Argonne NATIONAL LABORATORY

HEC I/O Measurement and Understanding

Phil Carns (carns@mcs.anl.gov)
Mathematics and Computer Science Division
Argonne National Laboratory

U.S. DEPARTMENT OF ENERGY

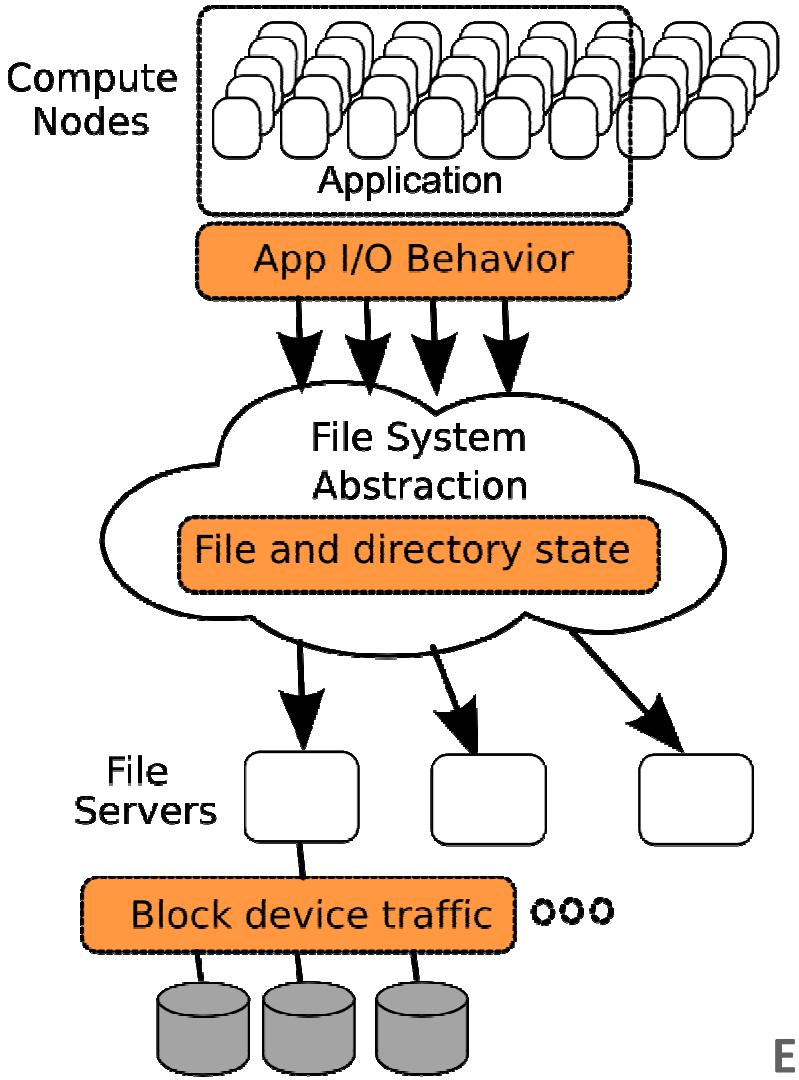
Understanding production workloads: Current technology

- Key questions for scientists:
  - How can we guide application tuning to improve productivity?
- Key questions for I/O researchers:
  - What tools and resources are (or are not) being used? How do we plan for the future?
- Key questions for administrators:
  - Where should tuning and support efforts be focused?

There is no reason we shouldn't be able to answer these questions now!

- Observing production workloads is critical
- Real-world workloads often include a variety of applications not represented by popular benchmarks, especially on open science systems

Where can we collect data? Three views



- **What did the application intend to do? How does it access its data?**
- What are the characteristics of data at rest? How large are the files and who owns them?
- What workloads are being serviced at the hardware level?

Each view may offer a different perspective on the same overall workload.

