Analysis of parallel I/O use on the UK national supercomputing service, ARCHER using

Cray's *LASSi* and EPCC *SAFE*

Andrew Turner, Dominic Sloan-Murphy (EPCC)
Karthee Sivalingam, Harvey Richardson (CERL)
Julian Kunkel (UoR)

Outline

- Introduction
- Methodology
- LASSi application analysis
- SAFE analysis of LASSi data
- Summary

Introduction

Motivation and some background

Motivation

For users: High-level **I/O metrics** on a per-job basis

- Better understanding of the different **I/O requirements** of different jobs
- Help identify any **slowdown** issues or performance bottlenecks
- More effectively **plan** their research workflow

For the service: High-level **I/O metrics** assessed across the service

- Overall view of **I/O usage** of the service
- Better understanding of **I/O requirements** of different user groups
- Assist I/O resource planning and setup
- Trend analysis and design of future services

I/O metrics

- High-level I/O statistics on a per job basis
- Routinely collected with no intervention from user
- No or little impact on job performance
- Ability to analyse slowdown issues
- Ability to examine particular jobs in more detail if required

Cray's LASSi tool meets these requirements

What is LASSi?

© Raksanand (CC BY-SA 4.0)

LASSi (**L**og **A**nalytics for **S**hared **S**ystem resource with **i**nstrumentation).

- Analyse slowdown of applications due to Lustre.
- Allows monitoring and profiling the I/O usage.
- How? Metric-based approach to study the I/O quantity and quality.
- Based on statistics available from LAPCAT – MySQL database with Lustre stats.

What can LASSi offer?

- A coarse I/O profile of each application running.
- Identification of abnormal
  - Filesystem I/O usage.
  - Application I/O usage.
- Identification of exact times when the filesystem is at risk of slowdown.
- Identification of exact applications causing the risk of slowdown.
- A prototype towards real-time analysis of risks and triggers.

Reporting and Analysis

- Provide **reporting interface** to users to inspect I/O metrics
- Link per-job I/O data to **metadata** (research area, application, etc.)
- Ability to perform **statistical analyses** across different periods and classifiers
- **Flexibility** to provide different analyses as requirements evolve

EPCC SAFE meets these requirements

What is SAFE?

SAFE (**S**ervice **A**dministration **F**or **E**PCC)

- Software framework designed to support:
  - User administration
  - Project administration
  - Query administration
  - Reporting and monitoring
- First version developed in 2002 by EPCC

Combining LASSi and SAFE

- SAFE designed to take many different data feeds: LASSi feed configured.
- Import historical LASSi data and setup regular feed from LASSi.
- Link LASSi data to other sources (ALPS, PBS, project/user management).
- Write reports to analyse overall use by different classifiers.

ARCHER filesystem

- Top500 #19 in 2013.
- Three Lustre filesystems: *fs2, fs3, fs4*
- Lustre I/O stats:
  - OSS: *read_kb, read_ops, write_kb, write_ops*
  - MDS: *open, close, mkdir, rmdir, sync…*

Methodology

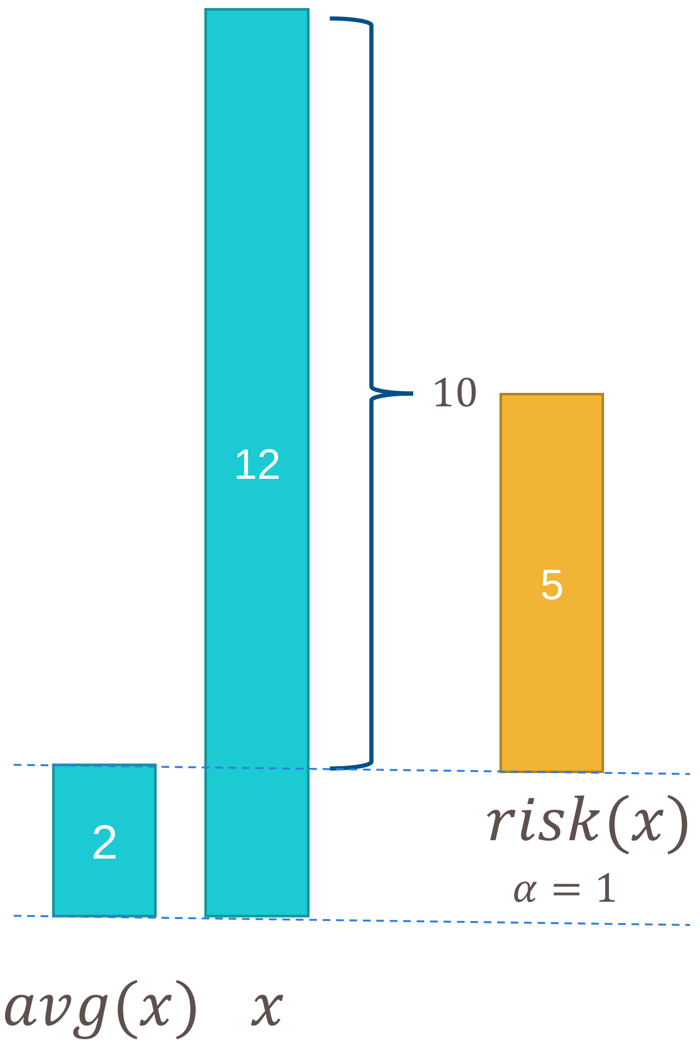
A different approach based on risks

- The simplest way to look at risks is perhaps:
  - In isolation, slowdown will happen only when an **application does more I/O than expected**
  - Also users will report slowdown only when they encounter **more I/O in a filesystem than expected**
- We will use this idea as a metric for risks

Risk metrics

$$risk_{fs}(x) = \frac{x - \alpha * avg_{fs}(x)}{\alpha * avg_{fs}(x)}$$

- x is any I/O operation OSS or MDS
- Risk is **calculated** for each application run
- We use averages for I/O operation for each filesystem
- We calculate risk as *scale of deviation from α times the avg on a filesystem*
- Higher value of risk denotes a higher risk of slowdown



Metrics for I/O

$$risk_{oss} = risk_{read\_kb} + risk_{read\_ops} + risk_{write\_kb} + risk_{write\_ops} + risk_{other}$$

$$\begin{aligned} risk_{mds} = {} & risk_{open} + risk_{close} + risk_{getattr} + risk_{setattr} + risk_{mkdir} \\ & + risk_{rmdir} + risk_{mknod} + risk_{link} + risk_{unlink} + risk_{ren} \\ & + risk_{getxattr} + risk_{setxattr} + risk_{statfs} + risk_{sync} + risk_{cdr} + risk_{sdr} \end{aligned}$$

