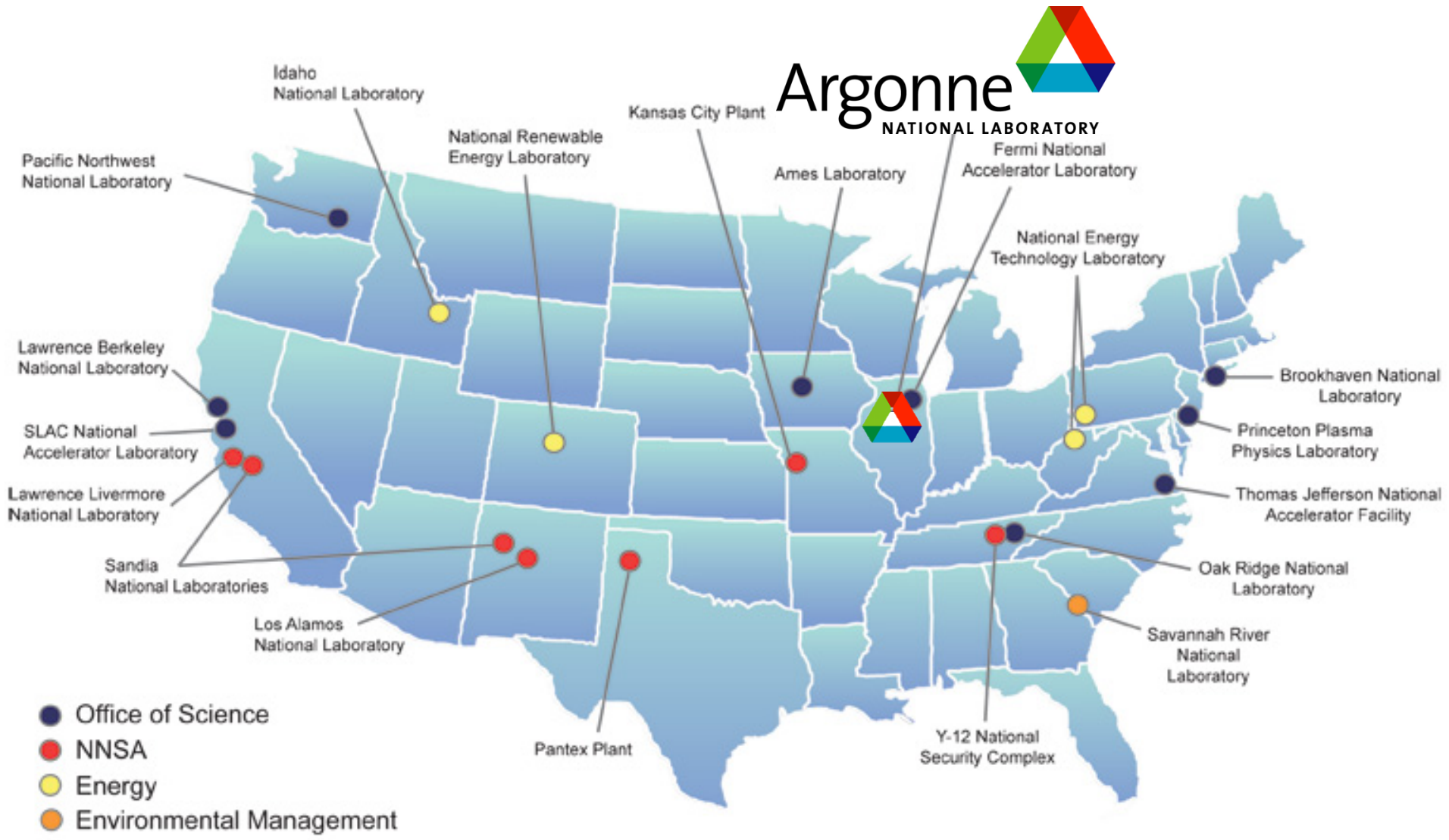
I/O at Argonne National Laboratory

Florin Isaila
Argonne National Laboratory - USA
Unversity Carlos III - Spain

Slides from Rob Ross, Phil Carns, Tom Peterka, Marc Snir, Justin Wosniak, Susan Coghlan

Argonne: Vital part of DOE National Laboratory System



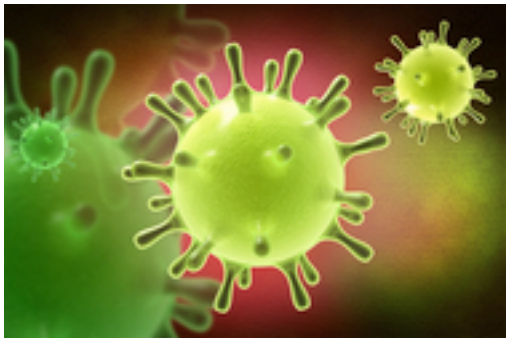
$10B, 22,000 people, 17 labs

Computing, Environment, and Life Sciences Directorate: Four Divisions

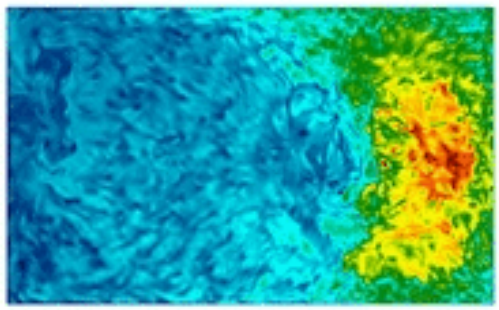
Argonne Leadership Computing Facility

Biosciences Division

Environmental Science Division



Mathematics and Computer Science Division



And three institutes and facilities

- ARM Climate Research Facility
- Computation Institute
- Institute for Genomics and Systems Biology

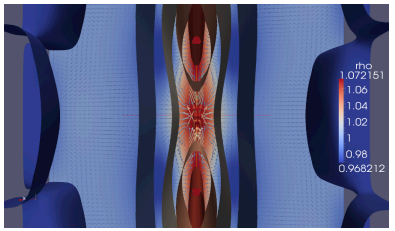
Mathematics and Computer Science Division Strategic Areas

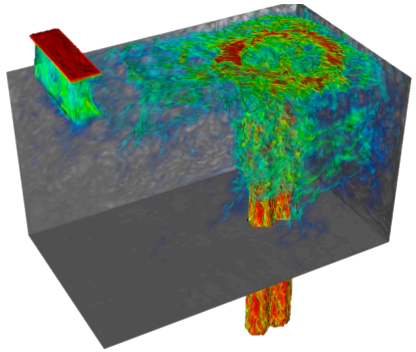
Extreme Computing: exploring new approaches to system software, fault tolerance, and innovative programming models for next-generation computers

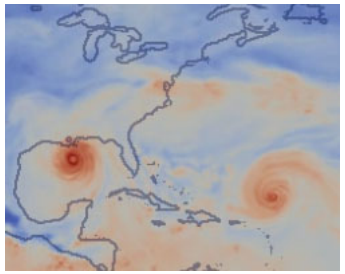
Big Data: formulating novel techniques for managing, storing, analyzing, and visualizing the enormous amounts of data produced by leadership-class computers and large experimental facilities

Applied Mathematics: formulating rigorous theory leading to fast algorithms, deployed in software on leading-edge computing platforms

Applications: working with scientists and engineers to apply our advanced algorithms and tools to applications critical to our society, such as life science, climate change, materials, and energy systems simulations







