I/O Profiling Towards the Exascale

holger.brunst@tu-dresden.de

ZIH, Technische Universität Dresden

NEXTGenIO & SAGE: Working towards Exascale I/O

Barcelona, May 19th, 2017

NEXTGenIO facts

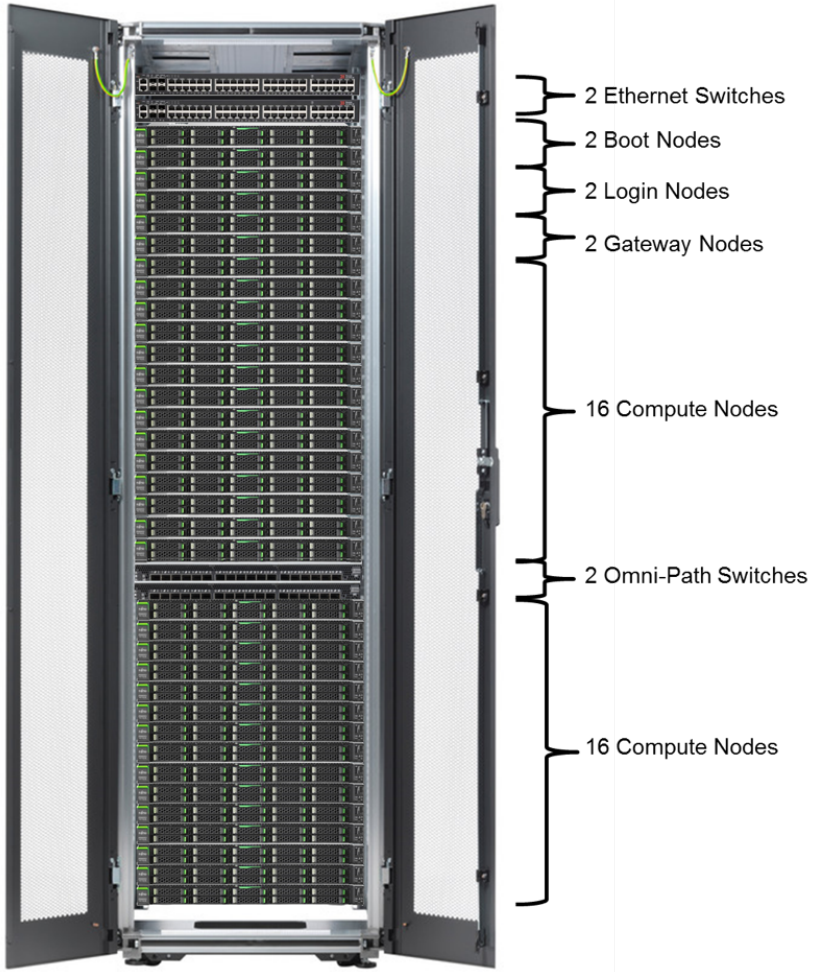
Project

- Research & Innovation Action
- 36 month duration
- €8.1 million

Partners

- EPCC
- INTEL
- FUJITSU
- BSC
- TUD
- ALLINEA
- ECMWF
- ARCTUR

Approx. 50% committed to hardware development



- Note: final configuration may differ

Intel™ DIMMs are a key feature

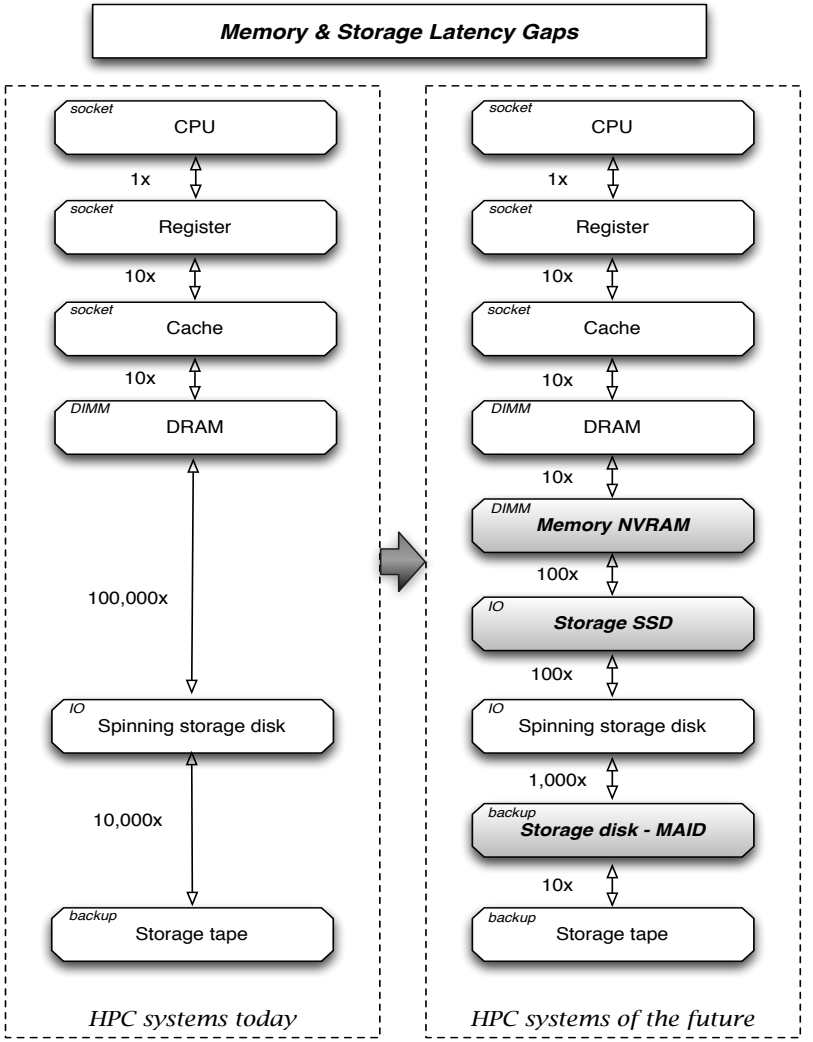
- Non-volatile RAM
  - 3D XPoint technology
- Much larger capacity than DRAM
- Slower than DRAM
  - By a certain factor
  - Significantly faster than SSDs ™
- 12 DIMM slots per socket
  - Combination of DDR4 and Intel™ DIMMs