I/O quality

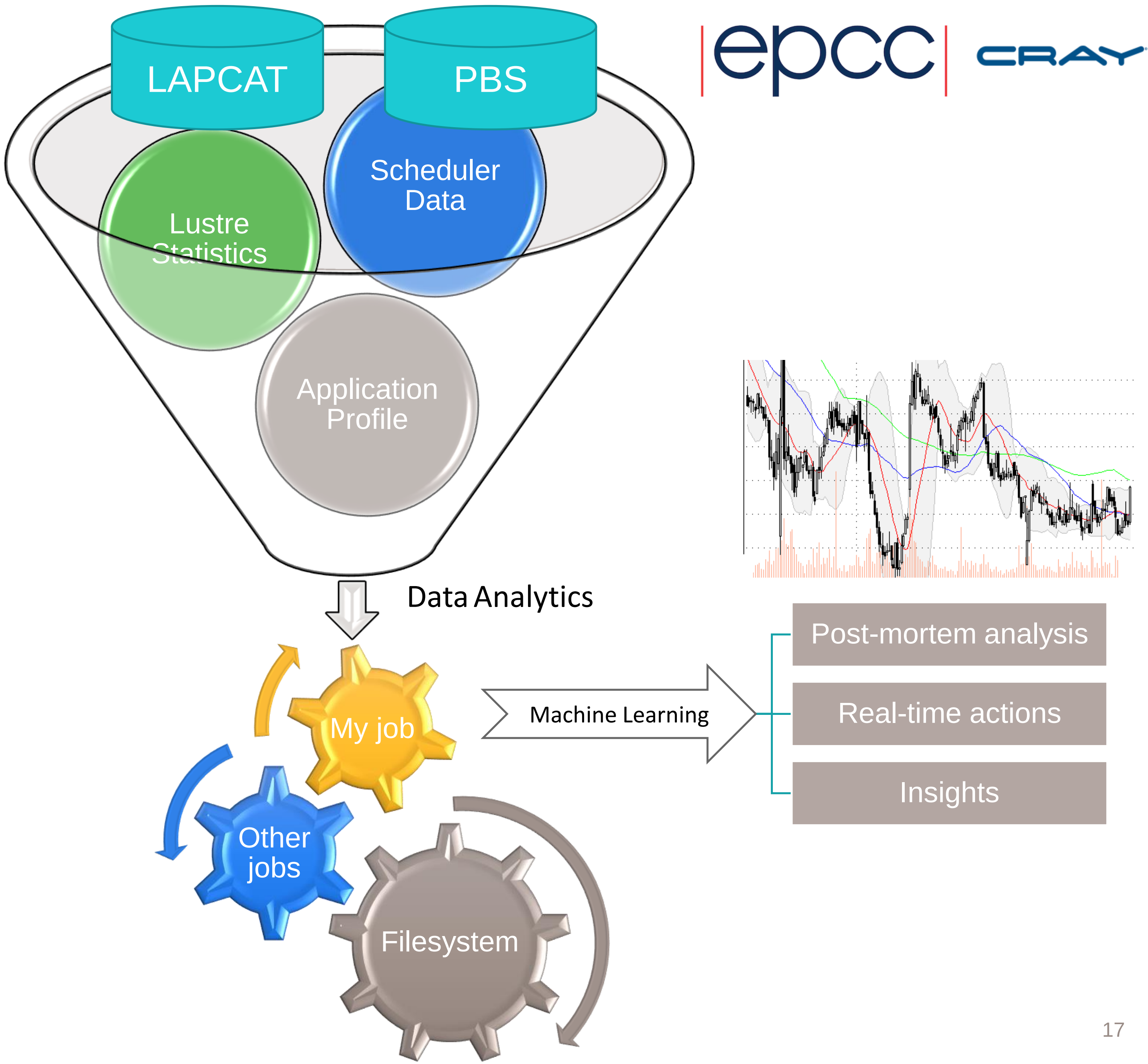
- ARCHER default stripe = 1MB
- 1MB aligned accesses are the optimal size per operation

$$read\_kb\_ops = \frac{read\_ops * 1024}{read\_kb}$$

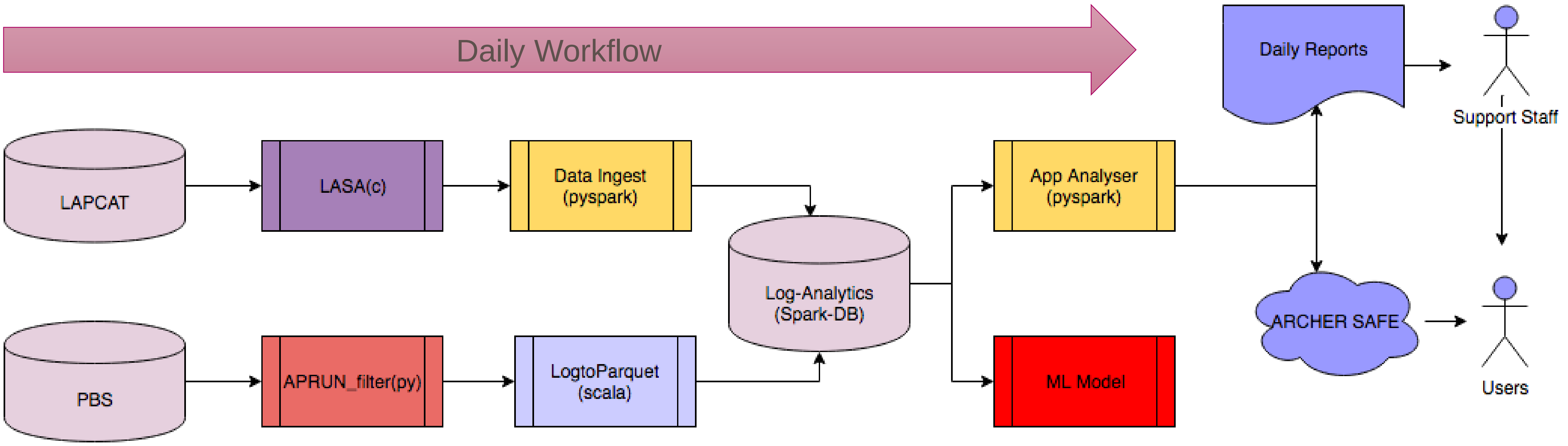
$$write\_kb\_ops = \frac{write\_ops * 1024}{write\_kb}$$