Mira vs. Aurora

System Feature	Mira	Aurora	Increase
Peak Performance	10 PF	180 PF	18x
Number of Nodes	49,152	>50,000	Similar
High Bandwidth On-Package Memory, Local Memory, and Persistent Memory	786 TB	>7 PB	8.9x
High Bandwidth On-Package Memory Bandwidth	2.5 PB/s	n/a	Significantly higher
Interconnect Aggregate Node Link BW	2 PB/s	n/a	Higher
Interconnect Bisection Bandwidth	24 TB/s	n/a	Significantly higher
File System Capacity	26 PB	>150 PB	5.8x
File System Throughput	300 GB/s	>1 TB/s	3.3x
Peak Power Consumption	4.8 MW	13 MW	2.7x
FLOPS/watt	2.1 GF/watt	>13 GF/watt	6.2x
Facility Area	1,536 sq. ft.	~3,000 sq. ft.	2x

Outline

- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

History: I/O research via prototyping

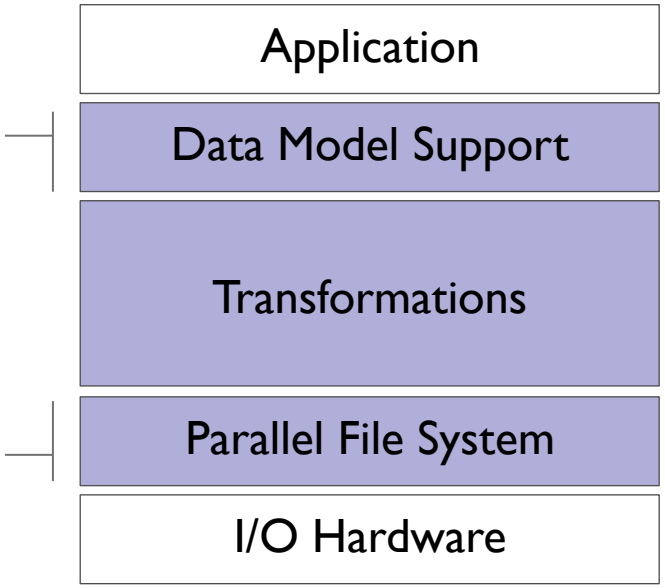
PVFS **ROMIO** **PnetCDF** **BMI** **IOFSL**
PVFS2 **LogFS**

Data Model Libraries map application abstractions onto storage abstractions and provide data portability.

HDF5, ***Parallel netCDF****, ADIOS*

Parallel file system maintains logical file model and provides efficient access to data.

PVFS*, Gfarm, GPFS, Lustre*



I/O Middleware organizes accesses from many processes, especially those using collective I/O.

*MPI-IO (****ROMIO****), PLFS,* ***LOGFS***

I/O Forwarding transforms I/O from many clients into fewer, larger request; reduces lock contention; and bridges between the HPC system and external storage.

IBM ciod, ***IOFSL****, Cray DVS*

Outline

- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

Darshan

Goal: collect I/O patterns of applications running on production HPC platforms

- Majority of applications – need to integrate into the build environment of the systems.
- Without perturbation – bounded use of resources (memory, network, storage); no communication or I/O prior to job termination; compression.
- Adequate detail – basic job statistics; file access information from multiple APIs.

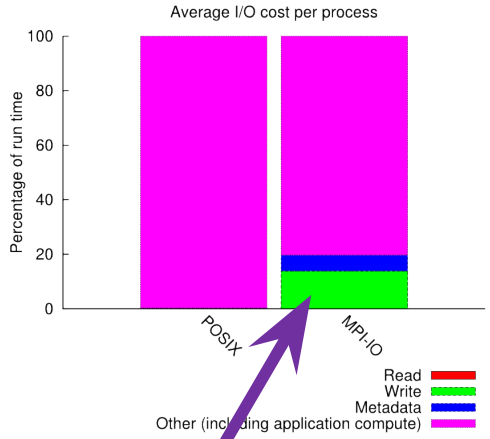
Darshan Specifics

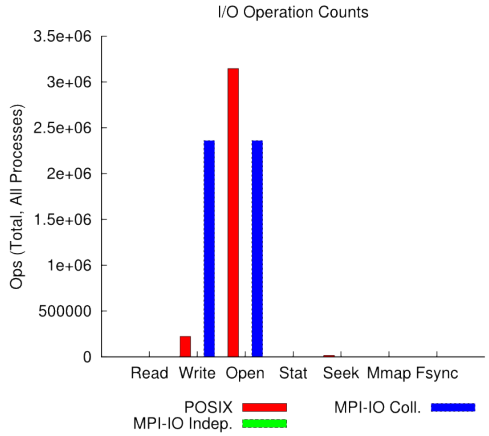
- Runtime library for characterization of application I/O
  - Instrumentation is inserted at build time (for static executables) or at run time (for dynamic executables)
  - Captures POSIX I/O, MPI-IO, and limited HDF5 and PNetCDF functions
- Minimal application impact
  - Bounded memory consumption per process
  - Records strategically chosen counters, timestamps, and histograms
  - Reduces, compresses, and aggregates data at MPI_Finalize() time
- Compatible with IBM BG, Cray, and Linux environments
  - Deployed system-wide or enabled by individual users
  - Instrumentation is enabled via software modules, environment variables, or compiler scripts
  - No source code modifications or changes to build rules
  - No file system dependencies
- Enabled by default at NERSC, ALCF, and NCSA

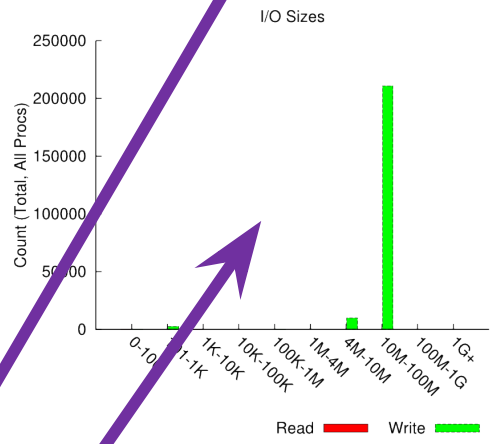
http://www.mcs.anl.gov/research/projects/darshan

