UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

A Multiplatform Study of I/O Behavior on Petascale Supercomputers

Huong Luu[*], Marianne Winslett[*], William Gropp[*], Robert Ross[+], Philip Carns[+], Kevin Harms[+], Prabhat[^], Suren Byna[^], Yushu Yao[^]

[*]: University of Illinois at Urbana – Champaign
[+]: Argonne National Laboratory
[^]: Lawrence Berkeley National Laboratory

illinois.edu

As apps read/write more & more data, I/O becomes more important for performance.

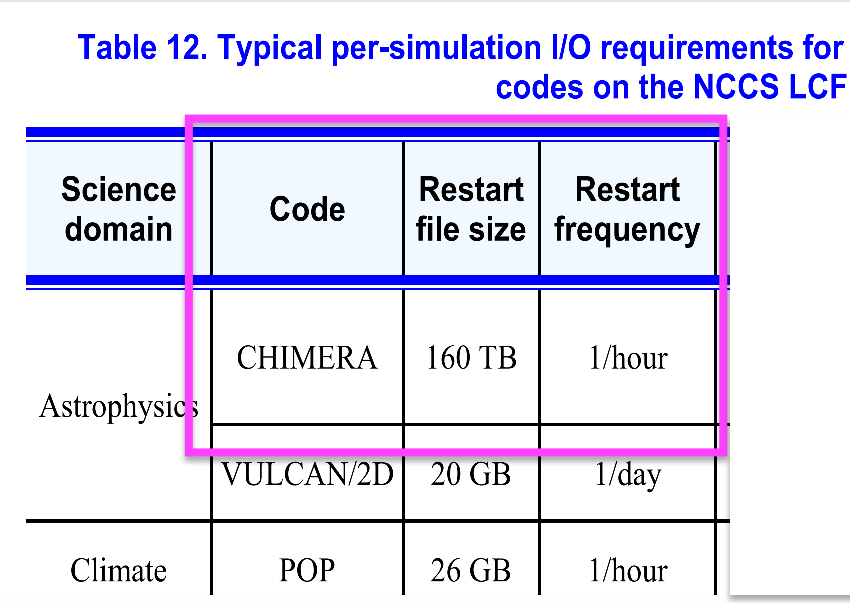
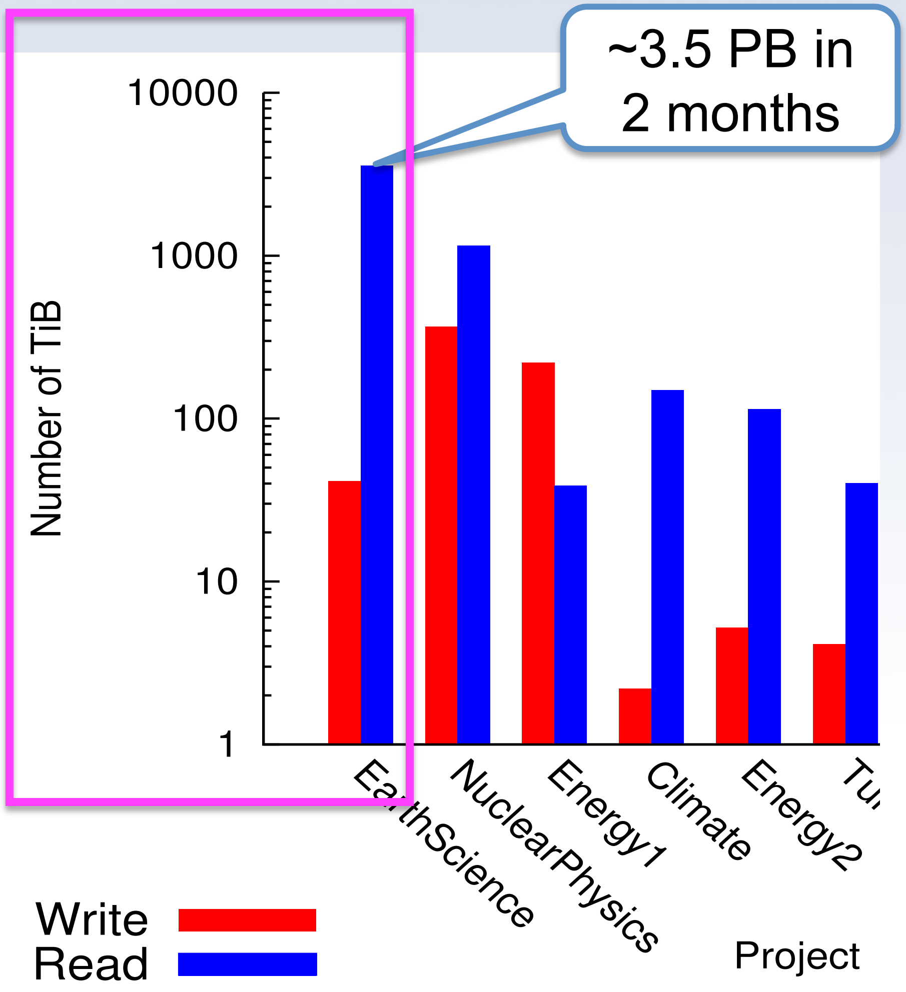


Table 12. Typical per-simulation I/O requirements for codes on the NCCS LCF

Science domain	Code	Restart file size	Restart frequency
Astrophysics	CHIMERA	160 TB	1/hour
	VULCAN/2D	20 GB	1/day
Climate	POP	26 GB	1/hour



~3.5 PB in 2 months

illinois.edu

We study the I/O behavior of thousands of applications on 3 large-scale supercomputers.

- Application-specific, platform-wide, cross-platform analysis.
- Portrait of state of HPC I/O usage.
- Application I/O analysis + visualization procedure.
- Help improve system utilization.

We analyze I/O logs captured by Darshan, a lightweight I/O characterization tool.

- Instruments I/O functions at multiple levels
- Reports key I/O characteristics
- Does not capture text I/O functions
- Low overhead → Automatically deployed on multiple platforms.

We break down I/O time into 4 categories.

