SC20

Everywhere we are | more than hpc.

General I/O and Persistent Memory Hardware

Adrian Jackson
EPCC
a.jackson@epcc.ed.ac.uk
@adrianjhpc

epcc

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

I/O

- I/O essential for all applications/codes
  - Some data must be read in or produced
  - Instructions and Data
- Small parallel programs (i.e. under 1000 processors)
  - Cope with I/O overhead
- Large parallel programs (i.e. tens of thousand processors)
  - Can completely dominate performance
  - Exacerbate by poor functionality/performance of I/O systems
- Any opportunity for program optimisation important
  - Improve performance without changing program

Challenges of I/O

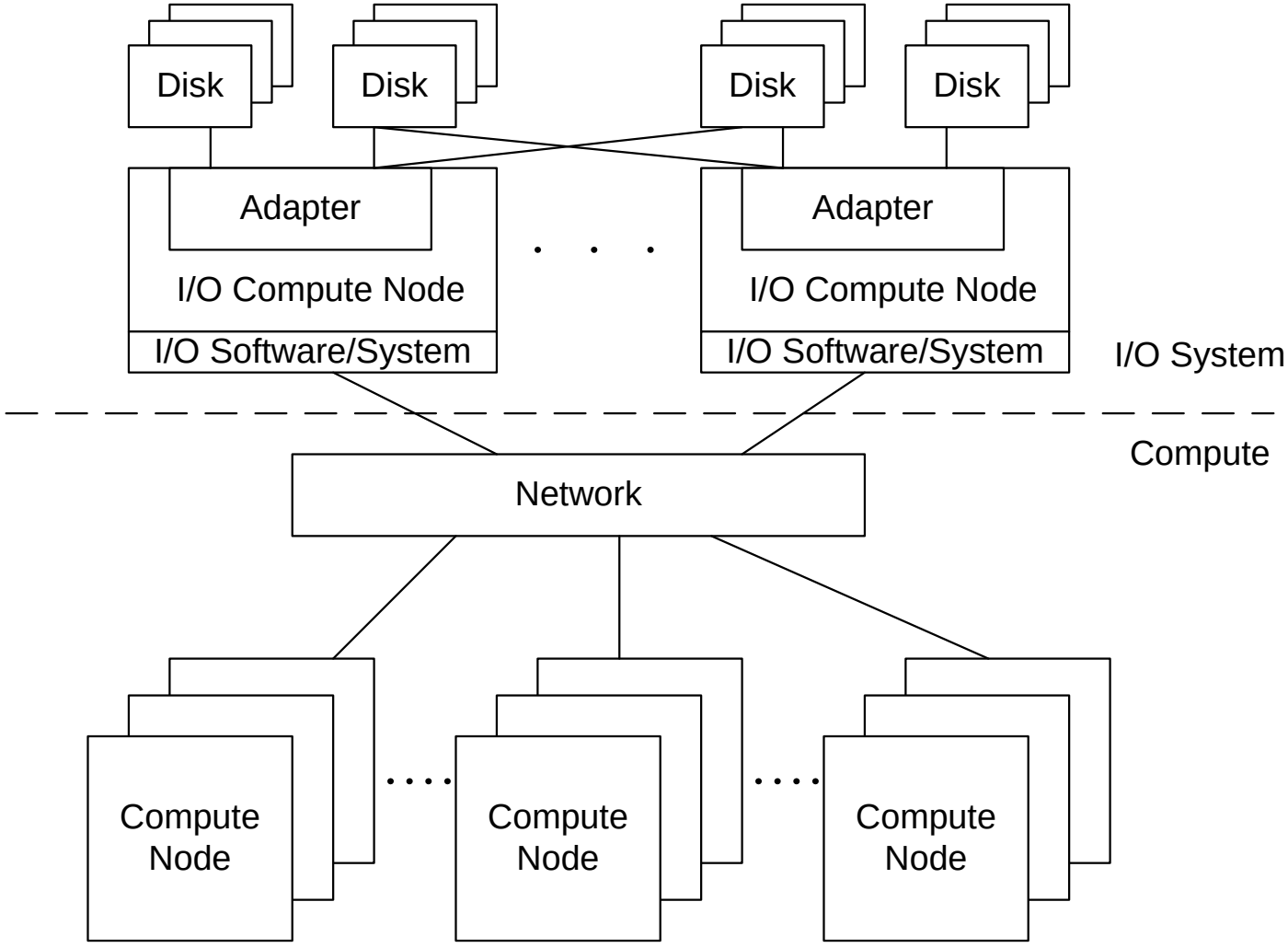
- Moves beyond process-memory model
  - data in memory has to physically appear on an external device
- I/O operations are generally optimised (or require for good performance) for bulk activities
- Files are very restrictive
  - Don’t often map well to common program data structures (i.e. flat file/array)
  - Often no description of data in file
- I/O libraries or options system specific
  - Hardware different on different systems
- Lots of different formats
  - text, binary, big/little endian, Fortran unformatted, ...
  - Different performance and usability characteristics
- Disk systems are very complicated
  - RAID disks, caching on disk, in memory, I/O nodes, network, etc…