Three usage models

- The “memory” usage model
  - Extension of the main memory
  - Data is *volatile* like normal main memory
- The “storage” usage model
  - Classic *persistent* block device
  - Like a very fast SSD
- The “application direct” usage model
  - Maps *persistent* storage into address space
  - Direct CPU load/store instructions

New members in memory hierarchy

- New memory technology
- Changes the memory hierarchy we have
- Impact on applications e.g. simulations?
- I/O performance is one of the critical components for scaling up HPC applications and enabling HPDA applications at scale

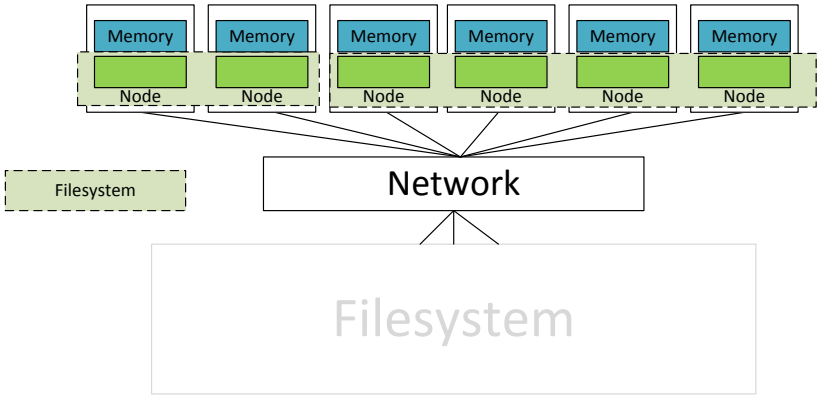


Remote memory access on top

- Network hardware will support remote access
- Data in NVDIMMs
  - To be shared between nodes
- Systemware
  - Support remote access
  - Data partitioning and replication

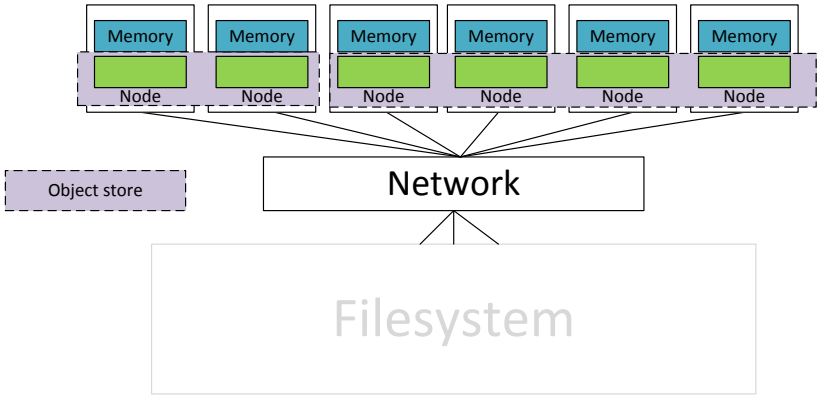
Using distributed storage

- Global file system
  - No changes to apps
- Required functionality
  - Create and tear down file systems for jobs
  - Works across nodes
  - Preload and postmove filesystems
  - Support multiple filesystems across system
- I/O Performance
  - Sum of many layers



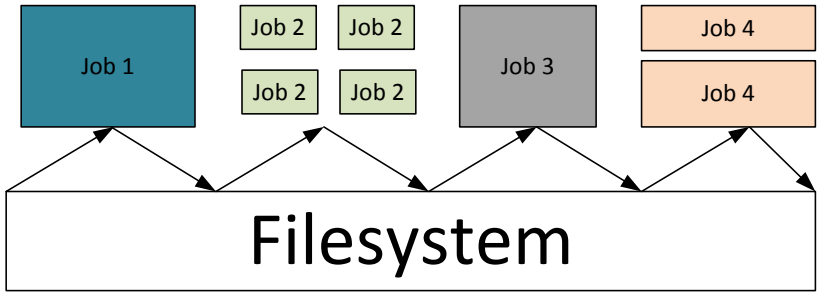
Using an object store

- Needs changes in apps
  - Needs same functionality as global filesystem
  - Removes need for POSIX functionality
- I/O Performance
  - Different type of abstraction
  - Mapping to objects
  - Different kind of Instrumentation



Towards workflows

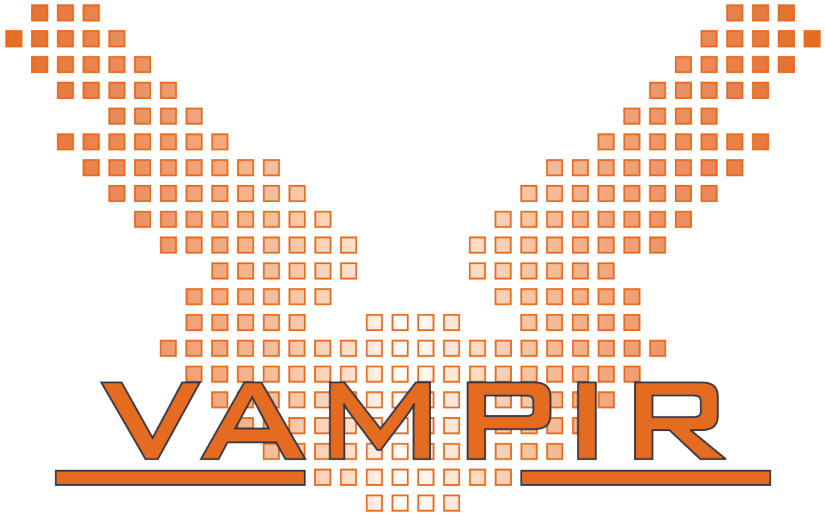
- Resident data sets
  - Sharing preloaded data across a range of jobs
  - Data analytic workflows
  - How to control access/ authorisation/security/ etc….?
- Workflows
  - Producer-consumer model
  - Remove file system from intermediate stages
- I/O Performance
  - Data merging/integration?



