IN5050:
Programming heterogeneous multi-core processors

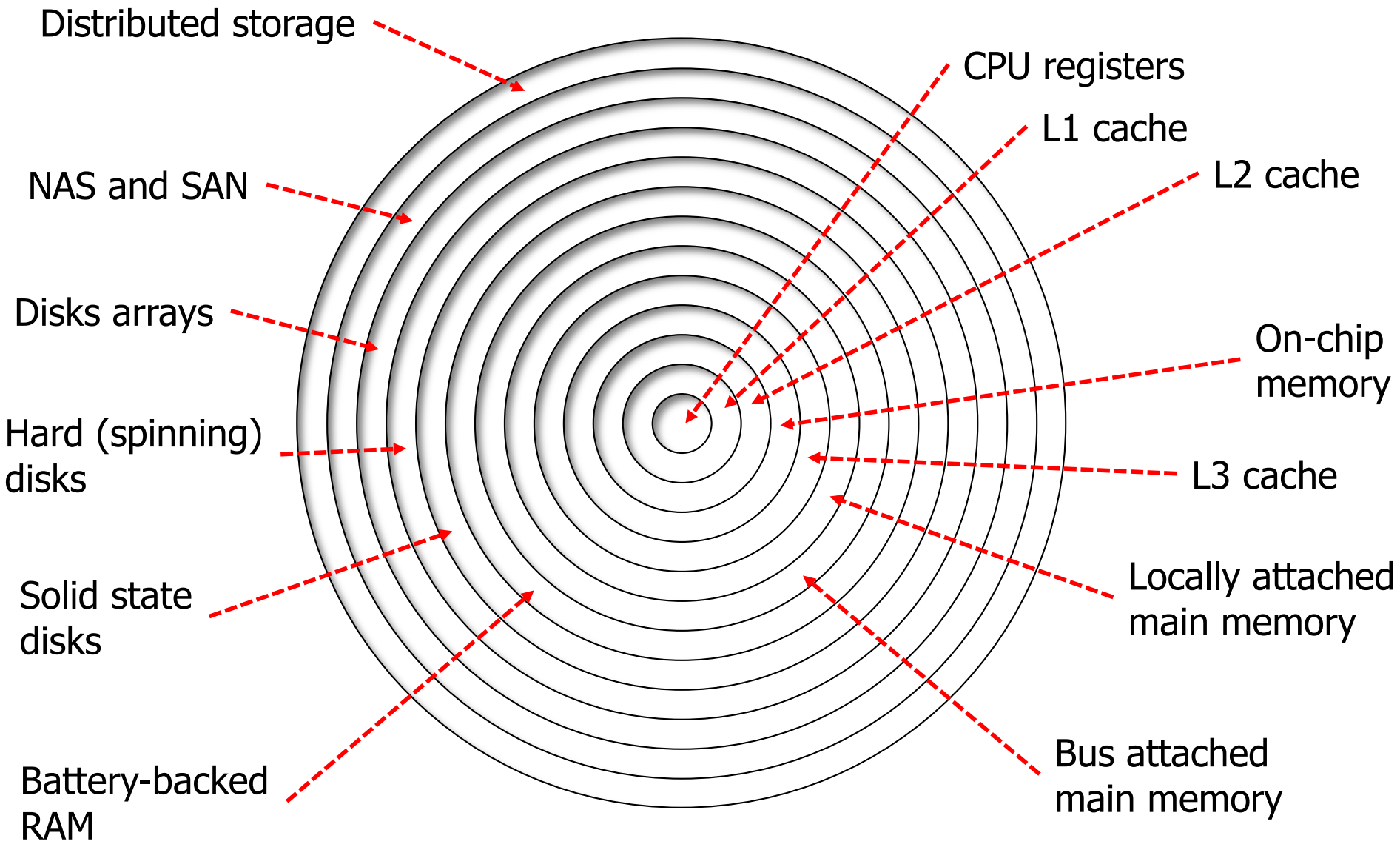


Memory Hierarchies

Carsten Griwodz

February 9, 2026

Hierarchies at scale



Registers

- Register sets in typical processors

processor	Generic	Special	Float	SIMD
MOS 6502	2	1xAccu 1x16 SP		
Intel 8080	7x8 or 1x8 + 3x16	1x16 SP		
Intel 8086	8x16 or 8x8 + 4x16			
Motorola 68000	8x32	8x32 address	8x32	
ARM A64 (each core)	31x64	zero-register	32x64	
Intel Core i7 (each core)	16x (64|32|16|8)			32x (512|256|128)
Cell PPE (each core)	32x64		32x64	32x128
Cell SPE (each core)				128x128
CUDA CC 5.x (each SM)	1x (64|32)			128x512

