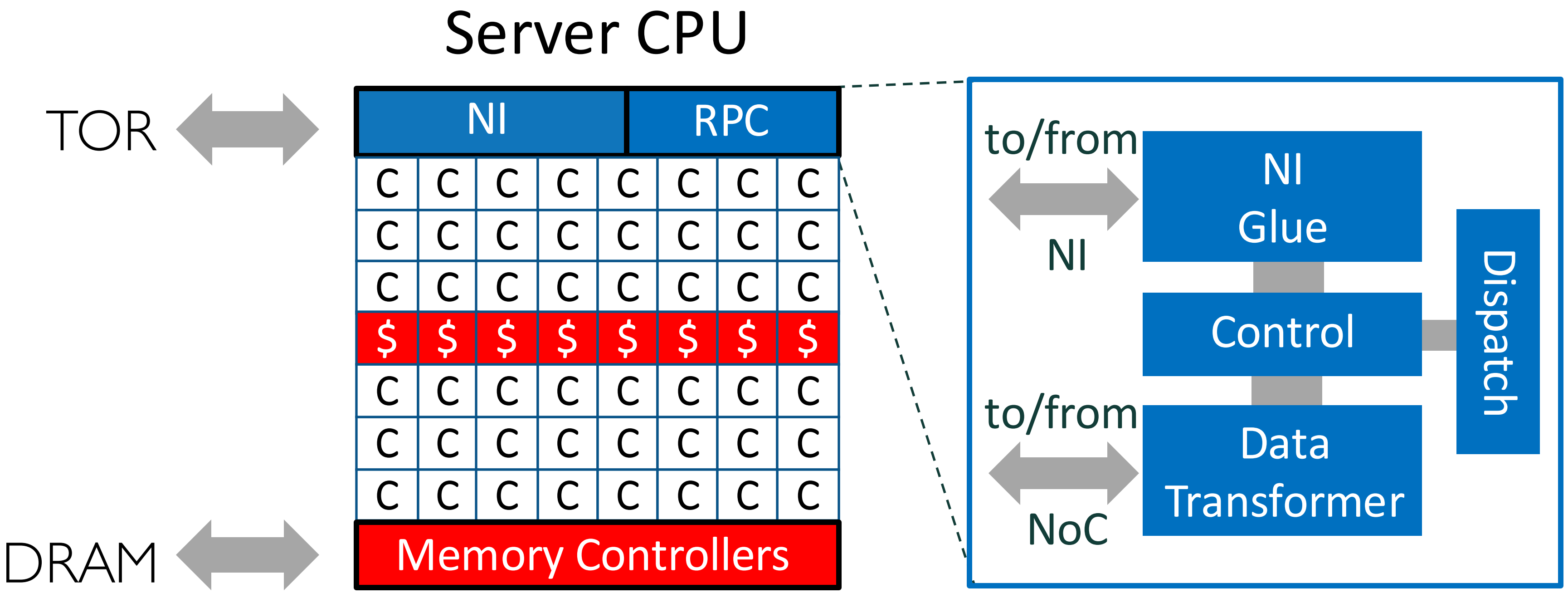
RPC ACCELERATORS



- Wire time and protocol stacks have shrunk
- RPC dominates CPU cycles in µServices
- E.g., data transformation @ ~2.4Gbps w/ Thrift on Xeon

CEREBROS RPC PROCESSOR

[ISCA’15’19’20,ASPLOS’20,MICRO’21]



Socket-integrated NICs:

- Zero-copy transfers
- Single-queue semantics (RPCValet)
- HW-terminated protocol (NebuLa)

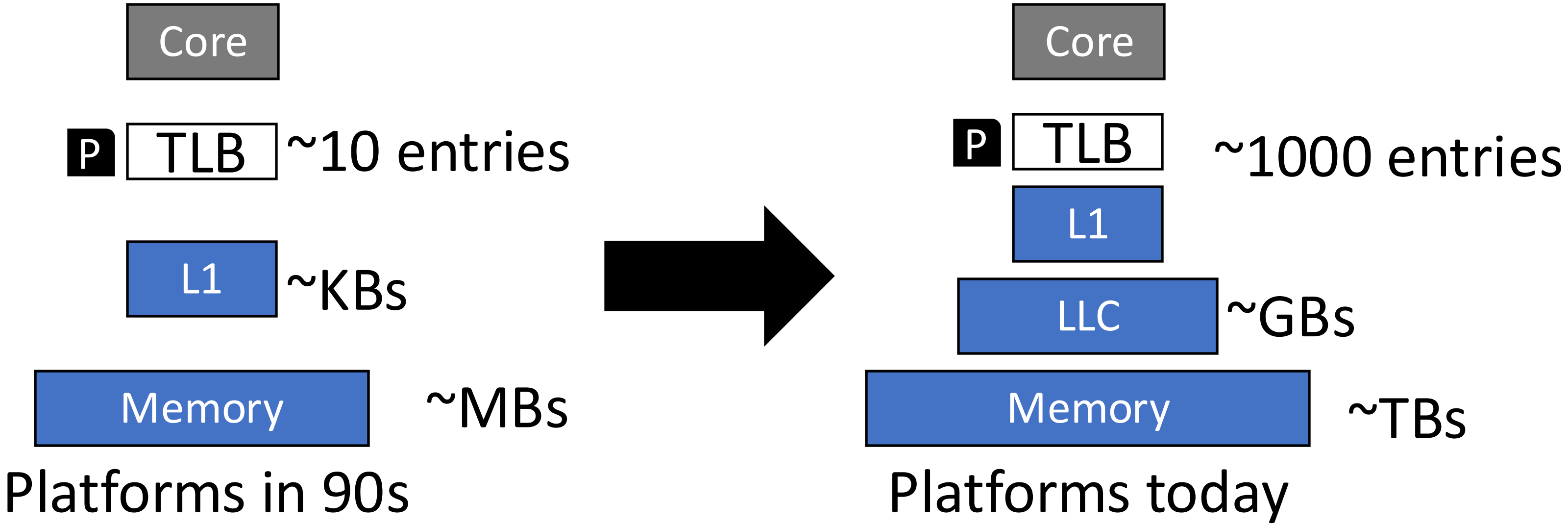
RPC Processor:

- Thrift ”schema” as interface (Optimus Prime)
- Dispatch, load balancing, affinity schedulin
- RPC at line rate = 100 Gbps

OUTLINE

- ~~Overview~~
- Post-Moore servers
  - ~~Today's servers~~
  - ISA opportunities
    - ~~CPU~~
    - ~~Network~~
    - Memory
- Wrapping up

THE VM BOTTLENECK: TLBs



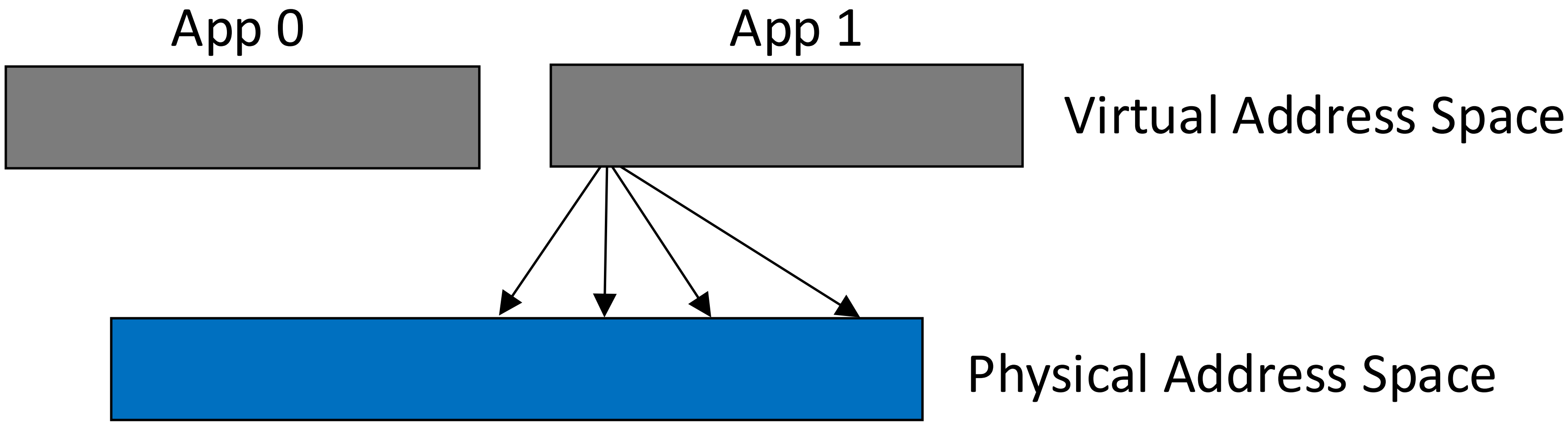
Product	Year	Cores	Cache capacity	TLB entries	Coverage (4KB)
Intel P4	2000	1	256KB SRAM	64	256KB
Intel KabyLake	2016	4	128MB eDRAM	1536	6MB
Apple M1	2020	8 (4+4)	16MB SRAM	3096	12MB (16KB)
AMD Zen3	2021	64 (8x8)	256MB SRAM	2048	8MB
Intel Sapphire Rapids	2022	56 (14x4)	64GB HBM2	?	?

VIRTUAL MEMORY

- Classic programming abstraction
  - Provides process isolation using private address spaces
  - Provides memory management without application involvement
- Ubiquitous in all modern computing devices (servers, desktops, mobile)

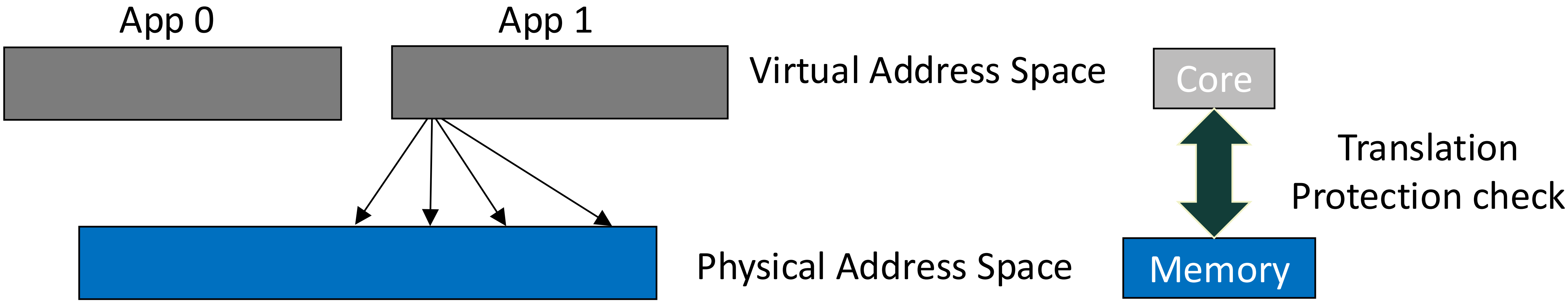
Essential abstraction for programming and memory management

VIRTUAL MEMORY 101 (OS)



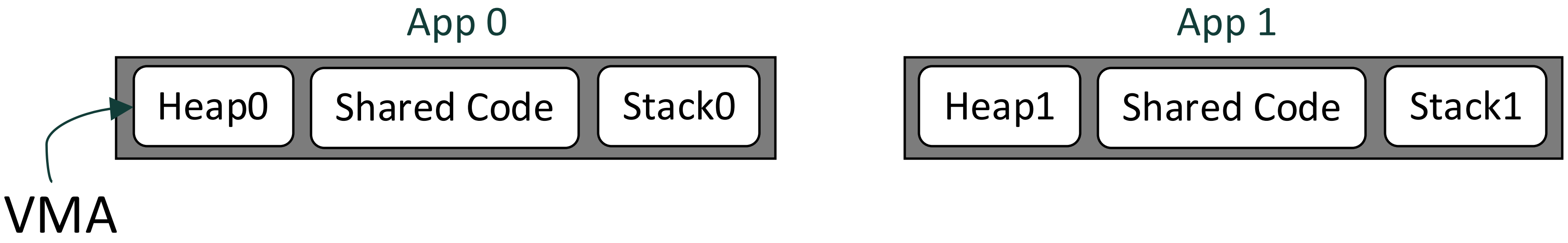
- Operating System (OS) provides
  - Virtual address space for applications
  - Physical address space for memory
  - Mapping of virtual addresses to physical addresses