Tools have three key objectives

- Analysis tools need to
  - Reveal performance interdependencies in I/O and memory hierarchy
  - Support workflow visualization
  - Exploit NVRAM to store data themselves
  - (Workload modelling)

Vampir & Score-P



How to meet the objectives?

- File I/O, NVRAM performance
  - Monitoring (data acquisition)
    - Sampling
    - Tracing
  - Statistical analysis (profiles)
  - Time series analysis
- Multiple layers
  - Simultaneously
  - Topology context
- Workflow support
  - Merge and relate performance data

- Data sources

Tapping the I/O layers

- I/O layers
  - POSIX
  - MPI-I/O
  - HDF5
  - NetCDF
  - PNetCDF
  - File system (Lustre, Adios)
- Data of interest
  - Open/Create/Close operations (meta data)
  - Data transfer operations

What the NVM library tells us

- Allocation and free events
- Information
  - Memory size (requested, usable)
  - High Water Mark metric
  - Size and number of elements in memory
- NVRAM health status
  - Not measurable at high frequencies
- Individual NVRAM load/stores
  - Remain out of scope (e.g. memory mapped files)

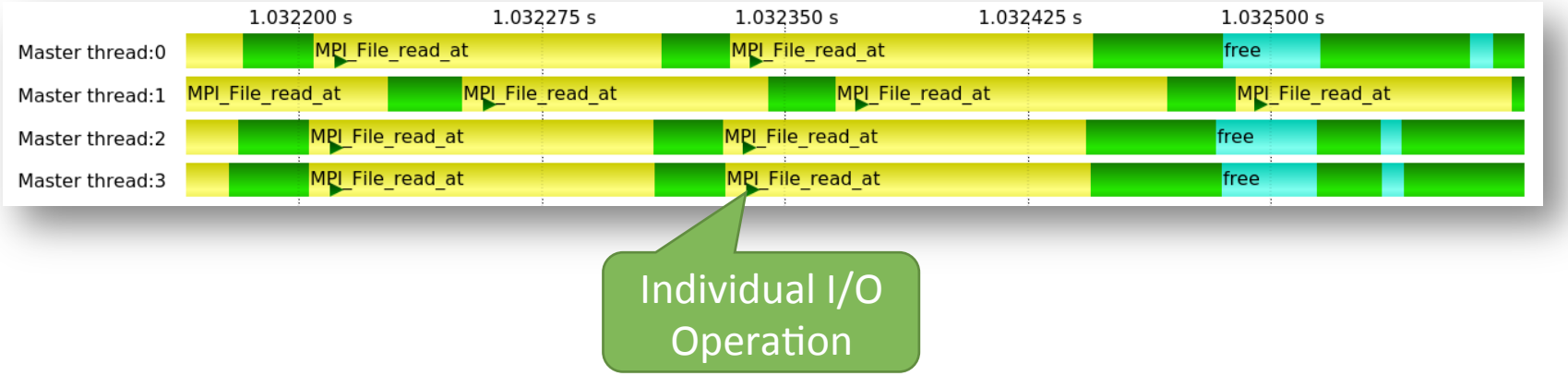
Memory Access Statistics

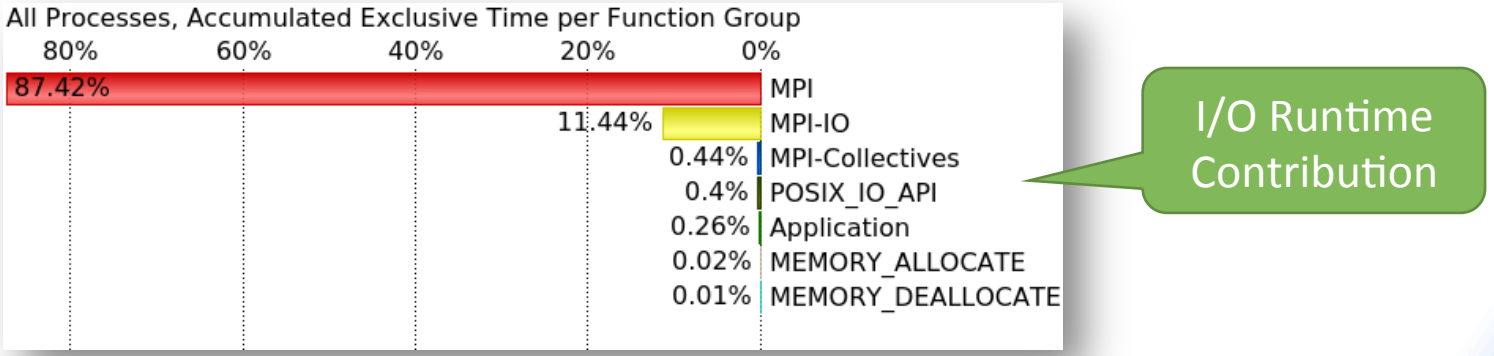
- Memory access hotspots for using DRAM and NVRAM?
  - Where? When? Type of memory?
- Metric collection needs to be extended
  1. DRAM local access
  2. DRAM remote access (on a different socket)
  3. NVRAM local access
  4. NVRAM remote access (on a different socket)

