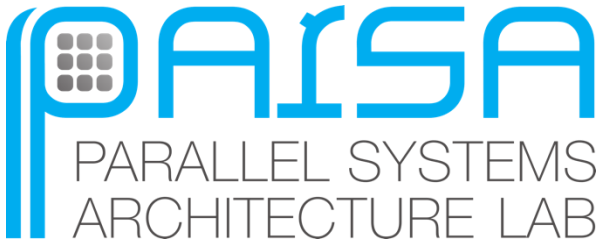
VIRTUAL MEMORY FOR POST-MOORE SERVERS

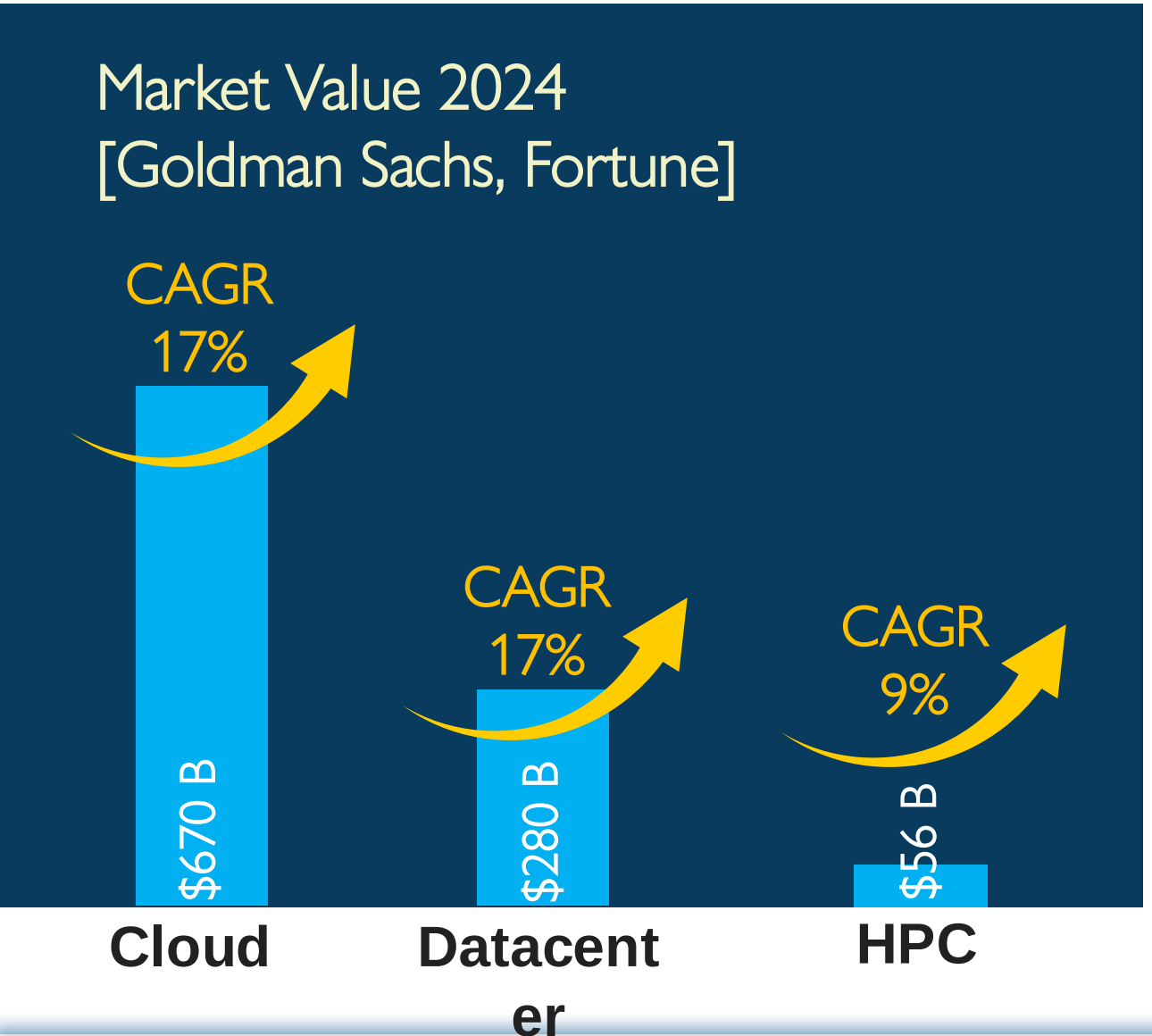
Babak Falsafi



parsa.epfl.ch

CLOUD & DATACENTER GROWTH

PARALLEL SYSTEMS ARCHITECTURE LAB

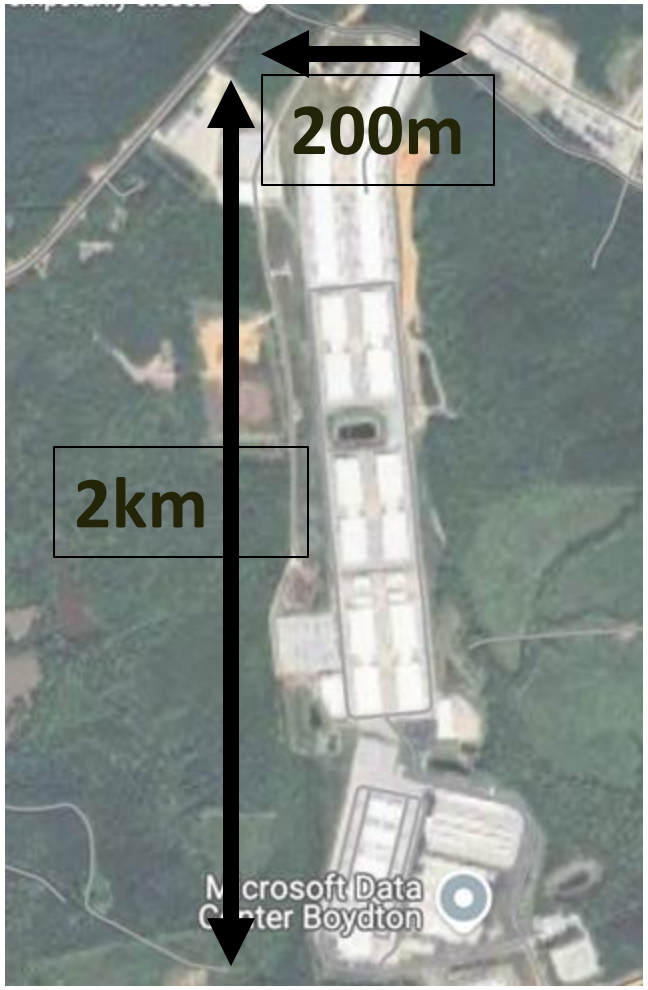


- Data → fuel for digital economy
- Exponential demand for digital services
- Many apps (e.g., AI) with higher exponential demand