Registers

- Register sets in typical processors

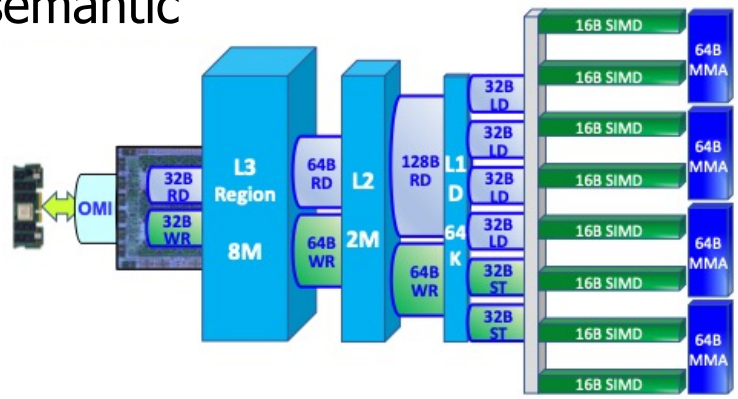
processor	Generic	Special	Float	SIMD
CUDA CC 8.x (128 SMs per chip)	4 x 16384 x 32	PC per SIMD cell -> nearly generic		4 x 512 x 1024
CUDA CC 12.x (12.0: 192 SMs per die)	II	II	II	II
IBM Power 1	32x32	1 link, 1 count	32x32	
OpenPOWER	32x64		32x64	32x128 SIMD
IBM Power 10 (15x or 30x per chip)	1xSMT 32x64	4x64 Matrix (4 ML pipes)	32x64	32x128 SIMD 8x512 SIMD
IBM Power 11 (16x or 30x per chip)	1xSMT 32x64	4x128 Matrix 4x2x512 Accu	?	64x128 SIMD

Registers

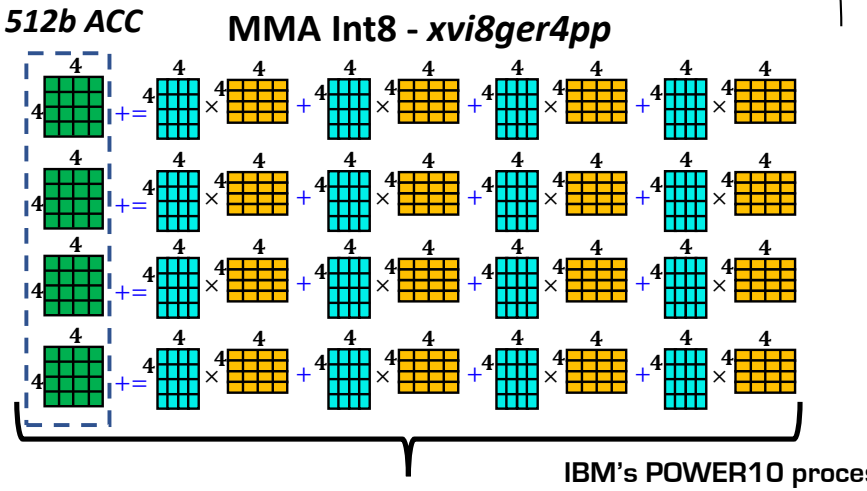
- Register sets in typical processors

processor	Generic	Special	Float	SIMD
IBM Power 10 (15x or 30x per chip)	1xSMT 32x64	4x64 Matrix (4 ML pipes)	32x64	32x128 SIMD 8x512 SIMD

Single use-only with a weird load/store semantic



Note: CUDA SM's similar but more flexible TensorCore is counted as a compute unit

