Analyzing I/O Performance on a NEXTGenIO Class System

holger.brunst@tu-dresden.de

ZIH, Technische Universität Dresden

LUG17, Indiana University, June 2nd 2017

NEXTGenIO Fact Sheet

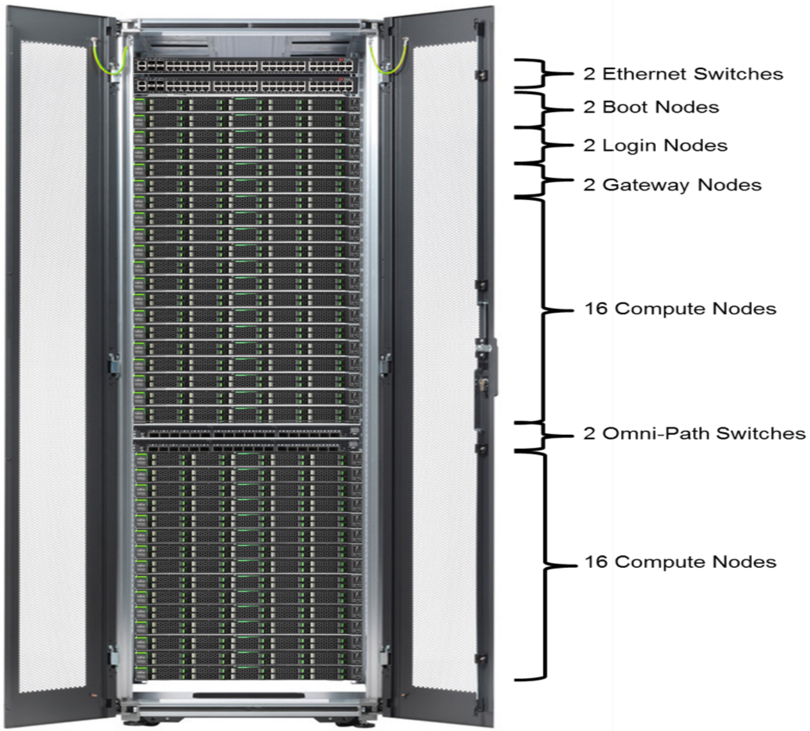
Project

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

Partners

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

Approx. 50% Committed to Hardware Development



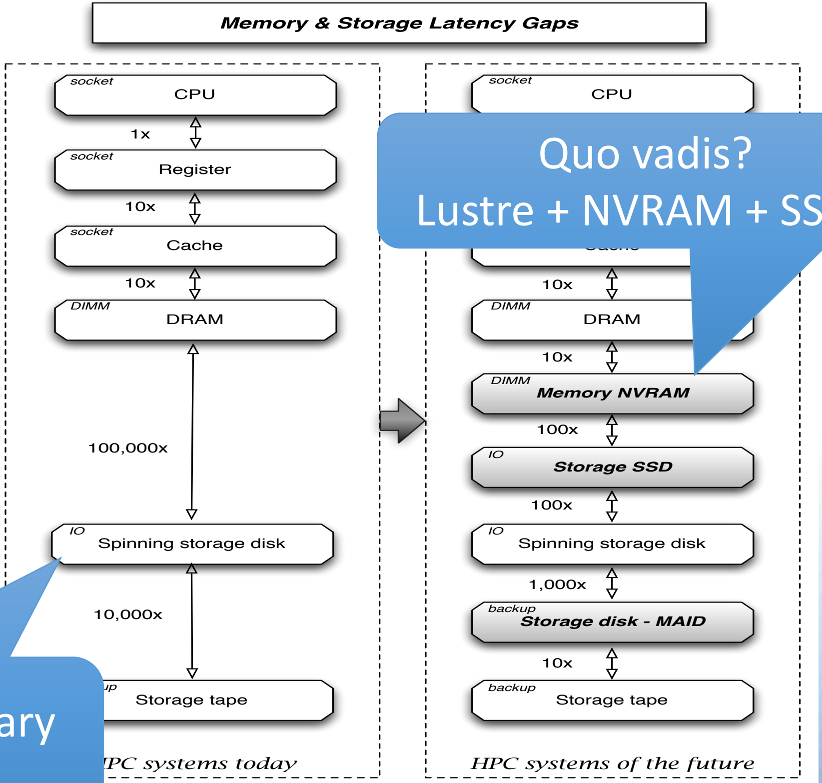
- Intel™ DIMMs are a Key feature
  - Non-volatile RAM
  - Much larger capacity than DRAM
  - Slower than DRAM
    - By a certain factor
    - Significantly faster than SSDs ™
  - 12 DIMM slots per socket
    - Combination of DDR4 and Intel™ DIMMs
- How can files systems like Lustre benefit?

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 applications

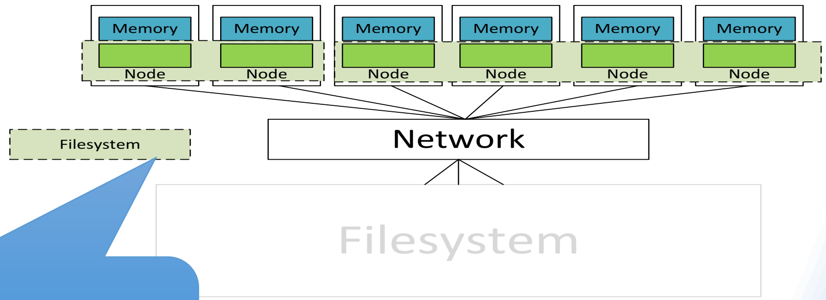


Quo vadis?
Lustre + NVRAM + SSD?