Access to PMU using perf

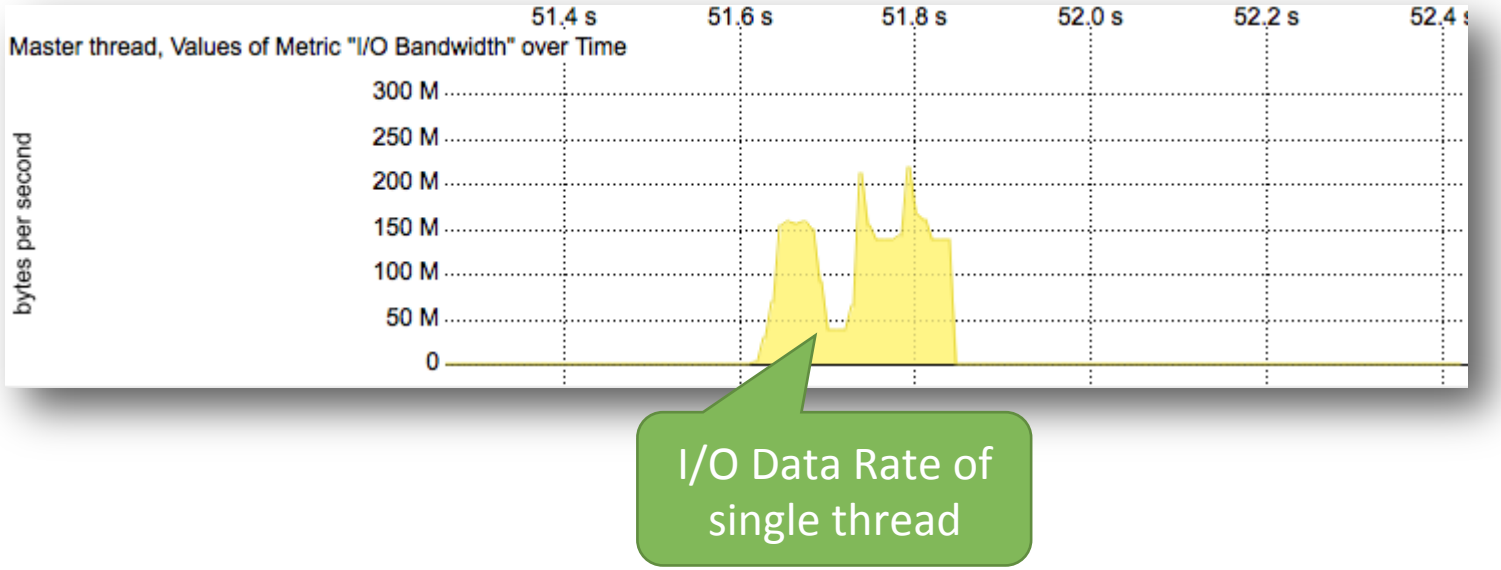
- Architectural independent counters
  - May introduce some overhead
    - MEM_TRANS_RETIRED.LOAD_LATENCY
    - MEM_TRANS_RETIRED.PRECISE_STORE
    - Guess: It will also work for NVRAM?
- Architectural dependent counters
  - Counter for DRAM
    - MEM_LOAD_UOPS_L3_MISS_RETIRED.REMOTE_DRAM
    - MEM_LOAD_UOPS_L3_MISS_RETIRED.LOCAL_DRAM
    - MEM_LOAD_UOPS_*.REMOTE_NVRAM ?
    - MEM_LOAD_UOPS_*.LOCAL_NVRAM ?