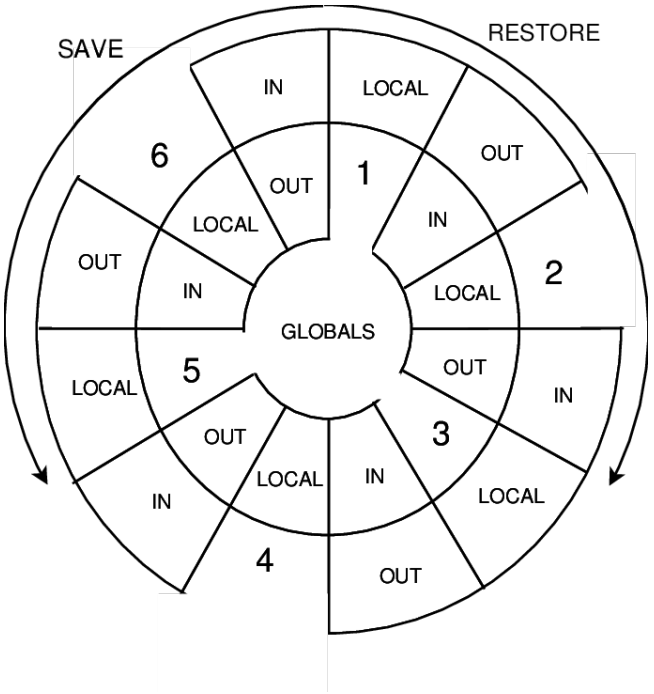


IBM's POWER10 processor

Willian Stark, Brian Thompto, *Hot Chips 32, 2020*

Registers

- Register windows
  - Idea: compiler must optimize for a subset of actually existing registers
  - Function calls and context changes do not require register spilling to RAM
  - Berkeley RISC: 64 registers, but only 8 visible per context
  - Sun Sparc: 3x8 in a shifting register book, 8 always valid



- **very useful** if OS should not yield on kernel calls
- **nice** with *upcall* concept
- **otherwise:** totally useless with multi-threading

CPU caches

- Effects of caches

		Latency		
		Intel Broadwell Xeon E5-2699v4	Intel Broadwell Xeon E5-2640v4	Intel Arrow Lake
Cache level	L1	4 cyc	4 cyc	4 cyc
	L1.5			9 cyc
	L2	12-15 cyc	12-15 cyc	17 cyc
	L3	4MB: 50 cyc	4 MB: 49-50 cyc	4MB: >80 cyc
Random load from RAM regions of size ...	16 MB	21 ns	26 ns	reportedly 70-100ns (up to 127ns)
	32-64 MB	80-96 ns	75-92 ns	
	96-128 MB	96 ns	90-91 ns	
	384-512 MB	95 ns	91-93 ns	

CPU caches

- Effects of caches

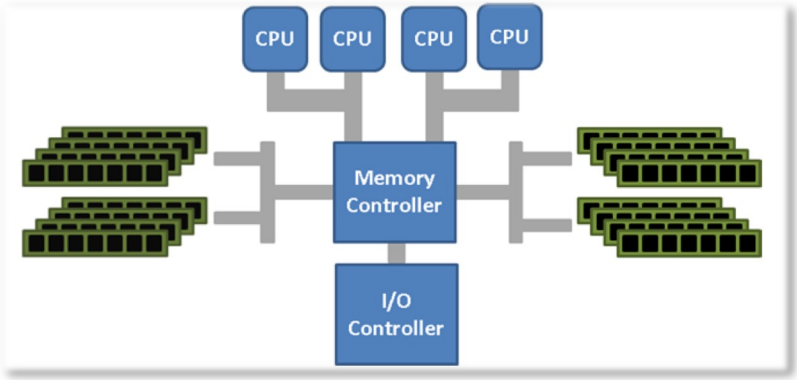
		Latency		
		Intel Broadwell Xeon E5-2699v4	Intel Broadwell Xeon E5-2640v4	IBM POWER8
Cache level	L1	4 cyc	4 cyc	3 cyc
	L1.5			
	L2	12-15 cyc	12-15 cyc	13 cyc
	L3	4MB: 50 cyc	4 MB: 49-50 cyc	8MB: 27-28 cyc
Random load from RAM regions of size ...	16 MB	21 ns	26 ns	55 ns
	32-64 MB	80-96 ns	75-92 ns	55-57 ns
	96-128 MB	96 ns	90-91 ns	67-74 ns
	384-512 MB	95 ns	91-93 ns	89-91 ns

Cache coherence

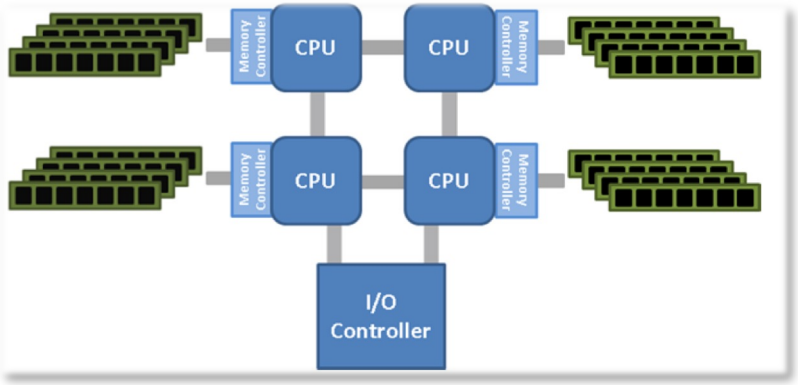
- Problems
  - User of one cache writes a cacheline, users of other caches read the same cacheline
    - Needs write propagation from dirty cache to other cache
  - Users of two caches write to the same cacheline
    - Needs write serialization among caches
    - Cacheline implies that sharing be true or false