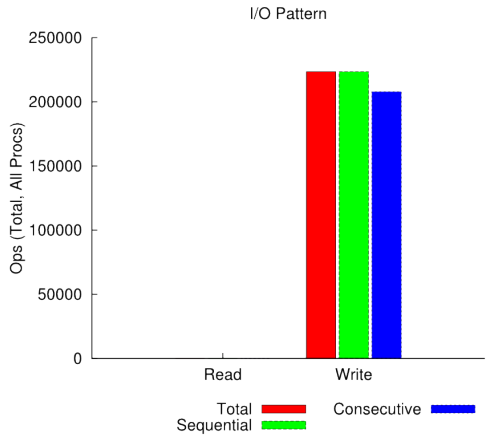
Job Level

jobid: 149563	uid: 6729	nprocs: 786432	runtime: 2751 seconds









% of runtime in I/O

Access size histogram

Most Common Access Sizes

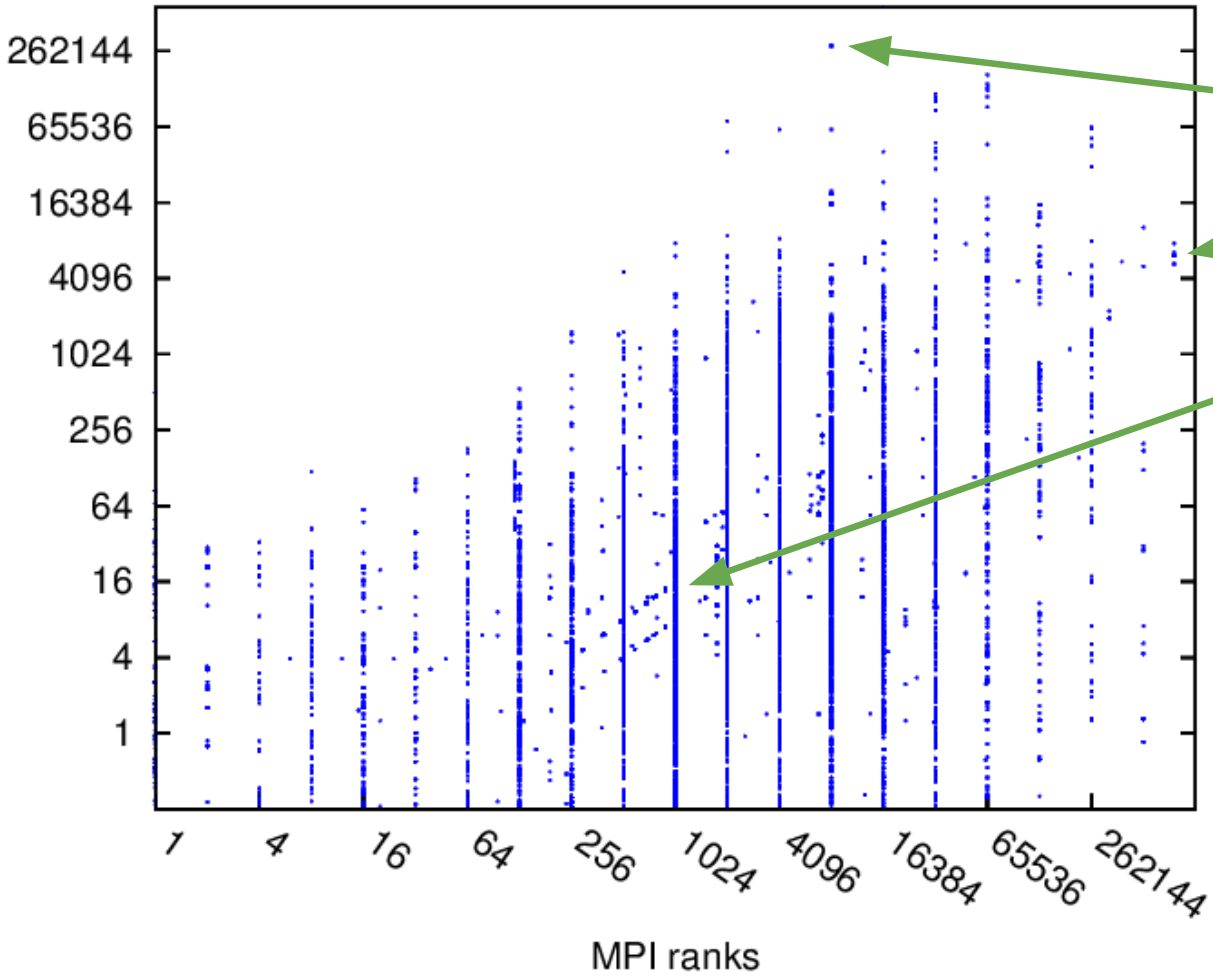
access size	count
16777216	210977
8388608	9866
256	2598
68	9

File Count Summary
(estimated by I/O access offsets)

type	number of files	avg. size	max size
total opened	17	199G	1.6T
read-only files	1	2.0K	2.0K
write-only files	13	260G	1.6T
read/write files	0	0	0
created files	13	260G	1.6T

System Level: Aggregated View of Data Volume

Job size vs. data volume for Mira BG/Q system in 2014
(~128,000 logs as of October, ~8 PiB of traffic)



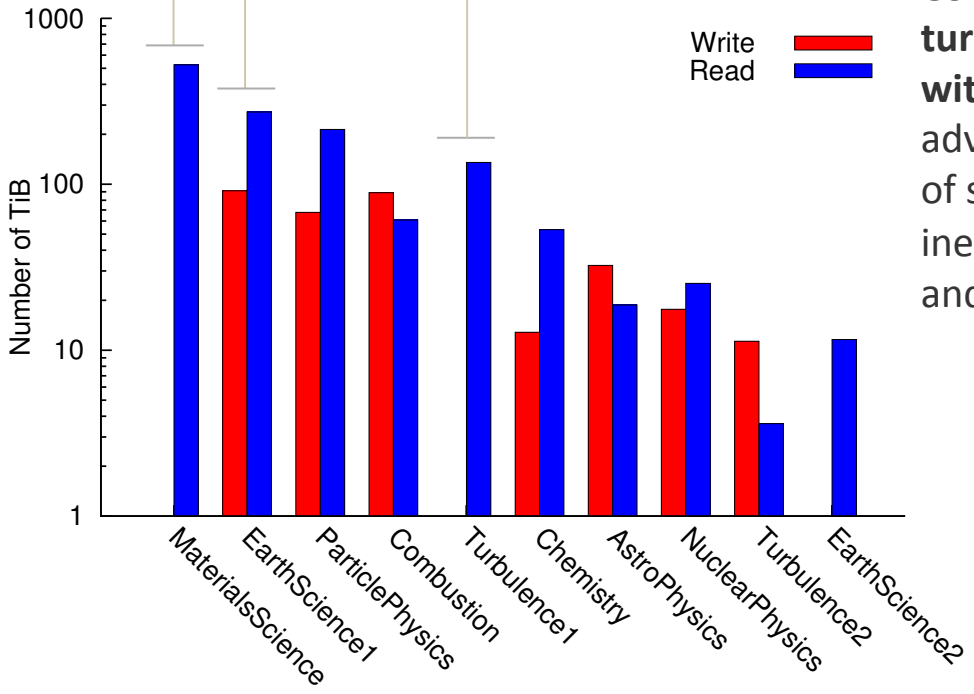
- Biggest by volume: ~300 TiB
- Biggest by scale: 768K processes
- Probably some scaling experiments?
- Most jobs use power of 2 numbers of processes on Mira

System Level: I/O Mix

Matching large scale simulations of dense suspensions with empirical measurements to better understand properties of complex materials such as concrete.

Processing large-scale seismographic datasets to develop a 3D velocity model used in developing earthquake hazard maps.

Comparing simulations of turbulent mixing of fluids with experimental data to advance our understanding of supernovae explosions, inertial confinement fusion, and supersonic combustion.



Top 10 data producer/consumers instrumented with Darshan over the month of July, 2011 on Intrepid BG/P system at Argonne. Surprisingly, three of the top producer/consumers almost exclusively read existing data.