- Optimal quality when it is equal to 1
- Poor quality when it is greater to 1

ARCHER & LASSi

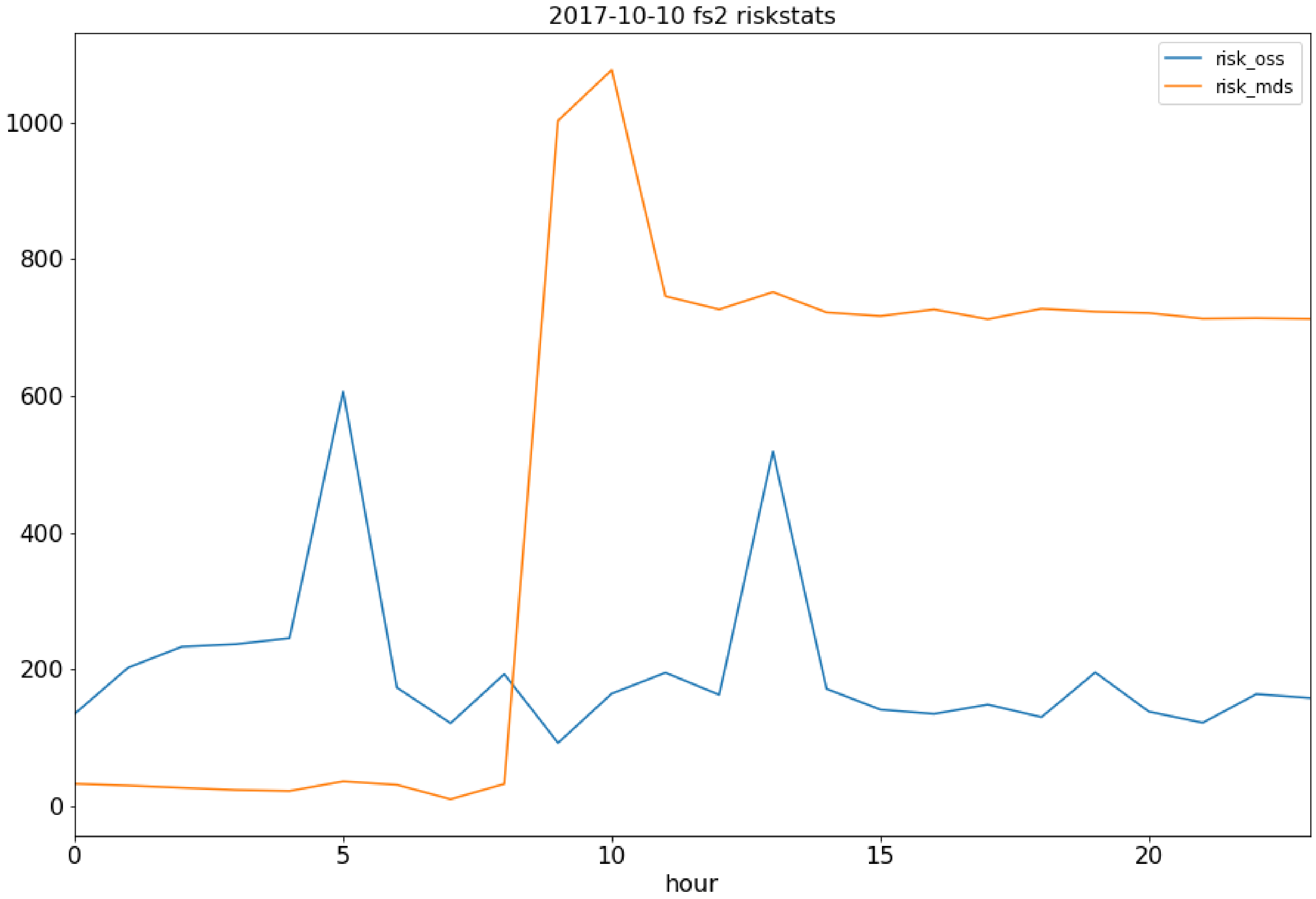


Architecture



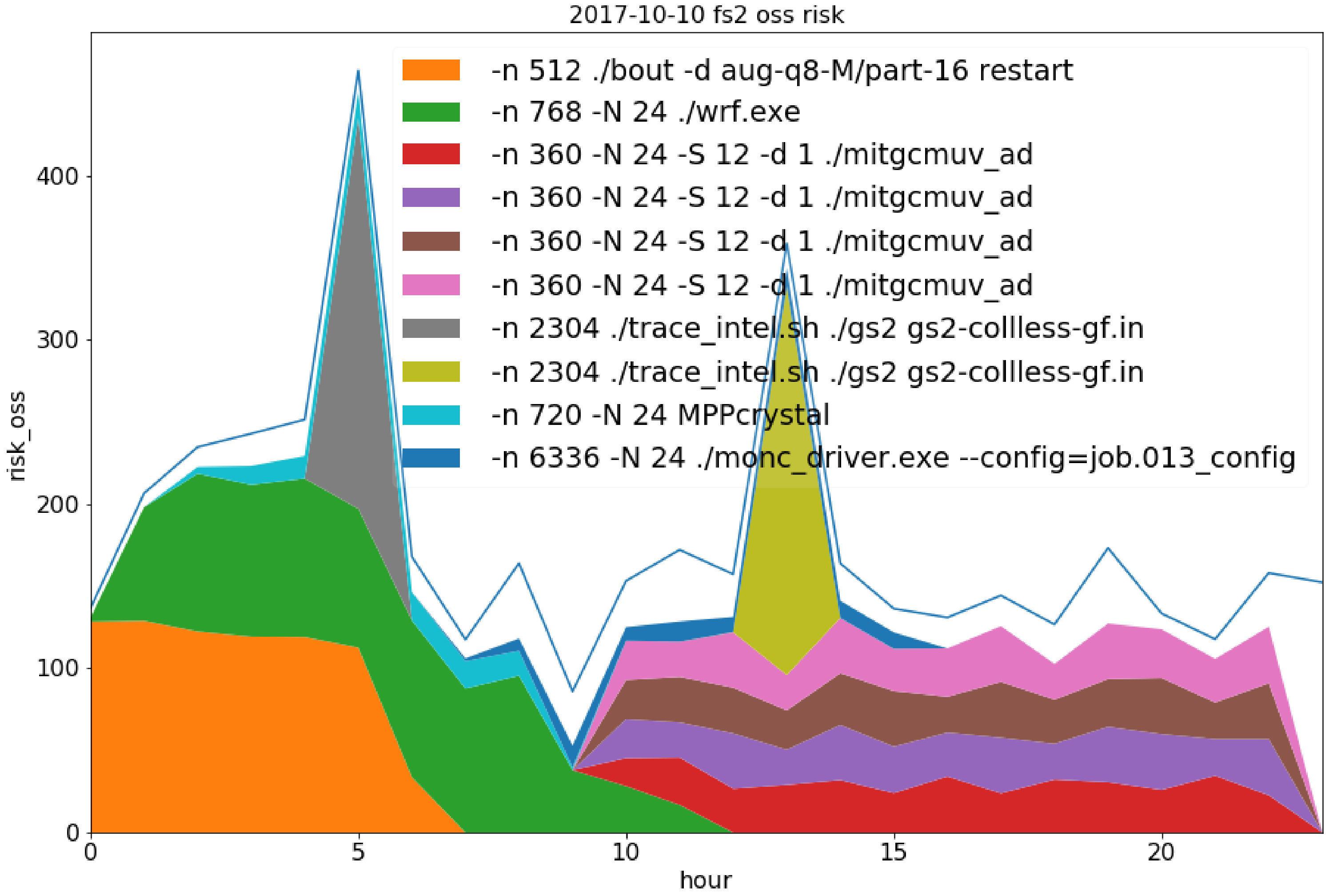
Examples of displays for helpdesk (risk)