VIRTUAL MEMORY 101 (HW)



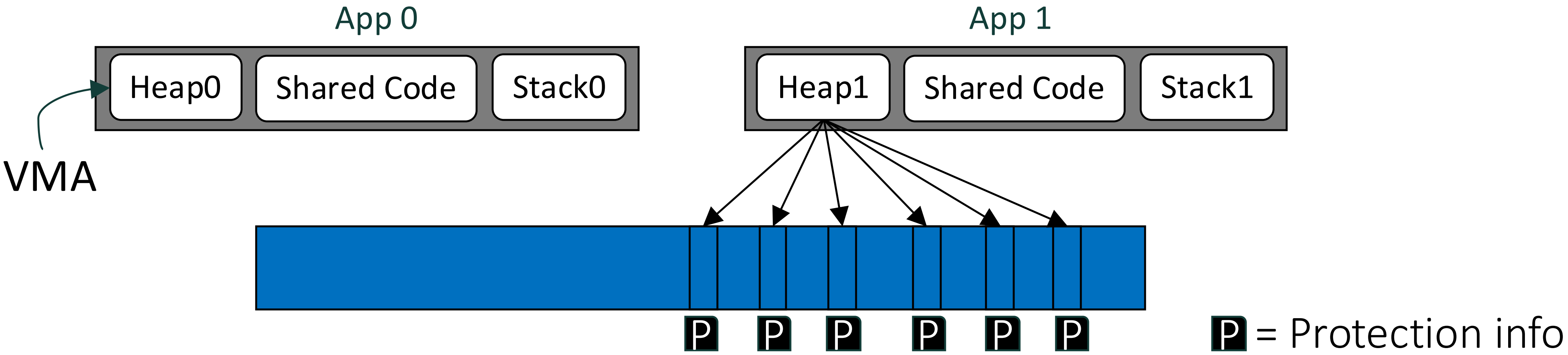
- Architectural support is required for
  - Translating virtual addresses to physical addresses
  - Performing protection checks

HOW ARE ADDRESS SPACES ORGANIZED?



- Virtual address space
  - Organized using Virtual Memory Areas (VMAs)
  - Protection is defined at a VMA granularity

HOW ARE ADDRESS SPACES ORGANIZED?

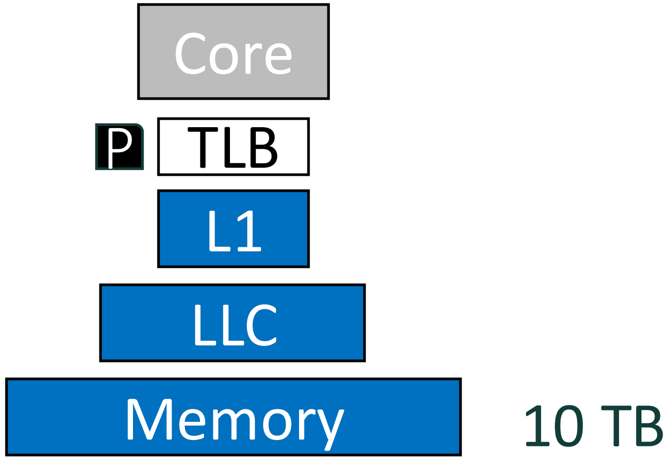


- Physical address space
  - Organized using fixed-size pages for efficient capacity management
  - VMAs are divided and mapped to numerous pages

Protection and translation information is replicated for pages

PROTECTION CHECK IN HARDWARE

- Translation Lookaside Buffer (TLB)
  - Cache mappings for recently used pages
  - Protection check in a fraction of a clock
- TLBs scalability challenge
  - Memory capacity has grown from MBs to ~10 TB
  - Cache hierarchies have grown up to ~10 GB
  - 1000s of TLB entries can reach only MB of coverage
  - End of Moore’s law prohibits further silicon scaling



Platforms today

TLBs cannot provide the required coverage

PRIOR/TODAY'S SOLUTIONS

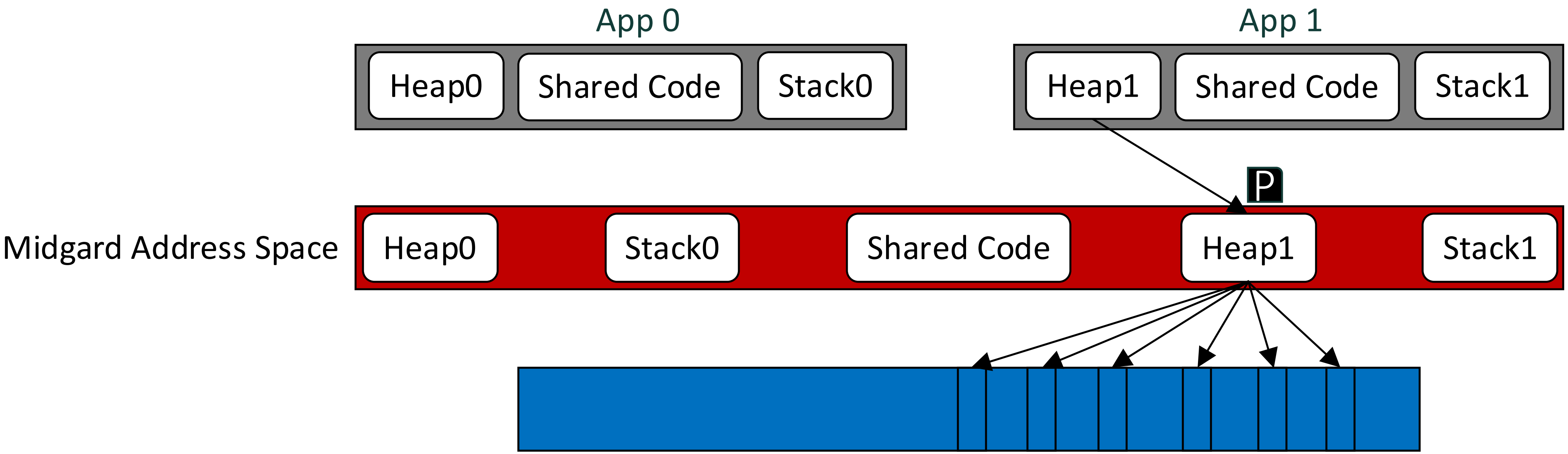
- Contiguity in the physical address space
  - Huge pages
  - Direct segments [Basu, ISCA'13]
  - Memory defragmentation [Yan, ISCA'19]