Interface	Throughput Bandwith (MB/s)
SATA	600
NVMe	2,000+
Fibre	6,000+

Parallel I/O

- Lots of different methods for providing high performance I/O
- Hard to support multiple processes writing to same file
  - Basic O/S does not support
  - Data cached in units of disk blocks (eg 4K) and is *not coherent*
  - Not even sufficient to have processes writing to distinct parts of file
- Even reading can be difficult
  - 1024 processes opening a file can overload the filesystem limit on file handles etc….
- Data is distributed across different processes
  - Dependent on number of processors used, etc…
- Parallel file systems may allow multiple access
  - but complicated and difficult for the user to manage

Parallel Filesystems



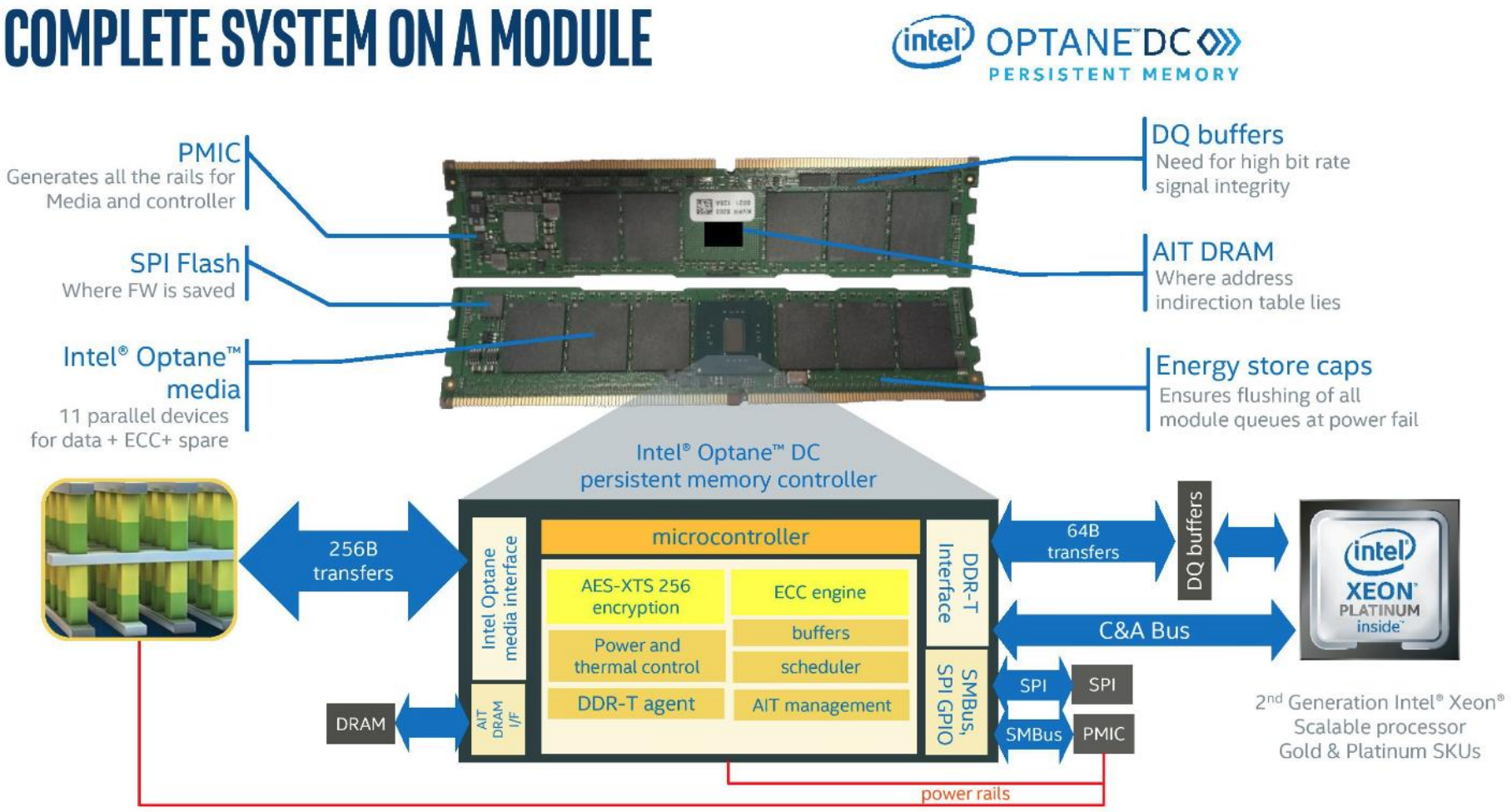
Persistent/Non-volatile memory

- Persistent/Non-volatile memory stores data after power off
  - SSDs (NAND Flash) are common examples
  - Similar technology in memory cards for your phones, cameras, etc…
- These store data persistently but are generally slow and less durable than volatile memory technologies (i.e. DDR memory)
- Traditional non-volatile technology is accesses through a block device interface
  - Chunks of data at a time (i.e. 4kb), asynchronous access