Sharing data with the community

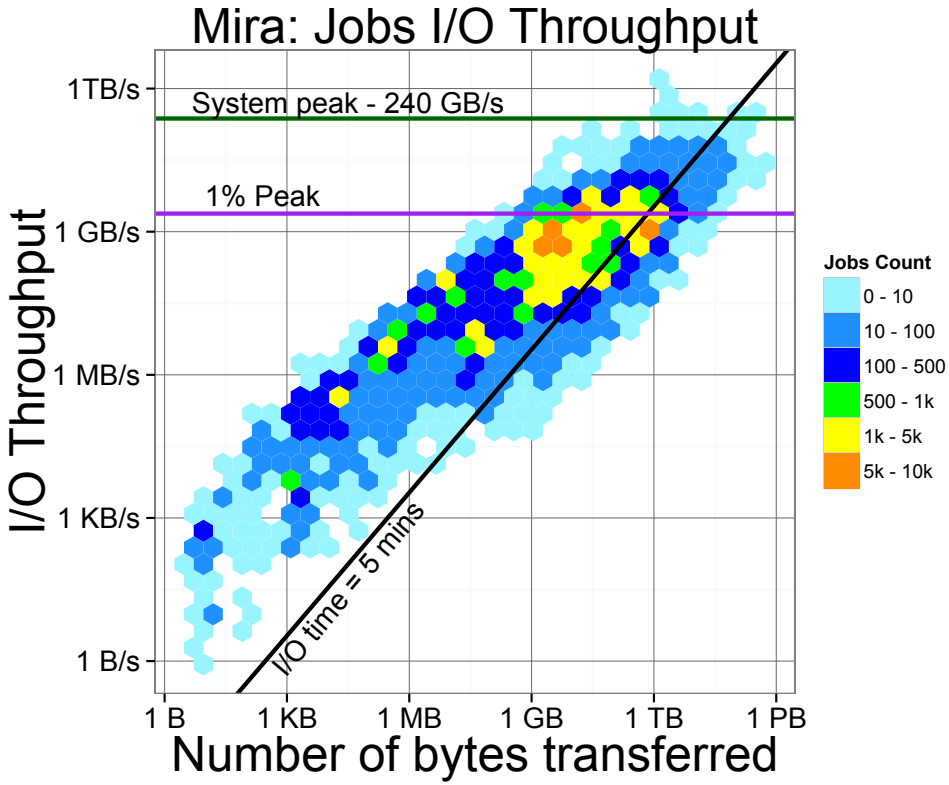
- Conversion utilities can anonymize and re-compress data
- Compact data format in conjunction with anonymization makes it possible to share data with the community in bulk
- The ALCF I/O Data Repository provides access to production logs captured on Intrepid
- Logs from 2010 to 2013, when machine was retired

ALCF I/O Data Repository Statistics	
Unique log files	152,167
Core-hours instrumented	721 million
Data read	25.2 petabytes
Data written	5.7 petabytes

http://www.mcs.anl.gov/research/projects/darshan/data/

Darshan Future(s)

- Mining Darshan data
  - Ongong work under M. Winslett (UIUC)
- Reconstructing workloads
  - How well can we reproduce the I/O operations from the compressed Darshan log?
- Refining Darshan
  - Supporting new interfaces
  - Capturing more information (e.g., I/O patterns)
  - More compact grammar-based representation of logs



Thanks to Huong Luu (UIUC) for providing this figure.

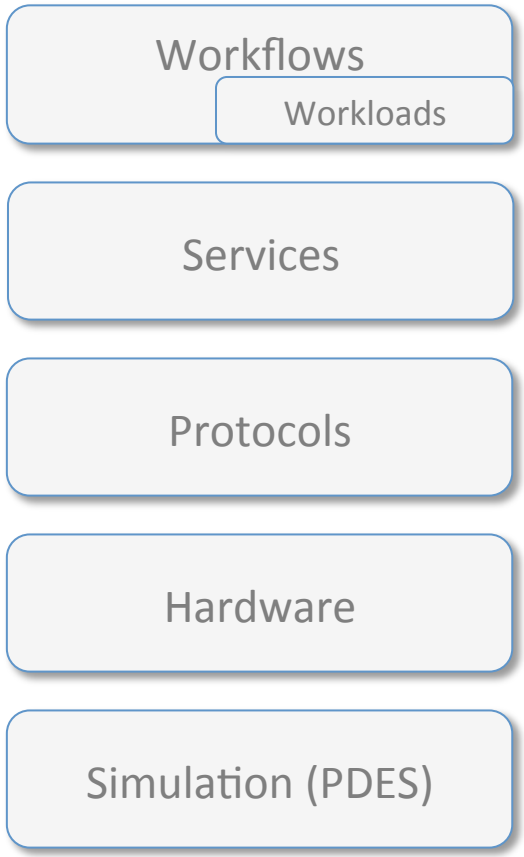
Outline

- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

CODES: Enabling Co-Design of Exascale Storage Architectures and Science Data Facilities

The goal of the CODES project is use highly parallel simulation to explore the design of exascale storage architectures and distributed data-intensive science facilities.

- Set of models, tools, and utilities intended to simplify complex model specification and development
  - Commonly-used models (e.g., networking)
  - Facilities to inject application workloads
- Using these components to understand systems of interest to DOE ASCR



CODES simulations

- Based on **ROSS** (Rensselaer Optimistic Simulation System) developed by Rensselaer Polytechnic Institute
- HPC networks
  - N-ary k-cube network (IBM Blue Gene, Cray XT5 and Cray XE6 series.)
  - Dragonfly (Cray XC30 system.)
- Burst buffer architectures
  - N. Liu, J. Cope, P. Carns, C. Carothers, R. Ross, G. Grider, A. Crume, and C. Maltzahn. On the role of burst buffers in leadership-class storage systems. In Proceedings of the 2012 IEEE Conference on Massive Data Storage (MSST), Pacific Grove, CA, April 2012.
- Resource and Job Management in Data Analytics Environments
  - N. Liu, X. Yang, X. Sun, J. Jenkins, R. Ross. YARNsim: Hadoop YARN Simulation System. Accepted to CCGrid 2015.
- Understanding Behavior of Metagenomics Workflows on Multi-Cloud Systems
  - W. Tang, J. Jenkins, F. Meyer, R. Ross, R. Kettimuthu, L. Winkler, X. Yang, T. Lehman, and N. Desai. Data-Aware Resource Scheduling for Multicloud Workflows: A Fine-Grained Simulation Approach. In Proceedings of CloudCom 2014. December, 2014.

http://www.mcs.anl.gov/research/projects/codes/

Outline

- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

Aesop Programming Language

Goal: Increase team productivity by making programming of distributed services easier.

- New programming language (+ support libraries)
  - Based on C with concurrency extensions
  - Designed for implementing distributed network services
  - Aims to be highly productive – sequential flow while programming
  - Aims to be fast – concurrent execution
- Implemented as Source-To-Source translator
  - Translator written in Haskell, injects macro calls into the source.
  - Outputs plain C
- Git repository at `git://git.mcs.anl.gov/aesop`

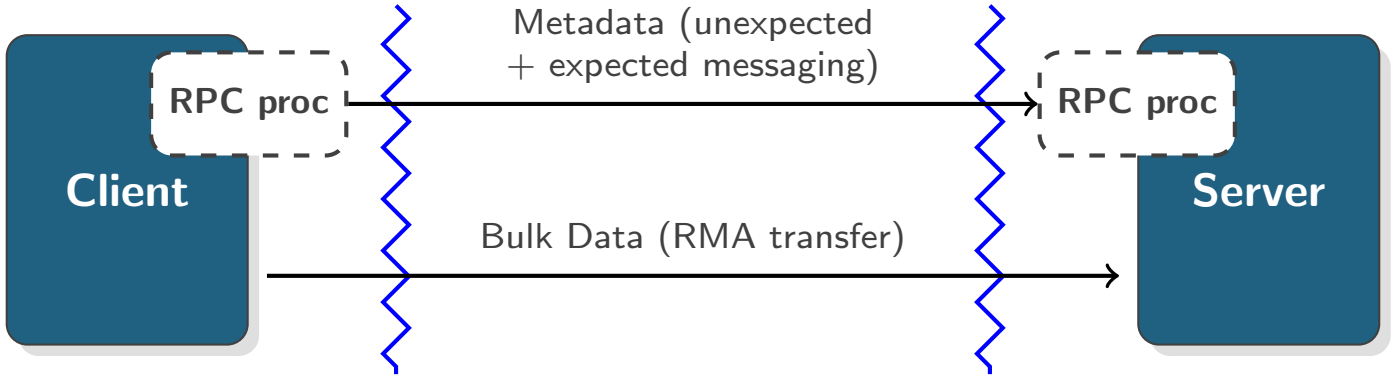
D. Kimpe, P. Carns, K. Harms, J. M. Wozniak, S. Lang, and R. Ross. AESOP: Expressing concurrency in high- performance system software. In Proceedings of the 7th IEEE International Conference on Networking, Architecture and Storage (NAS), June 2012.