DATACENTERS ARE BACKBONE OF CLOUD

PARALLEL SYSTEMS ARCHITECTURE LAB

- 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 DC, 300MW

CLOUDS AT VARIOUS SCALES

PARALLEL SYSTEMS ARCHITECTURE LAB

SCALE-OUT DATACETNERS

PARALLEL SYSTEMS ARCHITECTURE LAB

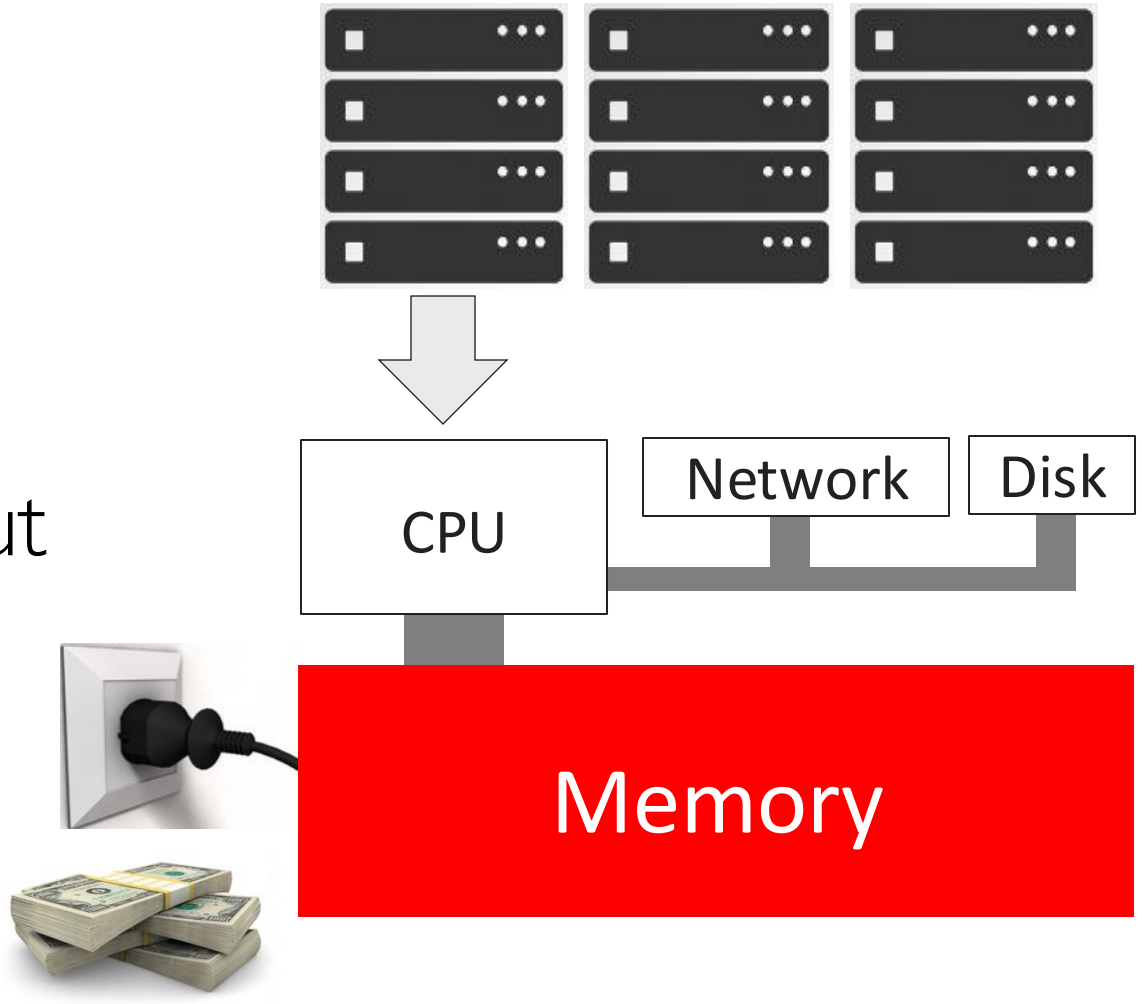
Cost is the primary metric (~50%)