- Risk of slowdown of fs2 on 10th October 2017
- OSS (blue line) means read/write operations
- MDS (orange line) means metadata operations



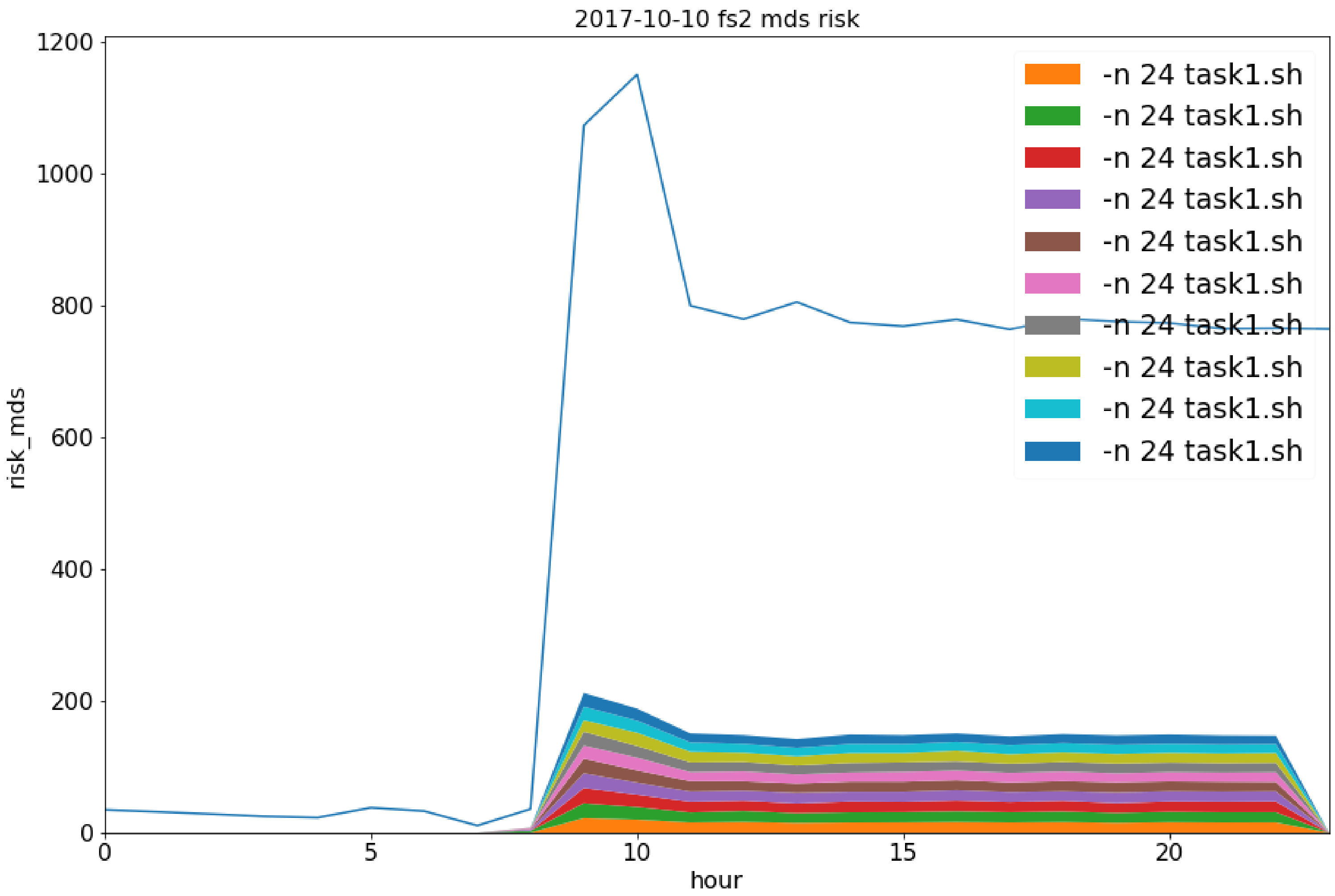
Example of displays for helpdesk – daily risk to OSS

- Risk of slowdown of fs2 on 10th October 2017
- Top 10 applications that contribute to the OSS risk