  ✓ Achieve faster translation/protection

  X Software development/runtime overhead
- Virtual hierarchies
  - In-cache address translation [Wood, ISCA'86] has synonym/homonym problems
  - VBI [Hajinazar, ISCA'20] not software compatible

Previous proposals help, but do not solve the problem

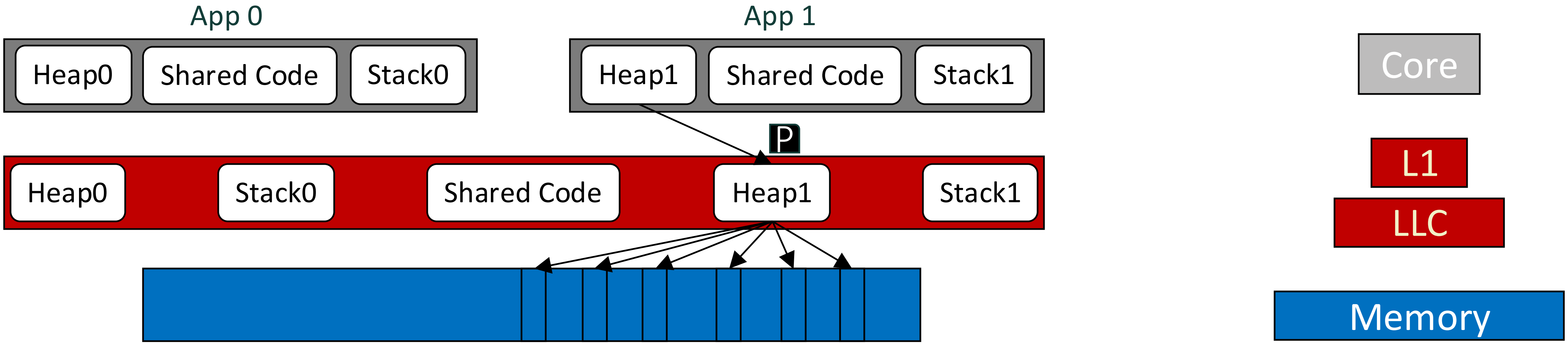
MIDGARD ADDRESS SPACE



- A sparse intermediate address space that retains VMAs
  - Protection check and contiguous translation at VMA granularity
  - OS deduplicates shared VMAs, ensuring no synonyms/homonyms

Midgard provides an address space for the cache hierarchy

MIDGARD-ADDRESSED CACHE HIERARCHY

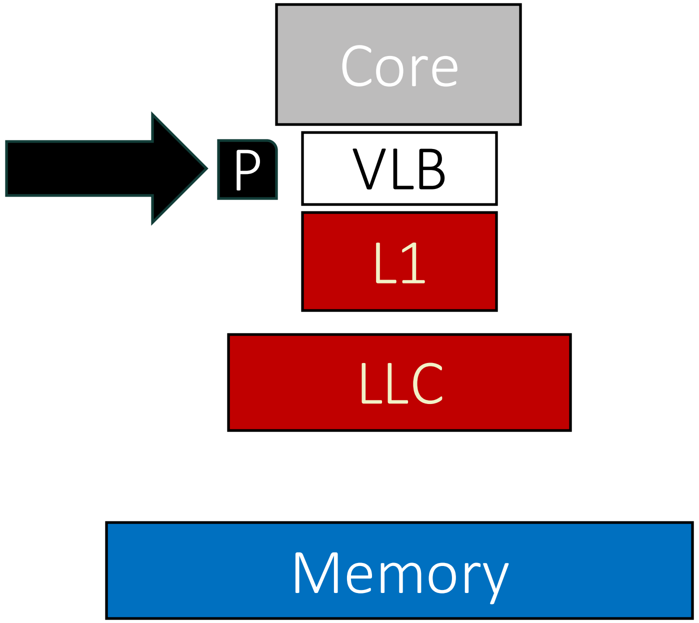


- Cache hierarchy now uses Midgard addresses
  - Virtual to Midgard translation is fast because of VMAs
  - Protection is implemented at a VMA granularity
  - Midgard to Physical translation is only required on cache misses

Midgard optimizes the common-case cache accesses

VIRTUAL-TO-MIDGARD TRANSLATION

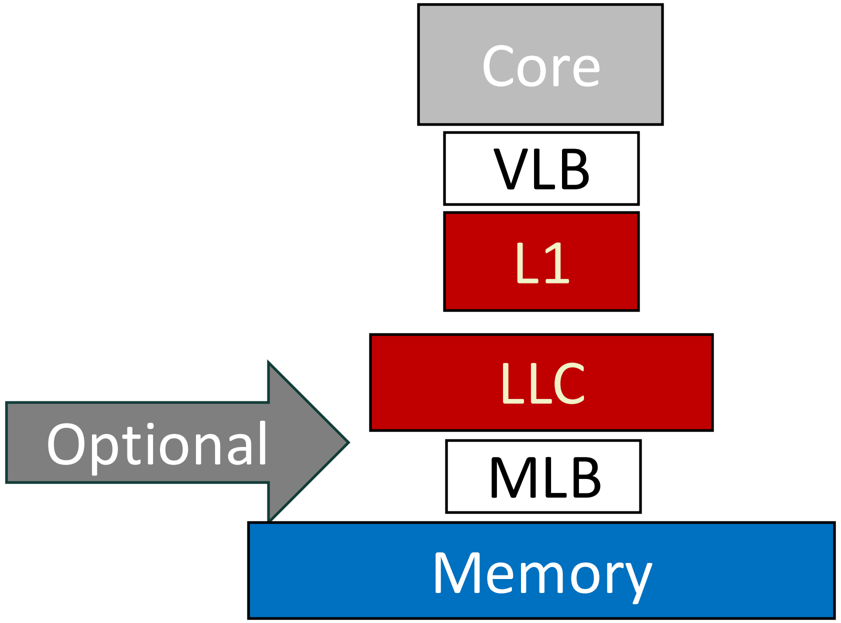
- Translation and protection at VMA granularity
  - Process-private VMA table contains mappings
  - Each process typically contains ~100 VMAs
  - E.g., range tables, B-trees
- Virtual Lookaside Buffer (VLB)
  - Cache VMA mappings to benefit from locality
  - Only ~10 VMAs are frequently accessed