Online services hosted in memory

Divide data up across servers

Design server for low cost, scale out

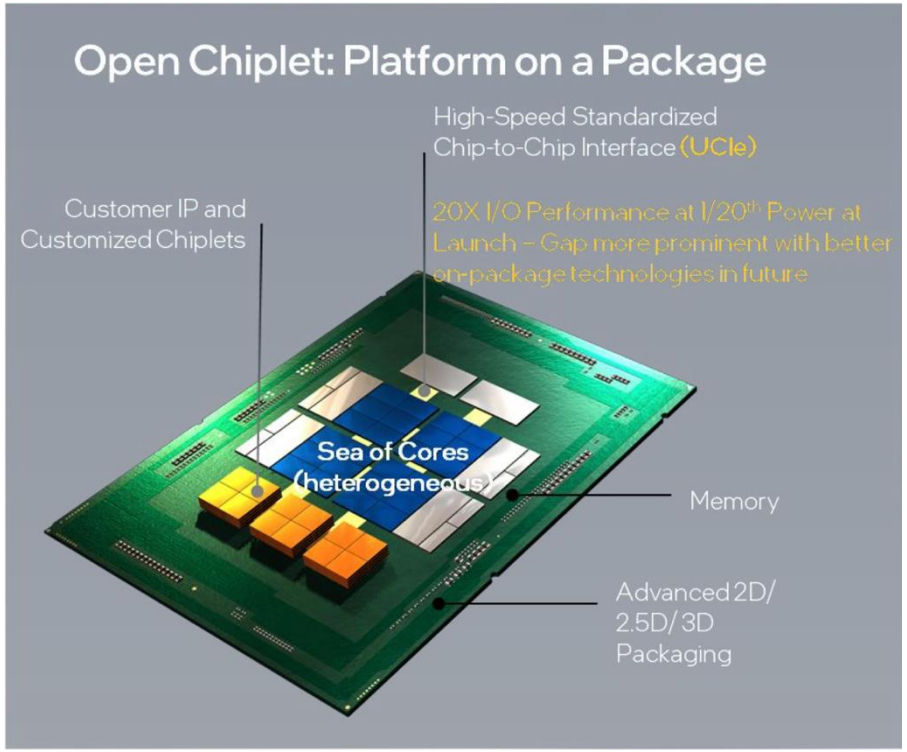
☞Memory most precious silicon



OPPORTUNTY: LD/ST INTERCONNECTS

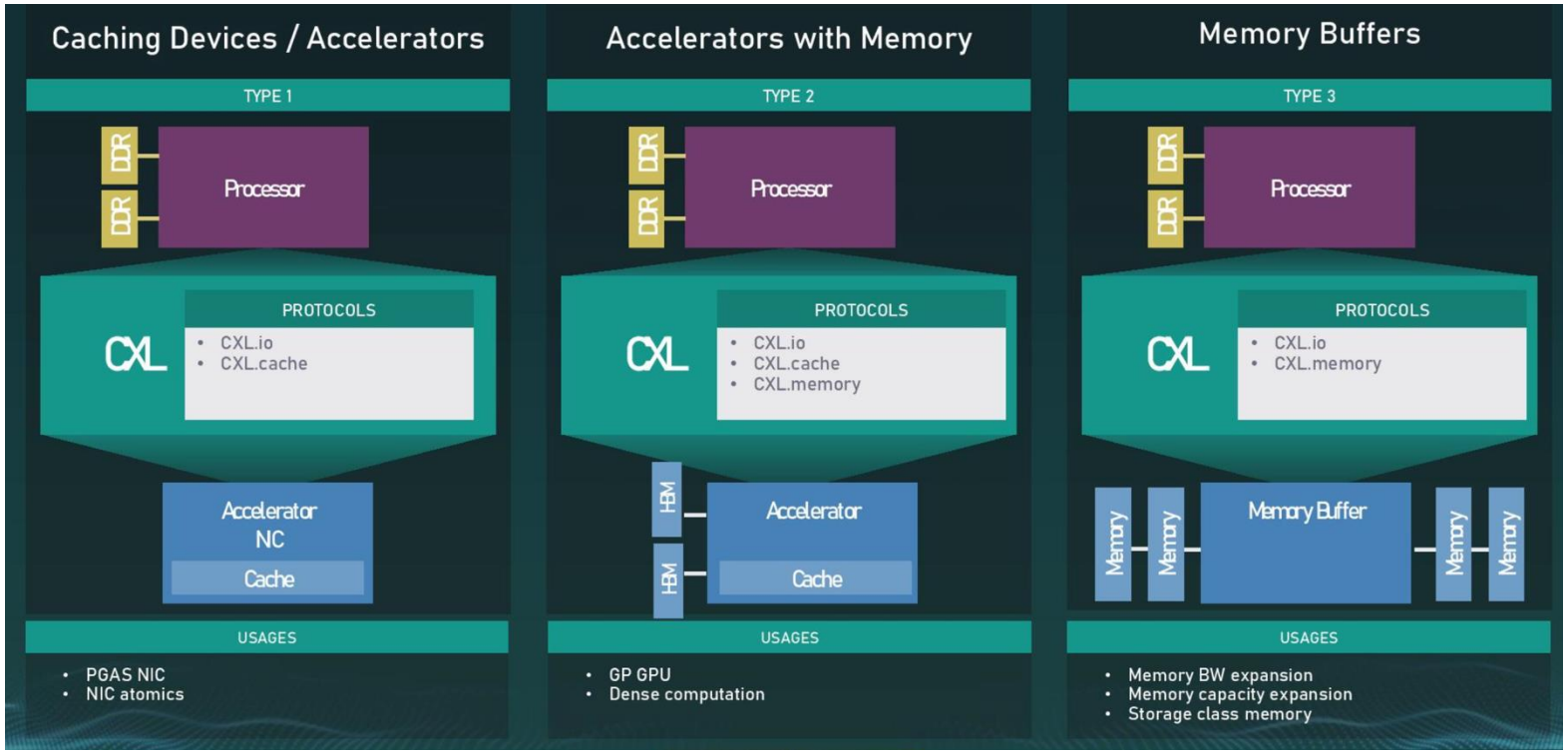
PARALLEL SYSTEMS ARCHITECTURE LAB

UCIE/HBM



- Low on-package latency
- Higher per-core bandwidth

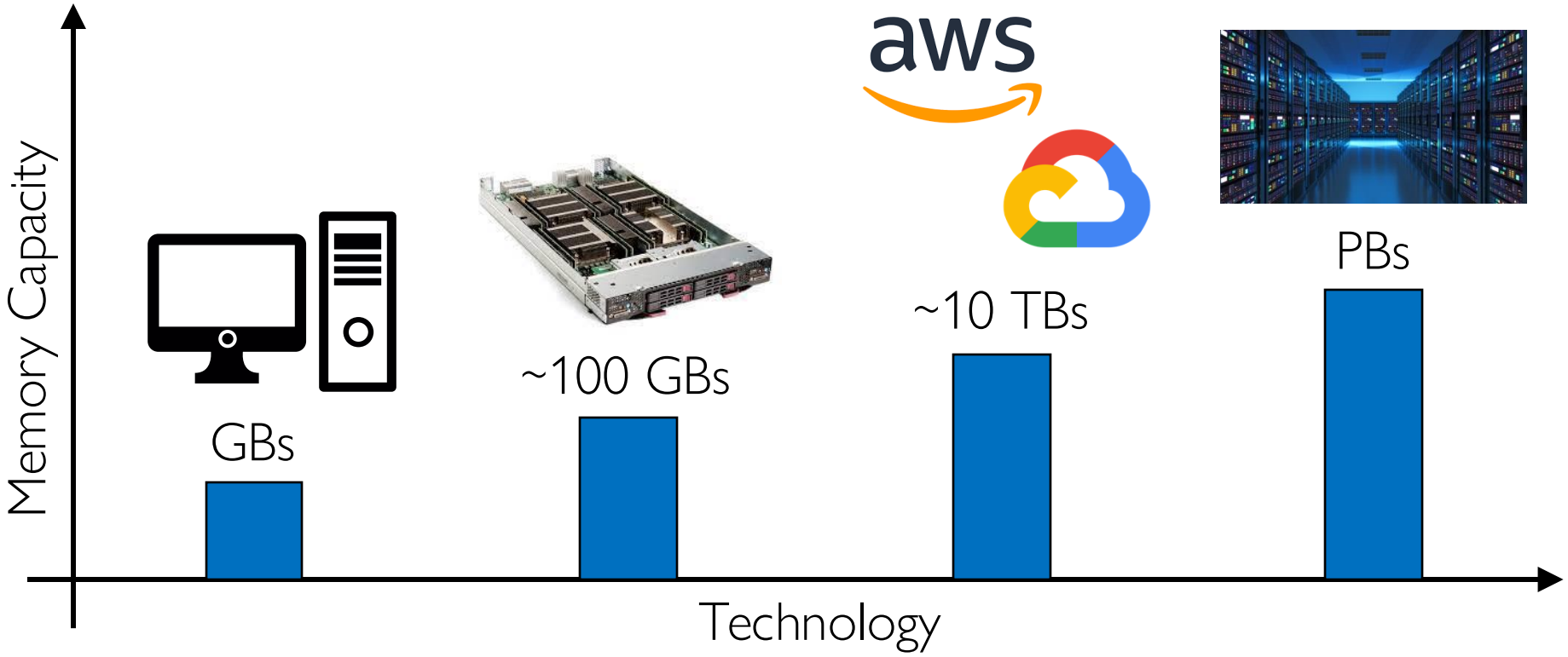
CXL/NVLINK



- Better per-core bandwidth scalability
- Shared memory with latency/bandwidth trade-offs

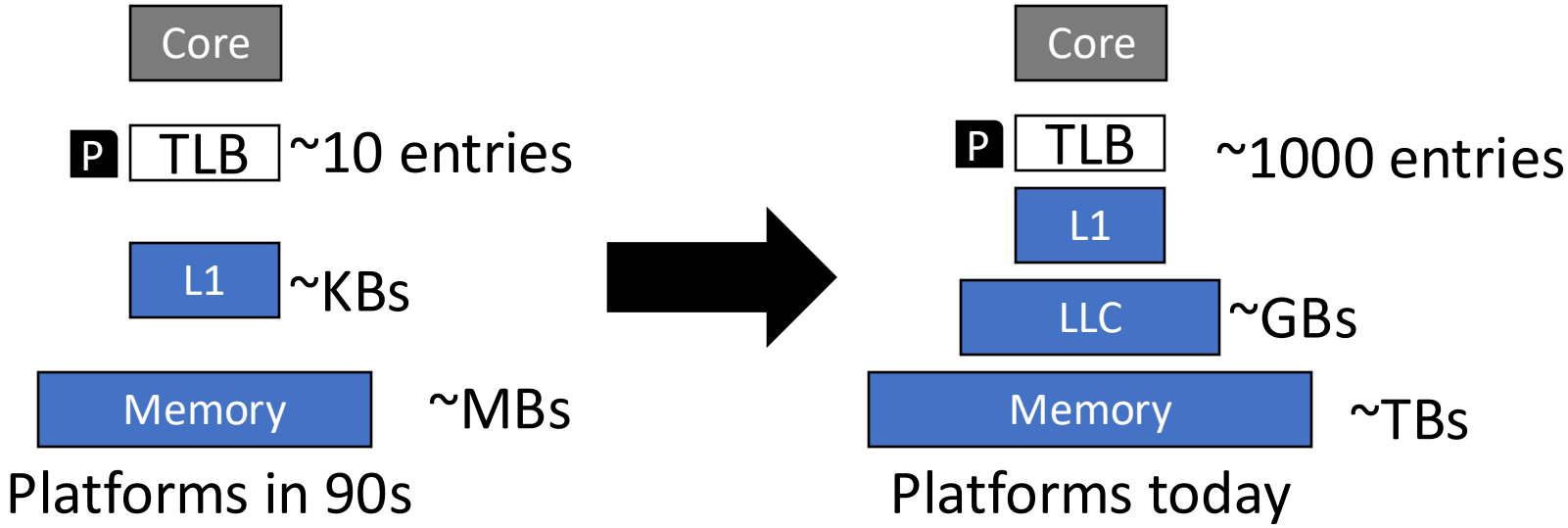
MEMORY CAPACITY IS INCREASING

PARALLEL SYSTEMS ARCHITECTURE LAB



Future servers will have access to PBs of memory capacity

CHALLENGE: VIRTUAL MEMORY

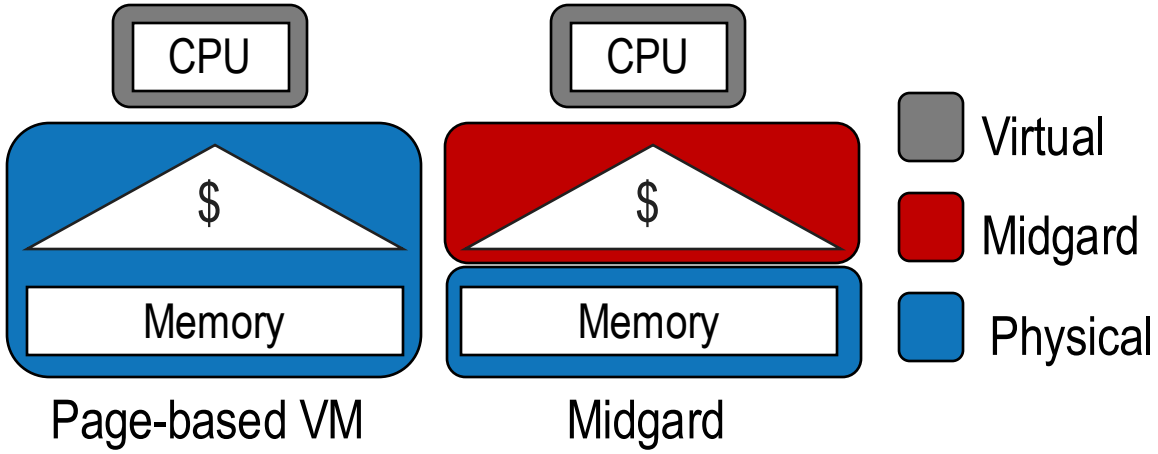


Product	Year	Cores	Cache capacity	TLB entries	Coverage (4KB)
Intel P4	2000	1	256KB SRAM	64	256KB
Intel KabyLake	2016	4	128MB eDRAM	1536	6MB
Apple M1	2020	8 (4+4)	16MB SRAM	3096	12MB (16KB)
AMD Zen3	2021	64 (8x8)	256MB SRAM	2048	8MB
Intel Sapphire Rapids	2022	56 (14x4)	64GB HBM2	?	?