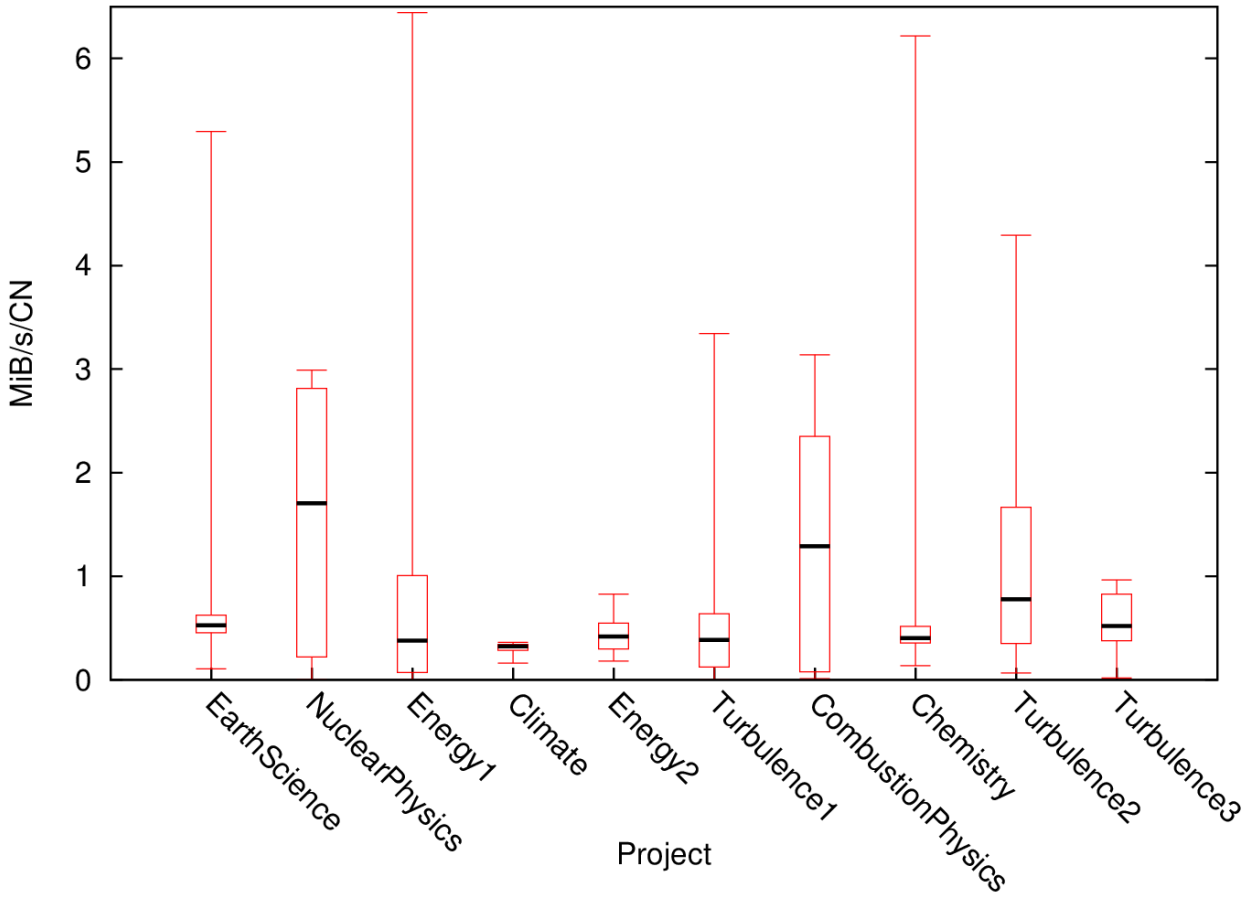
Understanding application I/O behavior with Darshan

- Transparently capture as much I/O access pattern information as possible from production jobs
  - Enabled by default since January 2010 on the Intrepid BG/P system at Argonne
  - Currently instruments **~ 60% of all core hours** consumed on the system
- Darshan provides application-level characterization with file and process granularity
  - Does *not* capture full I/O traces; instead it records counters, histograms, access patterns, and strategically chosen timestamps related to I/O activity
  - Low overhead at runtime; reduction, compression, and storage is performed at MPI_Finalize() time
- Not as in depth as some tools, but its low overhead allows us to observe production activity that would otherwise go unnoticed
- Repository of anonymized logs available for public use