- Cache coherence algorithms
  - Snooping
    - Based on broadcast
    - Write-invalidate or Write-update
  - Directory-based
    - Bit vector, tree structure, pointer chain
    - Intel Knights Mill: cache coherent interconnect with MESIF protocol

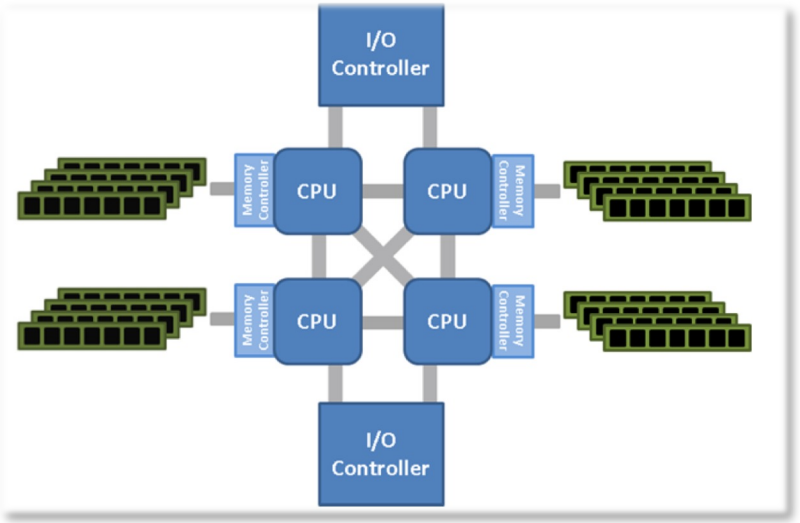
Main memory



Classical memory attachment,
e.g. Front-side bus
-
uniform addressing
but bottlenecked

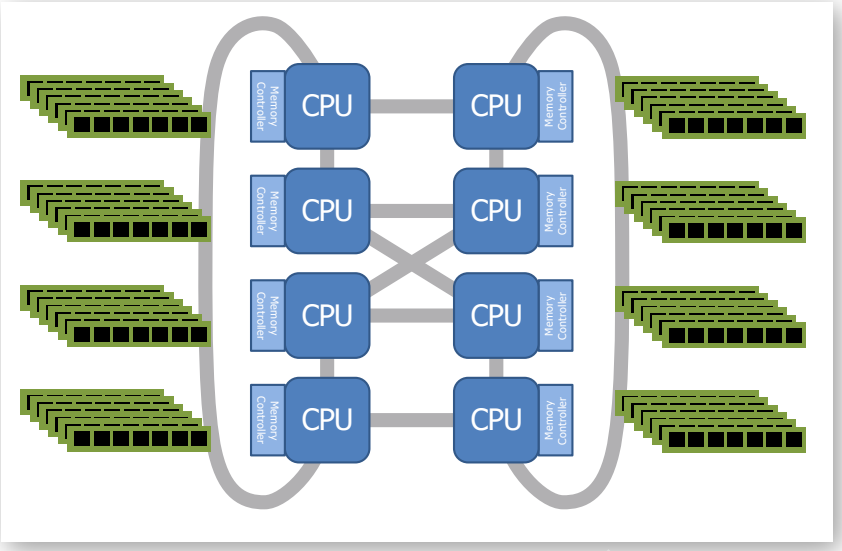


AMD HyperTransport
memory attachment



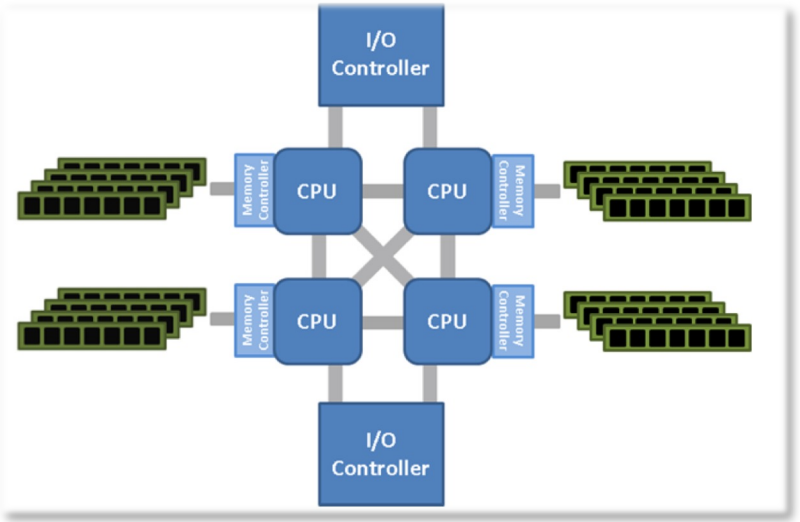
Intel Ultra Path Interconnect
memory attachment