I/O operations over time

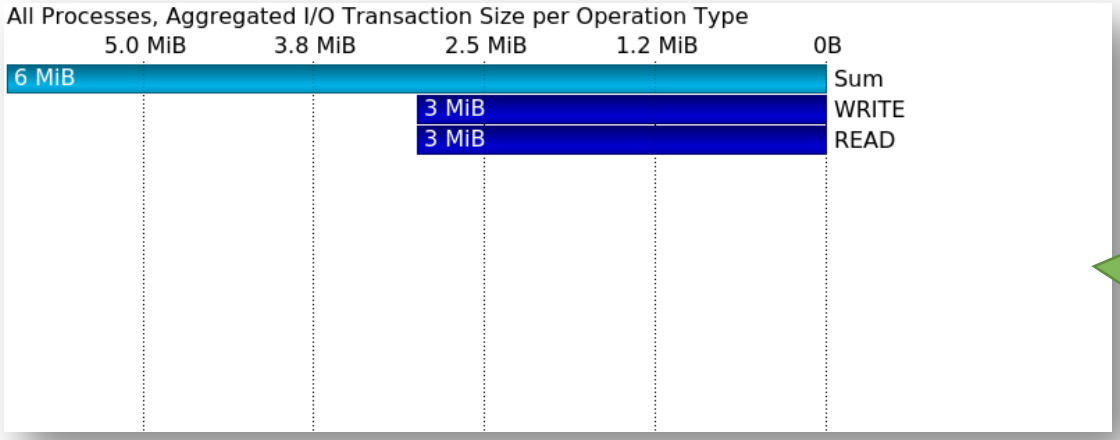




I/O data rate over time

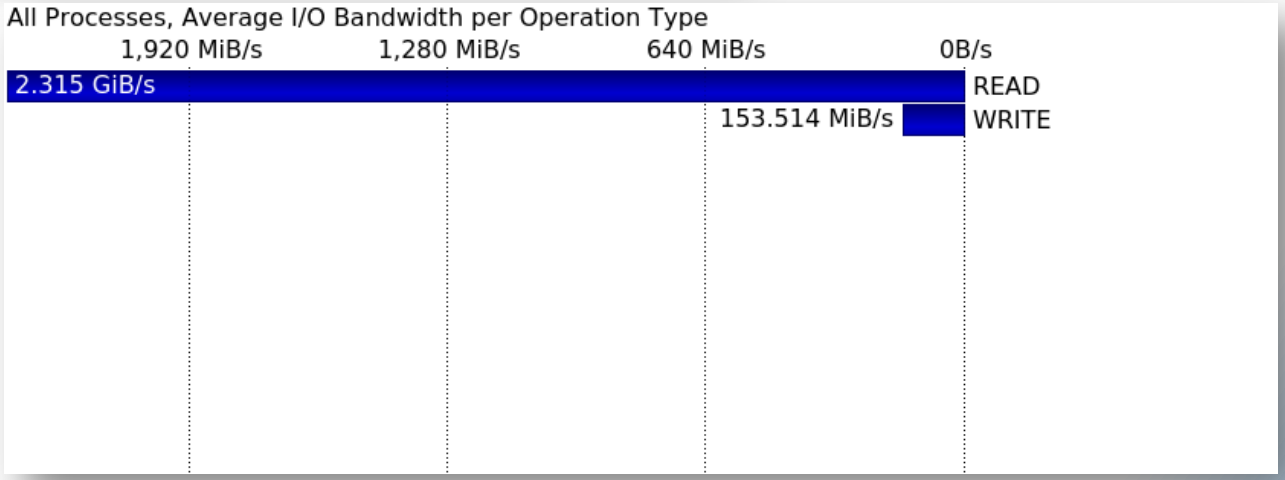


I/O summaries with totals

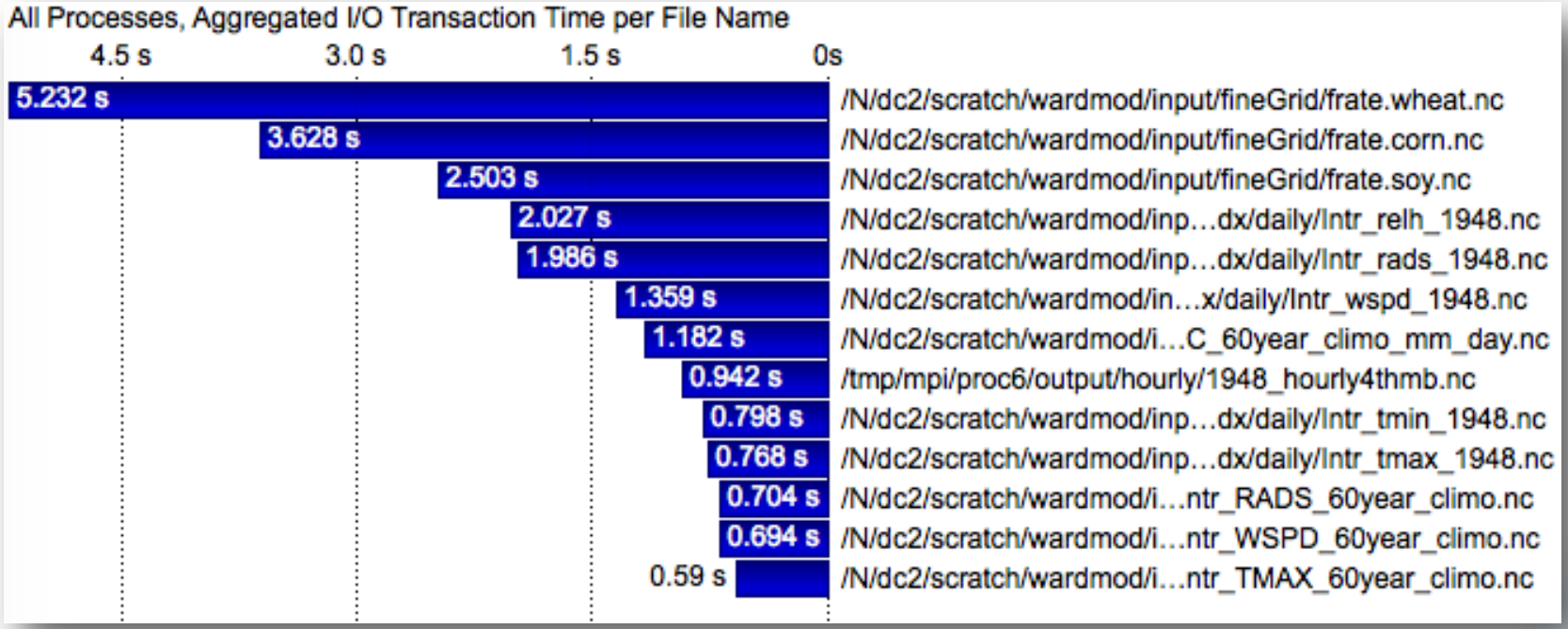


Other Metrics:

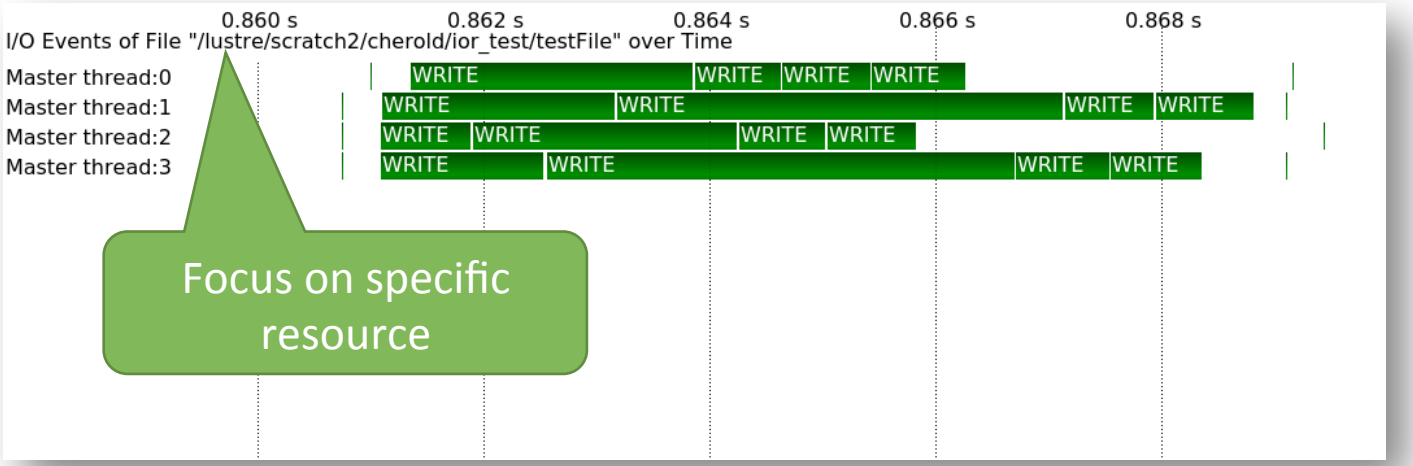
- IOPS
- I/O Time
- I/O Size
- I/O Bandwidth