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

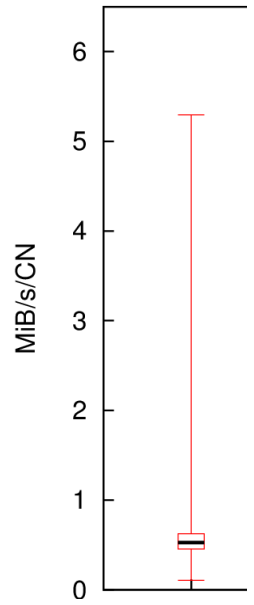
Observing behavior across application domains

- Normalized metric to compare I/O performance across jobs
- Accuracy is limited, but it provides some indication of which applications are most interesting
- Many different I/O strategies used by different applications!



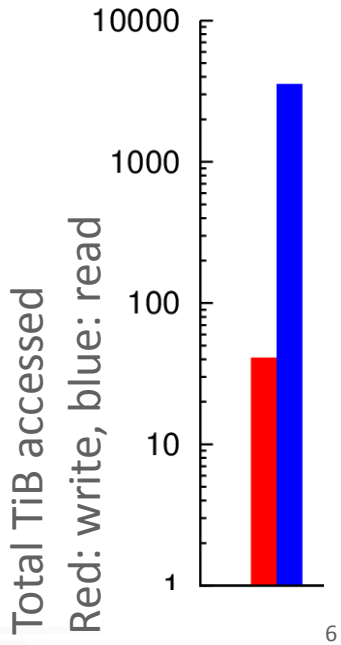
Investigating and tuning individual applications

- Darshan includes tools that help users and catalysts investigate the behavior of individual jobs in greater detail

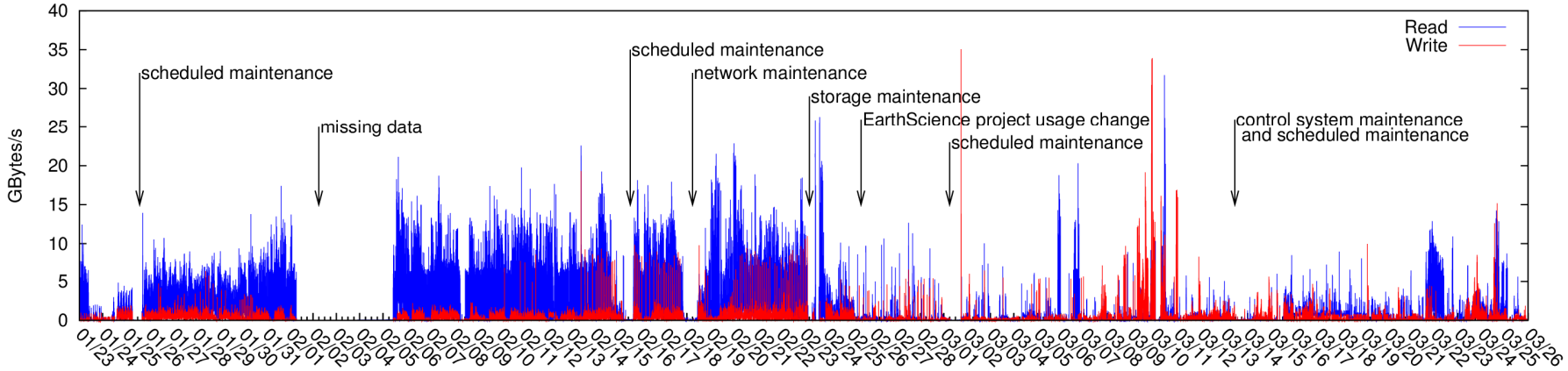


- This example project accessed 3.5 PiB of data over 2 months
- Low performance, despite some positive characteristics like relatively large access size
- File usage:
  - Over 1 million files accessed by some 4096 process script jobs
  - One read or write per file
- 95% of I/O time was spent performing metadata operations
- Write-only files holding back performance despite high read ratio