NVDIMMs

- JEDEC standard on non-volatile memory DIMMs
- NVDIMM-F
  - Traditional flash solution with controller on board
  - NAND flash performance and size
- NVDIMM-N
  - DRAM with Flash for backup
  - Separate power supply (i.e. super capacitors) allow data to be copied to flash on power failure
  - Limited by DRAM size and capacitor sizes
  - DRAM performance and size
- NVDIMM-P
  - Channel support for mixed memory types
  - Protocol to enable transactional access
  - Different access latencies allowed between media types
  - Intel Optane DCPMM, technically, does not implement the NVDIMM-P standard, but it is conceptually NVDIMM-P
  - What I call Byte-Addressable Persistent Memory (B-APM)

Intel Optane DCPMM

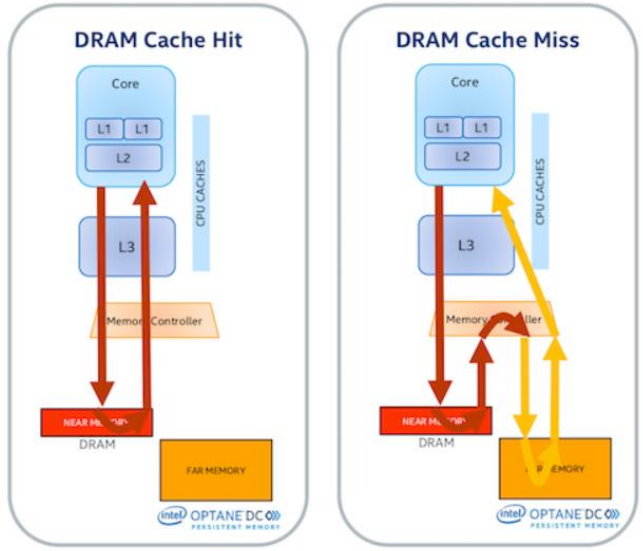


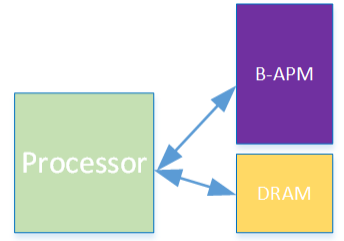
Intel Optane DCPMM

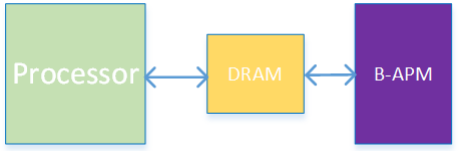
- Persistent/Non-volatile RAM/B-APM
  - Optane memory
- Much larger capacity than DRAM
  - Hosted in the DRAM slots, controlled by a standard memory controller
- Slower than DRAM by a small factor, but significantly faster than SSDs
- Read/write asymmetry and other interesting performance factors
- High endurance (5 year warranty)