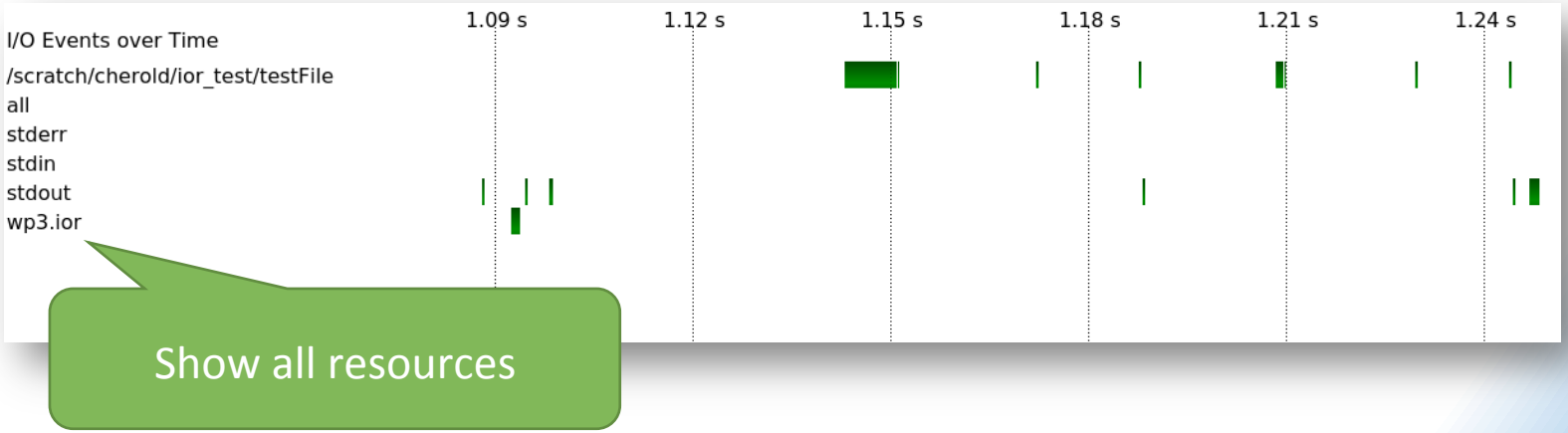


I/O summaries per file



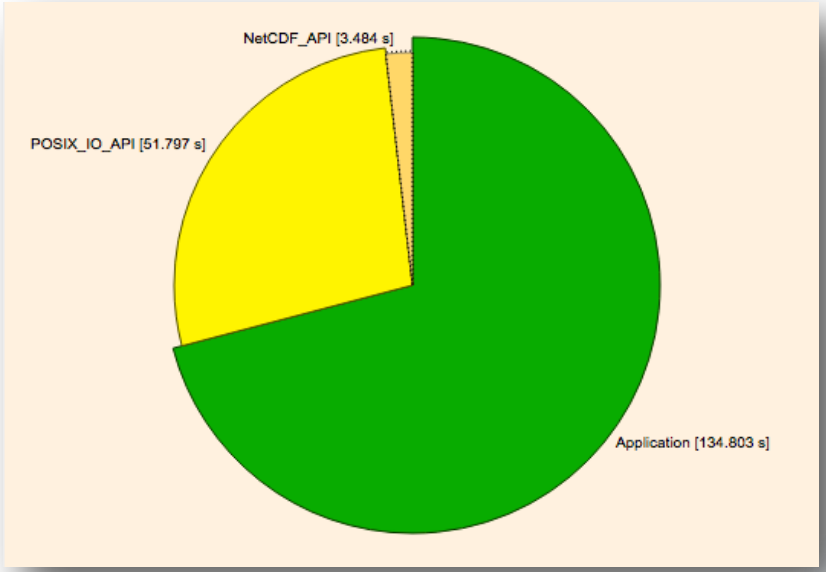
I/O operations per file



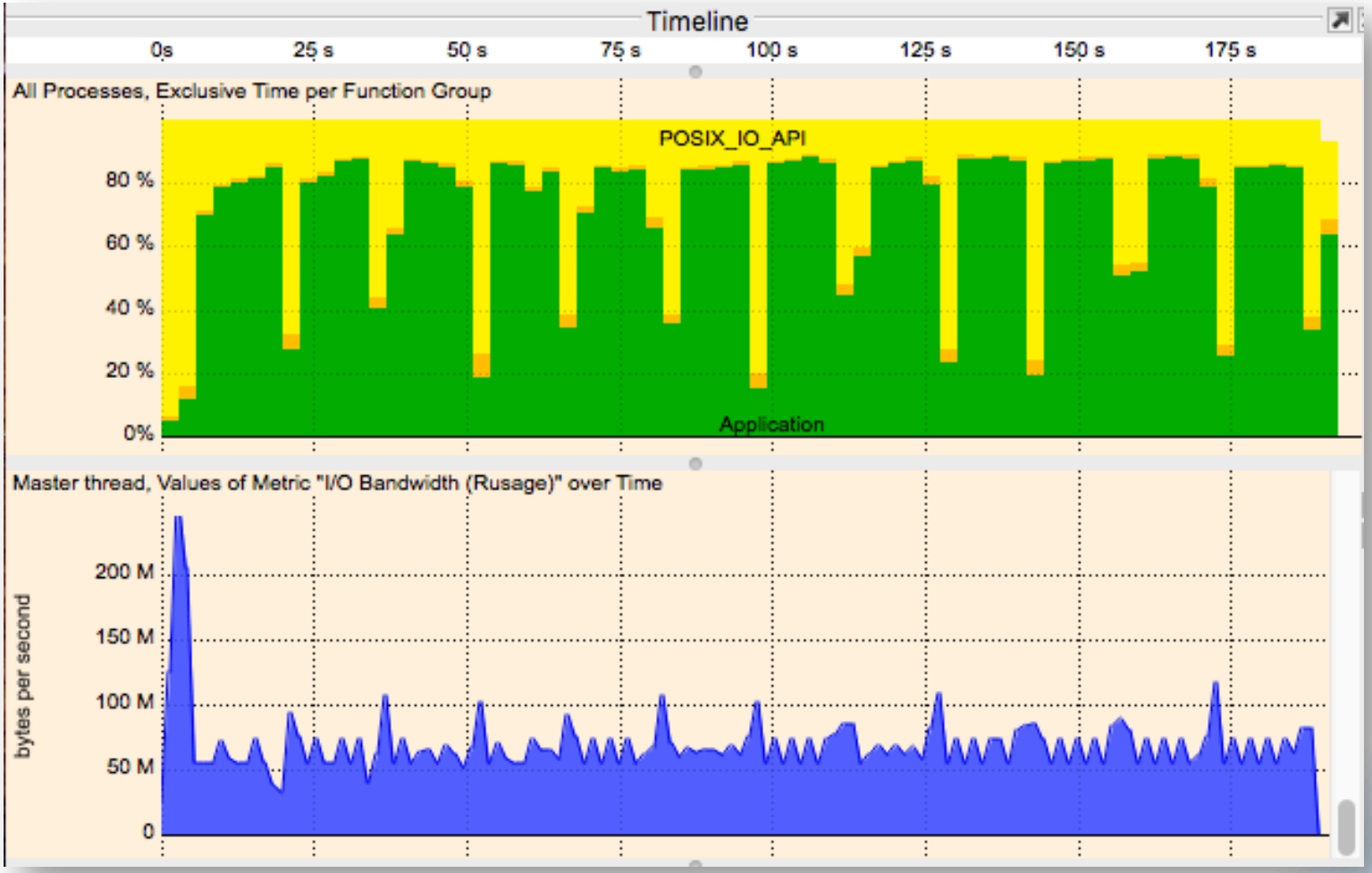


Taken from my daily work...