Lustre serves as primary file system

Using Distributed Storage

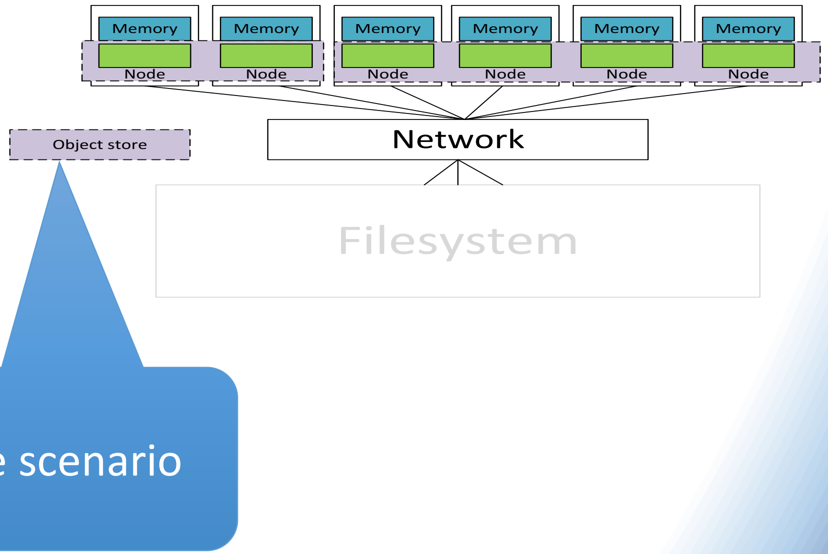
- Lustre global file system
  - No changes to apps
  - Support for NVRAM?
- Required functionality
  - Create and tear down file systems for jobs
  - Works across nodes
  - Preload and postmove filesystems
  - Support multipl
    across system
- I/O Performance
  - Sum of many la



e.g. Lustre

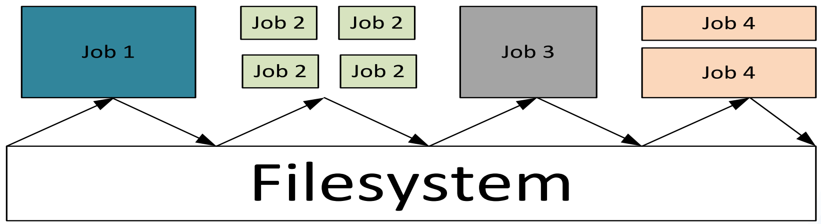
Using an Object Store

- 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



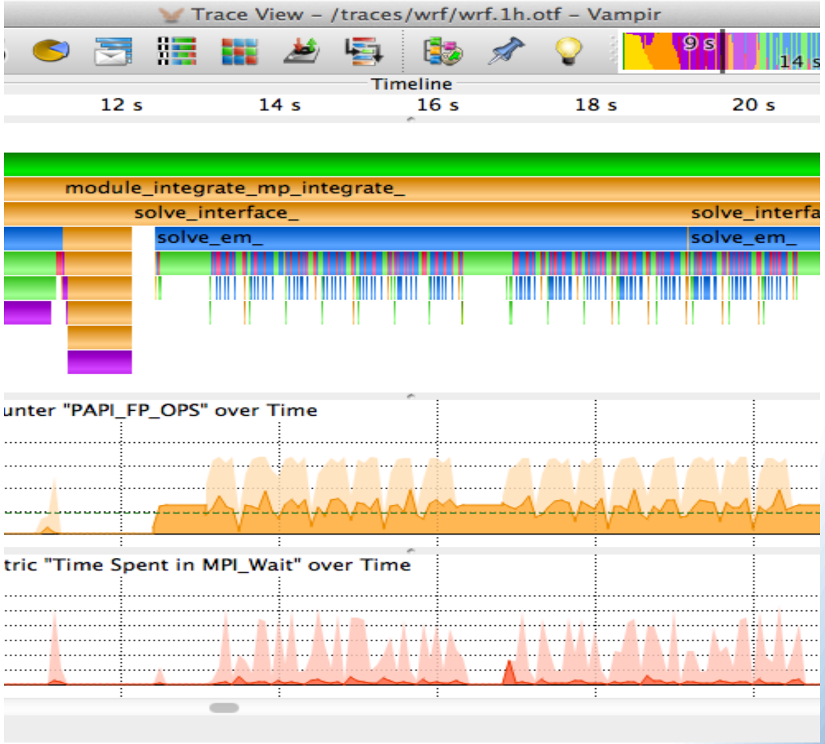
Towards Workflows

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



New Tool 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

Score-P
Scalable performance measurement infrastructure for parallel codes

VAMPIR

Tapping I/O Layers

- I/O layers
  - Lustre File System
    - Client side
    - Server side
  - Kernel
  - POSIX
  - MPI-I/O
  - HDF5
  - NetCDF
  - PNetCDF

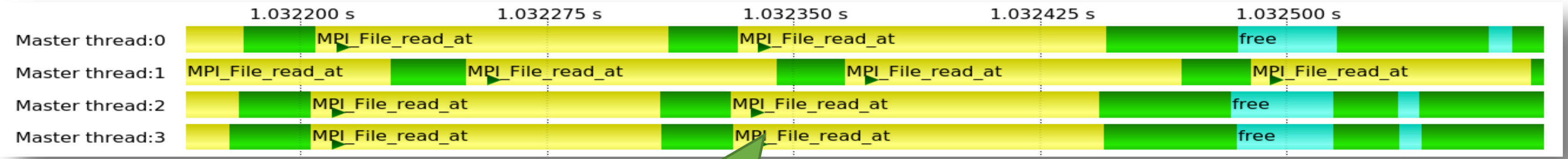
Lustre I/O performance data

- Event data AND aggregated numbers
  - Open/Create/Close operations (meta data)
  - Data transfer operations
- Client side
  - procfs via PAPI-Components counters, Score-P (user space)
- Server side
  - Data base (custom made, system space)
  - Mapping to nodes/applications

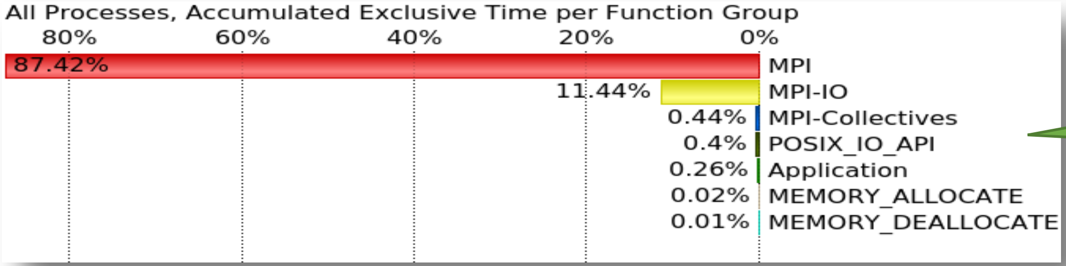
What the NVM Library Tells Us

- 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)