Intel Optane DCPMM

- Requires specific processors to support the hardware
  - PM-enabled memory controller required
  - Deal with different latency memory bus traffic
- Has different modes of operation
- 1LM – App Direct
  - Both memories are visible to the program
  - Using the PM requires program changes
- 2LM – Memory Mode
  - DRAM used as Last Level Cache (LLC) for PM
  - Transparent exploitation but no persistence

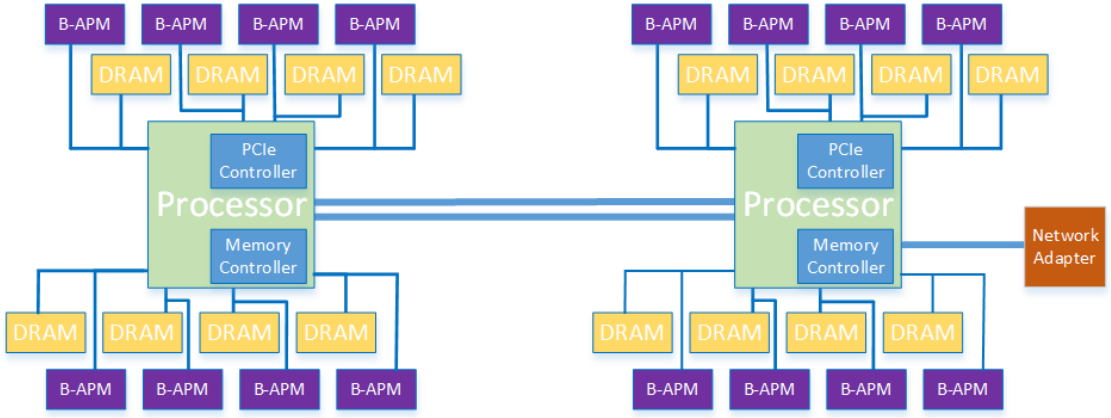






Placement

- As B-APM is in-node, placement is important
  - Configuration can be variable
  - Currently need one DRAM per memory channel
  - Can match DRAM-B-APM or have more B-APM
- Capacity (both DRAM and B-APM) affected
- Memory controller deals with mixed configuration
  - Variable latency on memory DIMMs
  - Requires asynchronous or non-blocking memory operations
- Optane DIMMs can be striped on non-striped
  - One memory area per DIMM, or one memory area per socket, striped across DIMMs



Optane DCPMM

- Cache coherent data accesses
  - Byte addressable (cache line)

- Requires reboot before switching platform mode
  - Memory mode (2LM)
  - App direct (1LM)
    - `fsdax`
      - Filesystem block device for creating namespace
    - `devdax`
      - Character device, no namespace
    - Performance identical once file is created
  - Can partition system to have both memory and app direct spaces