I/O time: largest I/O time among all its processes

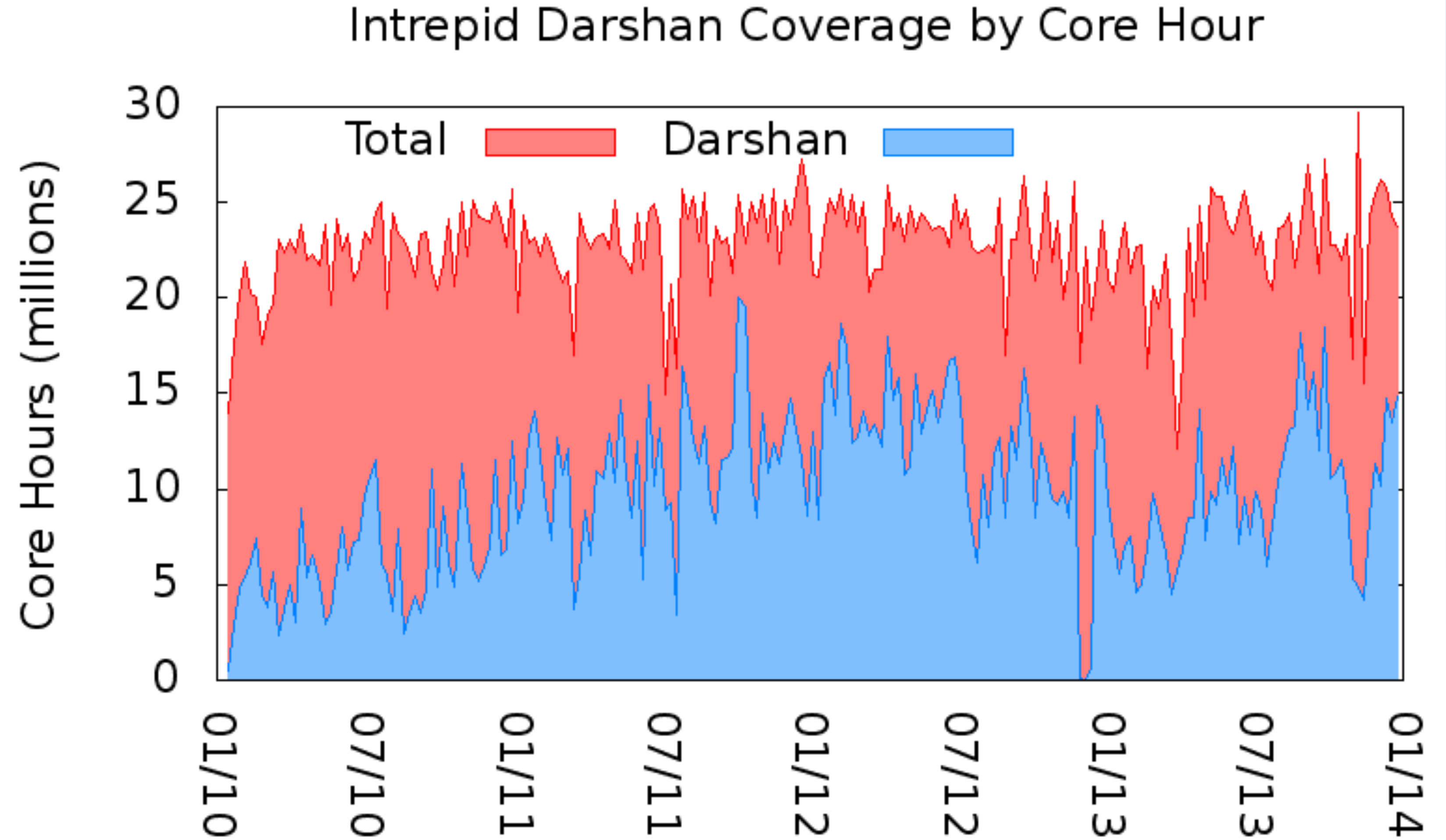
$$\text{Aggregate I/O throughput} = \frac{\text{Total bytes}}{\text{I/O time}}$$

	Global file	**Non-global file**
Metadata (Open, close, seek, …)	🟪	🟥
Data transfer (read, write)	🟩	🟦

I/O log dataset: 3 platforms, >1M jobs, >6 years combined.

	Intrepid	Mira	Edison
Architecture	BG/P	BG/Q	Cray XC30
Peak Flops	0.557 PF	10 PF	2.57 PF
Cores	160K	768K	130K
Total Storage	6 PB	24 PB	7.56 PB
Peak I/O Throughput	88 GB/s	240 GB/s	168 GB/s
File System	GPFS	GPFS	Lustre
# of jobs	**239K**	**137K**	**703K**
Time period	**4 years**	**18 months**	**9 months**

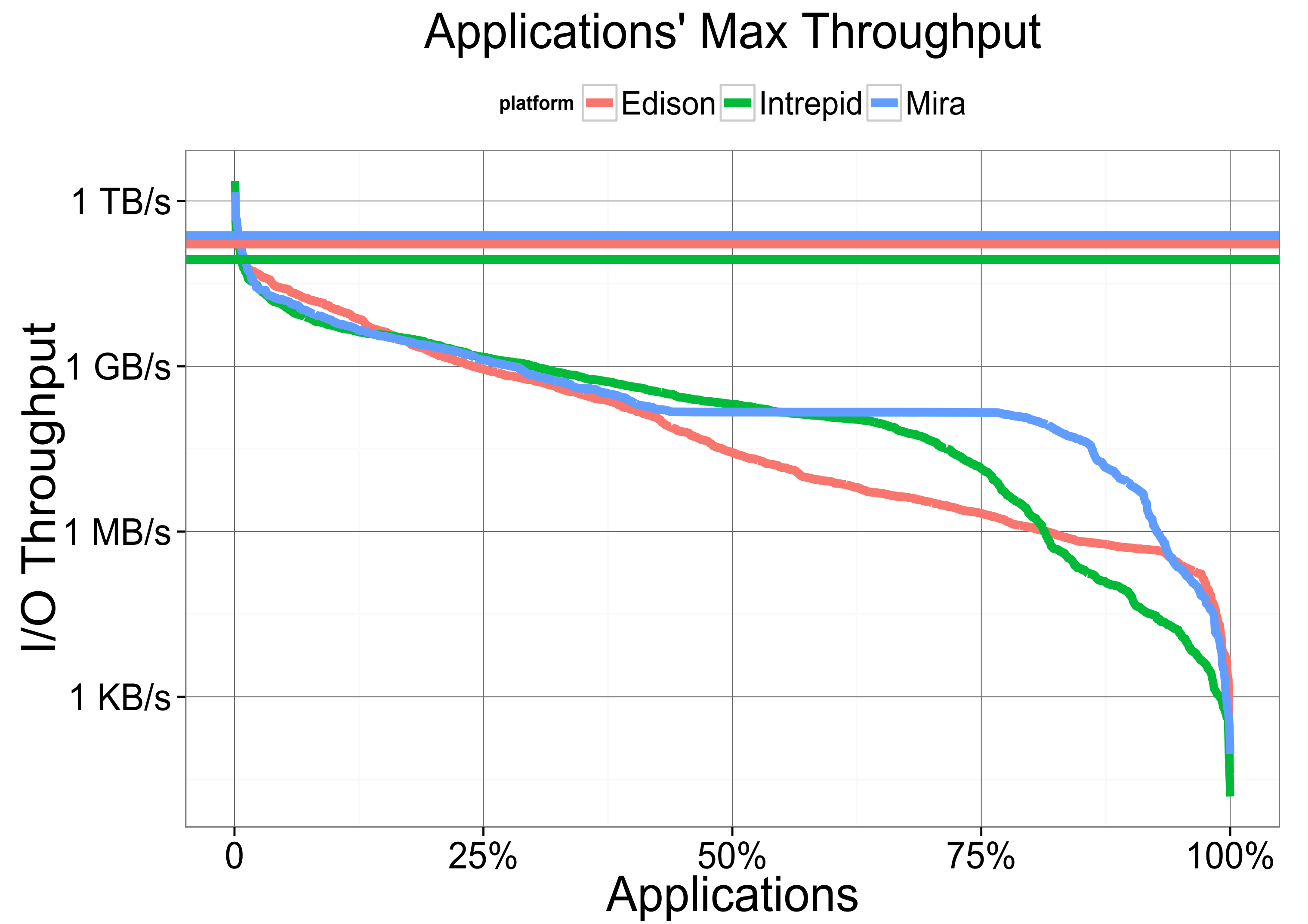
I/O log dataset: 3 platforms, >1M jobs, >6 years combined.