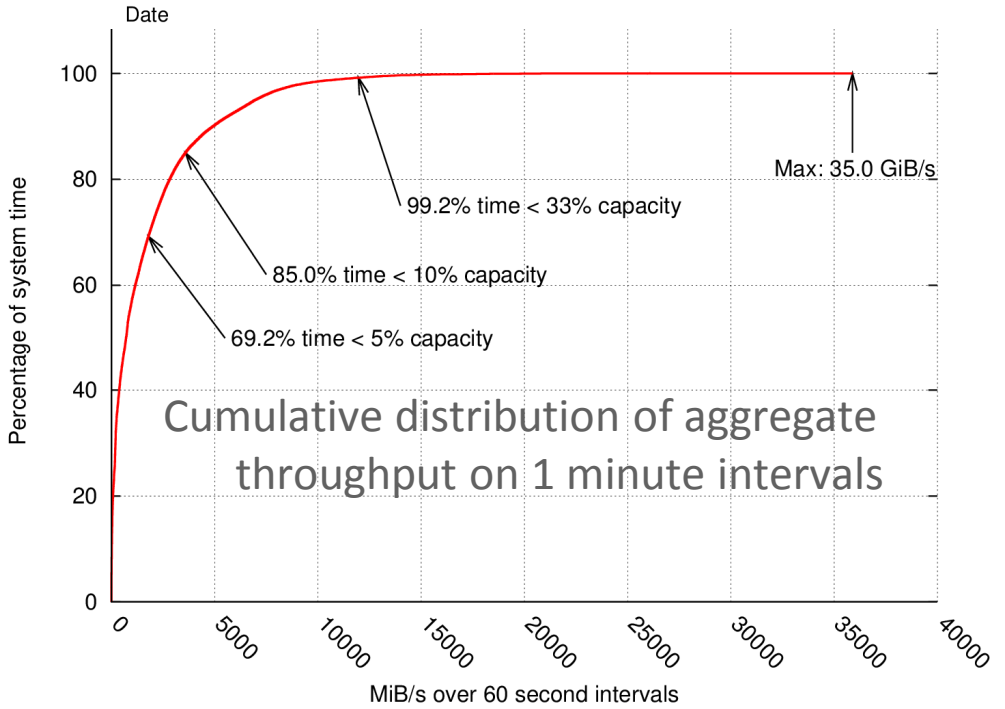
- *… but interpreting behavior and suggesting improvements is a manual process that requires knowledge of the storage system and I/O tuning experience.*



What about production behavior at the system level?



- Several off the shelf tools exist, and relatively modest instrumentation can yield interesting information
- This data was collected using "iostat"
- Allows us to observe bursty behavior and idle time that could be leveraged by system software algorithms



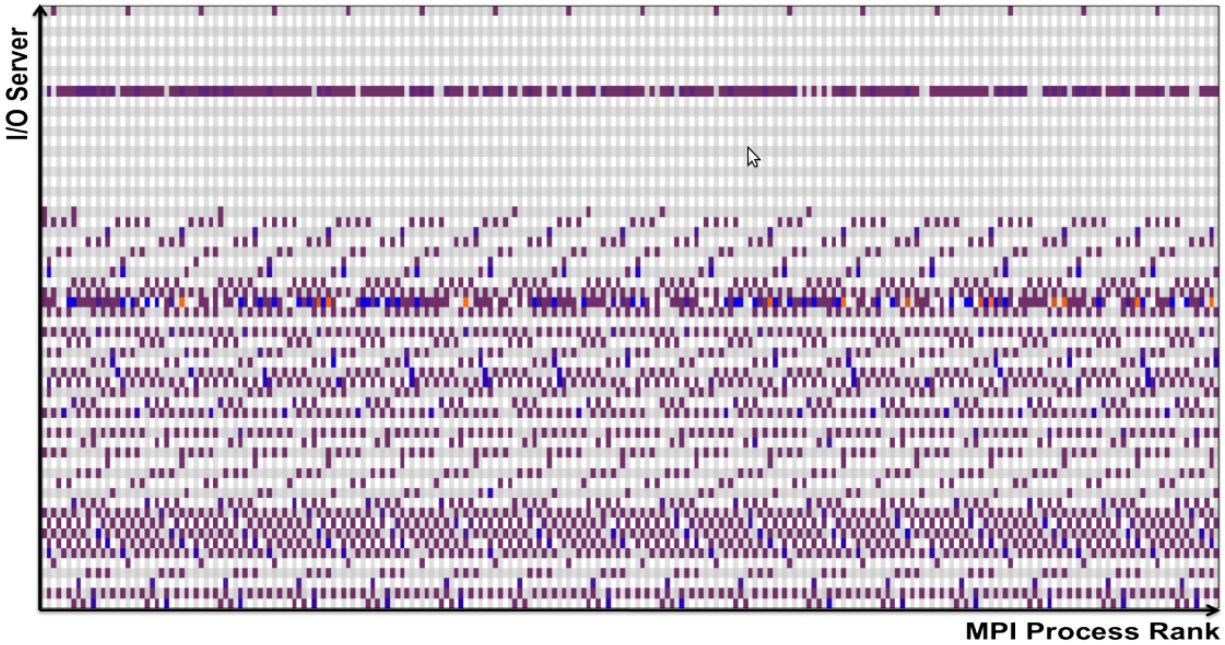
What if we drill down with in-depth instrumentation?