Only ~10 VLB entries required per core

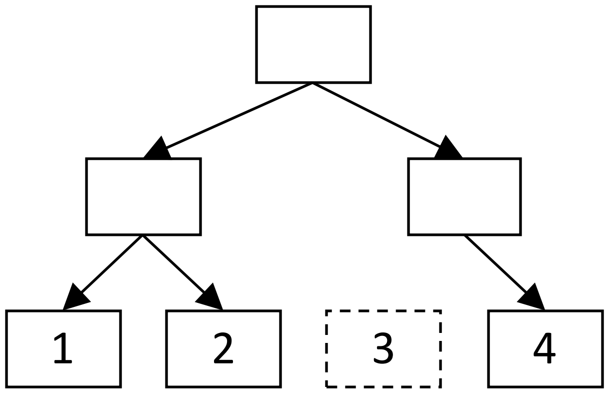
MIDGARD-TO-PHYSICAL TRANSLATION

- Cache hierarchy filters most of the memory accesses
  - Translation required only for cache misses
  - Larger cache hierarchy requires fewer translations
- Translations stored in Midgard page table
  - Shared by all the processes/cores
  - Little temporal locality left in the translation requests
  - Optionally cache in Midgard Lookaside Buffers (MLBs)

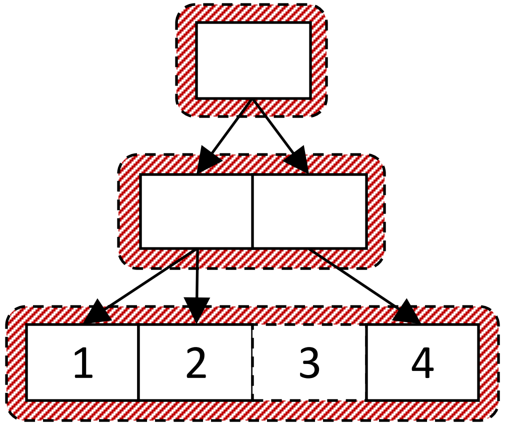


Page table walk required only on cache misses

MIDGARD PAGE TABLE



Layout in Physical Memory



Layout in Midgard

- Page table can be mapped to Midgard to ease the walk
    - Sparse Midgard address space allows reserving contiguous space for every level
    - Direct lookup of any entry in the cache hierarchy (like TLBs, MMU caches)

Cache hierarchy can directly serve Midgard page table entries

CPU SUPPORT

- Midgard (store) page faults are detected late in the pipeline [Qiu, ISCA’99]
  - After a store ends up retiring and is in the store buffer

- Precise exception handling requires keeping all retired state [Gniady, ISCA’99]
  - Post-retirement speculation needs a lot of silicon (e.g., 20KB of state)

- **Imprecise store exceptions** [Gupta, ISCA’23]
  - Microarchitecture + OS co-design to handle late store exceptions
  - Obviates the need for post-retirement speculation
  - Formalism to guarantee maintaining memory consistency