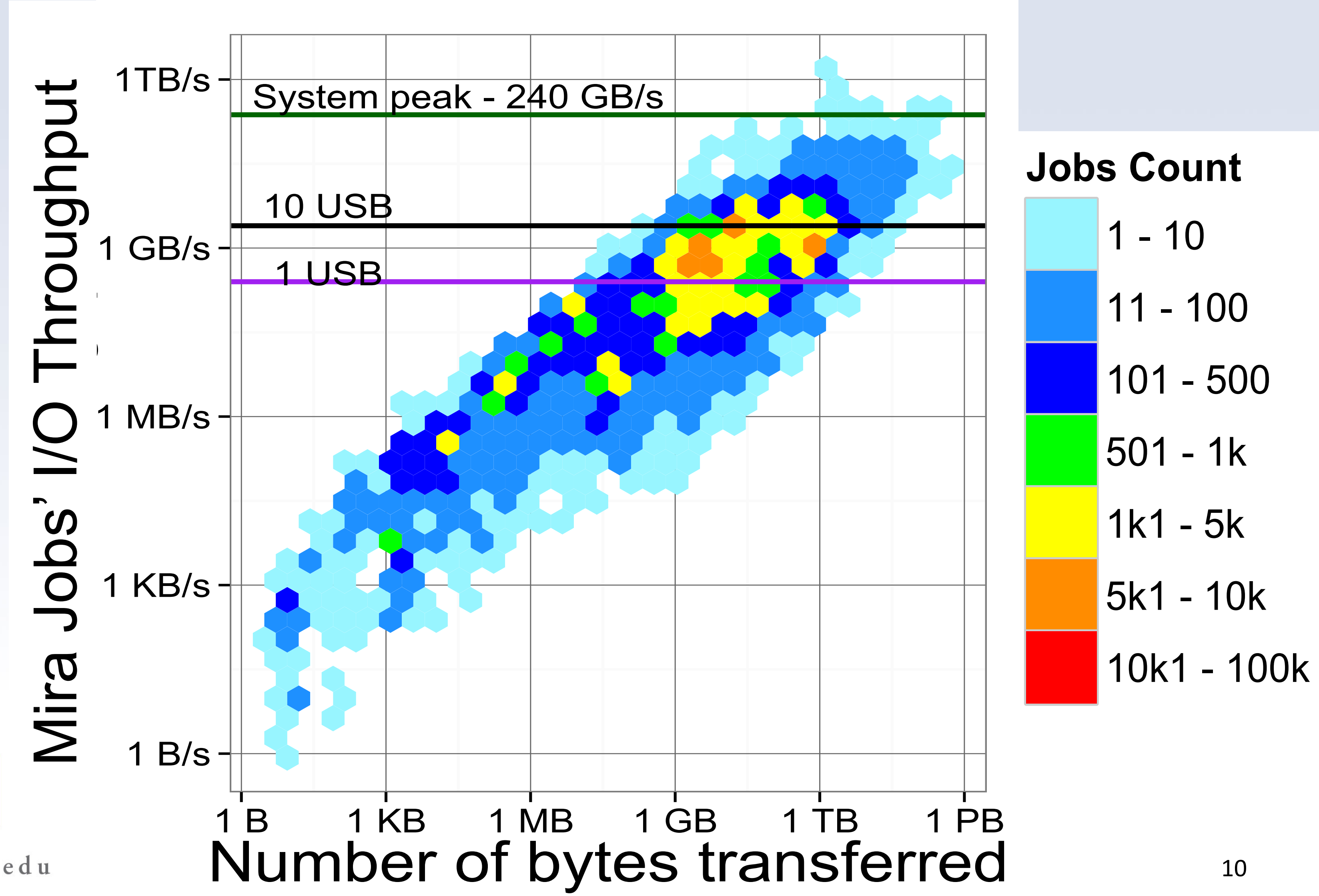
Observations from

PLATFORM-WIDE ANALYSIS

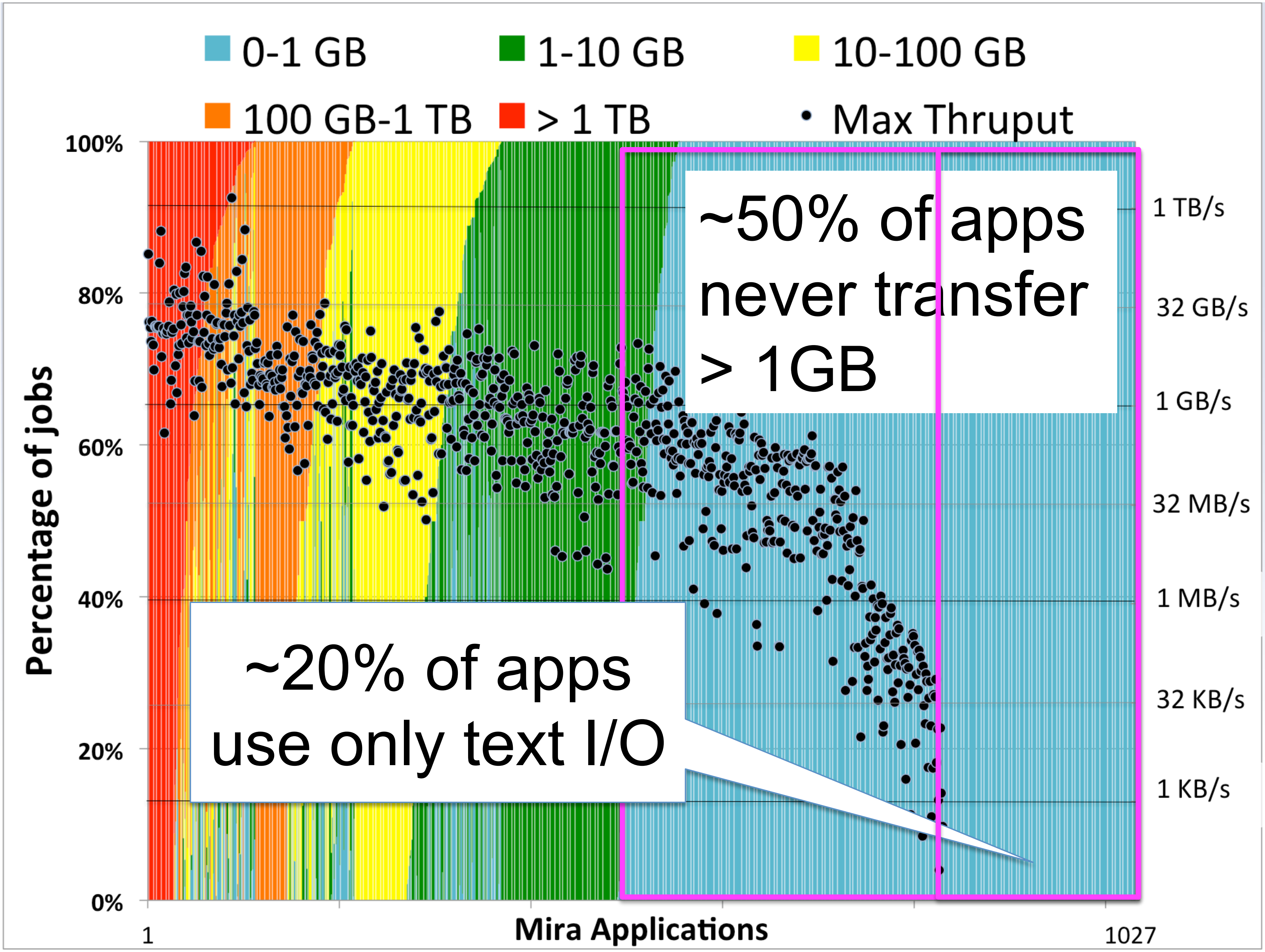
Very low I/O throughput is the norm.



Most jobs transfer little data. Many big-data jobs also have very low thruput.