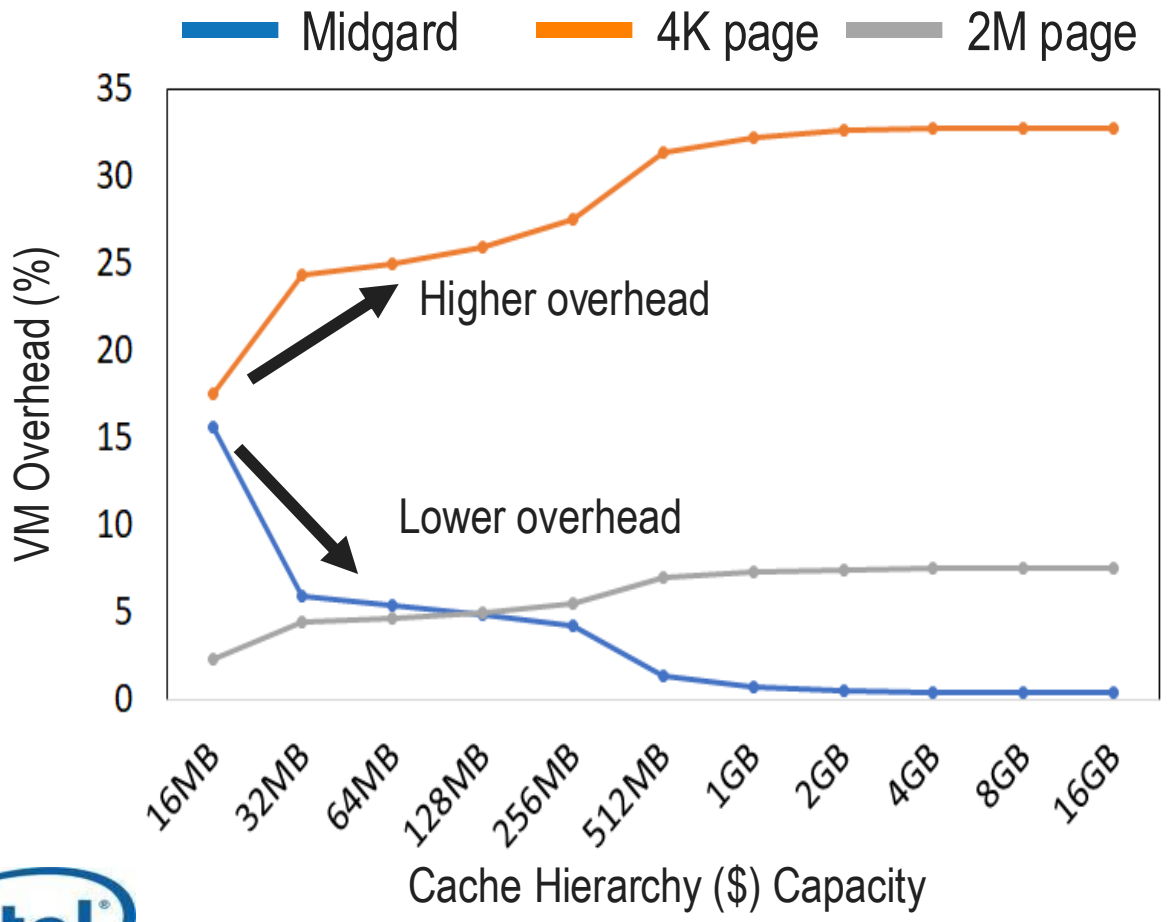
VIRTUAL MEMORY WITHOUT TLB

midgard

midgard.epfl.ch



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



intel

VIRTUAL MEMORY WITHOUT TLB

PARISA
PARALLEL SYSTEMS ARCHITECTURE LAB

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

EPFL

THE UNIVERSITY OF EDINBURGH

Yale University
LUX ET VERITAS

Intel Transformative Server Architecture Center

intel

ROADMAP

PARALLEL SYSTEMS ARCHITECTURE LAB

- ~~Overview~~
- Virtual Memory
- Midgard
- 128-bit Address Space
- Summary

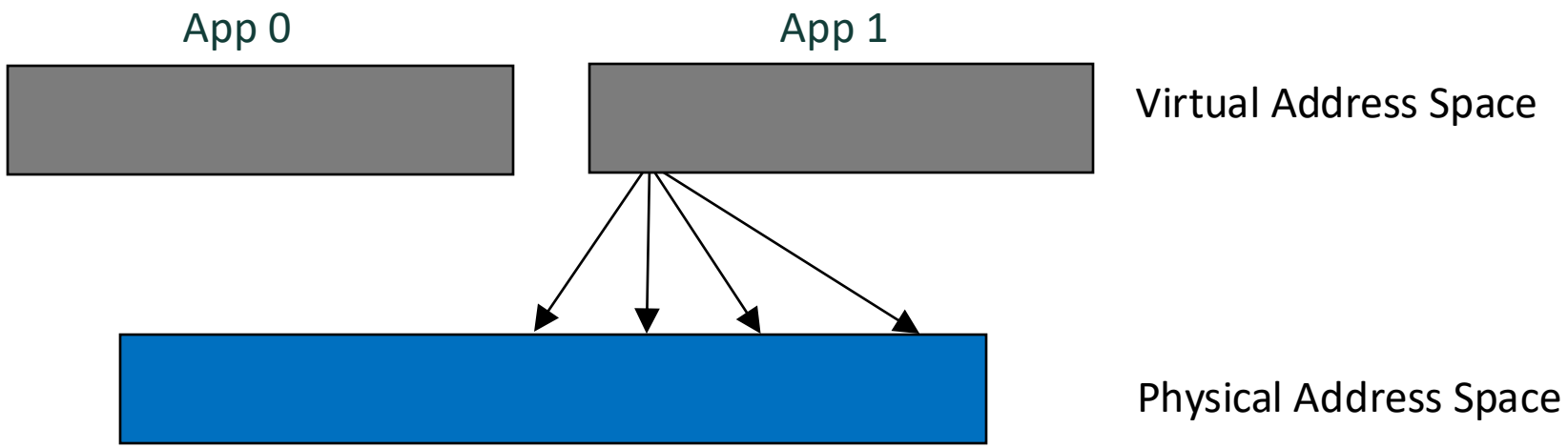
VIRTUAL MEMORY

PARALLEL SYSTEMS ARCHITECTURE LAB

- 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)

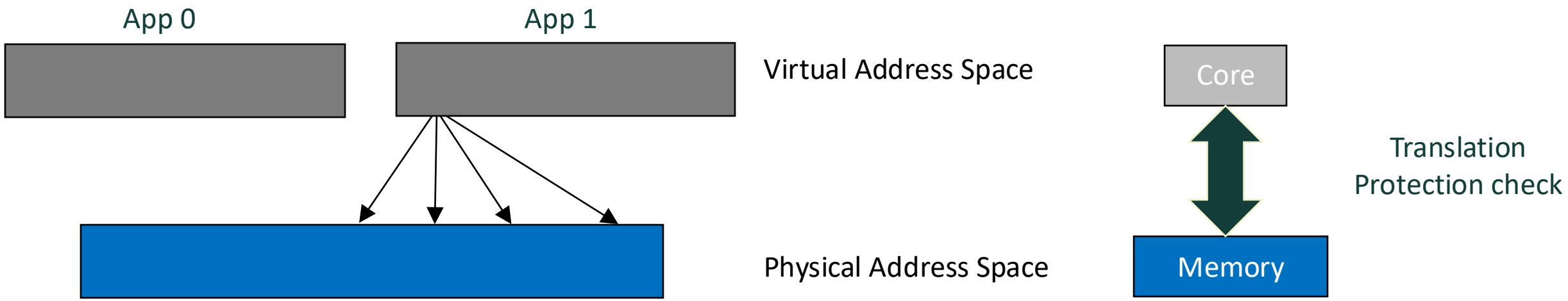
Essential abstraction for programming and memory management