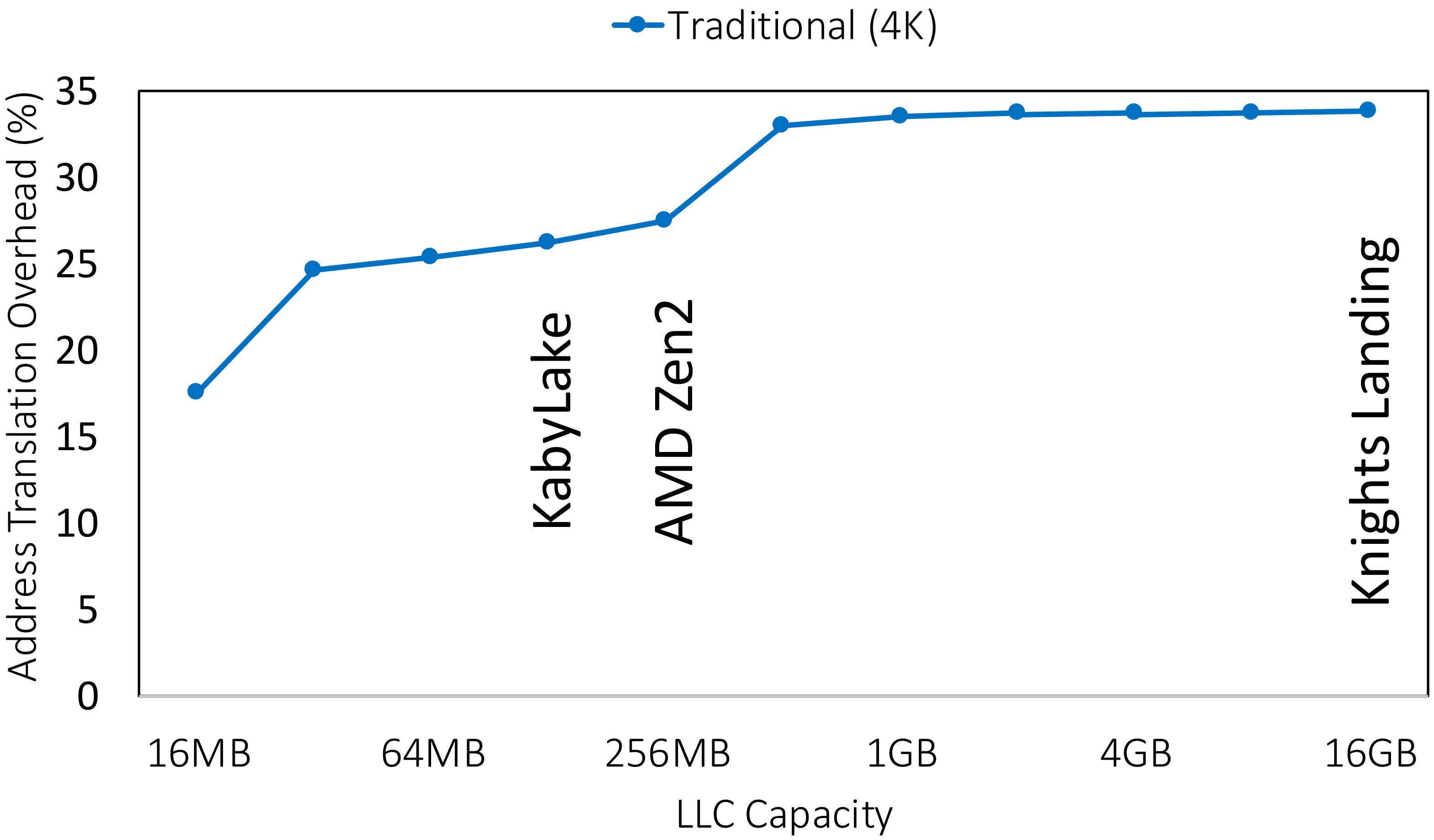
METHODOLOGY

- Trace analysis of memory accesses with QFlex
- AMAT analysis to quantify VM overhead
- Workloads: GAP benchmark suite, Graph500

- 16 ARM cores
- 256GB of dataset
- Baseline TLB: 64-entry L1, 1024-entry L2
- Midgard: 16-entry VLB, no MLB by default

POST-MOORE VM PERFORMANCE

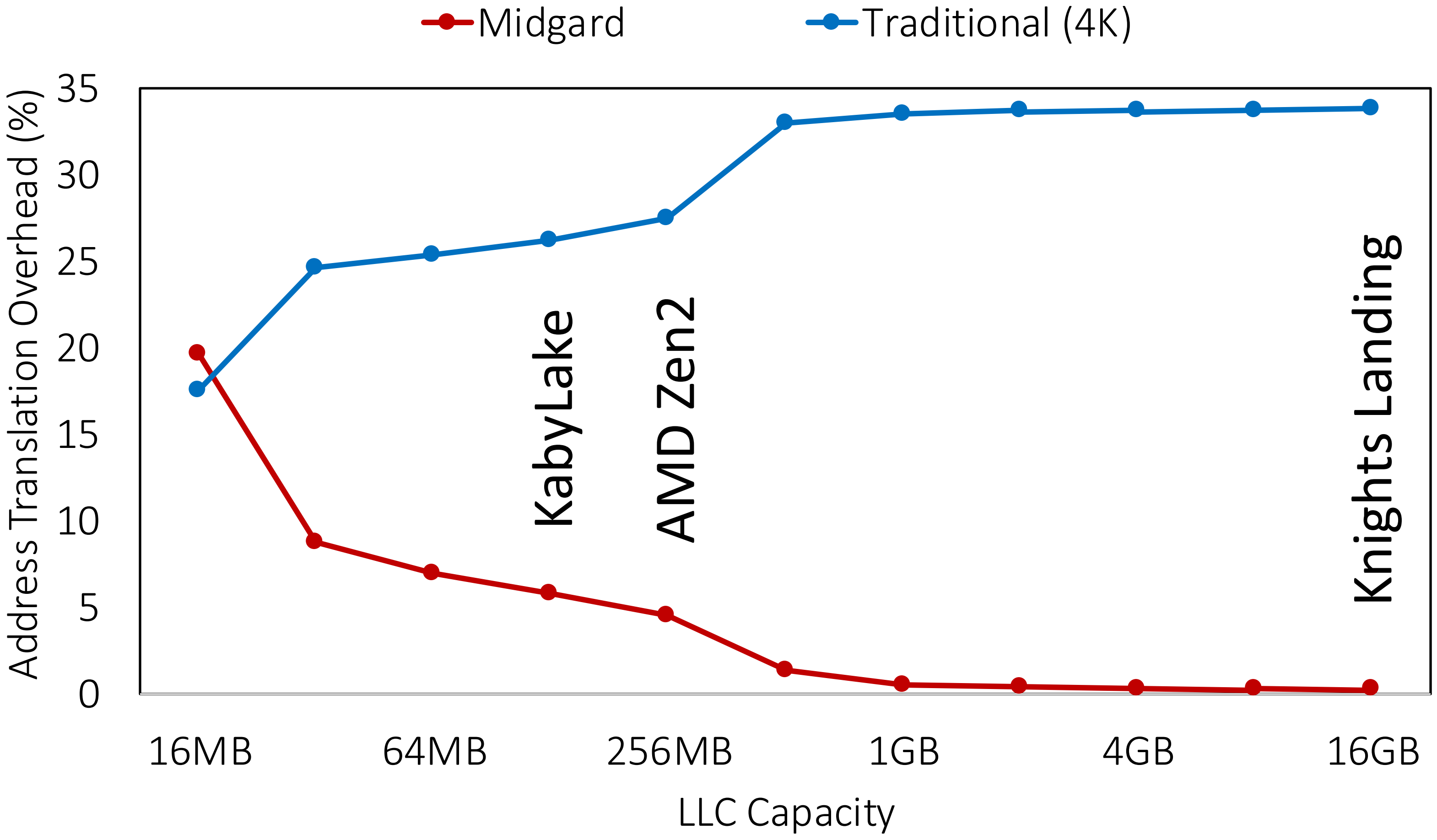
- As cache hierarchy capacity increases, time spent in data accesses goes down, thus increasing VM overhead