I/O Operations over Time

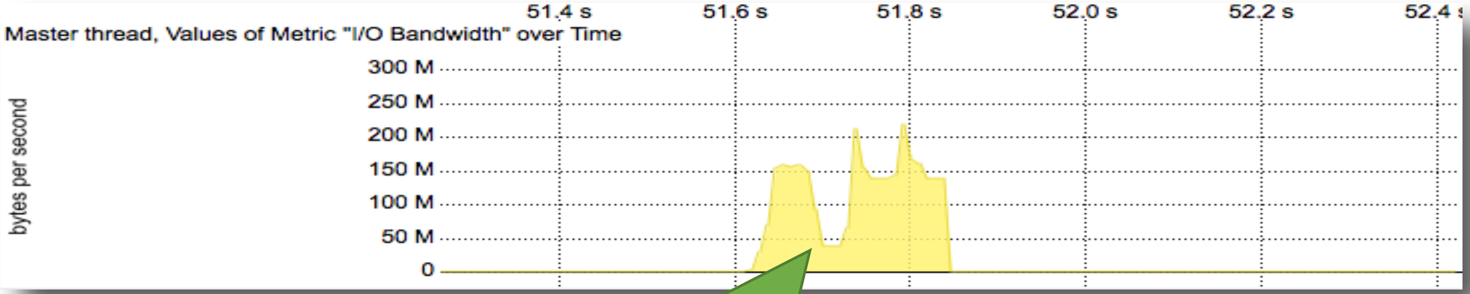


Individual I/O Operation



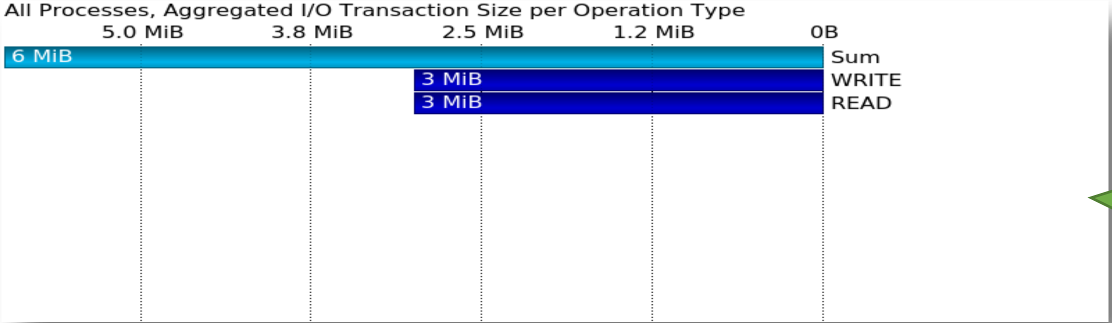
I/O Runtime Contribution

I/O Data Rate over Time

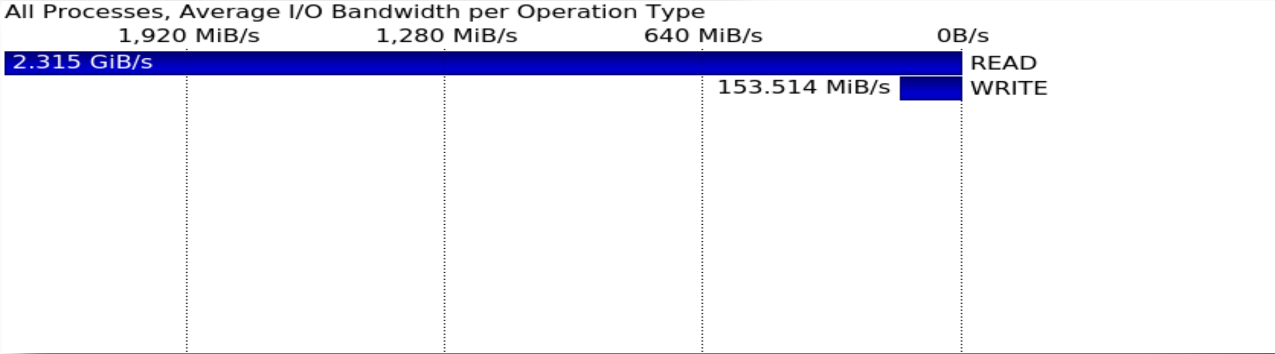


I/O Data Rate of single thread