Optane DCPMM – NUMA issues

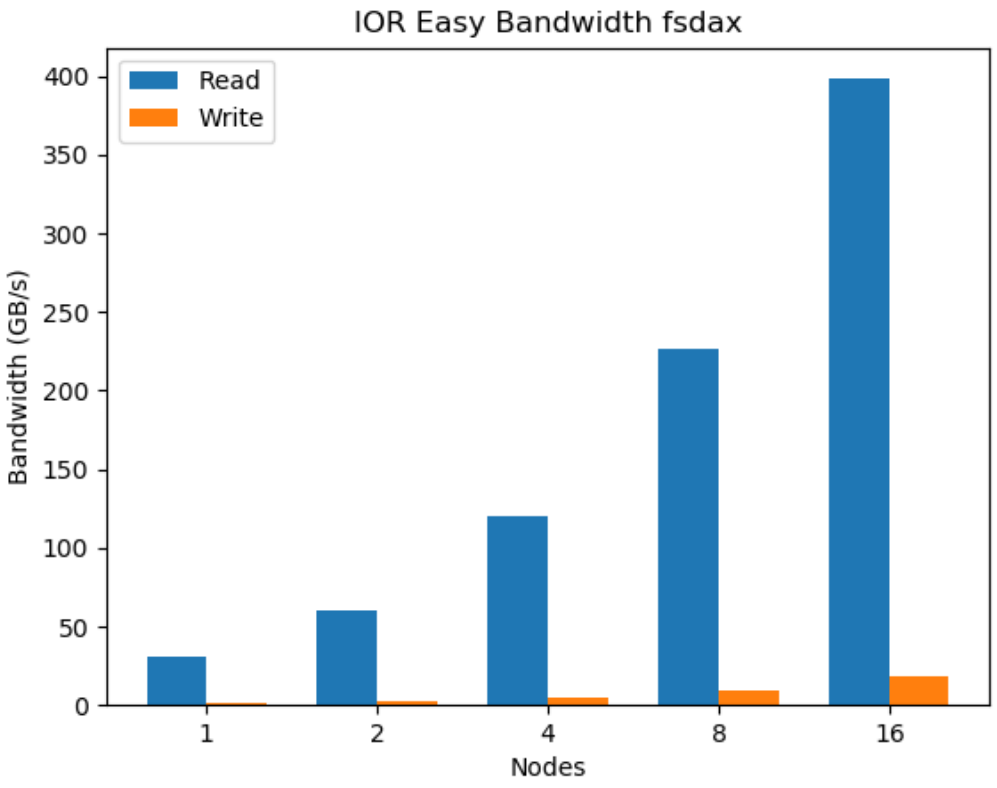
- Socket based systems means NUMA when not a single socket system
  - Performance dependent on using local memory
- Factor ~4x write performance for using local memory when fully populating nodes
- Factor ~2x read performance for using local memory when fully populating nodes
- Getting NUMA information in an application
  - Intel specific:

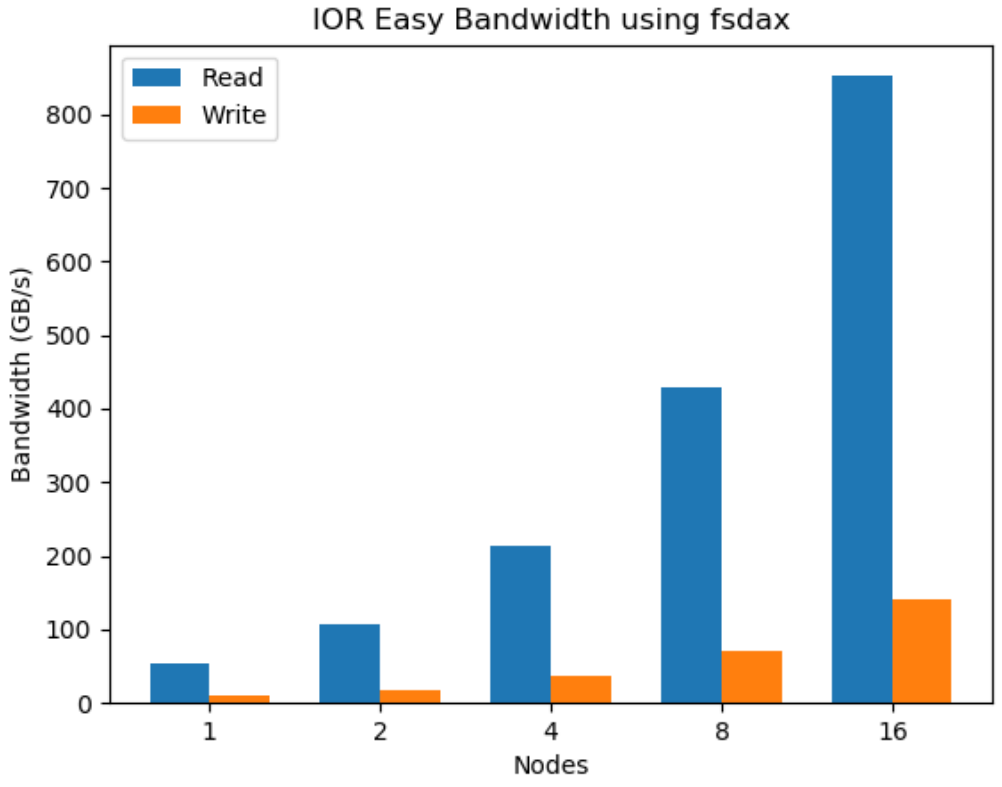
```
unsigned long GetProcessorAndCore(int *chip, int *core){
   unsigned long a,d,c;
   __asm__ volatile("rdtscp" : "=a" (a), "=d" (d), "=c" (c));
   *chip = (c & 0xFFF000)>>12;
   *core = c & 0xFFF;
   return ((unsigned long)a) | (((unsigned long)d) << 32);;
}
```