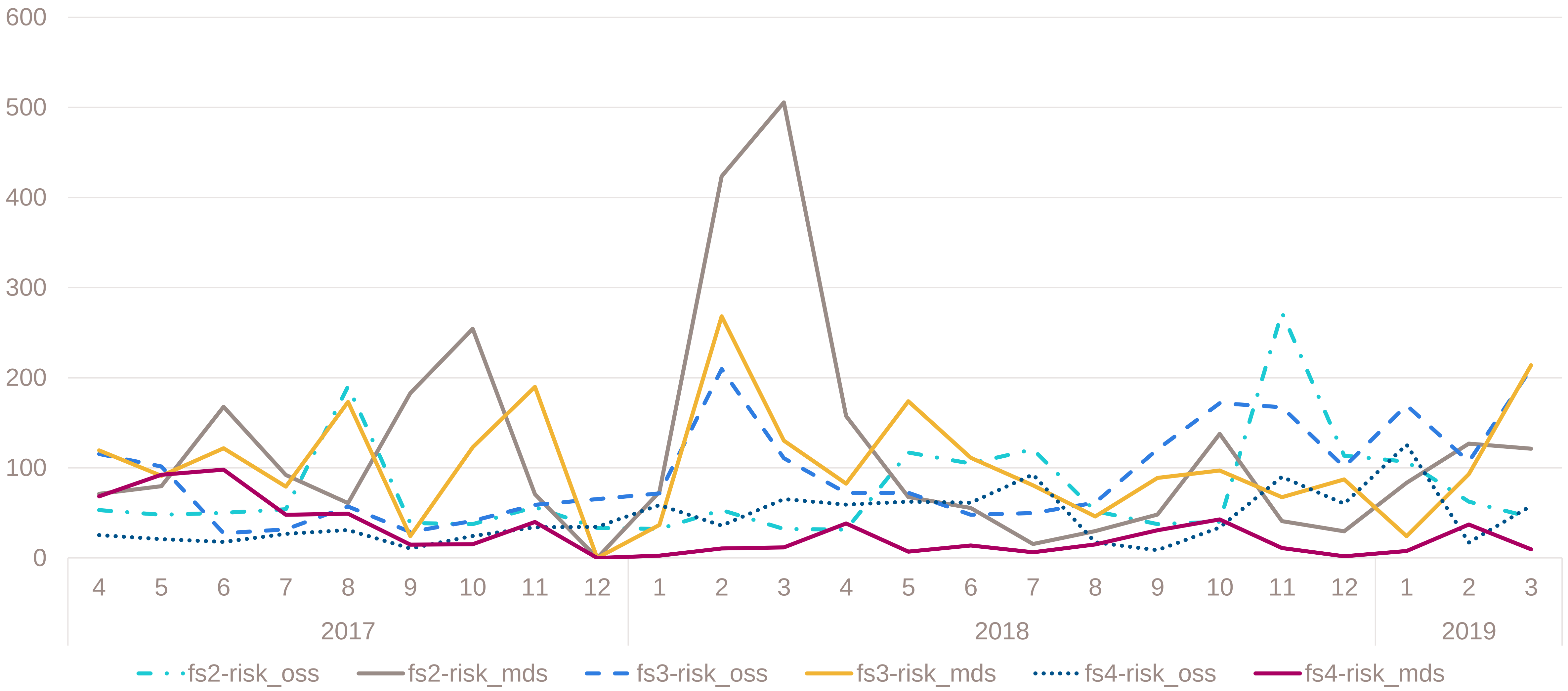


Example of displays for helpdesk - daily risk to mds

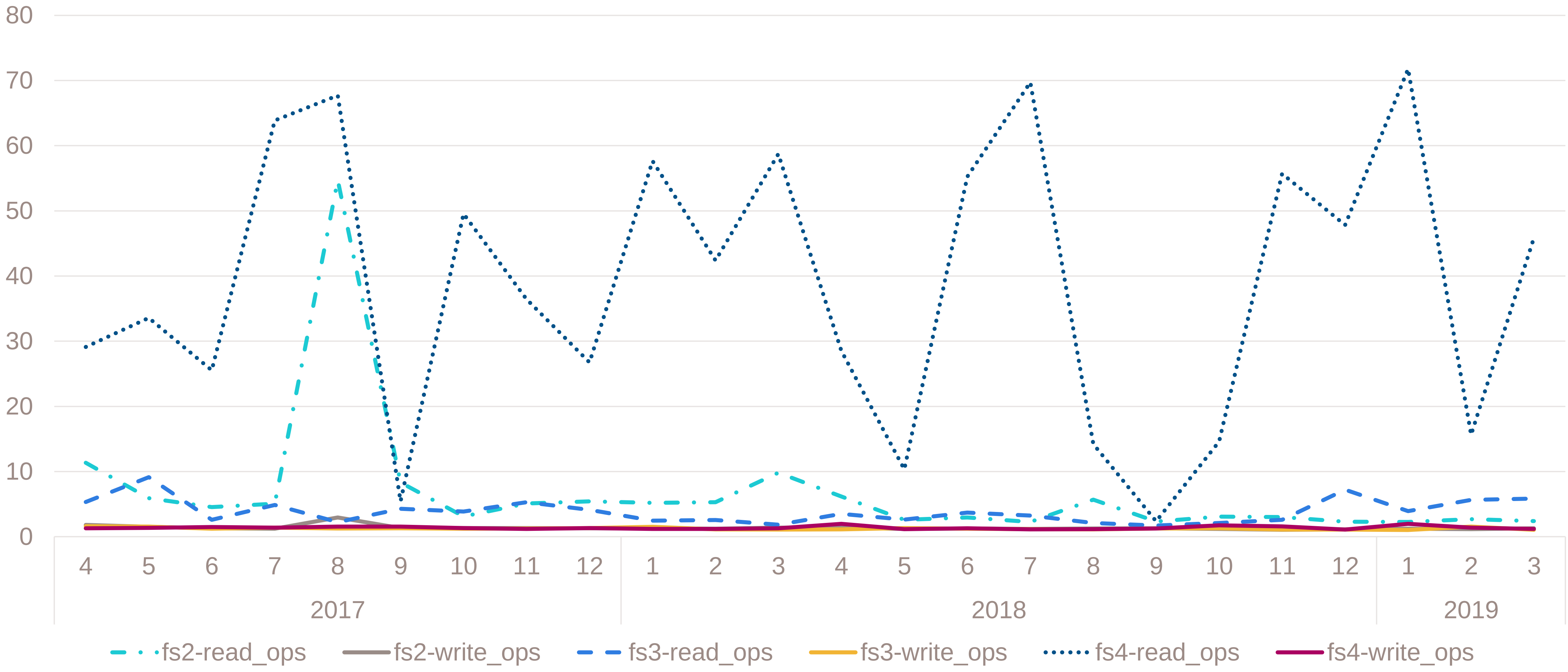
- Risk of slowdown of fs2 on 10th October 2017
- Top 10 applications that contribute to the MDS risk



Risk to filesystem over 24 months



Quality of I/O per fs over 24 months



LASSi application analysis

Applications I/O on ARCHER from April 2017 to March 2019 (two years)

Scatter plot of application groups

Three clusters:

- More read/write and less metadata

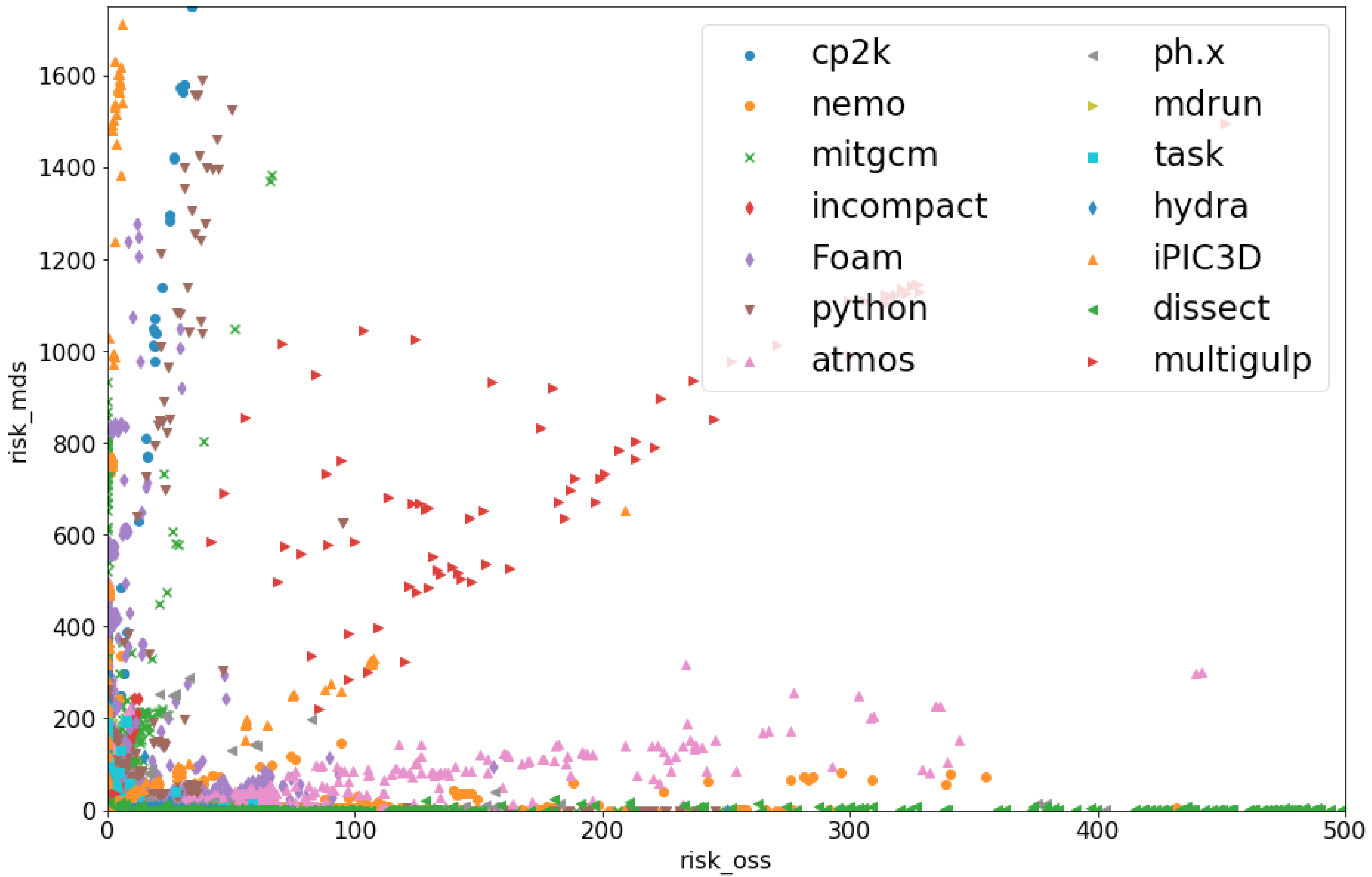
  Dissect, Atmos, Nemo

- More metadata and less read/write

  iPIC3D, Foam, cp2k, Python, mitgcm

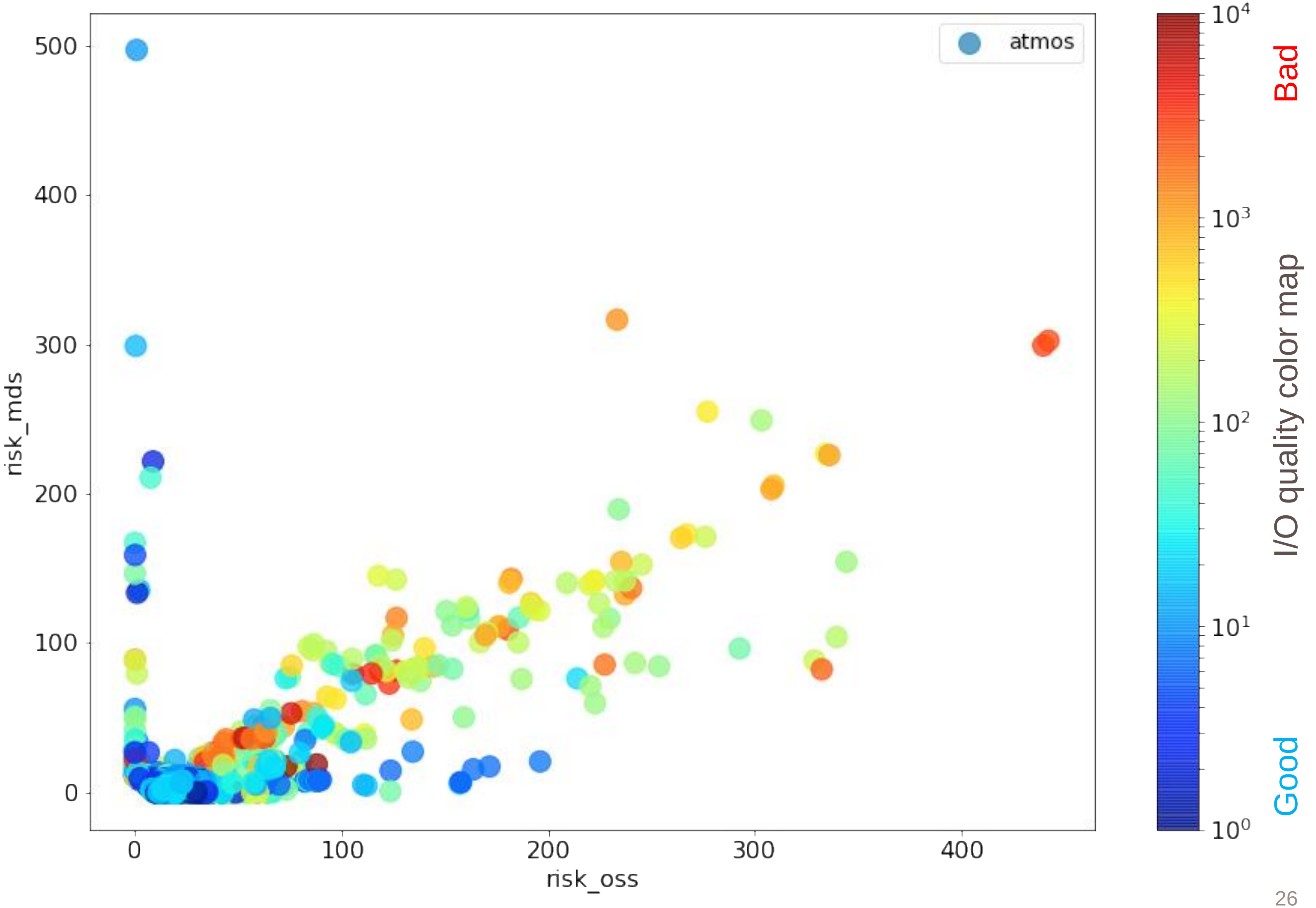
- Both metadata and read/write

  multigulp