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)

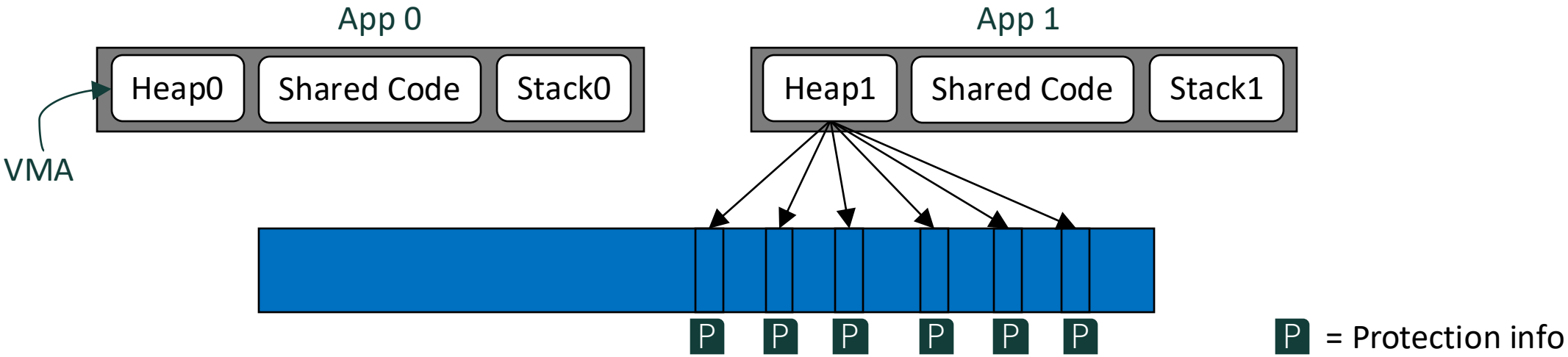


- Architectural support is required for
  - Translating virtual addresses to physical addresses
  - Performing protection checks

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

PERFORMANCE REQUIREMENTS

PARALLEL SYSTEMS ARCHITECTURE LAB

- Cores directly interact with cache hierarchy
  - Translation/protection should work at cache latency
  - VMAs could give us fast translation/protection at a large granularity
  - But we lost VMAs and divided them into numerous, small pages

- Page-based translation/protection
  - Require lookup of replicated information for each page
  - Lookups become expensive with larger cache/memory capacity

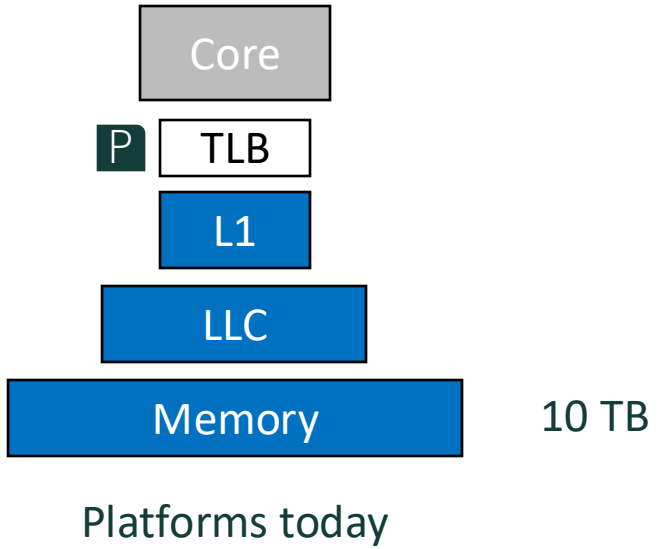
Translation/protection should match cache speed

HARDWARE SUPPORT TODAY

PARALLEL SYSTEMS ARCHITECTURE LAB

- Translation Lookaside Buffer (TLB)
  - Cache mappings for recently used pages
  - Accelerate translation and protection checks

- TLBs do not scale
  - Memory capacity has grown from MBs to ~10 TB
  - Cache hierarchies have grown up to ~10 GB
  - TLBs only have 1000s of entries i.e. ~10 MB coverage
  - End of Moore's law prohibits further silicon scaling



TLBs cannot provide the required coverage

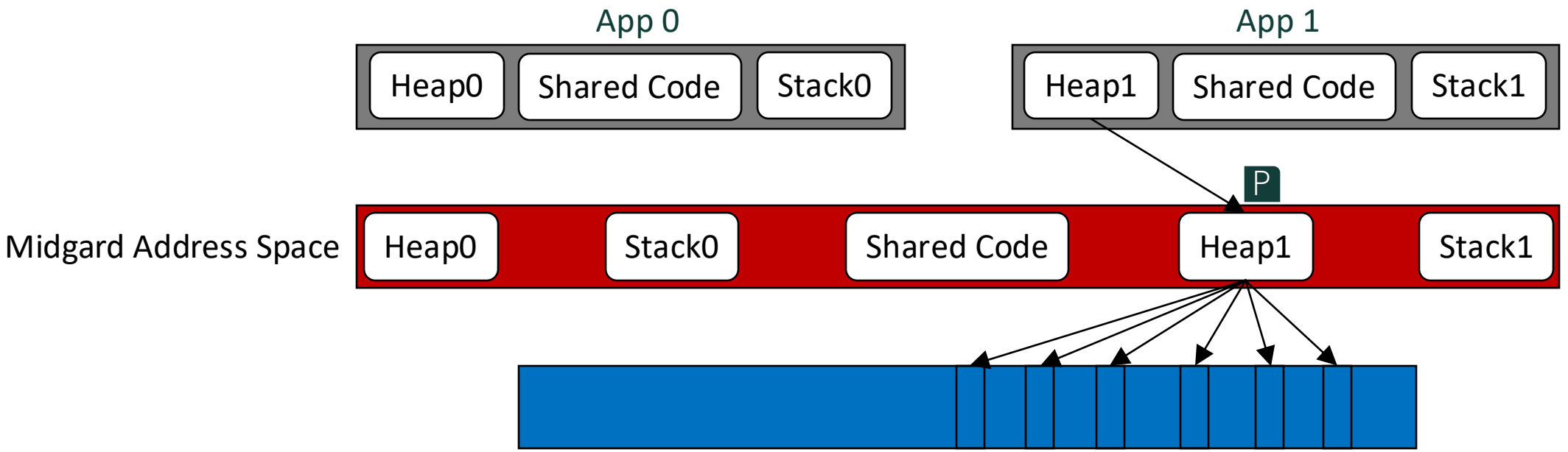
PRIOR WORK

PARALLEL SYSTEMS ARCHITECTURE LAB

- Aim to create contiguity in the physical address space
  - Huge pages
  - Direct segments [Basu, ISCA'13]
  - Memory defragmentation [Yan, ISCA'19]
- Contiguity helps achieve faster translation/protection [Skarlatos, ISCA'23]
  - At the price of higher runtime overhead
- Virtual hierarchies
  - In-cache address translation [Wood, ISCA'86]
  - VBI [Hajinazar, ISCA'20]