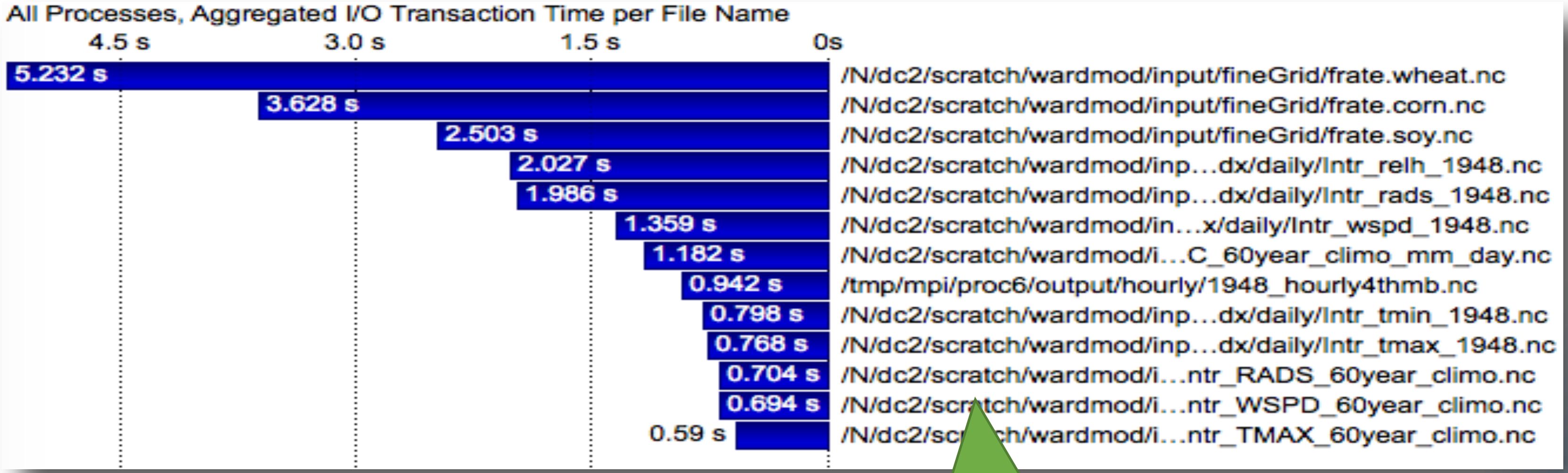
I/O Summaries with Totals



Other Metrics:
- IOPS
- I/O Time
- I/O Size
- I/O Bandwidth

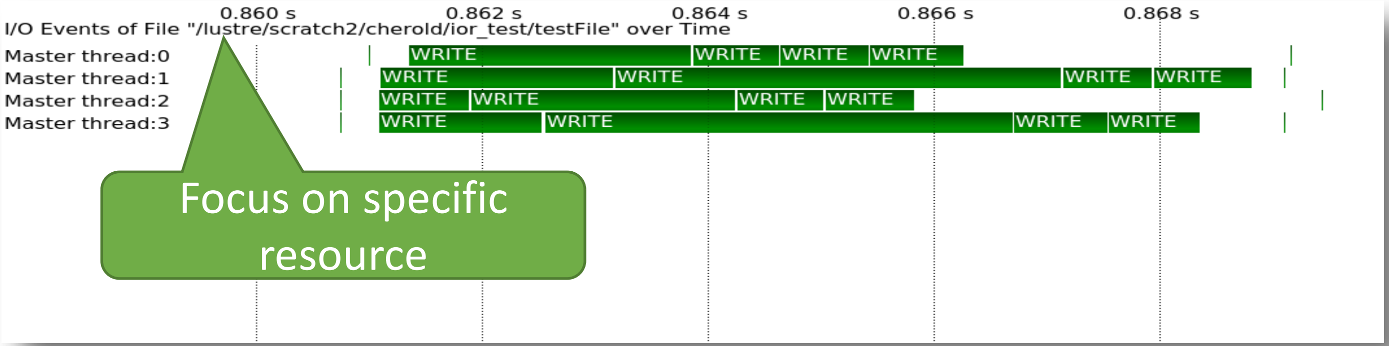


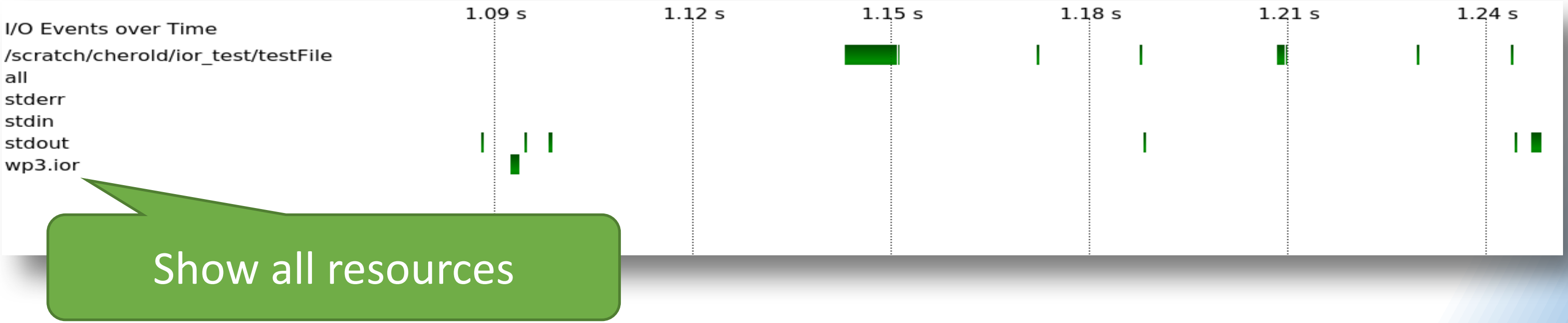
I/O Summaries per File



Aggregated data for specific resource