  - Processor agnostic

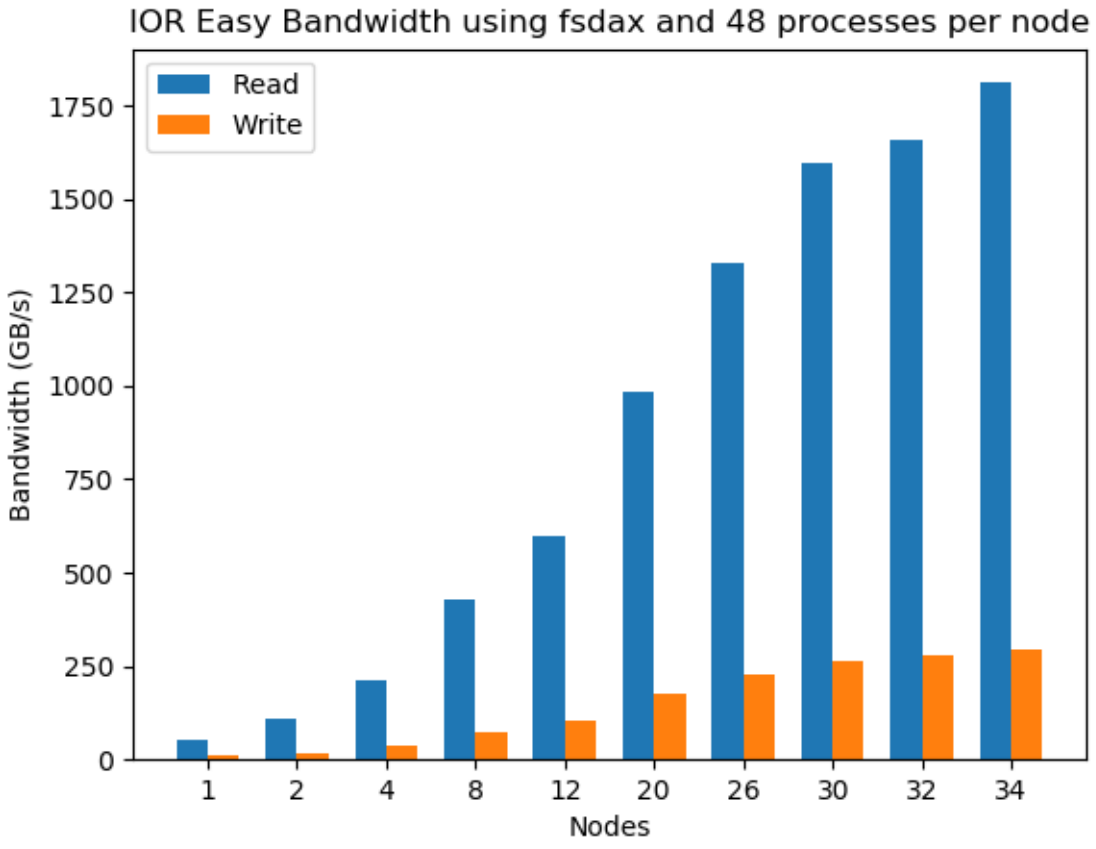
```
unsigned long GetProcessorAndCore(int *chip, int *core){
   return syscall(SYS_getcpu, core, chip, NULL);
}
```