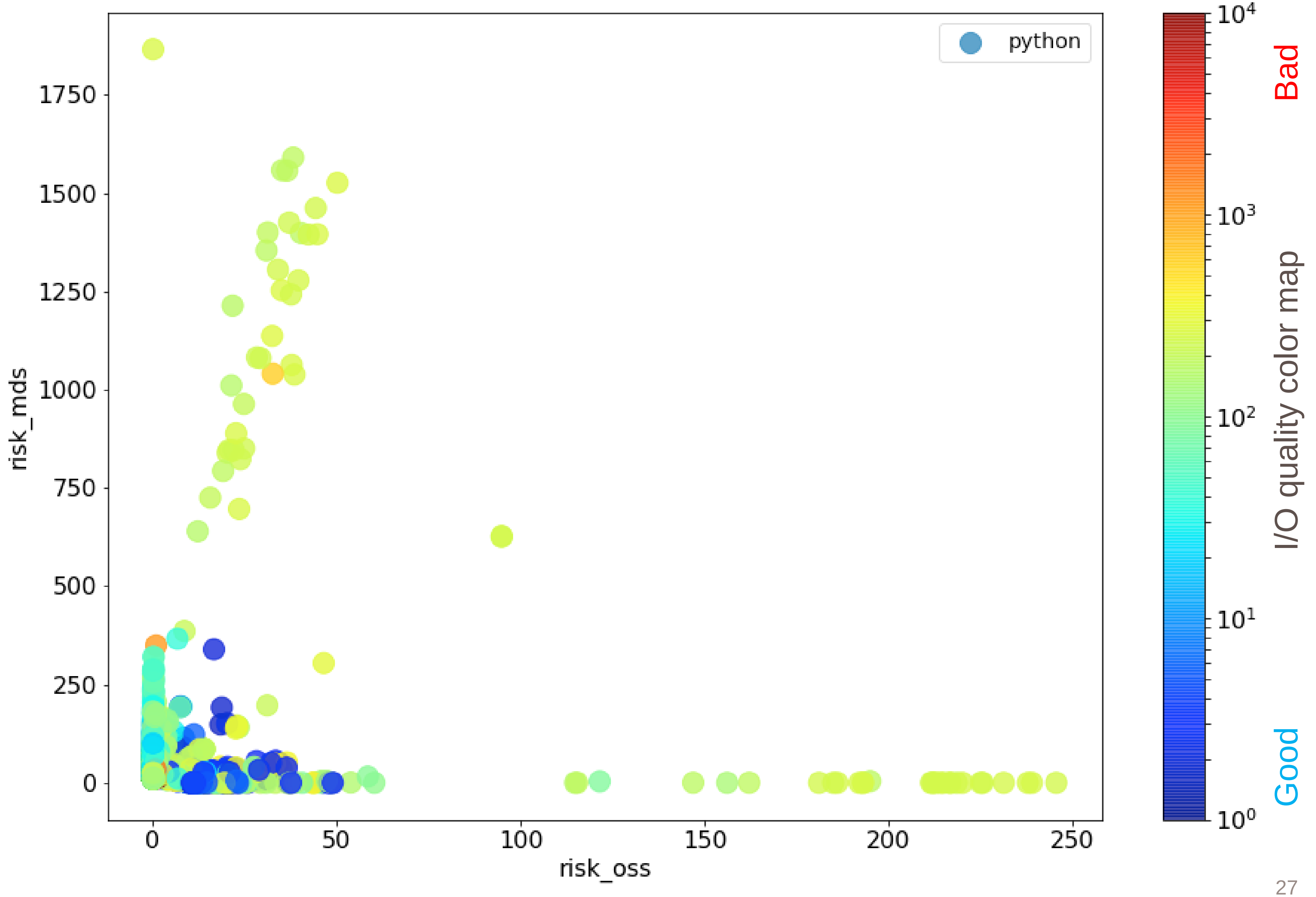
Risk/quality profile of Climate/NWP applications

- Blue denotes good-quality I/O.
- Red denotes bad-quality I/O.
- Clusters near the axis show good-quality I/O.
- Cluster away from the axis shows bad-quality I/O.