Previous proposals help, but do not solve the problem

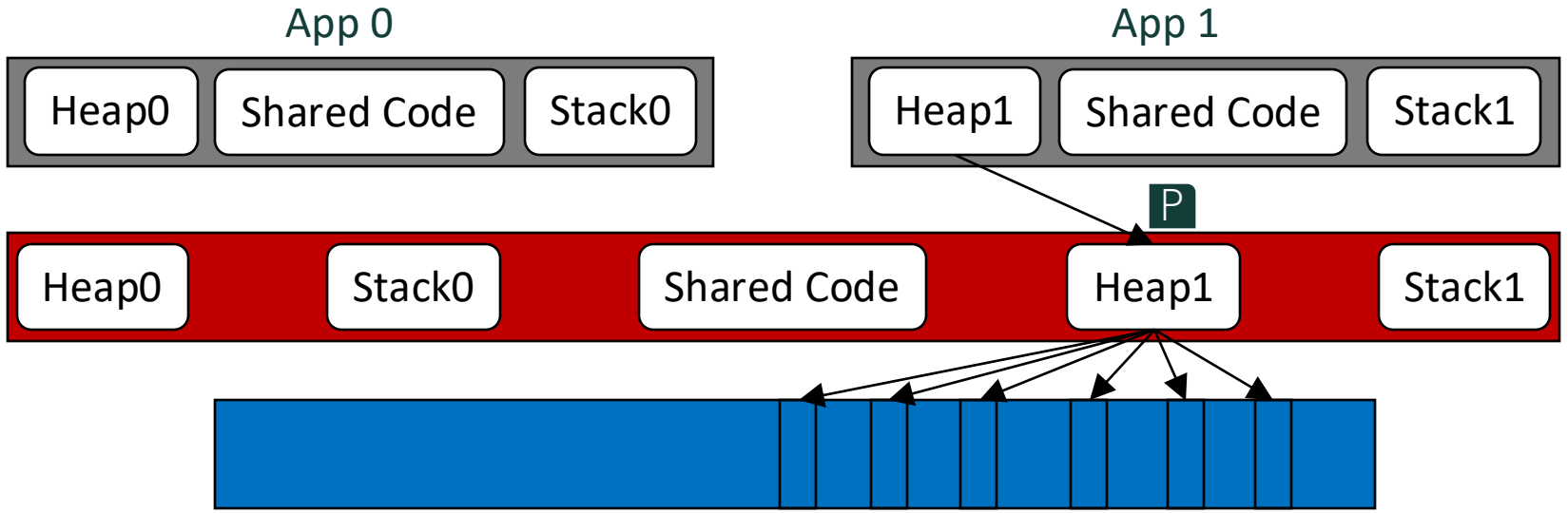
MIDGARD ADDRESS SPACE

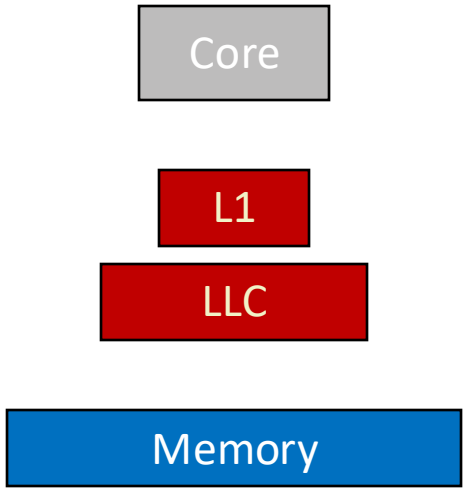


- A sparse intermediate address space that retains VMAs
  - Protection check and contiguous translation at VMA granularity
  - OS deduplicates shared VMAs, ensuring no synonyms/homonyms

Midgard provides an address space for the cache hierarchy

MIDGARD-ADDRESSED CACHE HIERARCHY





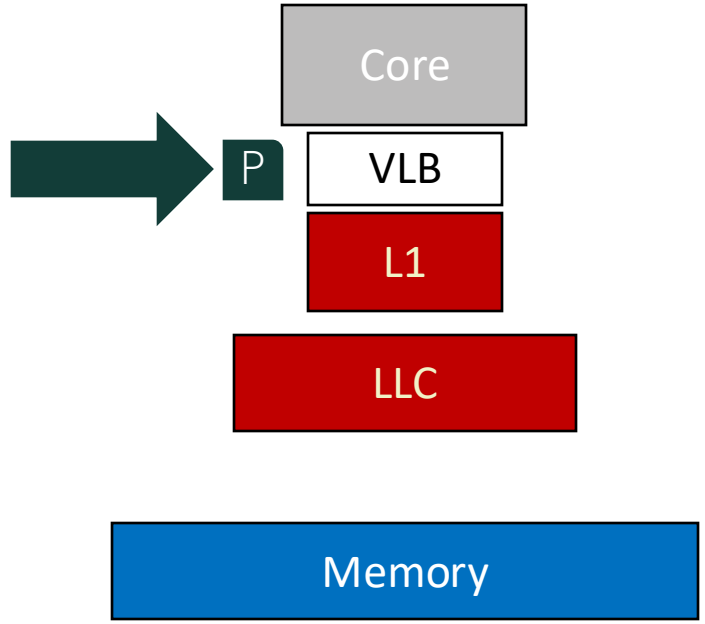
- Cache hierarchy now uses Midgard addresses
  - Virtual to Midgard translation is fast because of VMAs
  - Protection is implemented at a VMA granularity
  - Midgard to Physical translation is only required on cache misses

Midgard optimizes the common-case cache accesses

VIRTUAL-TO-MIDGARD TRANSLATION

PARALLEL SYSTEMS ARCHITECTURE LAB

- 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

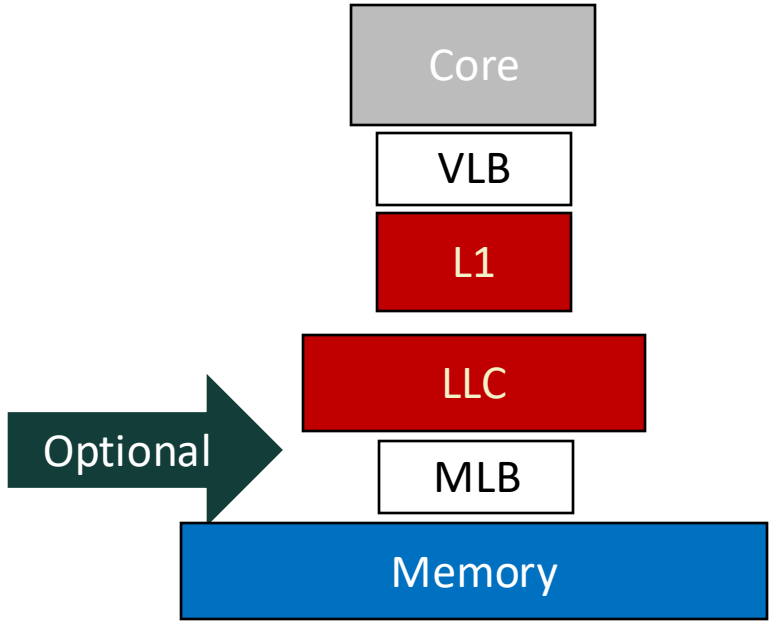


Only ~10 VLB entries required per core

MIDGARD-TO-PHYSICAL TRANSLATION

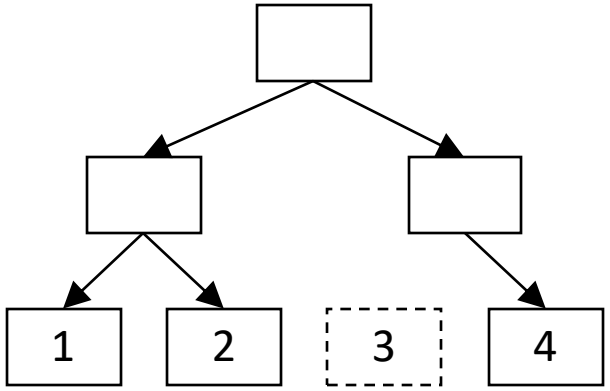
PARALLEL SYSTEMS ARCHITECTURE LAB

- 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
  - Spatial lookups (no temporal locality)
  - Optionally cache in Midgard Lookaside Buffers (MLBs)