Main memory



but only 3 channels per CPU,
so effectively that's a
cube-shape for 8 CPUs
(and 8 is max)

At the next hierarchy level, there are switches
Following the Omni Path Interconnect



Intel Ultra Path Interconnect
memory attachment

Types of memory: NUMA

- Same memory hierarchy level - different access latency
- AMD Hypertransport hypercube interconnect example
  - Memory banks attached to different CPUs
  - Streaming between CPUs in hypercube topology (Fully mesh in QPI helps)
  - Living with it
    - Pthread NUMA extensions to select core attachment
    - Problem with thread migration
    - Problem with some «nice» programming paradigms such as actor systems

Types of memory: non-hierarchical

- Historical Amiga
  - Chip memory
    - shared with graphics chips
    - accessible for Blitter
  - Local memory
    - onboard memory
  - Fast memory
    - not DMA-capable, only CPU-accessible (CPU *never* shares bus cycles)
  - DMA memory
    - accessible for Blitter
  - Any memory
    - on riser cards over Zorro bus, very slow

like a DMA engine only for memory, but also boolean operations on blocks of memory

PCIe's follow-up standard
CXL will bring us a lot of this !

Types of memory: throughput vs latency

- IXP 1200
  - Explicit address spaces
    - latency, throughput, plus price
    - Scratchpad – 32bit alignment, to CPU only, very low latency
    - SRAM – 32bit alignment, 3.7Gbps throughput read, 3.7 Gbps write, to CPU only, low latency
    - SDRAM – 64bit alignment, 7.4Gps throughput, to all units, higher latency
  - Features
    - SRAM/SDRAM operations are non-blocking
      hardware context switching hides access latency
      (the same happens in CUDA)
    - «Byte aligner» unit from SDRAM

not unified at all:
different assembler instructions for
Scratchpad, SRAM and SDRAM

- Intel sold the IXP family to Netronome
- renamed to NFP-xxxx chips, 100 Gbps speeds
- C compiler is still broken in 2020

Types of memory: throughput vs latency

- Netronome NFP-4000 (Agilio CX)
  - Grand-grand-grandchild of the IXP 1200

location	name		size	access time
Microengine	CS	code store	32 kB	
	LM	local memory	4 kB	1-3 cycles
Island	CLS	cluster-local scratch	64 kB	20-50 cycles
	CTM	cluster target memory	256 kB	50-100 cycles
Global	IMEM	internal memory	4 MB	150-250 cycles
	EMEM	external memory	2 GB	150-500 cycles

Hardware-supported atomicity

- Ensure atomicity of memory across cores

- Transactional memory
  - e.g. Intel TSX (Transactional Synchronization Extensions)
    - available in Core 4, 8 and 9, definitely removed since Core 10
    - *Hardware lock elision*: prefix for regular instruction to make them atomic
    - *Restricted transactional memory*: extends to an instruction group including up to 8 write operations
  - e.g. IBM Power 11
    - To be used conjunction with the compiler
    - Transaction tbegin (full/short), tend, tabort, tsuspend, tresume, tcheck
    - Transaction fails in case of access collision within a transaction, exceeding a nesting limit, or exceeding a writeback buffer size
  - Allow speculative execution with memory writes
  - Conduct memory state rewind and instruction replay of transaction whenever serializability is violated

Hardware-supported atomicity

- Ensure atomicity of memory across cores

- IXP hardware queues
  - push/pop operations on 8 queues in SRAM
  - 8 hardware mutexes
  - atomic bit operations
  - write transfer registers, read transfer registers for direct communication between microengines (cores)

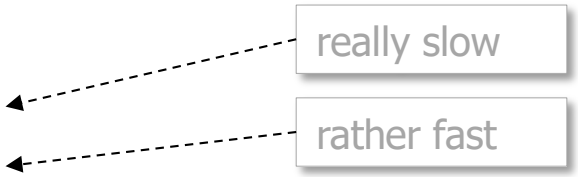
"push/pop" is badly named: they are queues, not stacks