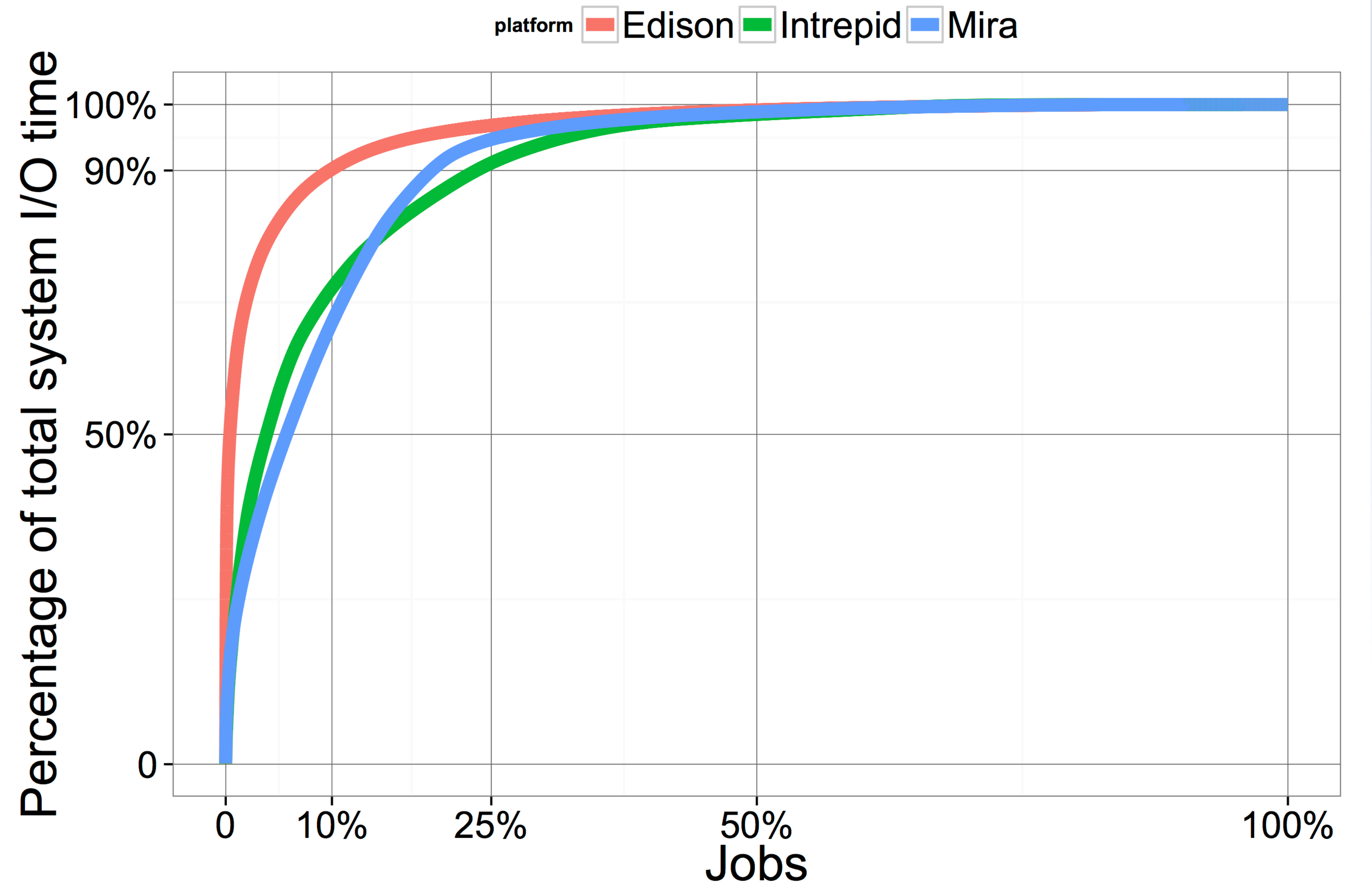
illinois.edu



~50% of apps never transfer > 1GB

~20% of apps use only text I/O

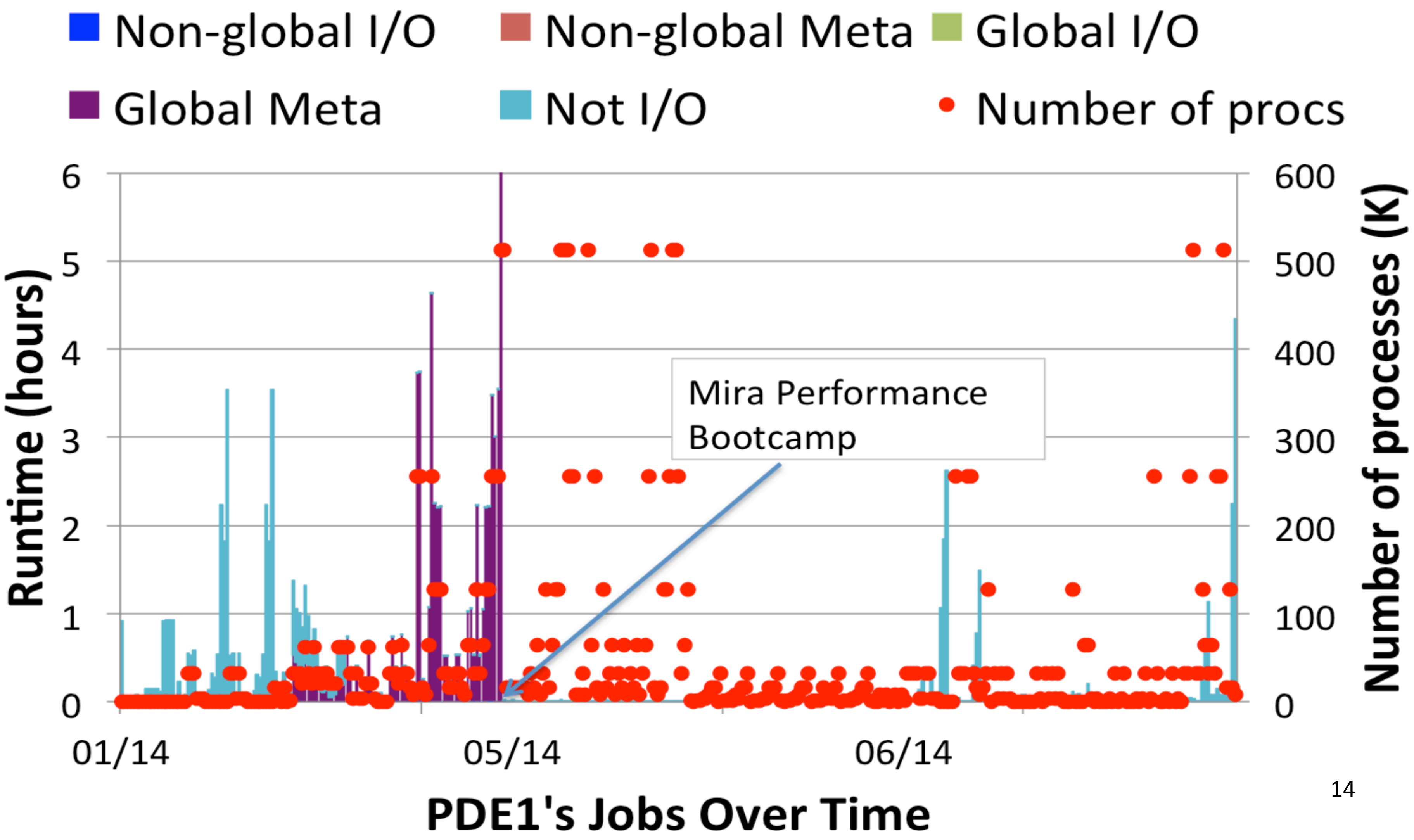
I/O time usage is dominated by a small number of jobs/apps.



Improving the performance of the top 15 apps can save a lot of I/O time.