Optane NUMA performance





Optane performance



Optane Performance asymmetry

- Read is ~3x slower than DRAM
- Write is ~7x slower than DRAM
- Read is 4x-5x faster than write for Optane
- Write queue issues can mean variable performance
  - Optane has active write management
  - On-DIMM controller
- Accesses are coalesced into blocks of i.e. 256 bytes

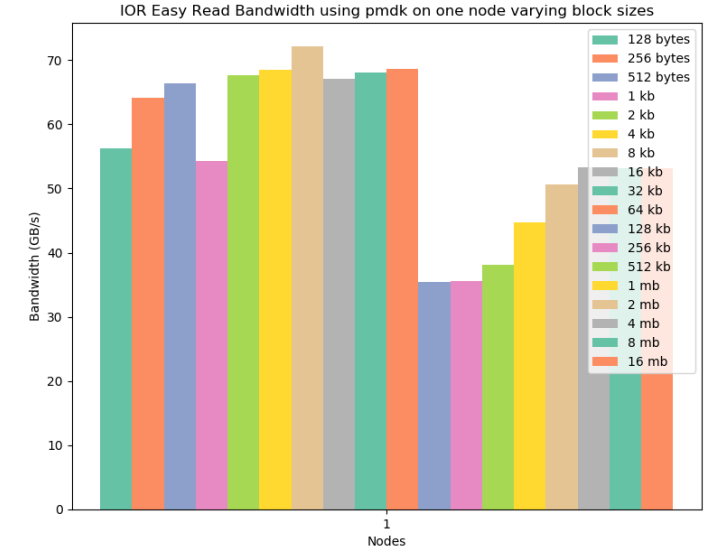
INTEL® OPTANE™ DC PERSISTENT MEMORY PERFORMANCE DETAILS

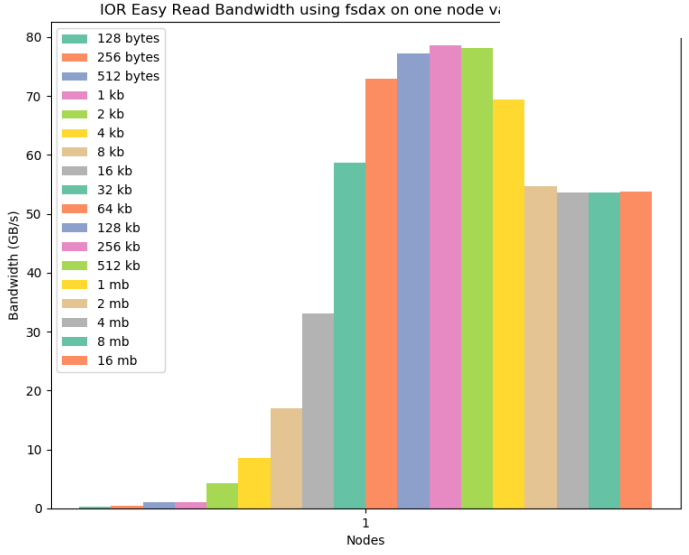
- Intel® Optane™ DC persistent memory is programmable for different power limits for power/performance optimization
  - 12W – 18W, in 0.25 watt granularity - for example: 12.25W, 14.75W, 18W
  - Higher power settings give best performance
- Performance varies based on traffic pattern
  - Contiguous 4 cacheline (256B) granularity vs. single random cacheline (64B) granularity
  - Read vs. writes
  - Examples:

Granularity	Traffic	Module	Bandwidth
256B (4x64B)	Read	256GB, 18W	8.3 GB/s
256B (4x64B)	Write		3.0 GB/s
256B (4x64B)	2 Read/1 Write		5.4 GB/s
64B	Read		2.13 GB/s
64B	Write		0.73 GB/s
64B	2 Read/1 Write		1.35 GB/s