- Cell hardware queues
  - Mailbox to send queued 4-byte messages from PPE to SPE and reverse
  - Signals to post unqueued 4-byte messages from one SPE to another

Hardware-supported atomicity

- Ensure atomicity of memory across cores

- CUDA
  - Atomic in device memory
  - Atomic in shared memory
  - Atomic in all managed memory (incl. host)
  - shuffle, shuffle-op, ballot, reduce
  - cooperative groups
    - sync freely defined subset of a threads in block
    - multiple blocks
    - multiple devices



Content-addressable memory

- Content addressable memory (CAM)
  - solid state memory
  - data accessed by content instead of address
- Design
  - simple storage cells
  - each bit has a comparison circuit
  - Inputs to compare: argument register, key register (mask)
  - Output: match register (1 bit for each CAM entry)
  - READ, WRITE, COMPARE operations
  - COMPARE: set bit in match register for all entries where Hamming distance between entry and argument is 0 at key register positions
- Applications
  - Typical in routing tables
    - Write a destination IP address into a memory location
    - Read a forwarding port from the associated memory location

TCAM (ternary CAM) merges argument and key into a query register of 3-state symbols

Swapping and paging

- Memory overlays
  - explicit choice of moving pages into a physical address space
  - works without MMU, for embedded systems
- Swapping
  - virtualized addresses
  - preload required pages, LRU later
- Demand paging
  - like swapping, but lazy loading

Swapping and paging

- More physical memory than CPU address range
  - Commodore C128
    - one register for page mapping
    - set page choice globally
  - X86 ”Real Mode”
    - Select page at the “instruction group level”
    - Two address registers, some bits from one for the prefix, some bits from the other for the offset
  - Pentium Pro
    - Physical address extension
    - MMU gives each process up to 2^32 bytes of addressable memory
    - But the MMU can manage up to 2^64 bytes of physical memory
  - ARMv7
    - Large Physical Address Extension
    - As above, but only up to 2^40 bytes of physical memory

DMA – direct memory access

- DMA controller
  - Initialized by a compute unit
  - Transferring memory without occupying compute cycles
  - Usually commands can be queued
  - Can often perform scatter-gather transfer, mapping to and from physically contiguous memory
- Modes
  - Burst mode: use the bus fully
  - Cycle stealing mode: share bus cycles between devices
  - Background mode: use only cycles unused by the CPU
- Challenges
  - Cache invalidation
  - Bus occupation
  - No understanding of virtual memory

SCSI DMA, CUDA cards, Intel E10G Ethernet cards: own DMA controller, priorities implementation-defined

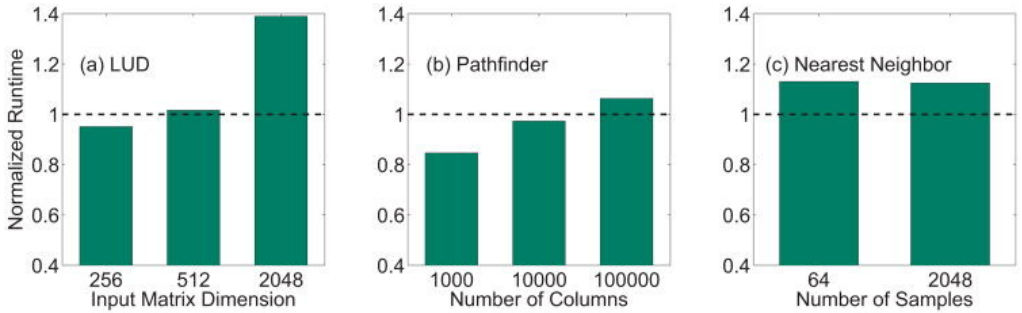
Amiga Blitter (bus cycles = 2x CPU cycles)

ARM11 – background move between main memory and on-chip high-speed TCM (tightly coupled memory)

Automatic CPU-GPU memory transfer

- CUDA Managed Memory
  - Allocates a memory block like malloc
  - Can be accessed from CPU and from GPU
    - Easy on shared-memory designs (laptops, special HPC designs) "unified memory"
    - Hard on computers where CPU and GPU have different memories
- DMA-capable
- Compiler and runtime decide when to move data

Rodinia benchmark,
Normalized to hand-coded speed
DOI 10.1109/HPEC.2014.7040988



Battery-backed RAM

- Battery-backed RAM is a form of disk
- Faster than SSD, limited by bus speed
- Requires power for memory refresh
- Not truly persistent



Nothing like this was recently marketed. Replaced by HDDs with a Flash buffer for persistence.