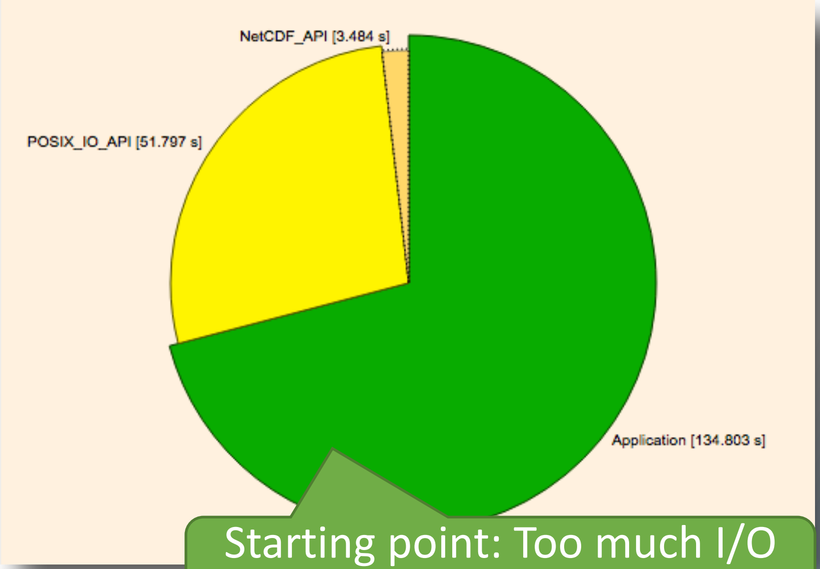
I/O Operations per File





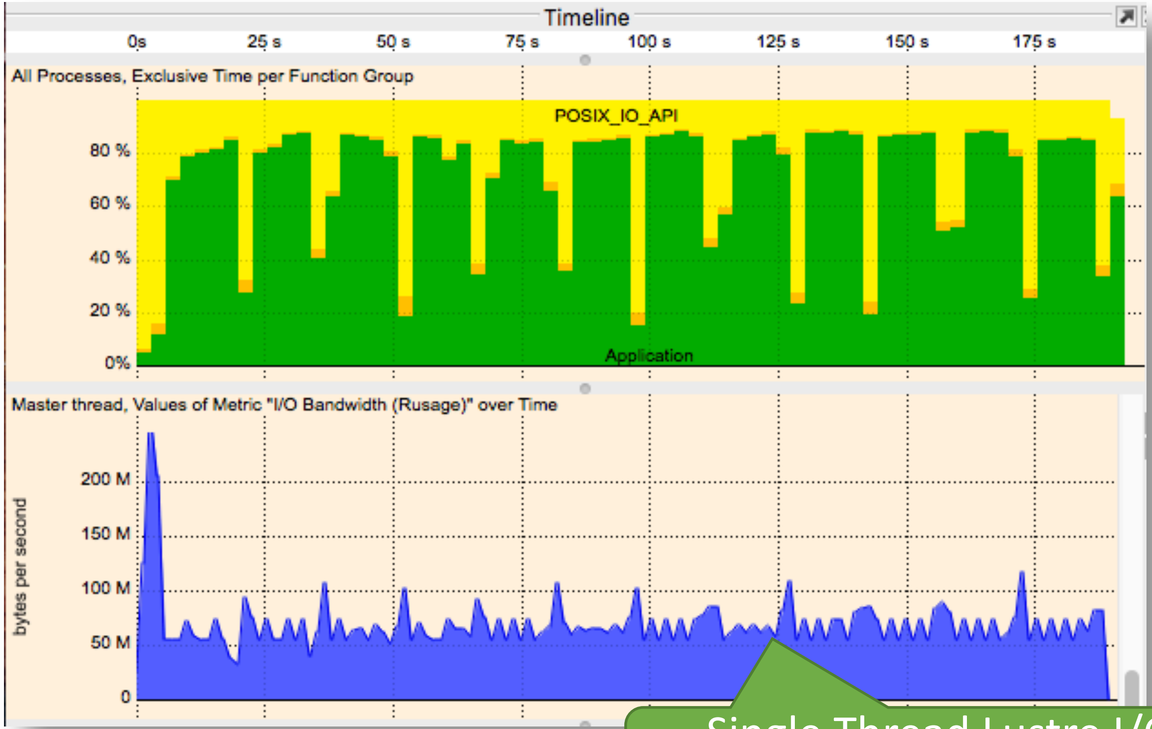
Taken from My Daily Work...

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



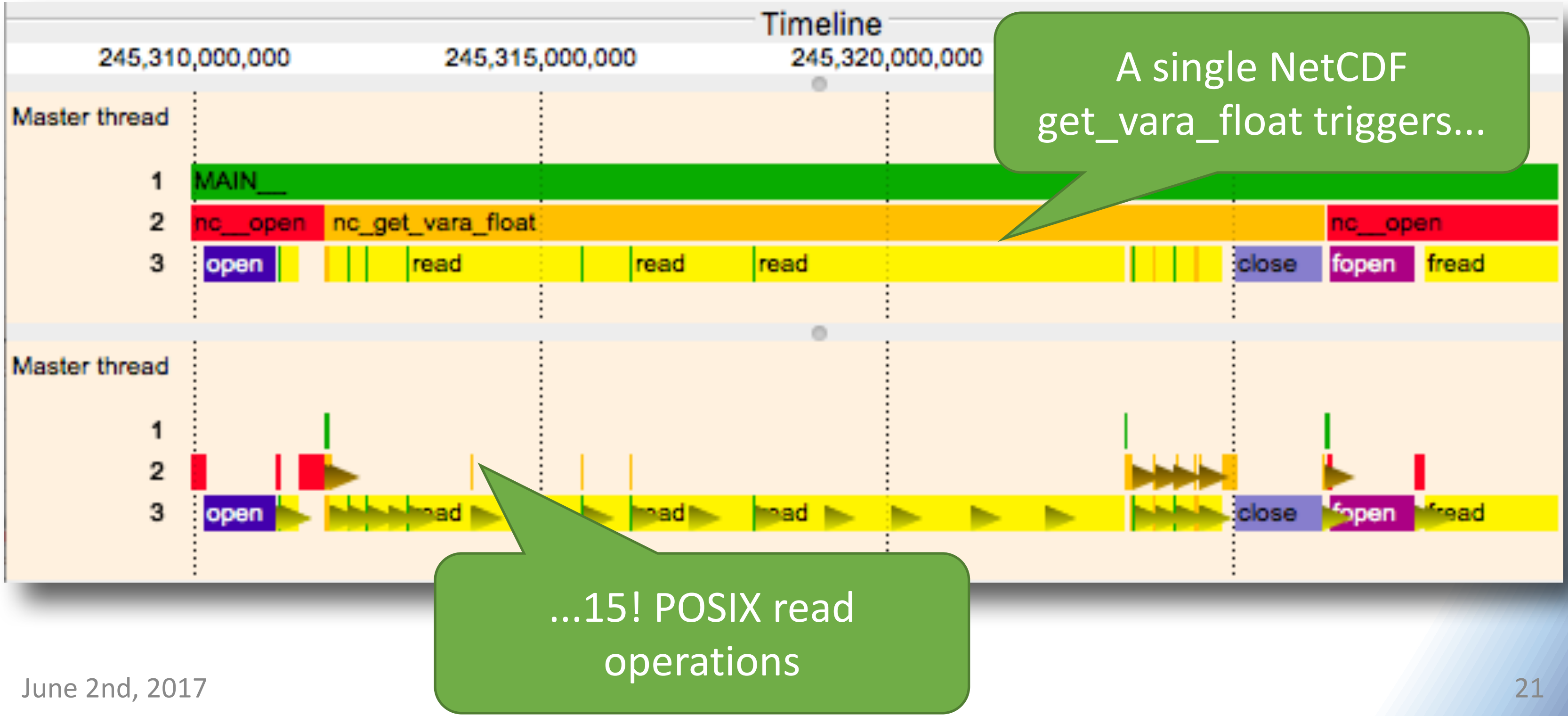
Starting point: Too much I/O (Bandwidth) on Lustre FS!!!