Mercury RPC Framework

The HDF Group

Mercury is an RPC system for use in the development of high performance system services. Development is driven by the HDF5 Group with Argonne participation.

- Portable across systems and network technologies
- Builds on lessons learned from IOFSL, Nessie, lnet, and others
- Efficient bulk data movement to complement control messages



Network Abstraction Layer

http://www.mcs.anl.gov/research/projects/mercury/

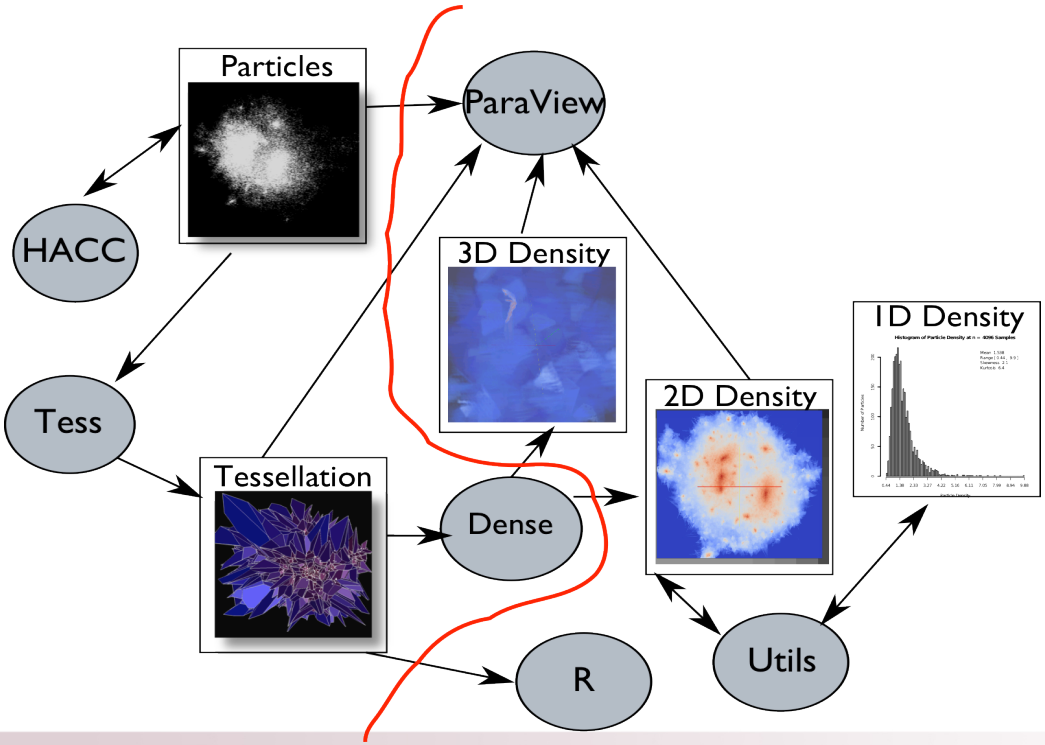
Outline

- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

Decaf Motivation

- All science problems have multiple connected steps, i.e., workflows.
- Challenging to automate these workflows on the current HPC infrastructures.

Example of a specific workflow in cosmology (HACC)

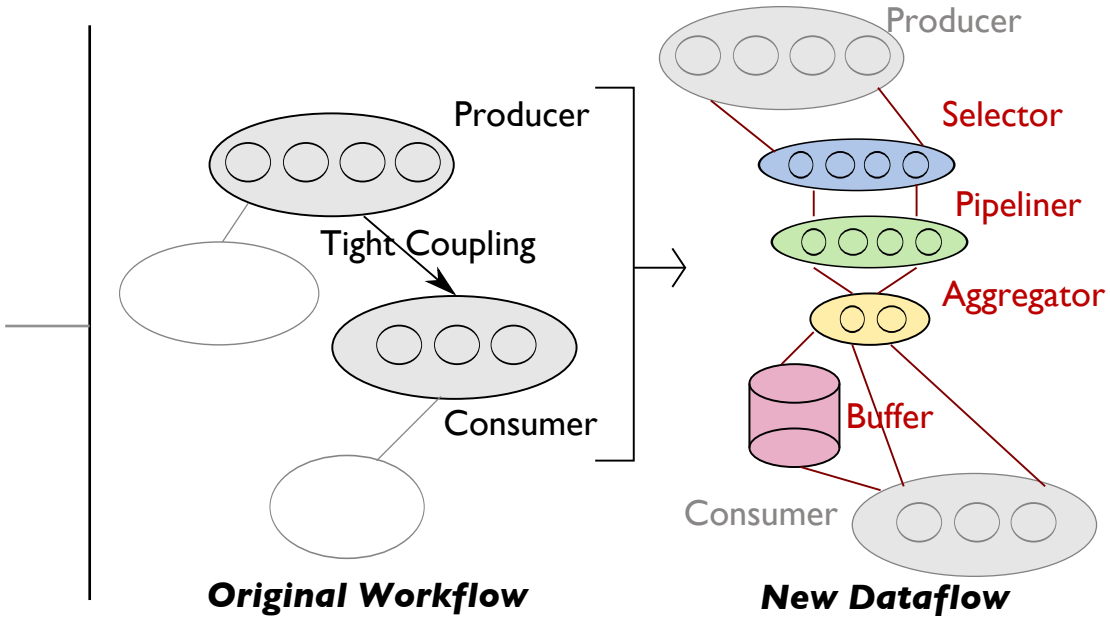


Decaf

- Decaf: software that expands one *workflow* link into a *dataflow*.
- Dataflow: one link in the workflow graph expanded to express control and data movement.
- It includes data selection, pipelining, buffering, and resilience.

Expanding one link in a workflow into a dataflow consisting of reusable primitives.

Producer, consumer, and dataflow are all parallel tasks.



Swift/T workflow language

- Swift/T: Language and runtime for dataflow applications

```
(int r) myproc (int i, int j)
{
    int f = F(i);
    int g = G(j);
    r = f + g;
}
```

  - F( ) and G( ) implemented in native code or external programs
  - F( ) and G( ) run concurrently in different processes
  - r is computed when they are both done