Hybrid BBRAM/Flash
- Speed of BBRAM
- Battery for short-term persistence
- SD card for long-term persistence

ALLONE Cloud Disk Drive 101 RAMDisk – 32GB capacity

SSD - Solid Stage Disks storage

- Emulation of «older» spinning disks
- in terms of latency between RAM and spinning disks
- limited lifetime is a major challenge



- With NVMe
  - e.g. M.2 SSDs
  - Memory mapping
  - DMA
  - could make serial device driver model obsolete



Persistent memory

- Intel Optane persistent memory (discontinued)
  - SSD package on the DRAM bus
  - Can be partly volatile or partly persistent
    - Reconfigure and reboot
  - Faster than SSD, slower than DRAM

- Drop-in replacement for RAM
  - Bandwidth and access times similar to RAM (factor 2-4)
  - Persistence of an NVMe disk
  - Desired for object-oriented paradigms for decades, e.g. OODBs
  - Too rare for a well-integrated programming paradigm



Persistent memory

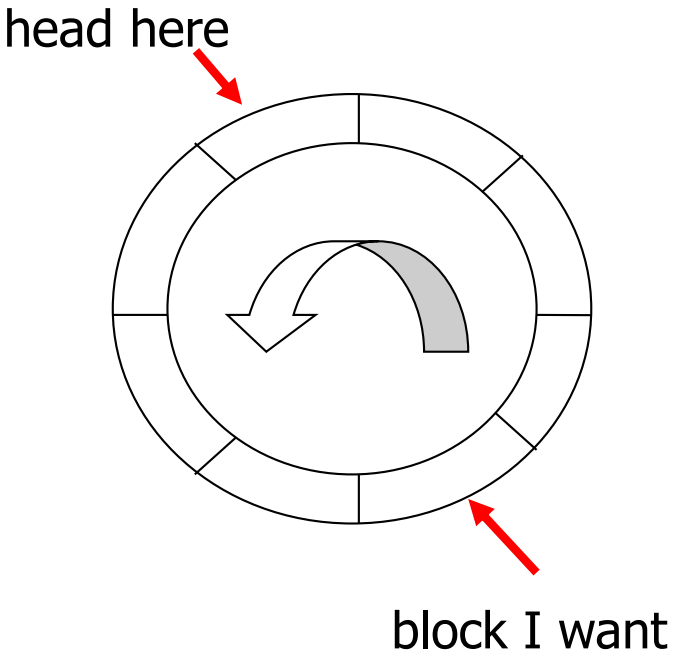
- Intel Optane persistent memory
  - SSD package on the DRAM bus
  - Can be partly volatile or partly persistent
    - Reconfigure and reboot
  - Faster than SSD, slower than DRAM

- Using persistent memory
  - ndctl: vmalloc
  - opmctl: command line
  - PMDK: libraries, C++ bindings
    - Persistence
    - Persistent smart pointers
    - Transactions
    - Synchronization
    - Containers



Hard disks

- Baseline for persistent storage
  - Persistent
  - Re-writable
  - Moderate throughput
  - Decent lifetime



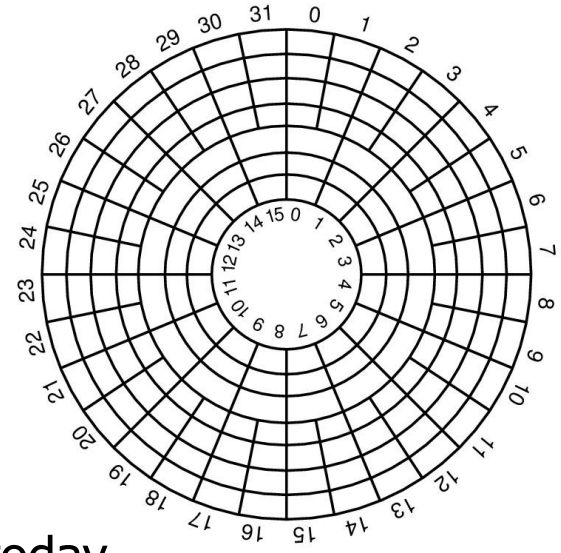
Average delay is 1/2 revolution

"Typical" average:

8.33 ms (3.600 RPM)
5.56 ms (5.400 RPM)
4.17 ms (7.200 RPM)
3.00 ms (10.000 RPM)
2.00 ms (15.000 RPM)