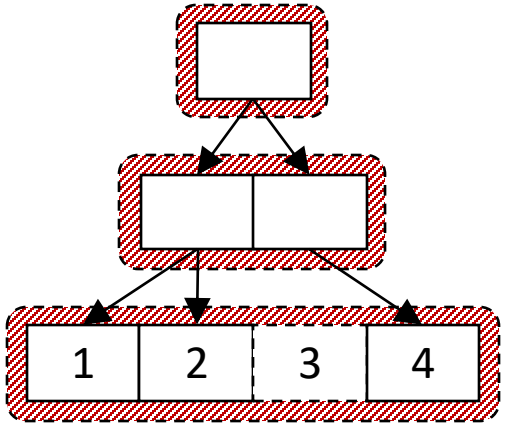


Page table walk required only on cache misses

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

MIDGARD PAGE FAULTS

PARALLEL SYSTEMS ARCHITECTURE LAB

- 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

PARALLEL SYSTEMS ARCHITECTURE LAB

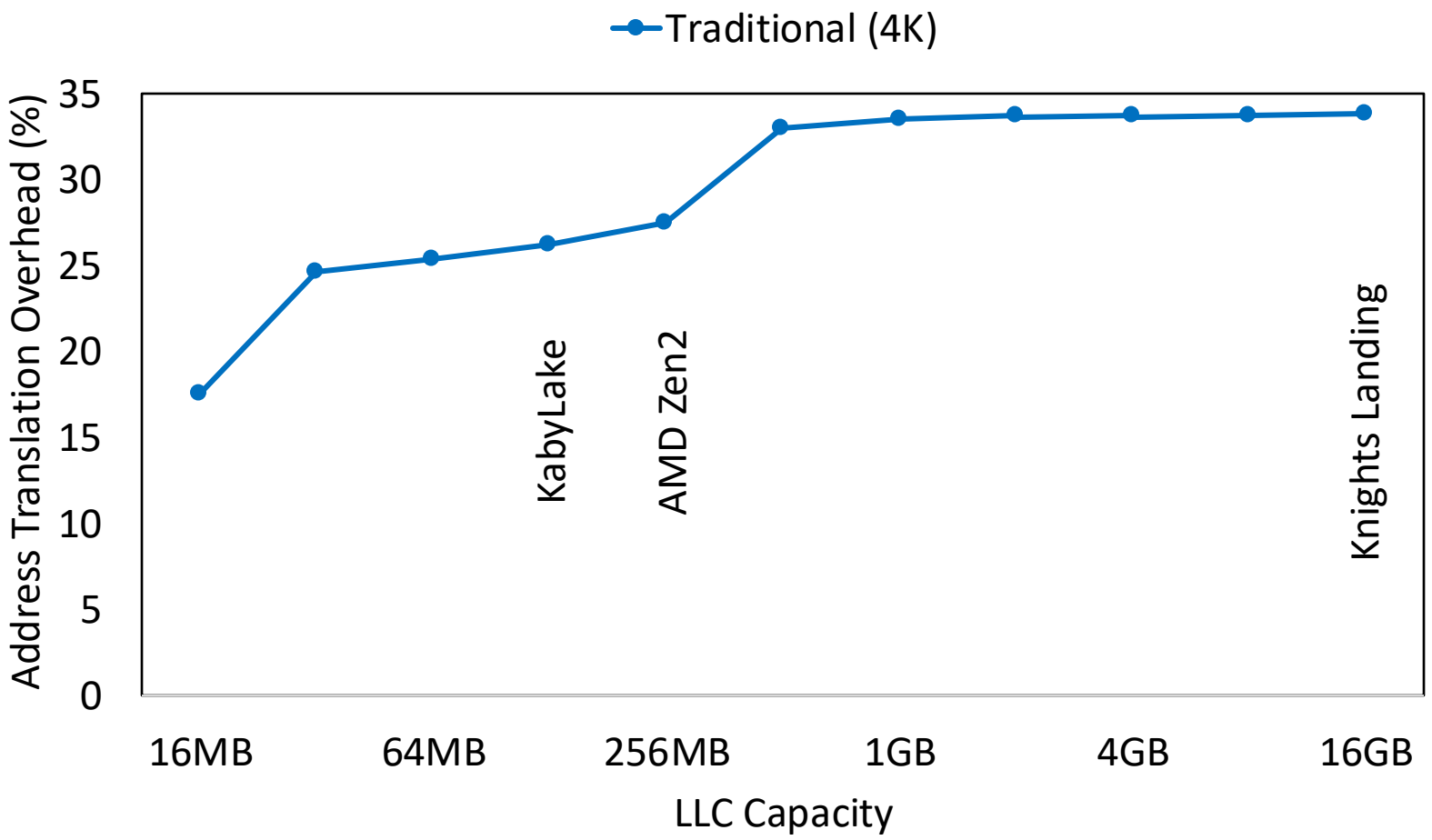
- Trace analysis of memory accesses with QFlex
- AMAT analysis to quantify VM overhead
- Workloads: GAP benchmark suite, Graph500