	Platform I/O time percent	Percent of platform I/O time saved if min thruput = 1 GB/s
Mira	83%	32%
Intrepid	73%	31%
Edison	70%	60%

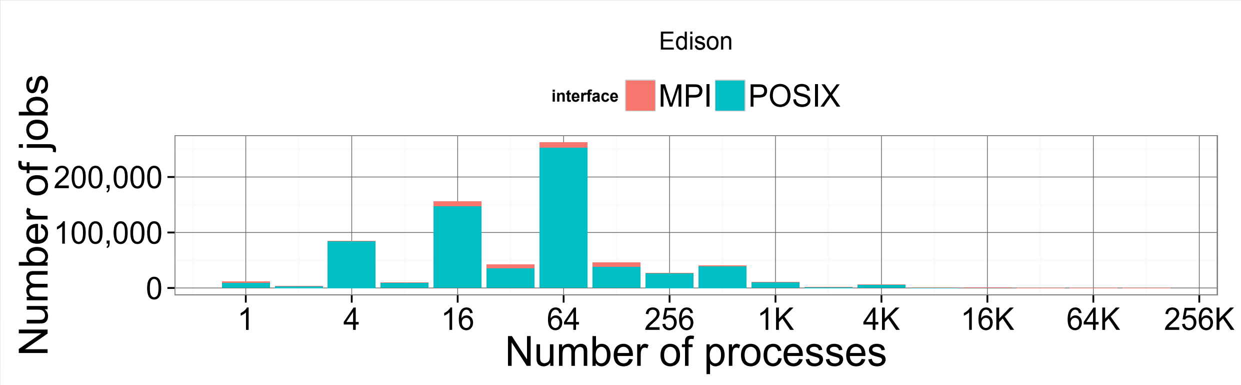
Early intervention by platform admins can help.

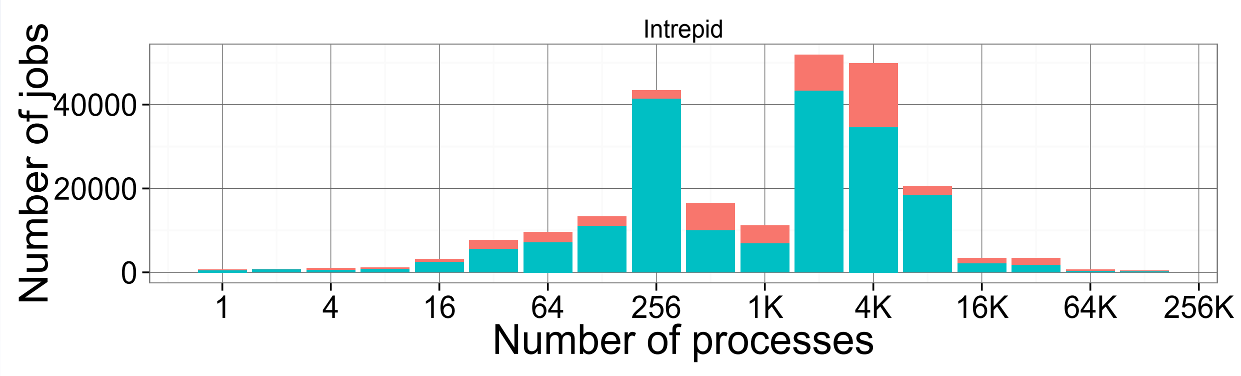


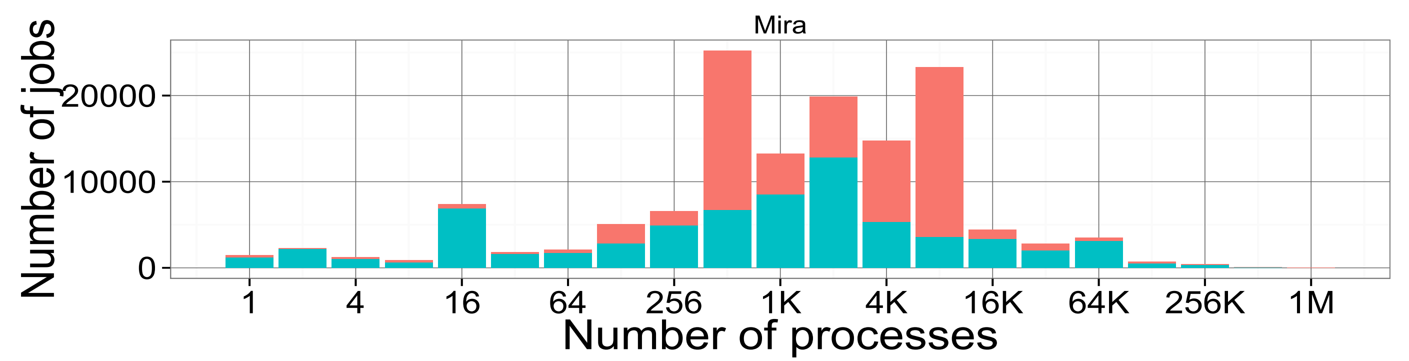
POSIX I/O is far more widely used than parallel I/O libraries.

POSIX-only:

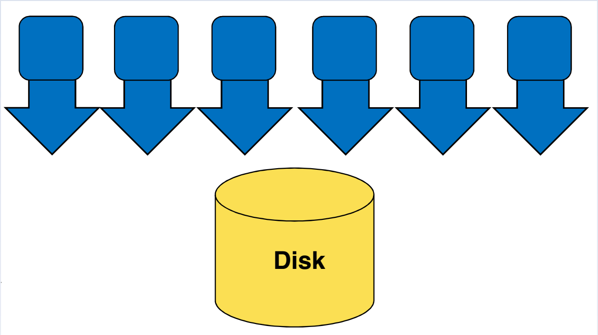
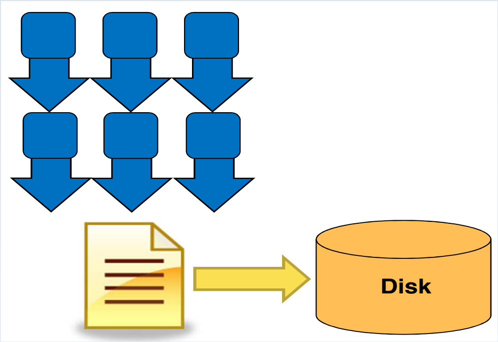
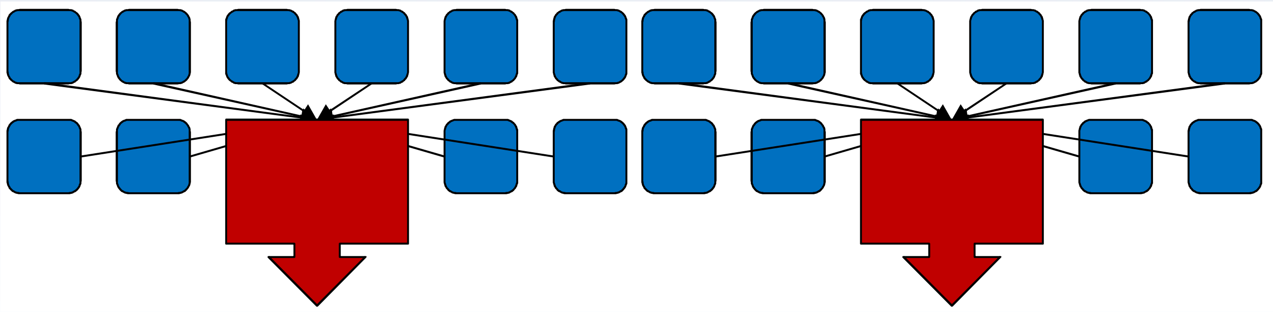
- Edison: 95%
- Intrepid: 80%
- Mira: 50%