Disk storage

- Partitions
  - Location matters on spinning disks
    - High sector density on outside of disk gives more throughput per rotation



Typical disk today
- Zoned Constant-Area Velocity Disk
- Spare sectors at the edge of each zone

- Priorities between disks
  - SCSI / SAS / FibreChannel disks
    - Prioritized Ids – gives DMA priority
  - PCIe-attached storage blades
    - Non-standardized priority

note that some zCAV disks are slowing down a bit on outer cylinders:
reduces transfer errors

Disk storage

- Filesystem considering location
  - JFS
    - PV - physical volume
      - a disk or RAID system
    - VG - volume group
      - physical volumes with softraid policy
      - enables physical move of disks
    - LV - logical volume
      - partition-like hot-movable entity on a VG
      - **can prefer specific disk regions**, also non-contiguous
      - another softraid layer
    - FS – file system
      - can grow and shrink

like Linux UUID for disks

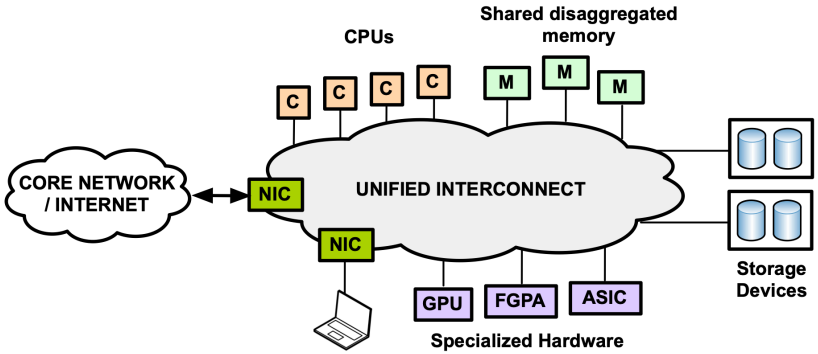
Tape drives

- Features
  - High capacity (LTO9: 10 TB uncompressed/tape)
  - Excellent price/GB (0,10 NOK)
  - Very long lifetime (1970s tapes still working)
  - Decent throughput due to high density (1 TB/hr)
  - Immense latency
    - Robotic tape loading
    - Spooling
- Famous example
  - StorageTek Powderhorn
- Google uses Tape for backup

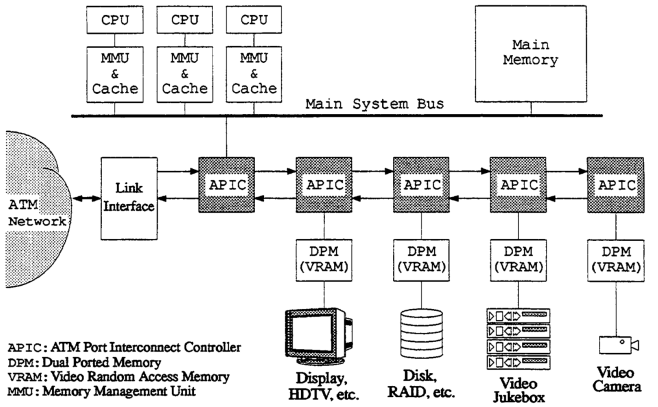
Resource disaggregation

- Build computers from infrastructure inside a cluster
  - Build virtual machine
  - From physical hardware
  - Page mapping across a network infrastructure
  - Different attachment for every kind of device
  - Memory integrated by an MMU
    - MMU is part of the CPU

- Ancient vision
  - ATM as internal and external interconnect
  - i.e. a unified interconnect



Network support for resource disaggregation in next-generation datacenters
Sangjin Han, Norbert Egi, Aurojit Panda, Sylvia Ratnasamy, Guangyu Shi, Scott J. Shenker
ACM HotNets 2013



Using an ATM interconnect as a high performance I/O backplanet
Z.D. Dittia, J.R. Cox, G.M. Parulkar
IEEE Hot Interconnects II, 1993

Networked file systems

- Remote file system
  - e.g. NFS - Network file system, CIFS – Common Internet file system
  - Limited caching, file locking, explicit mount like a separate file system
- Distributed file system
  - e.g AFS – Andrews file system, DFS – DCE file system
  - Aggressive caching, unique global directory structure
- Cluster file systems
  - e.g. HadoopFS, GPFS – General parallel file systems, Lustre
  - Sacrifice ease of use for bandwidth
- P2P file systems
  - e.g. OceanStore – files striped across peer nodes, location unknown
  - Sacrifice speed (latency and bandwidth) for resilience and anonymity