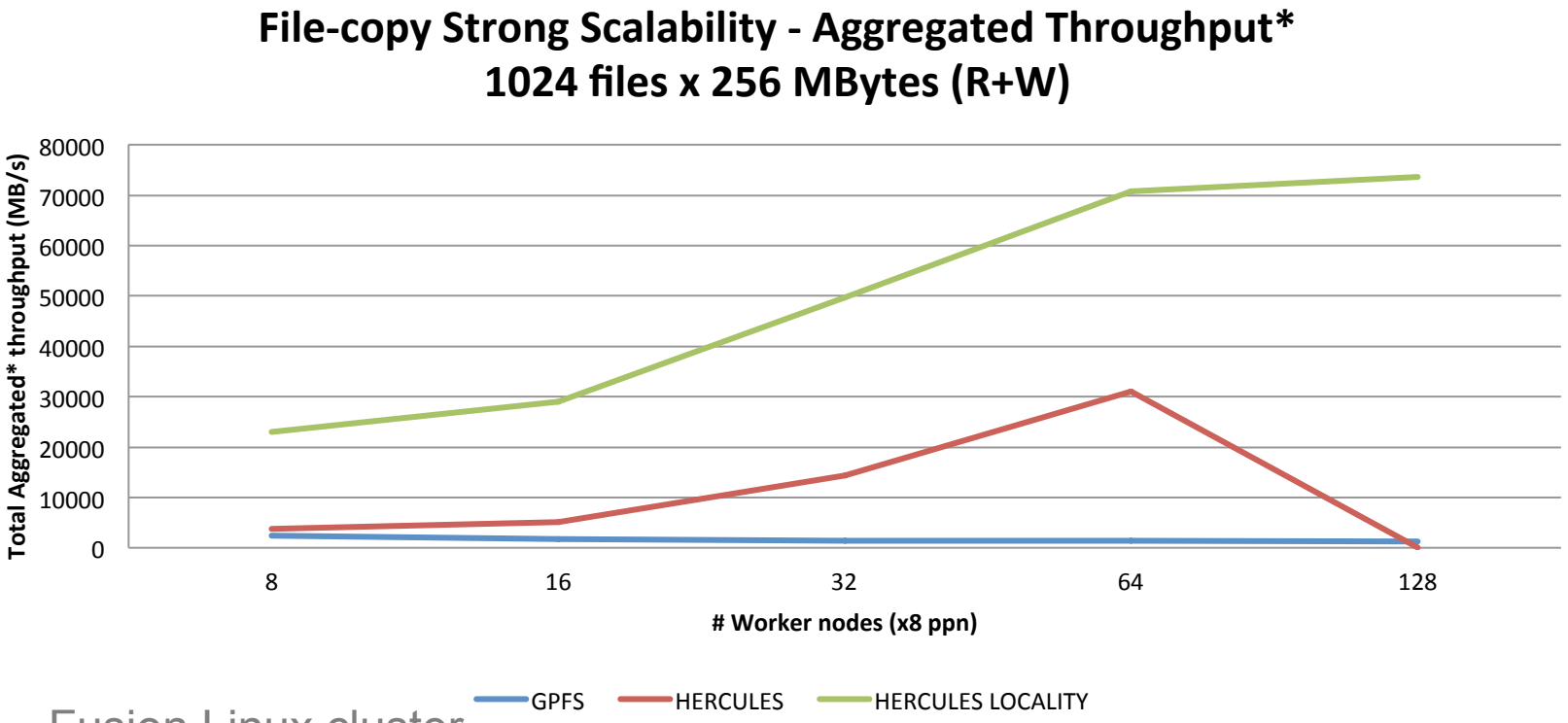
Data locality in Swift/T

- Joint work with University Carlos III (Spain)
- Problems
  - Load balancer is not locality-aware
  - Tasks communicate through the parallel file system (bottleneck)
- Objectives:
  - Improve the performance of inter-task communication
    - Data locality
  - Investigate the tradeoffs between data locality and load-balance in workflow execution

Approach

- Hercules
  - persistent key value store based on Memcached
  - On-demand deployment of servers on application nodes
- Data placement over the servers
  - Consistent hashing (original Memcached)
  - Locality-aware (implemented)
  - Load-aware (under implementation)
  - Capacity aware
- New Swift language constructs
  - Soft location: best effort task placement
  - Hard location: enforce data locality

Exploiting data locality in Swift/T workflows using Hercules. Francisco Rodrigo Duro, Javier Garcia Blas, Florin Isaila, Jesus Carretero, Justin M. Wozniak, and Robert Ross. Proc. NESUS Workshop 2014 .



Fusion Linux cluster

- 320 nodes, 2 x quad core, 36 GB RAM
- Infiniband QDR (4 GB/s) and gigabit ethernet
- GPFS: up to 2500 MB/s

Outline

- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

CLARISSE: Data staging coordination

- Joint work with University Carlos III (Spain)
- Challenges
  - Concurrent parallel data flows
  - Lack of data staging coordination
    - Among applications
    - Between applications and the system
  - Increasing storage hierarchy
  - Current HPC Storage I/O stack difficult to optimize

- Goal: offer novel mechanisms for data staging coordination to improve
  - Load balance
  - Resilience
  - Parallel I/O scheduling

F. Isaila, Garcia, J., Carretero, J., Ross, R. B., and Kimpe, D., "Making the Case for Reforming the I/O Software Stack of Extreme-Scale Systems", EASC 2013. Edinburgh, Scotland, 2013.

CLARISSE

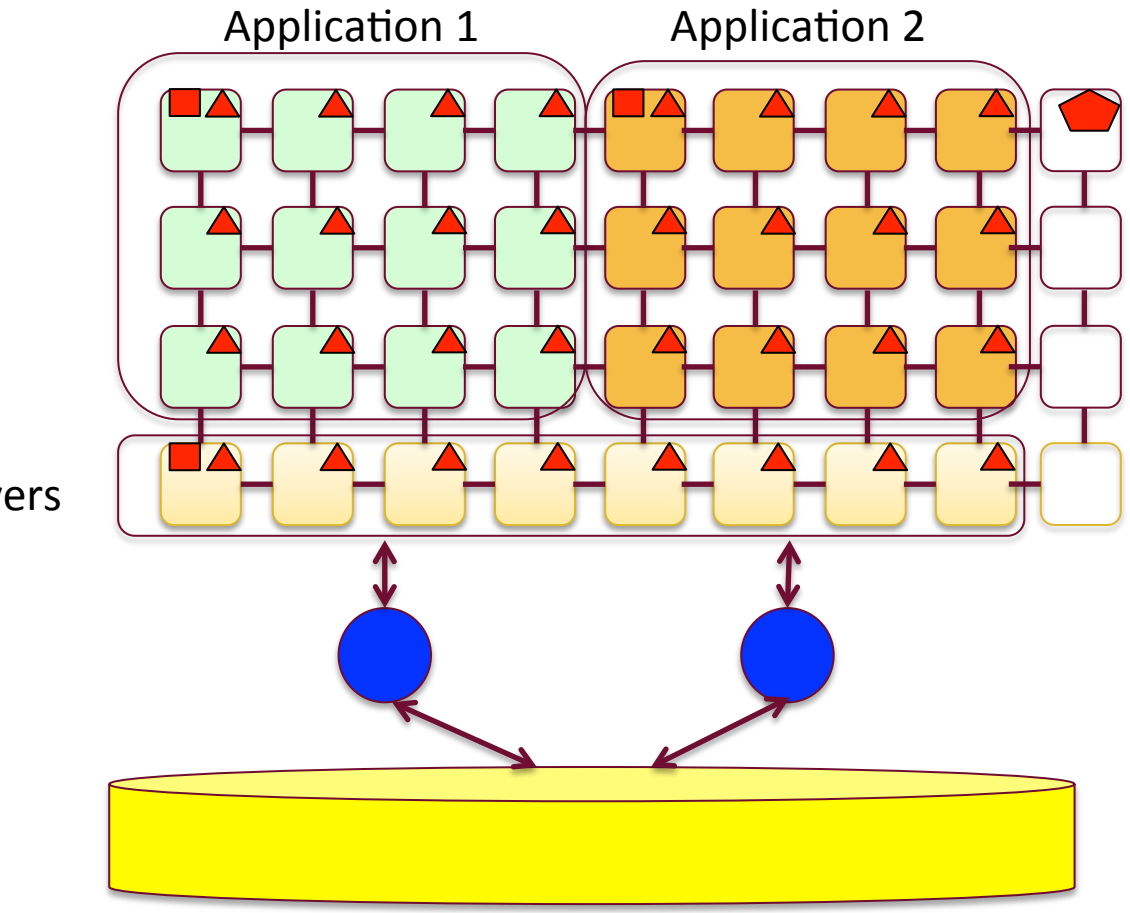
- Novel data staging library
- Separated data and control flow (SDN-like)
- Control backplane
  - On top of a pub/sub substrate (e.g. Beacon communication backplane)
  - Used for
    - I/O scheduling
    - Load balance
    - Resilience
    - Multiple stage coordination (e.g. : aggregation, storage I/O)
- Data path
  - Collective and independent I/O
  - Control backplane used for coordination
  - Adaptive buffering
  - Support for parallel workflows (Decaf)