Coarse Grained Time Series Reveal Some Clue, but...



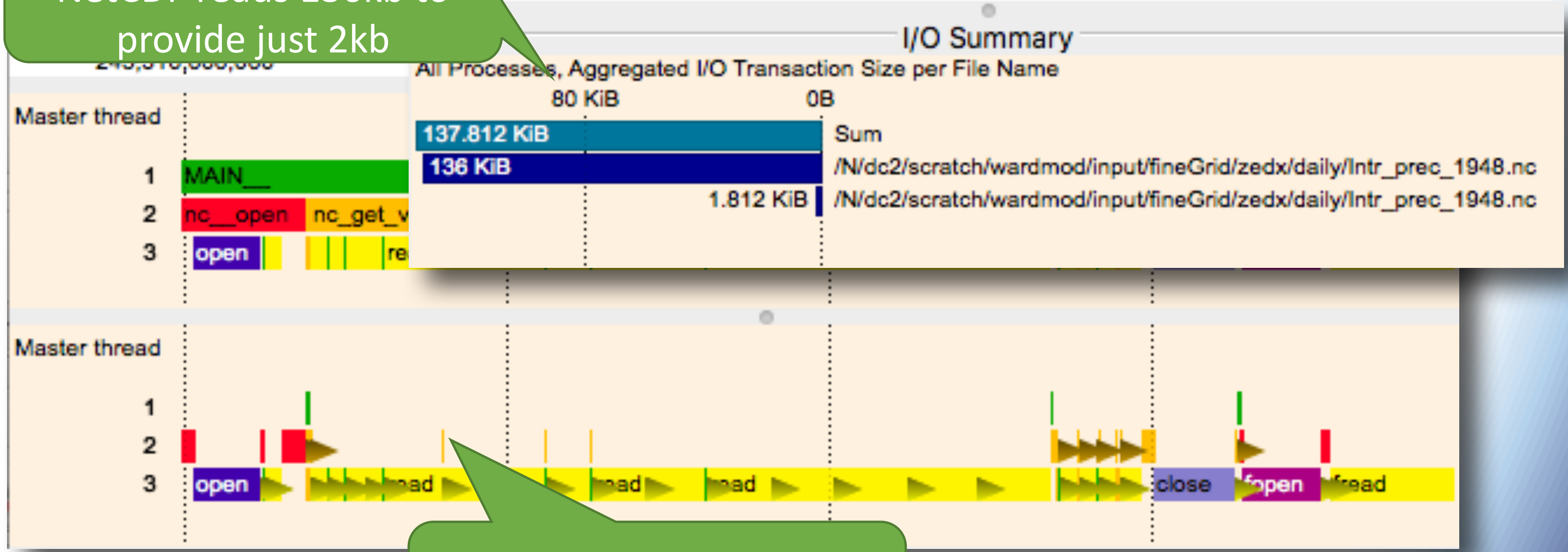
Single Thread Lustre I/O Bandwidth over time