Edison


Intrepid

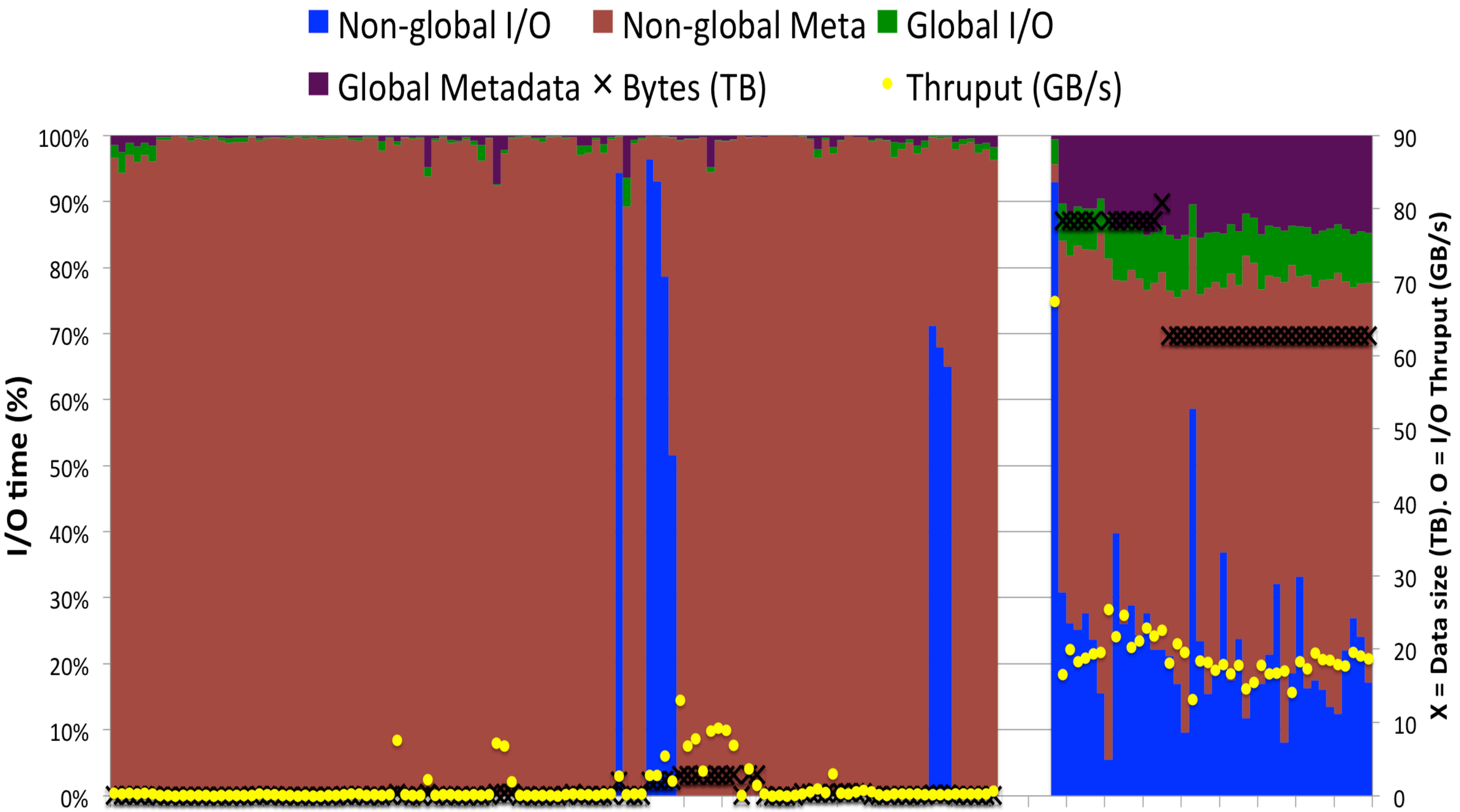

Mira

No major I/O paradigm is always good or bad.

	File-per-process	Global shared files
		
Subsetting I/O		

Minor I/O paradigms that will not scale: Text I/O, Serial I/O

E.g.: File-per-proc can work well if a job has enough data, even with >1M files.



illinois.edu

Physics4 is bad. **Plasma1 is good.**

APPLICATION-SPECIFIC ANALYSIS

Help application's users find

I/O bottlenecks

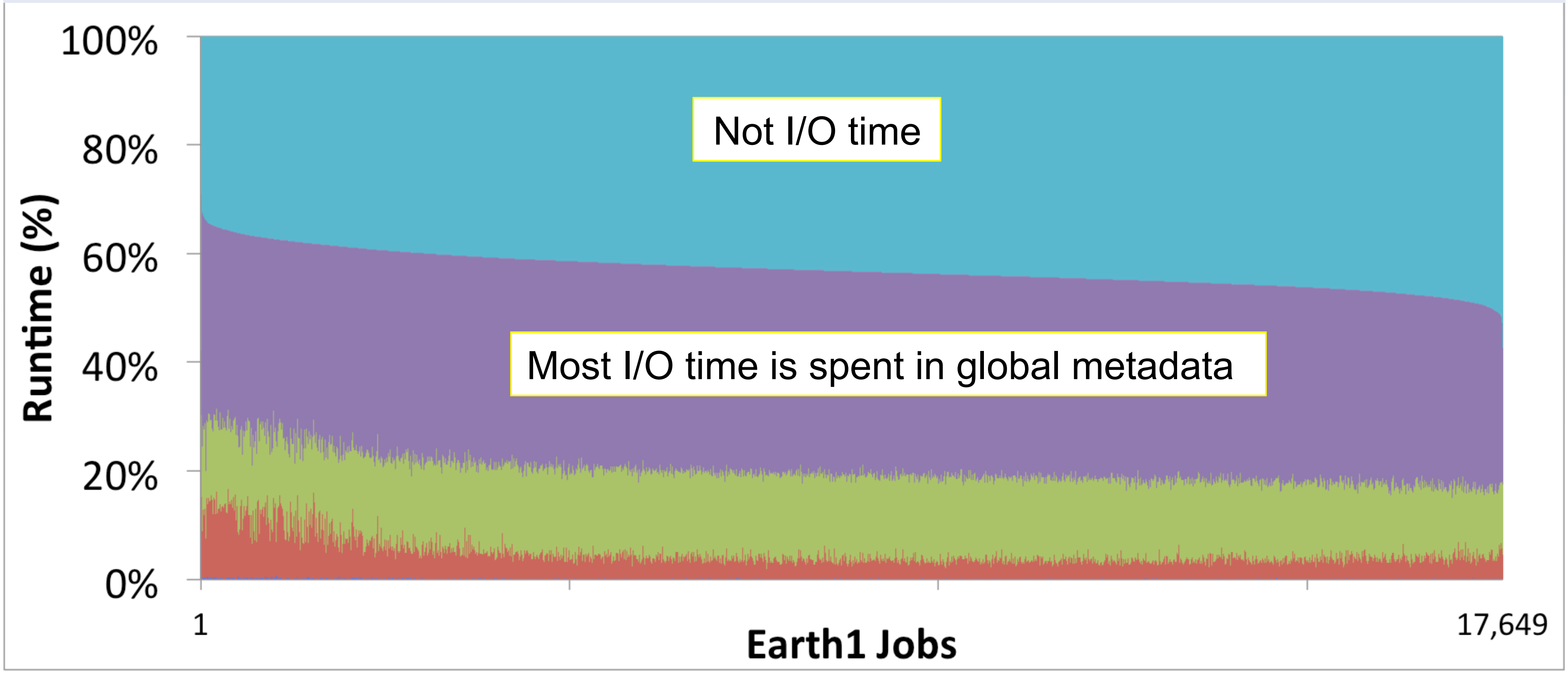
with simple analysis

and visualization procedure.