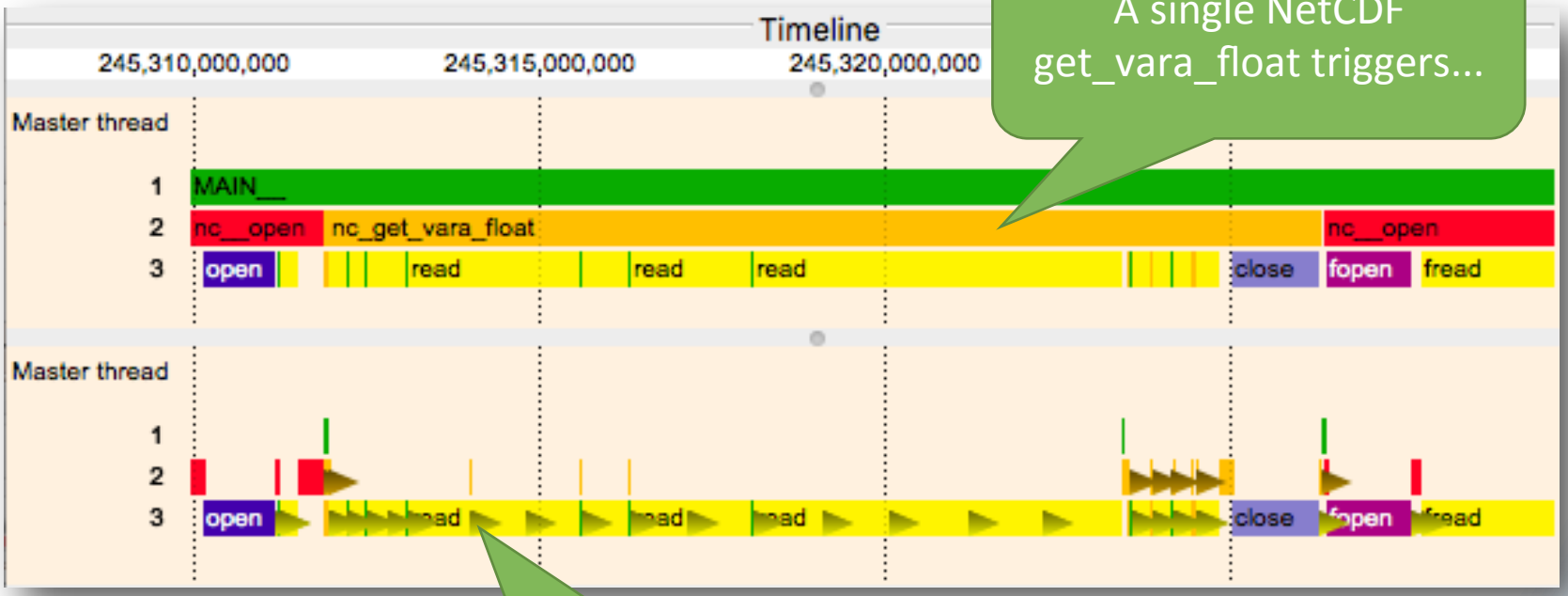
- Bringing the system I/O down
  - with a single (serial) application
- Higher I/O demand than IOR benchmark
- Why?



Coarse grained time series reveal some clue, but...