CLARISSE hierarchical control infrastructure

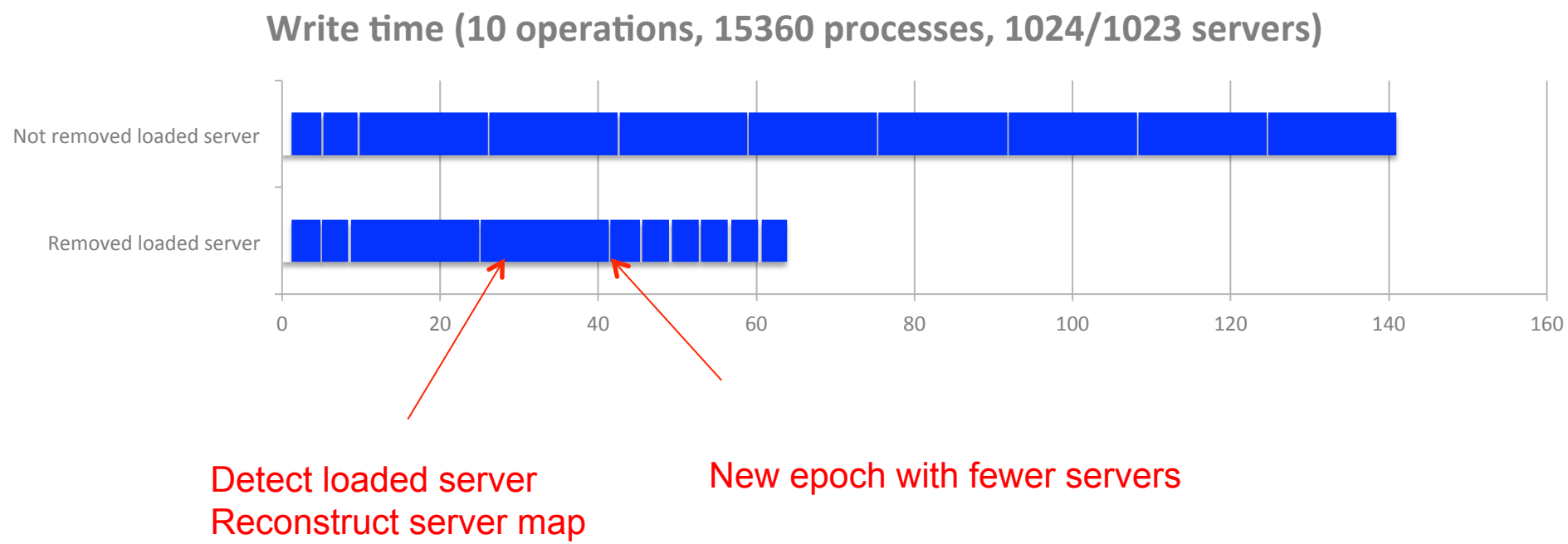
Node Controller

Application Controller

Global Controller



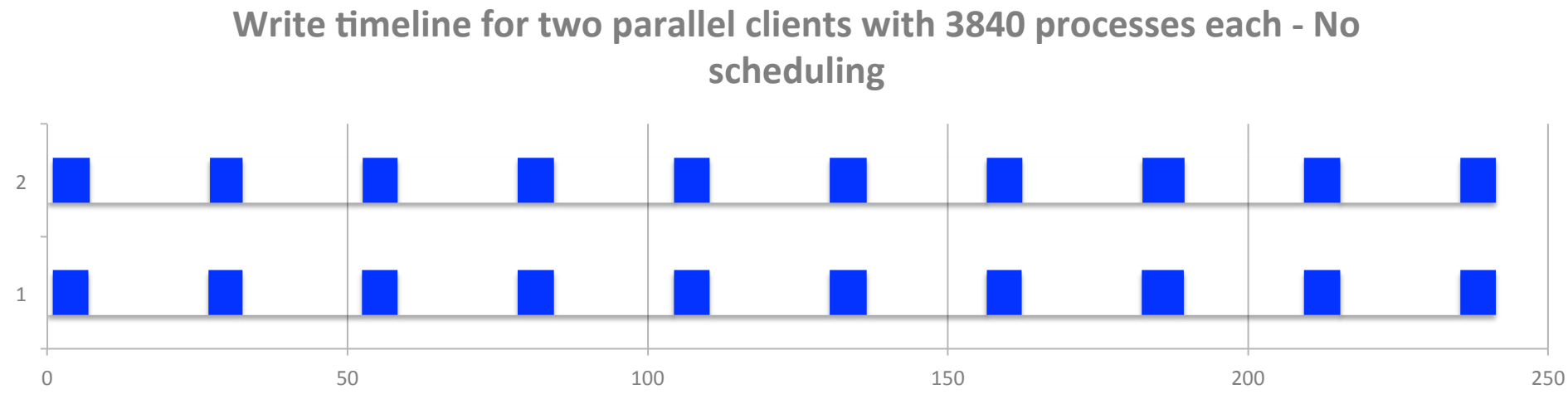
Dynamically eliminating loaded server

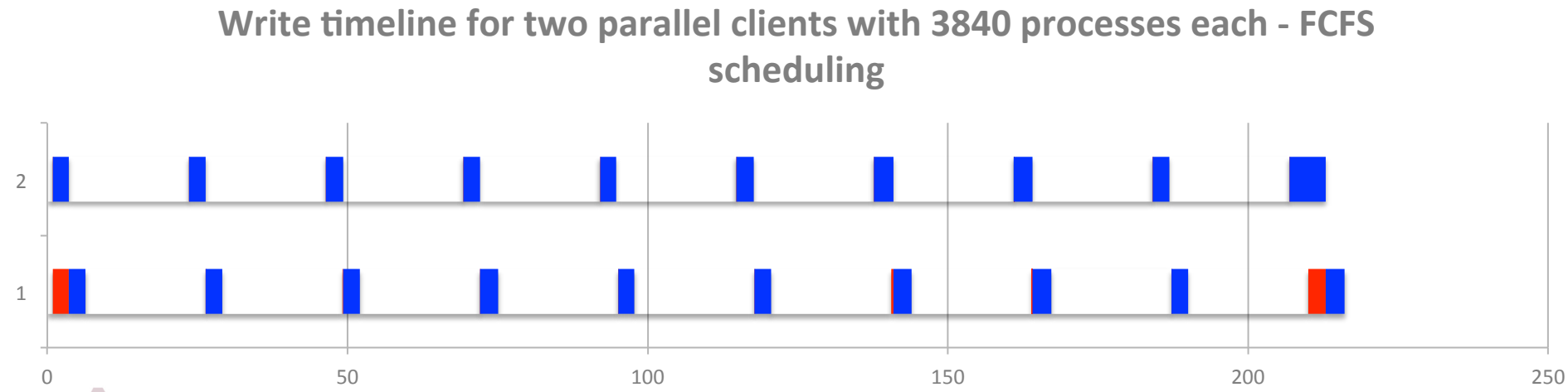


Detect loaded server
Reconstruct server map

New epoch with fewer servers

FCFS scheduling versus no scheduling



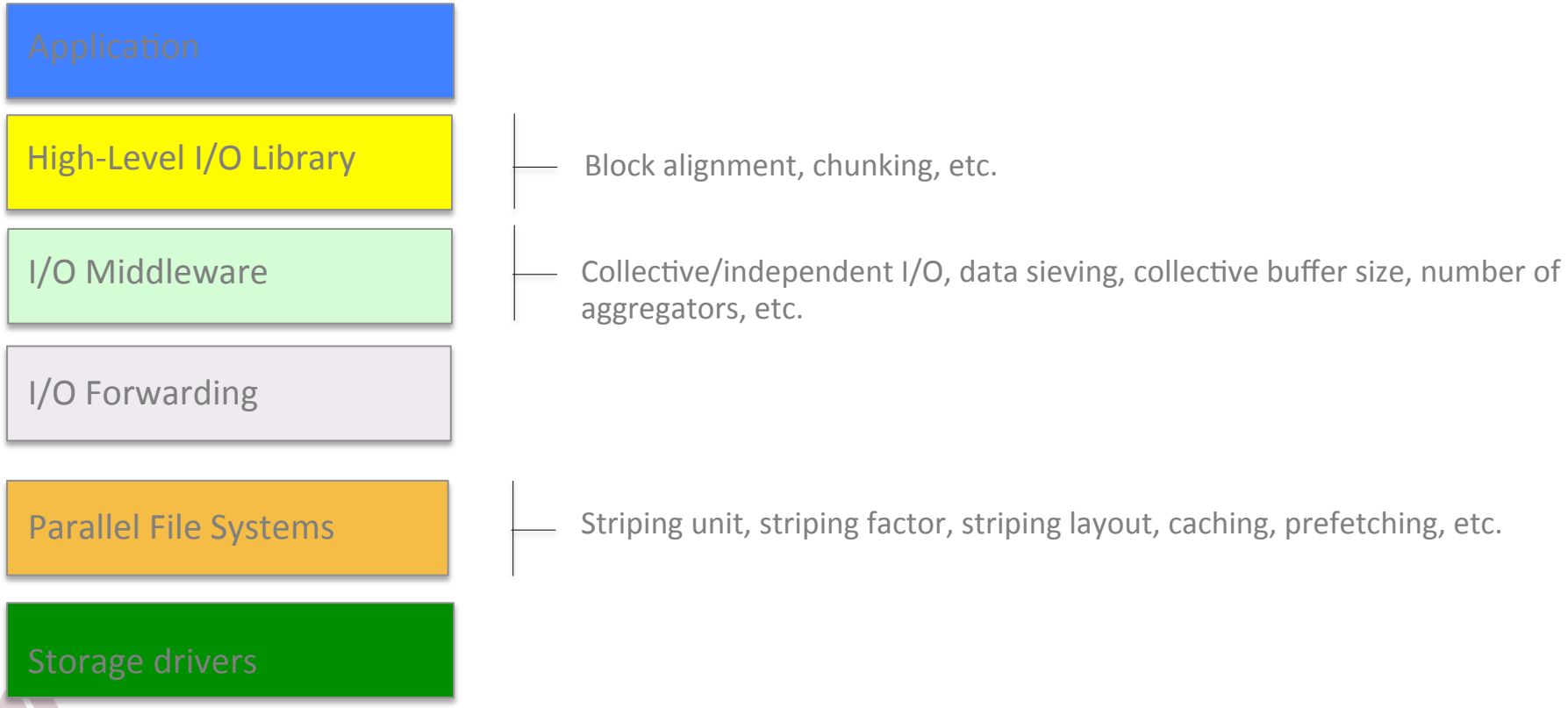


Outline

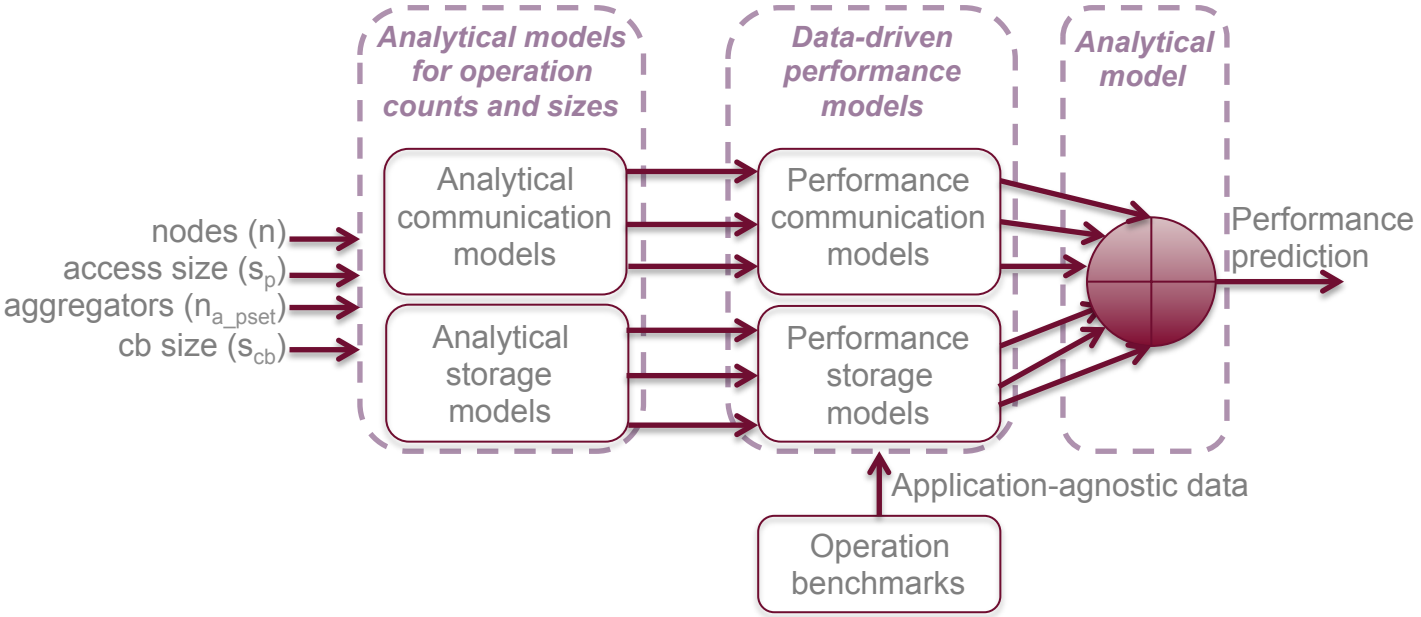
- A bit of history
- I/O characterization
- Simulation
- Tools
- Scientific workflows
- Data staging coordination
- Parallel I/O autotuning

Parallel I/O tuning motivation

- Huge parameter space of the storage I/O software stack
- Domain knowledge is increasingly harder: software and hardware complexity



Modeling ROMIO collective I/O



F.Isaila, P. Balaprakash, S. Wild, D. Kimpe, R.Latham, R.Ross, P.Hovland. Collective I/O tuning using analytical and machine learning models. In Proceedings of IEEE Cluster September 2015.

Conclusion

- Storage is part of a high end computing ecosystem.
- Historically storage I/O has been a second-class citizen.
- Big data requires global solutions
  - Fundamental redesign of the software storage I/O stack
  - Integration with resource management, scheduler, and workflow systems
  - Software defined solutions
- Integration of big data and HPC stacks