- 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

PARALLEL SYSTEMS ARCHITECTURE LAB

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

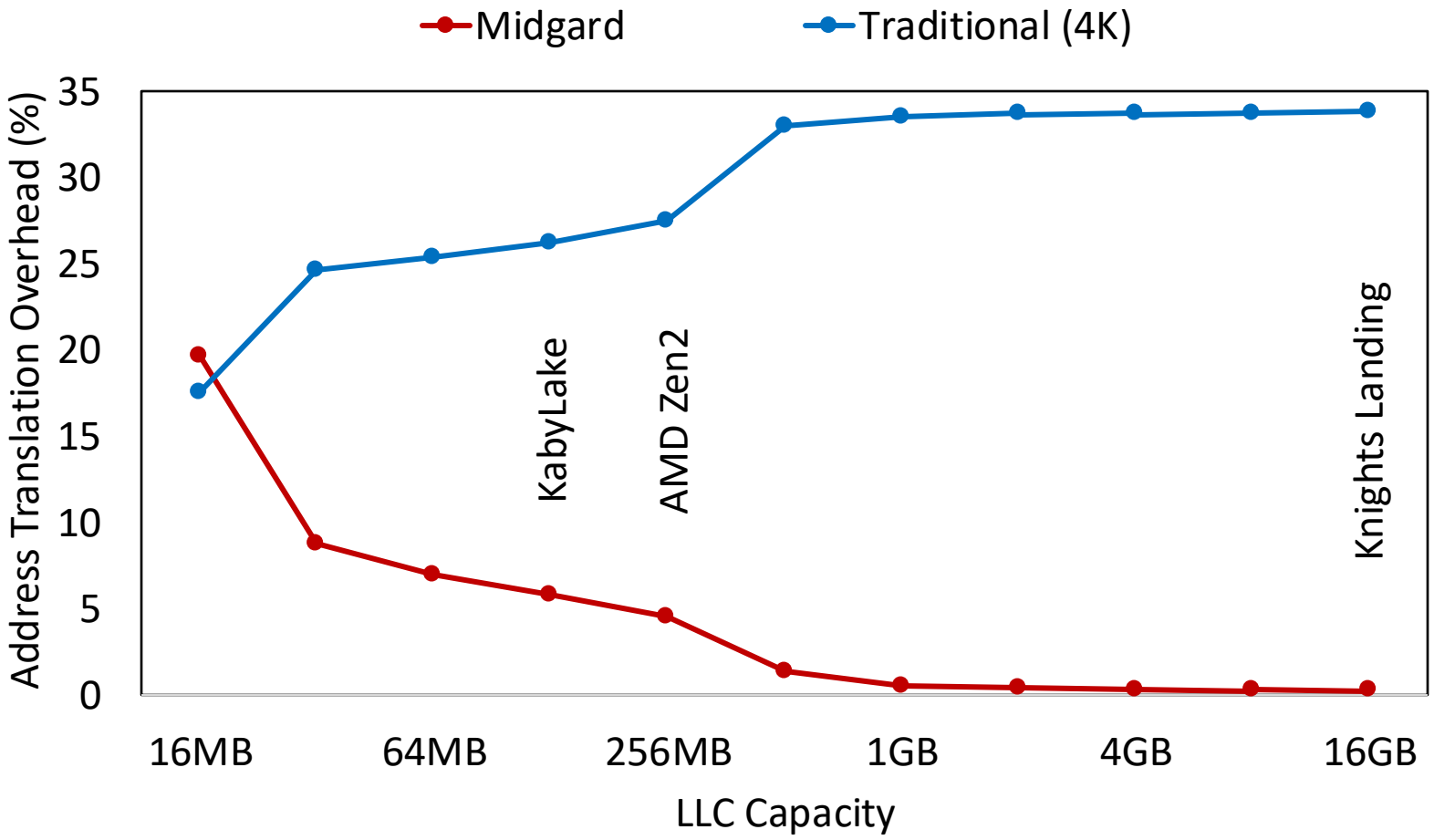


VM performance degrades as the cache hierarchy capacity increases

FUTURE-PROOFING VM WITH MIDGARD

PARALLEL SYSTEMS ARCHITECTURE LAB

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

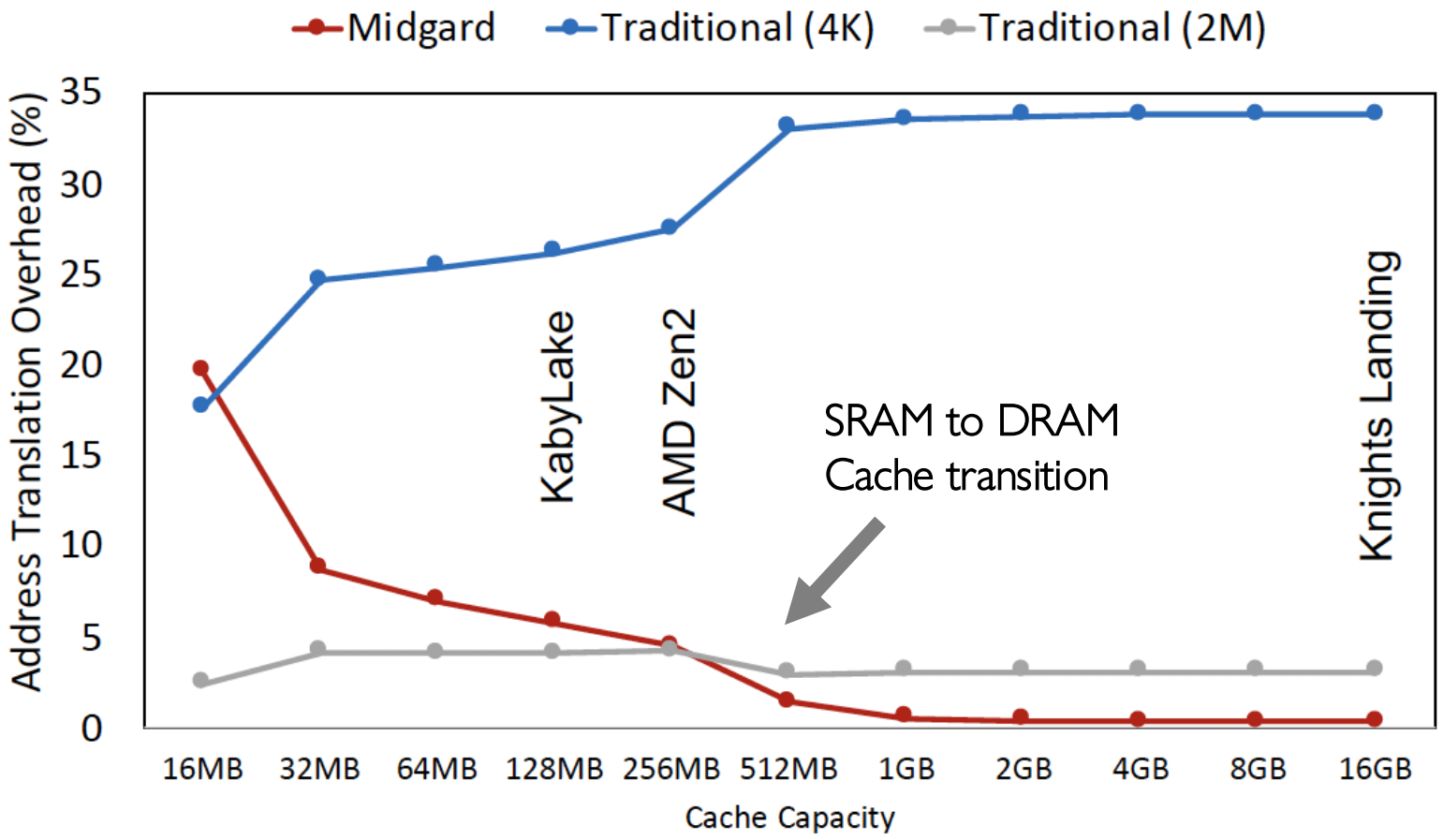


Midgard performance improves with the cache hierarchy capacity

CONSERVATIVE COMPARISON TO HUGE PAGES

PARALLEL SYSTEMS ARCHITECTURE LAB

- 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

ROADMAP

PARALLEL SYSTEMS ARCHITECTURE LAB

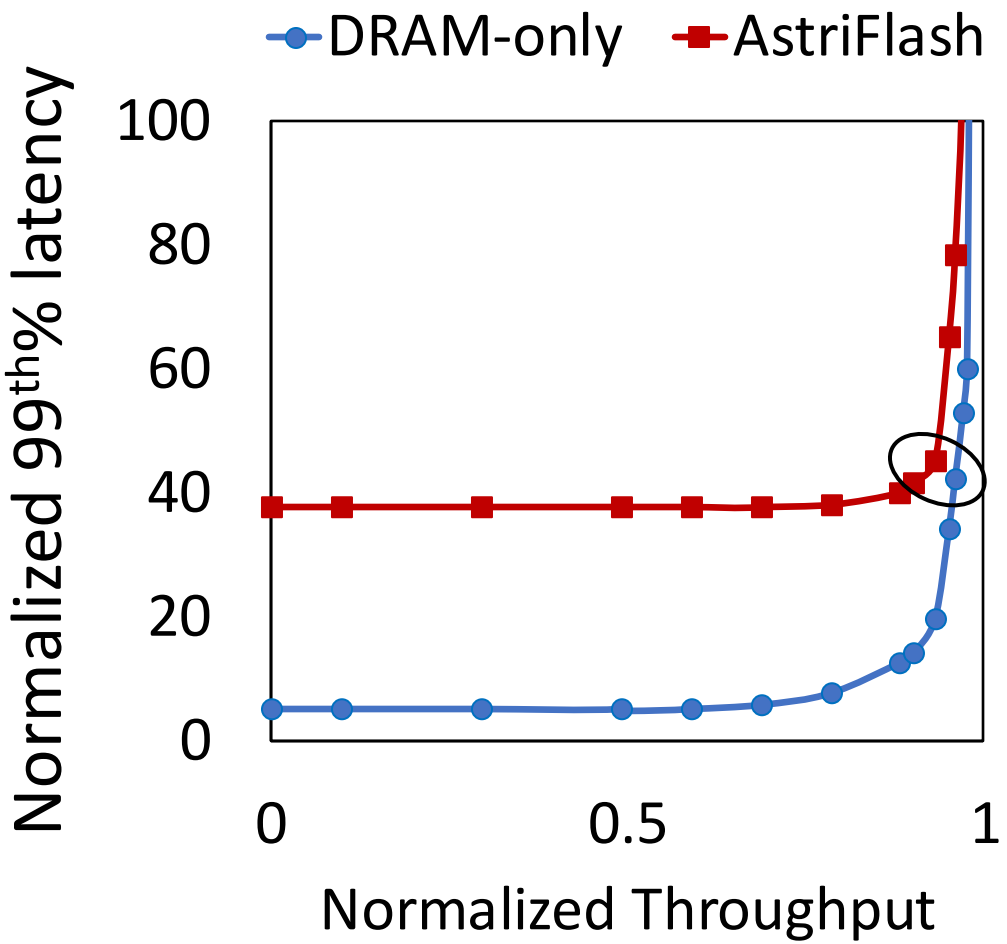
- ~~Overview~~
- ~~Virtual memory~~
- ~~Midgard~~
- 128-bit address spaces
- Summary

MAP FLASH INTO ADDRESS SPACE [HPCA'23]

PARALLEL SYSTEMS ARCHITECTURE LAB

	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

WHAT CAN WE DO WITH A 128-BIT ADDRESS SPACE?

PARALLEL SYSTEMS ARCHITECTURE LAB

- Many opportunities for intermediate address spaces
  - Flexibility in placement of VMAs in a 128-bit Midgard space
  - Manage VMAs at rack-scale w/ multiple racks
- Single-address space systems
  - OS in a single address space [Koldinger, ASPLOS'92]
  - FaaS at user level w/ hardware support (stay tuned)
  - Eliminate the OS overhead
- Map SSDs into the address space (see next)

SUMMARY

PARALLEL SYSTEMS ARCHITECTURE LAB

Midgard

- Cloud relies on virtual memory
- VM implementations have faltered with memory and cache scaling
- Midgard accelerates VMA management
- POSIX compatible

128-bits

- Great use-case for intermediate address spaces
- Single-address space systems

THANK YOU!

PARSA
PARALLEL SYSTEMS ARCHITECTURE LAB

For more information, please visit us at

parsa.epfl.ch

EPFL