Details make a difference

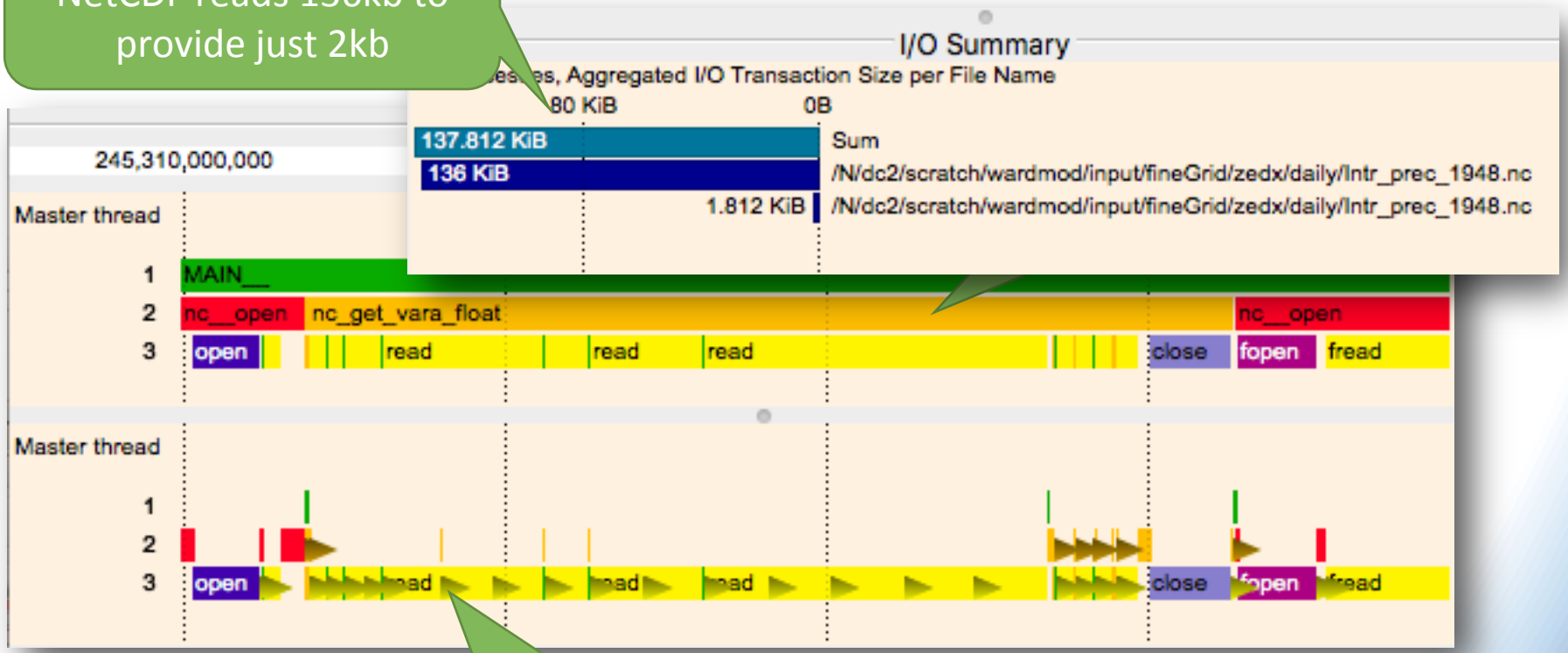


A single NetCDF get_vara_float triggers...

...15! POSIX read operations

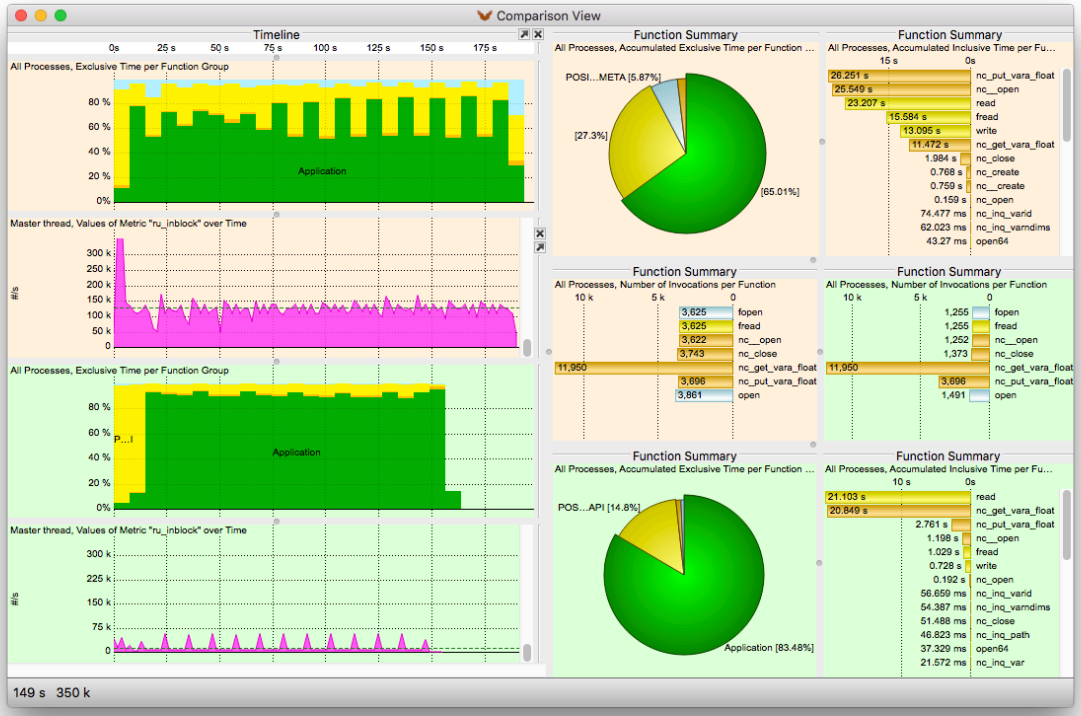
Approaching the real cause

Even worse:
NetCDF reads 136kb to provide just 2kb



...15! POSIX read operations

Before and after…



Summary

- NEXTGenIO developing a full hardware ***and*** software solution
- Performance focus
  - Consider complete I/O stack
  - Incorporate new I/O paradigms
  - Study implications of NVRAM
- Reduce I/O costs
- New usage models for HPC and HPDA