Details Make a Difference



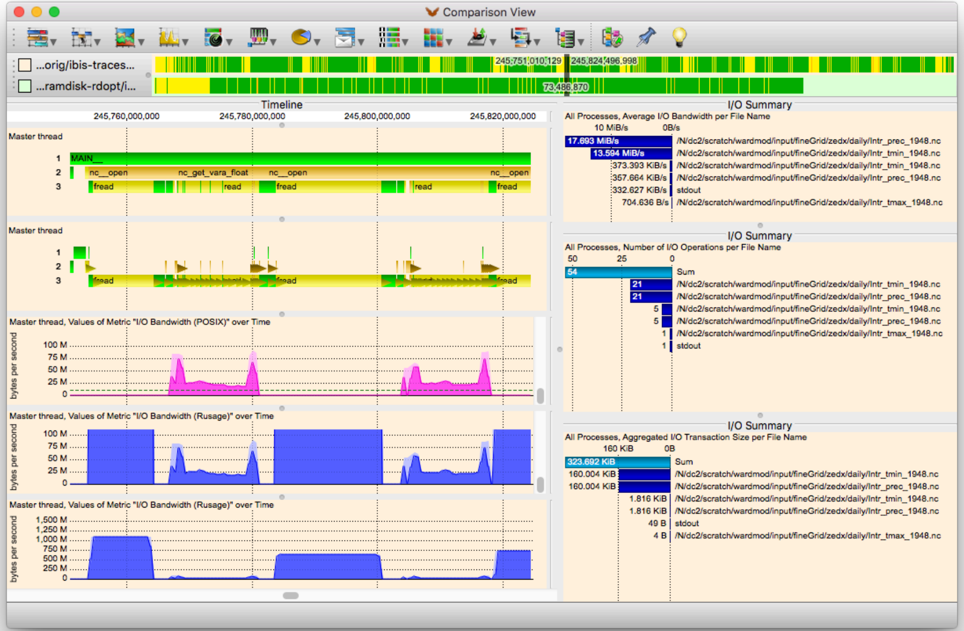
Approaching the Real Cause

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



...15! POSIX read operations

Impact of Lustre Prefetching



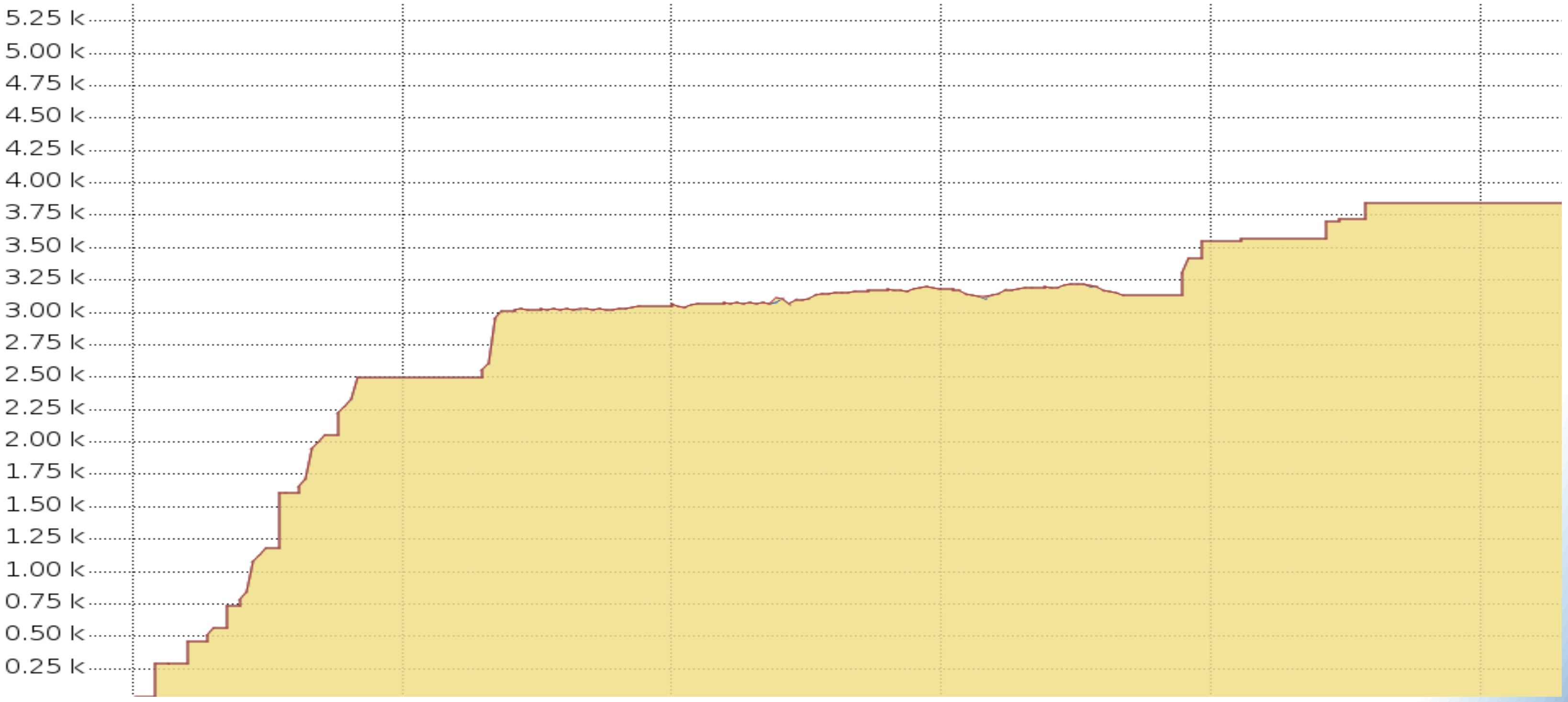
Before and After…

Summary

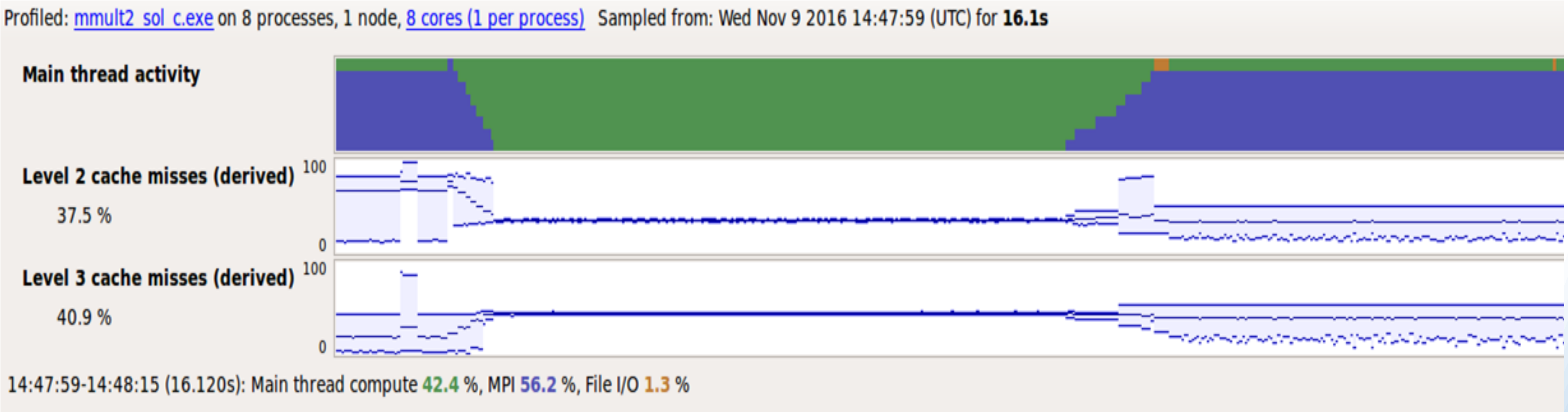
- NEXTGenIO developing a full hardware ***and*** software solution
  - Solution includes Lustre
- Performance focus
  - Consider complete I/O stack
  - Incorporate new I/O paradigms
  - Study implications of NVRAM
- Reduce I/O costs
- New usage models for HPC and HPDA