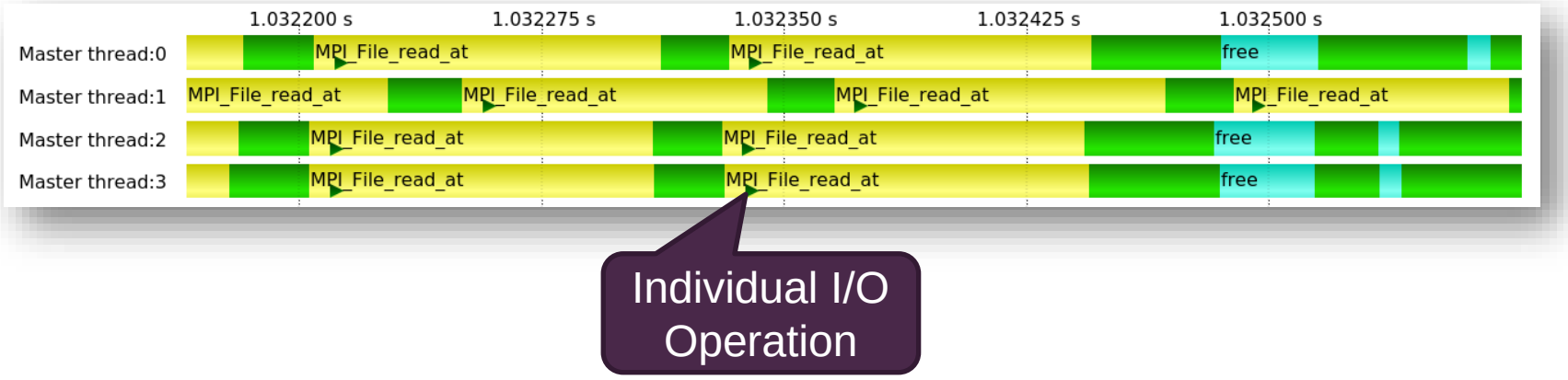
Byte Addressable/Granularity

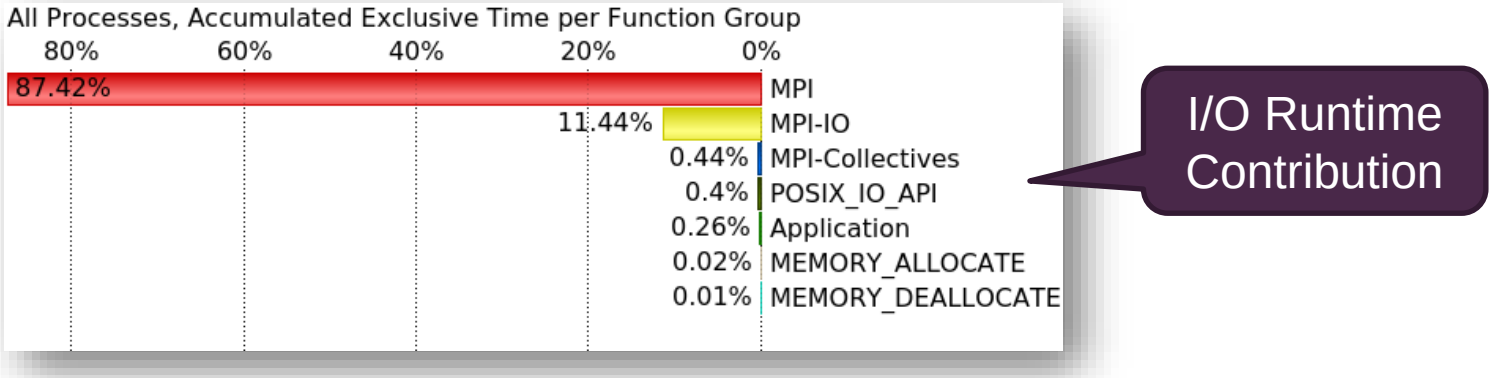
- A key feature of B-APM
  - Data access is the same cost for any size of operation
    - Not really, but much closer than for standard files
- Byte-addressable just like standard memory
  - Individual bytes accessible without large operations
  - In reality, cache-line level access
- I/O and data operations can be small
  - Restructure I/O/applications





Granularity





Summary

- Optane hardware is complicated
- Performance is workload dependent (when isn't it?)
- Targeted usage will be required for the best performance
- I/O performance has been problematic for a while anyway

Pricing

- List prices (old information now)

Optane DPCMM

Size (GB)	Cost (€)	€/GB
128	2223	17.36
256	7638	29.83
512	23199	45.31

DRAM

Size (GB)	Cost (€)	€/GB
8	347	43.37
16	444	27.78
32	855	26.71
64	2063	32.23
128	4959	38.74