VM performance degrades as the cache hierarchy capacity increases

FUTURE-PROOFING VM WITH MIDGARD

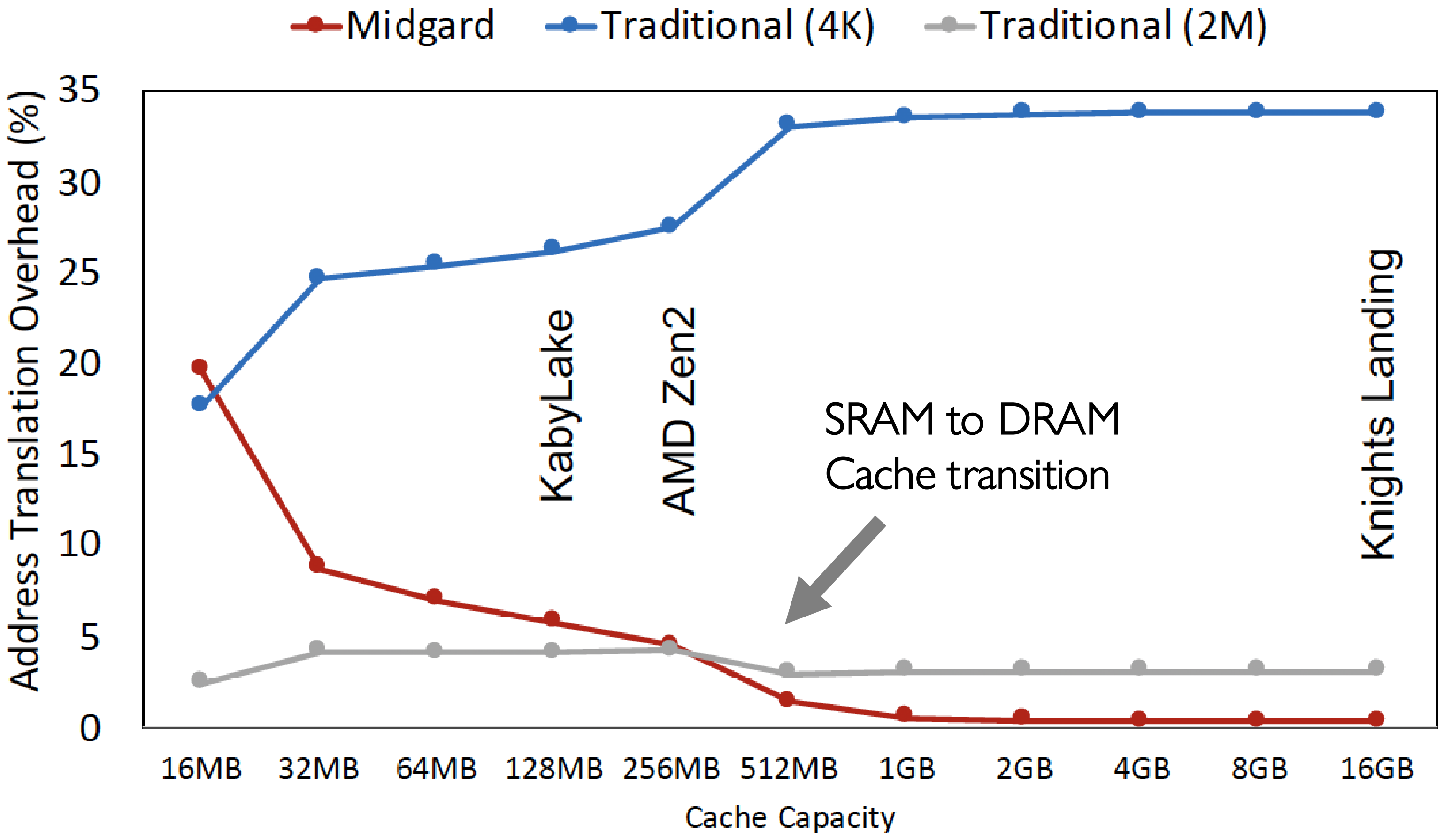
- For 16MB, Midgard has <5% performance overhead compared to traditional
- Secondary working sets fits in 32MB and 512MB LLC



Midgard performance improves with the cache hierarchy capacity

OPTIMISTIC COMPARISON TO HUGE PAGES

- Overhead of huge page transition ignored
- Overhead persists independent of page size as cache capacity grows