Earth1 – Mira’s #1 I/O Consumer

1. **Identify where the app spends most of its I/O time:**

Global metadata	**Non-global metadata**
Global data I/O	**Non-global data I/O**



Not I/O time

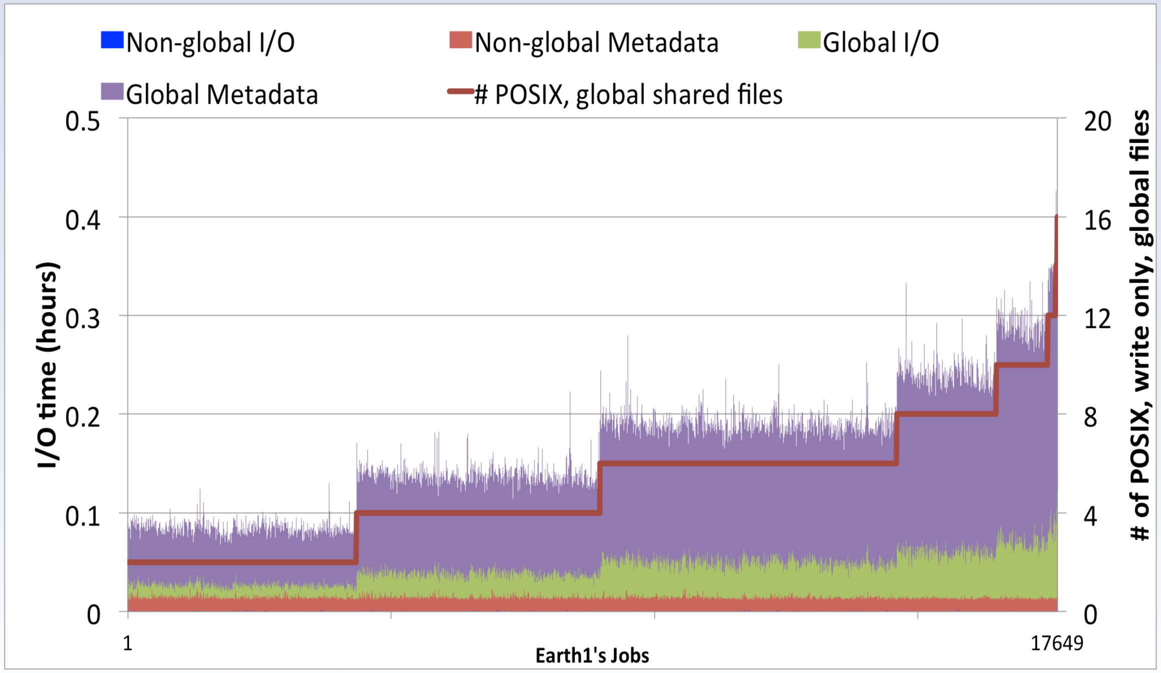
Most I/O time is spent in global metadata

Earth1 – Mira’s #1 I/O Consumer

2. Identify which files or file type consume most time.

One typical job

# files & Type	Bytes	Time
49158 Local, POSIX	619 GB	103s
35 Global shared	34 GB	596s
24 MPI, write only		27s
5 POSIX, read only		2s
6 POSIX, write only		567s



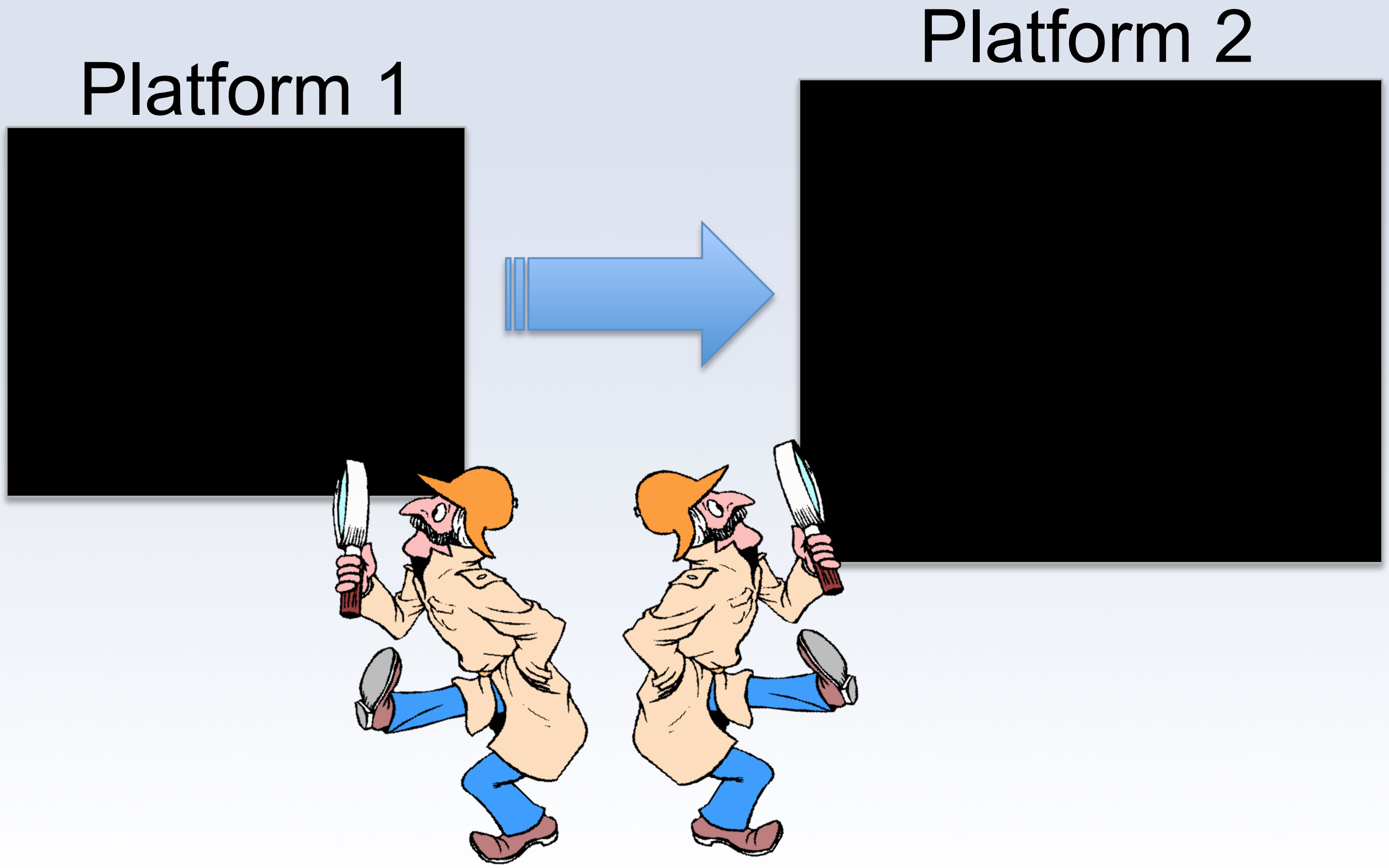
3. Examine per-file performance info

Each process writes in small pieces (< 256 KB) that do not align with file system block boundaries.