Supplementary Slides

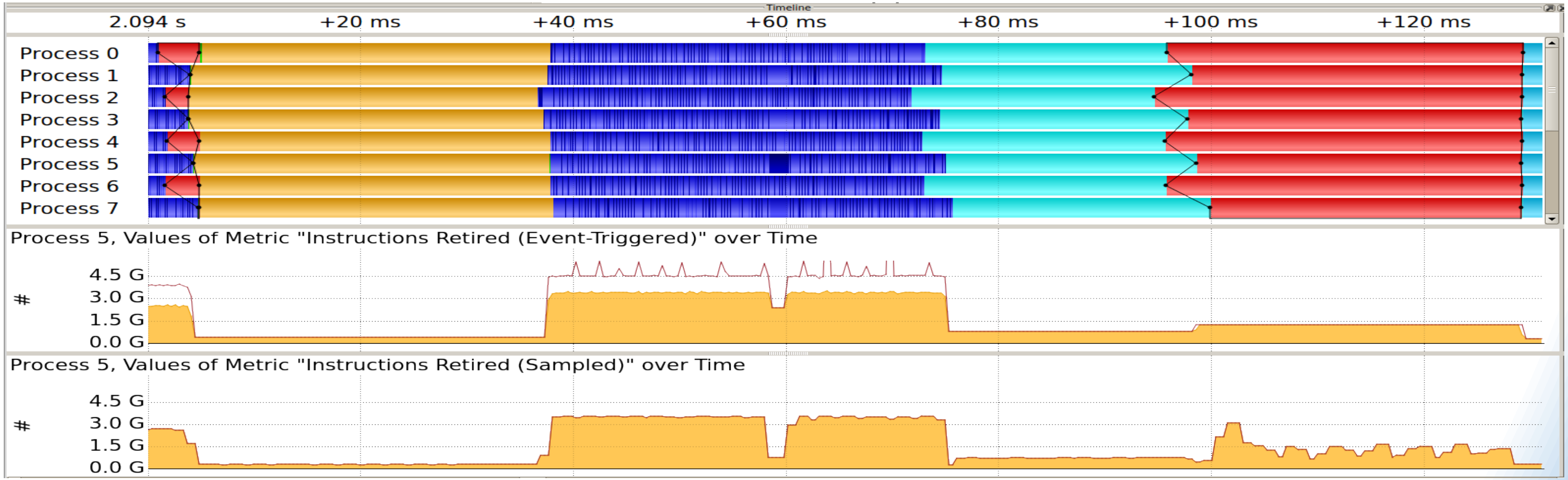
NVRAM allocation over time



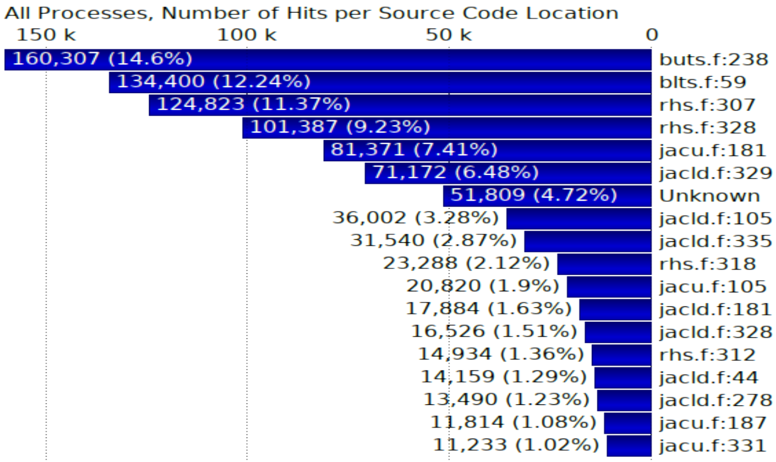
MAP Timeline

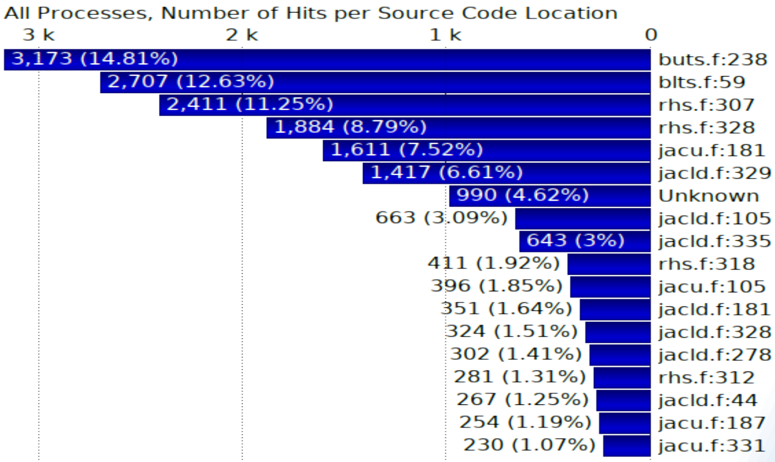


Vampir Timeline



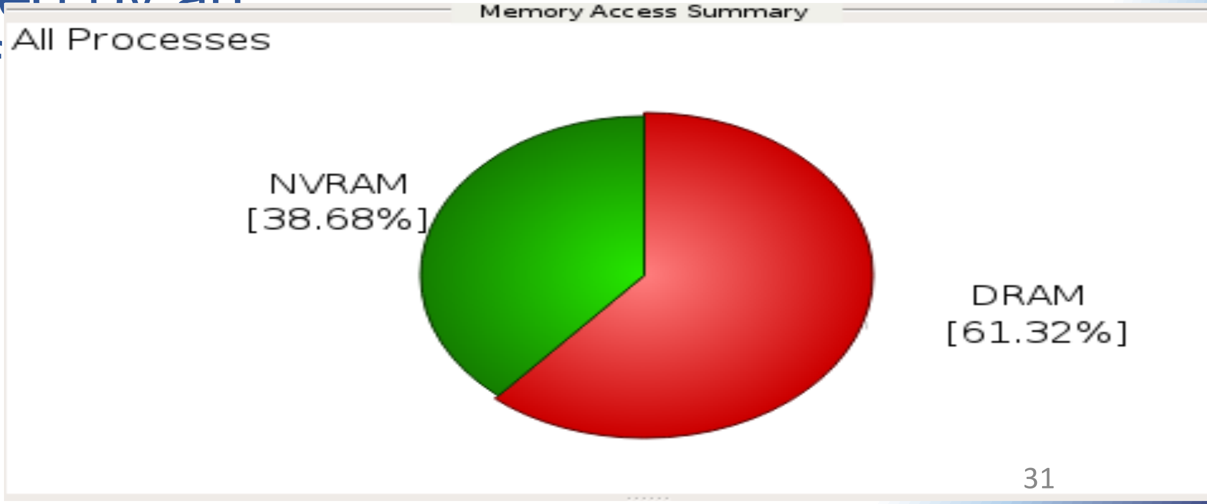
Future Stats





Visualization of memory access statistics

- Currently, Vampir has many different views presenting counter metrics
- Future: Absolute number of memory accesses performed by an application for diff types of memory



Future Stats