- IOVIS project (UC Davis and ANL)
- Opposite end of the spectrum from Darshan-level instrumentation
- Collects traces that link together:
  - MPI-IO operations
  - File system client operations
  - Network messages
  - File server operations

- Key challenges:
  - **How to capture and store the trace data efficiently**
  - **How to analyze and visualize the results in a meaningful way**

- Scale is a problem both for visualization and data capture
- Traditional time lines and Gantt charts are not clear with thousands of processes
- Requires new methods to interpret results and identify the source of I/O overhead

IOVIS example



- This matrix shows messages intensity between clients and servers using color opacity
- Servers on the Y axis
- Clients on the X axis

- Potential bottlenecks appear as dark horizontal lines
- Poor balance across servers in this example

C. Muelder, C. Sigovan, K.-L. Ma, J. Cope, S. Lang, K. Iskra, P. Beckman, and R. Ross. Visual analysis of I/O system behavior for high-end computing. In Proceedings of the third international workshop on Large-scale System and Application Performance, LSAP '11

Ongoing work on Measurement and understanding

- More deployments and comparisons
  - Hard to draw broad conclusions from a single system
  - Early success at LLNL (Richard Hedges), TACC (Bill Barth), and others
- Investigating trends discovered from early analysis
  - How bad is the I/O variance, and where does it come from?
  - What factors have the greatest impact on performance?
- Production ramifications
  - How can we improve turnaround time in analyzing data?
  - Can we utilize historical job information for I/O-aware scheduling?
- Tuning efforts for a variety of applications (Gpaw, Phasta, HSCD, etc.)

	Job classification					I/O time by slowest rank (seconds)				
job	samples	procs	files	GiB rd	GiB wr	mean	min	max	stddev	CV
EarthScience A	571	2048	57349	624.4	2.8	939.9	185.9	2034.5	186.7	19.9%
EarthScience B	171	4096	57349	927.8	2.8	1157.4	767.9	1564.4	111.0	9.6%
NuclearPhysics A	404	2048	4	4.3	0.0	23.1	19.1	248.8	21.3	92.1%
NuclearPhysics B	407	4096	4	4.3	0.0	24.3	19.1	1291.7	63.7	262.2%
NuclearPhysics C	837	4096	10809	791.1	326.8	450.4	396.4	3764.1	134.3	29.8%
NuclearPhysics D	809	4096	9815	626.8	122.5	246.7	225.4	658.3	29.5	12.0%
Climate A	8	3888	24455	353.8	100.5	720.1	662.1	835.6	56.9	7.9%

Open questions for Measurement and Understanding

- How can we be more proactive with the data we gather?
  - Automatically identify soft system failures, applications in need of assistance, etc.
  - Automatically tune system and library parameters
  - **Towards autonomous storage**
- How do we extrapolate I/O behavior to exascale?
  - A key issue for the CODES simulation project (ANL and RPI)
  - What workloads will be most important? What will the job mix look like?
  - Changes in concurrency and granularity
- How do we manage the data that we are collecting?
  - Fairly compact for individual jobs right now, but broad queries and analysis are not easy
- How much instrumentation can we perform in production?
  - Darshan might have been a little too conservative
  - If we had more data, what would we do with it?

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

Phil Carns

carns@mcs.anl.gov

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