Application-specific analysis

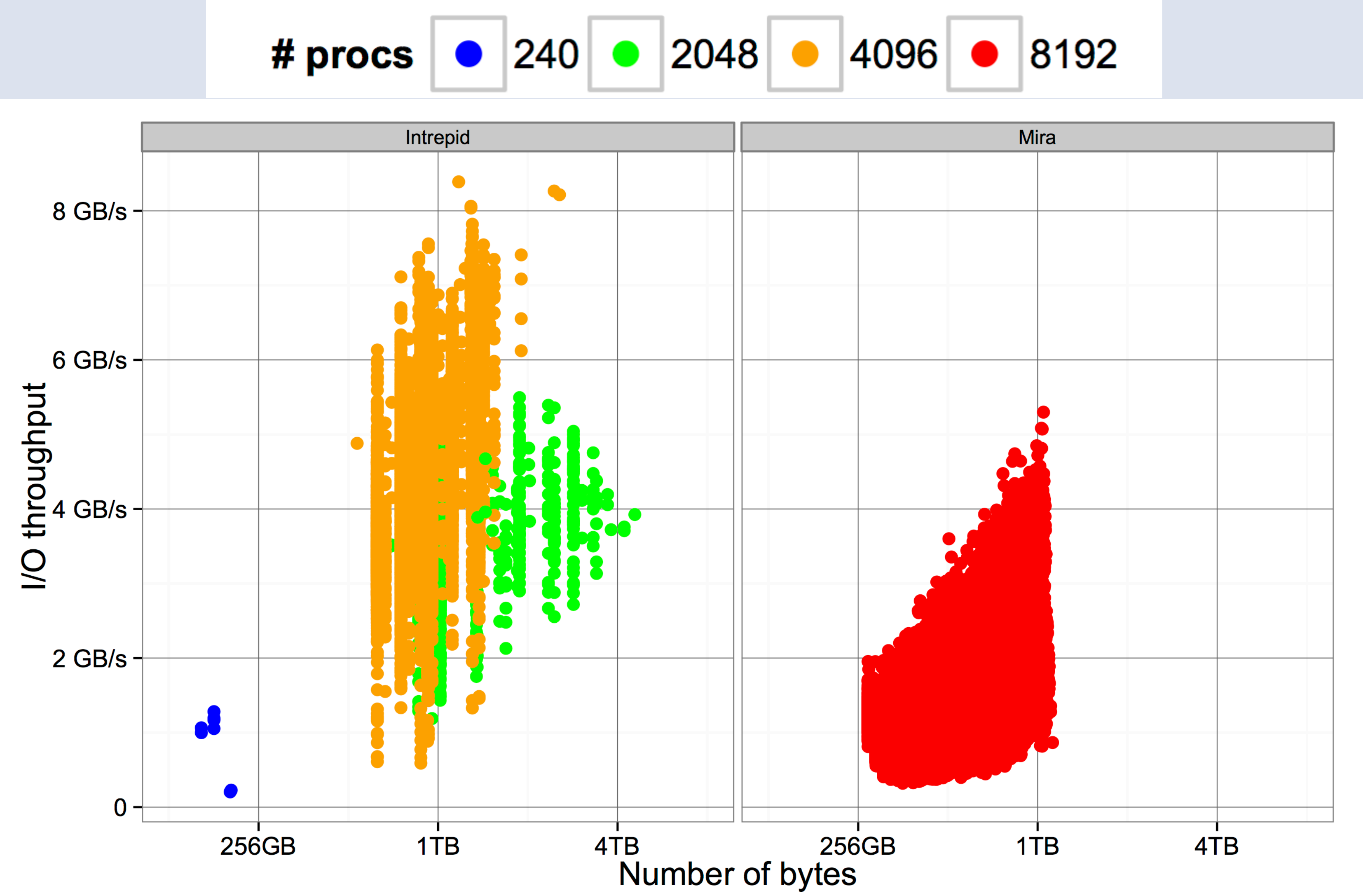
- Very simple and user-friendly.
- Quickly identify the I/O bottleneck/inefficiencies.
- User can follow up with a tracing/debugging tool.
- We are working with platform admins to make it available to all users.

CROSS-PLATFORM ANALYSIS

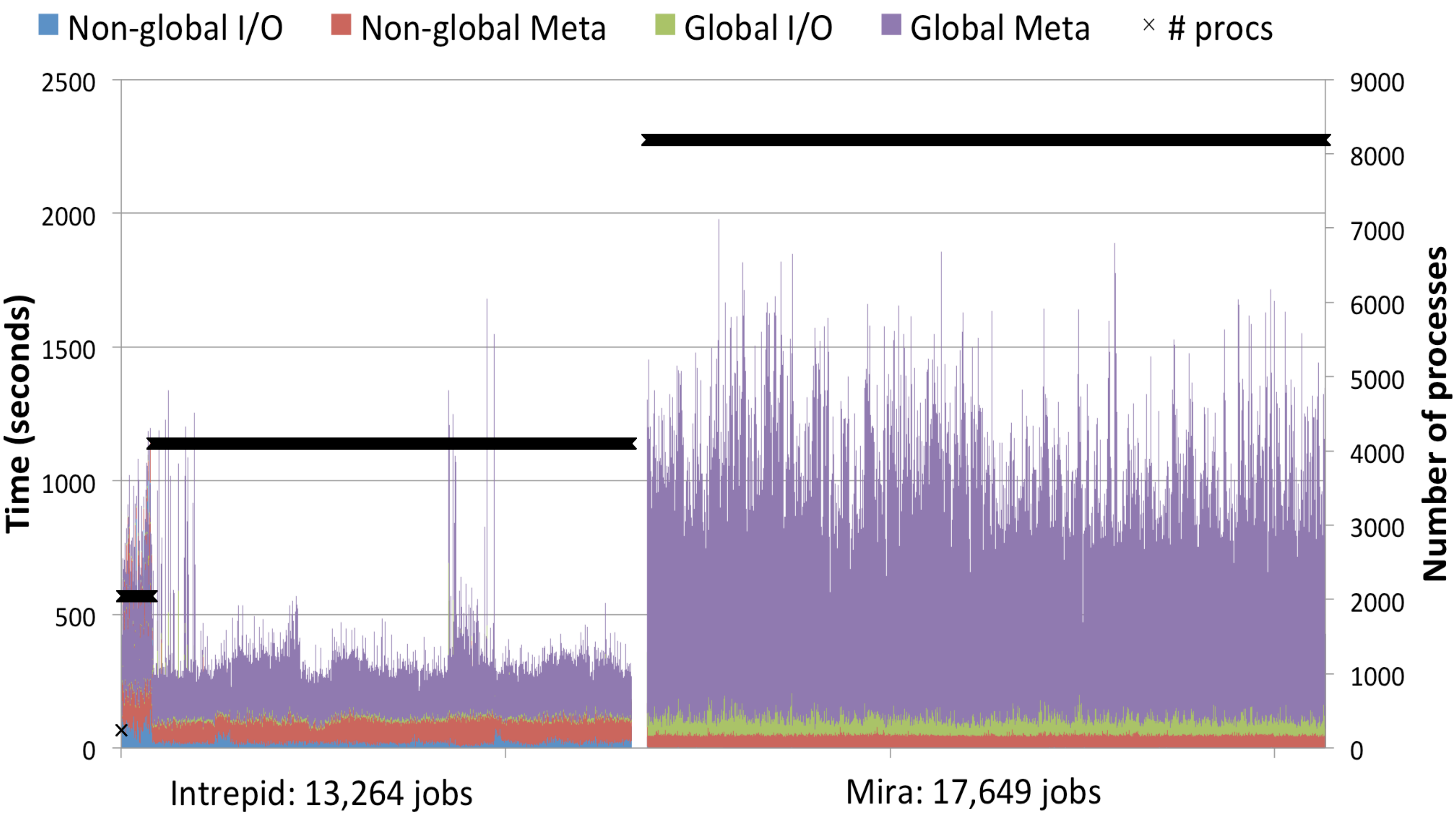


How does an app's scale (# procs, # bytes) and I/O thruput change? Why?

Earth1: Intrepid #4 → Mira #1.



Earth1's POSIX global shared files' metadata time didn't scale well.



Contributions

- Study I/O behavior of thousands of apps, >1M I/O logs, 6 years in combine, on 3 supercomputers.
- Application-specific, platform-wide, cross-platform analysis.
- Portrait of state of HPC I/O usage.
- Application I/O analysis + visualization procedure.
- Help improve system utilization.