VM performance degrades as the cache hierarchy capacity increases

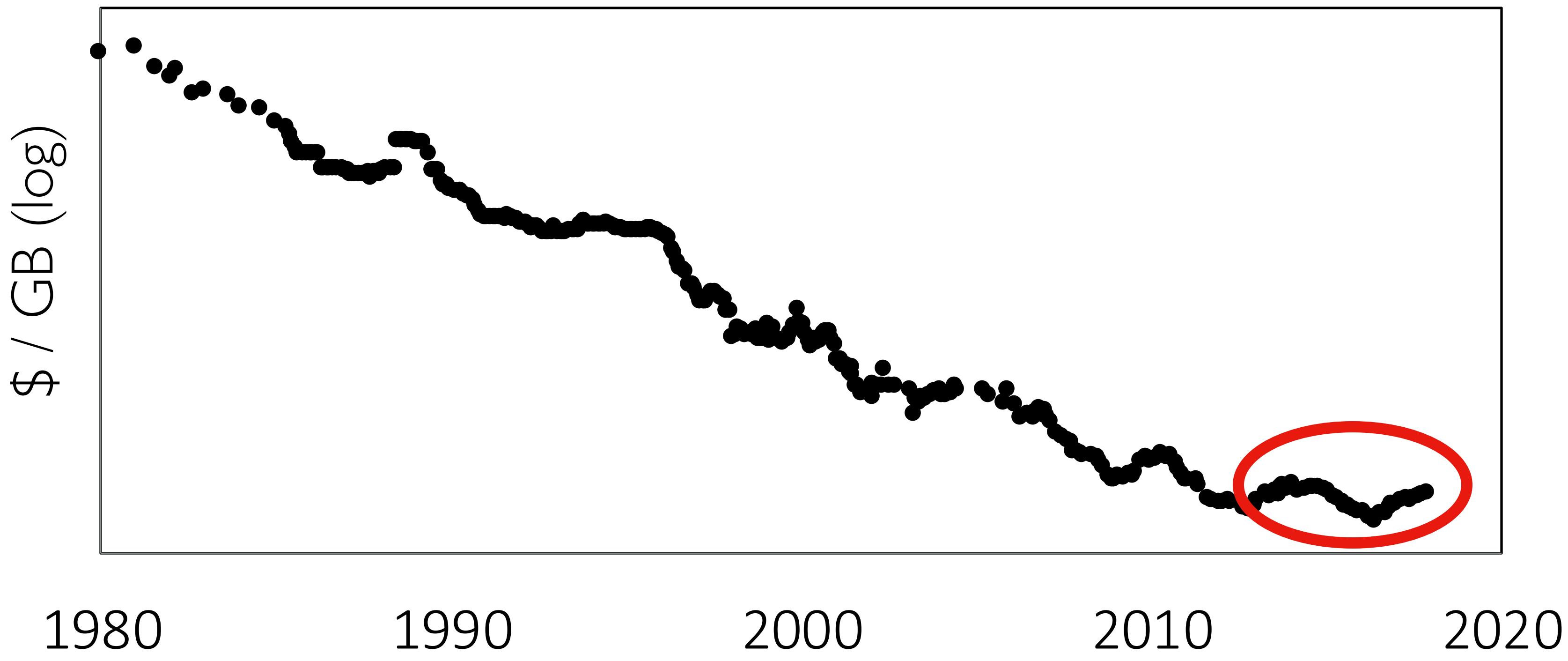
TB-SCALE MEMORY WITHOUT TLB

Midgard Roadmap:

- CPU microarchitecture/OS [ISCA’21’23]
- Compartmentalization [IEEE S&P’23]
- Virtualization/Containerization
- Accelerator ecosystem/IO
- Monolith/µservices/serverless
- ….

Intel Transformative Server Architecture Center

ONLINE SERVICES HOST DATA IN DRAM



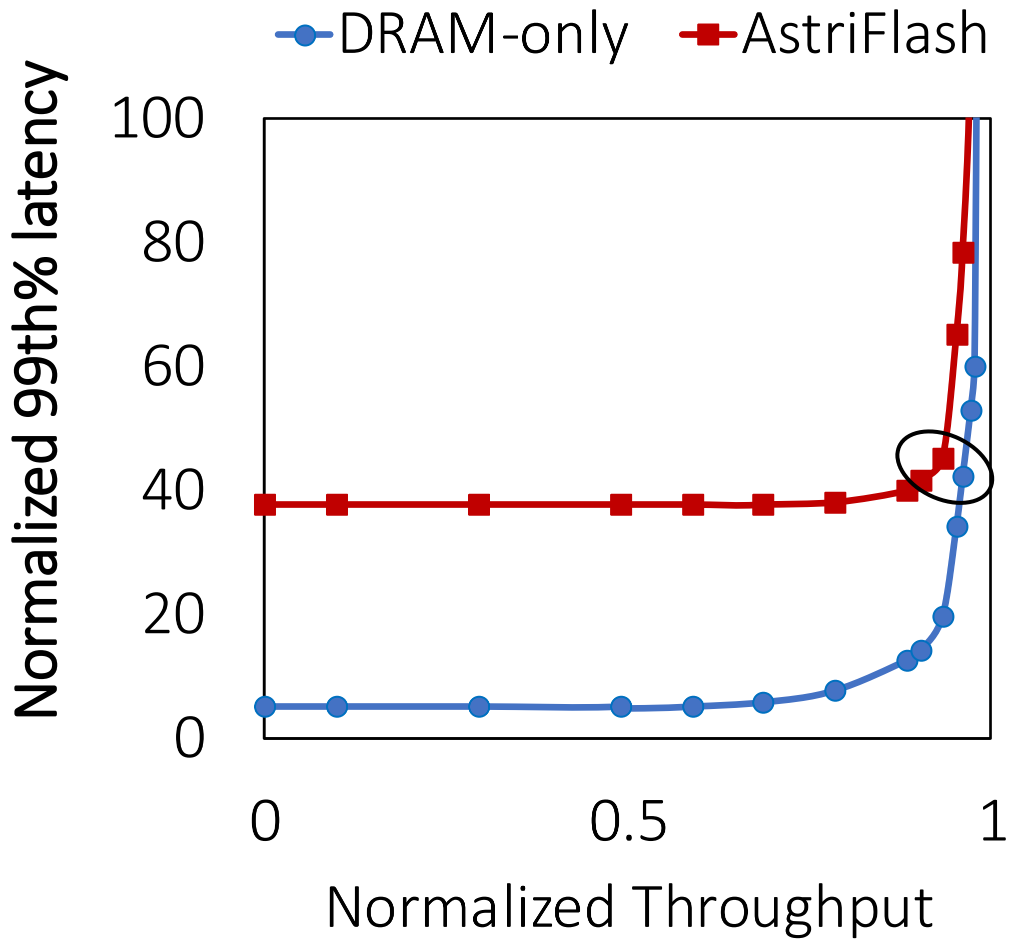
- Crucial for high performance and low tail latency
- DRAM is expensive and is not scaling in density

DRAM is about 50% of overall server cost

ONLINE SERVICES IN FLASH [HPCA'23]

	Cost	Latency
DRAM	1x	~100 ns
SCM	1/5x	1-10 µs
SSD	1/30x-1/50x	> 50 µs (OS)

- Host & serve mapped data from SSD
- Hardware-managed DRAM cache
- Co-design to eliminate OS overhead
  - paging
  - threading



Maintains tail latency with only 5% lower throughput

SUMMARY

Post-Moore datacenters:

- Integration + Specialization + Approximation
- Revisit legacy abstractions, SW/HW interfaces
- Holistic algorithm/SW/HW co-design
- Division of control vs. data plane

Datacenter sustainability:

- Best practices
- Metrics

THANK YOU!

PARALLEL SYSTEMS ARCHITECTURE LAB

For more information, please visit us at

parsa.epfl.ch

EPFL