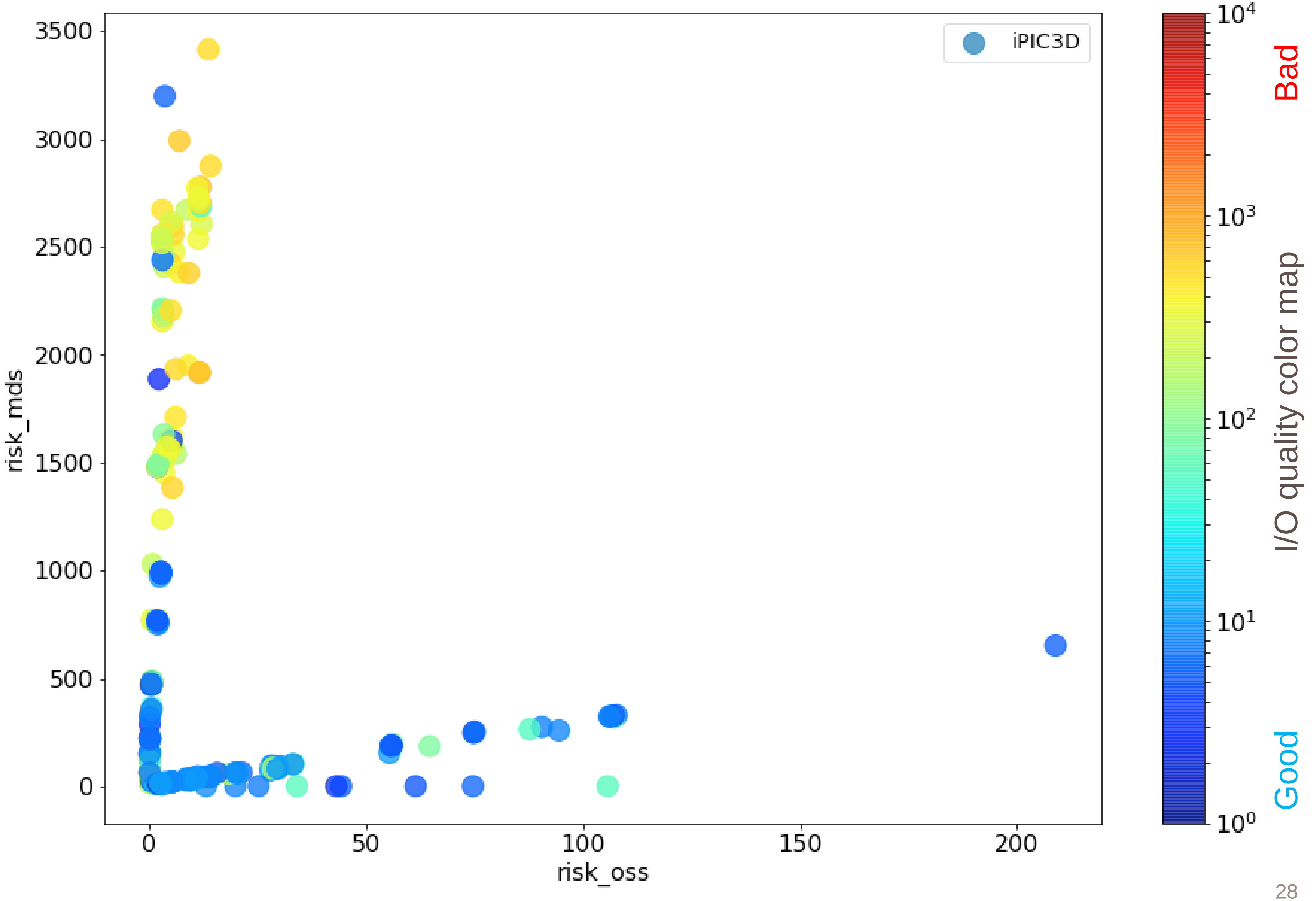
Risk/quality profile of Python applications

- Poor quality of I/O in general.
- Some applications with low risk perform good-quality I/O.



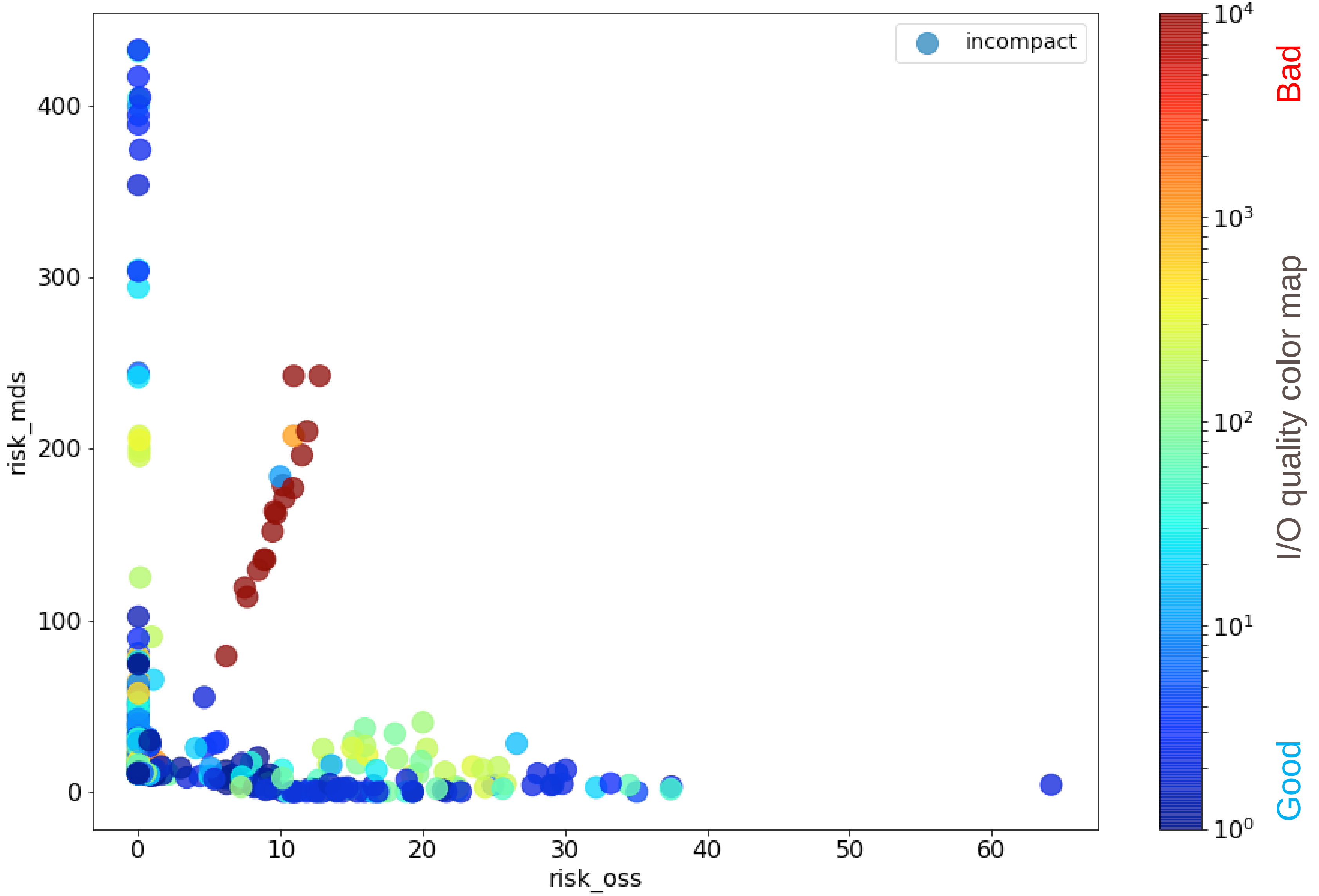
Risk/quality profile of *iPIC3D*

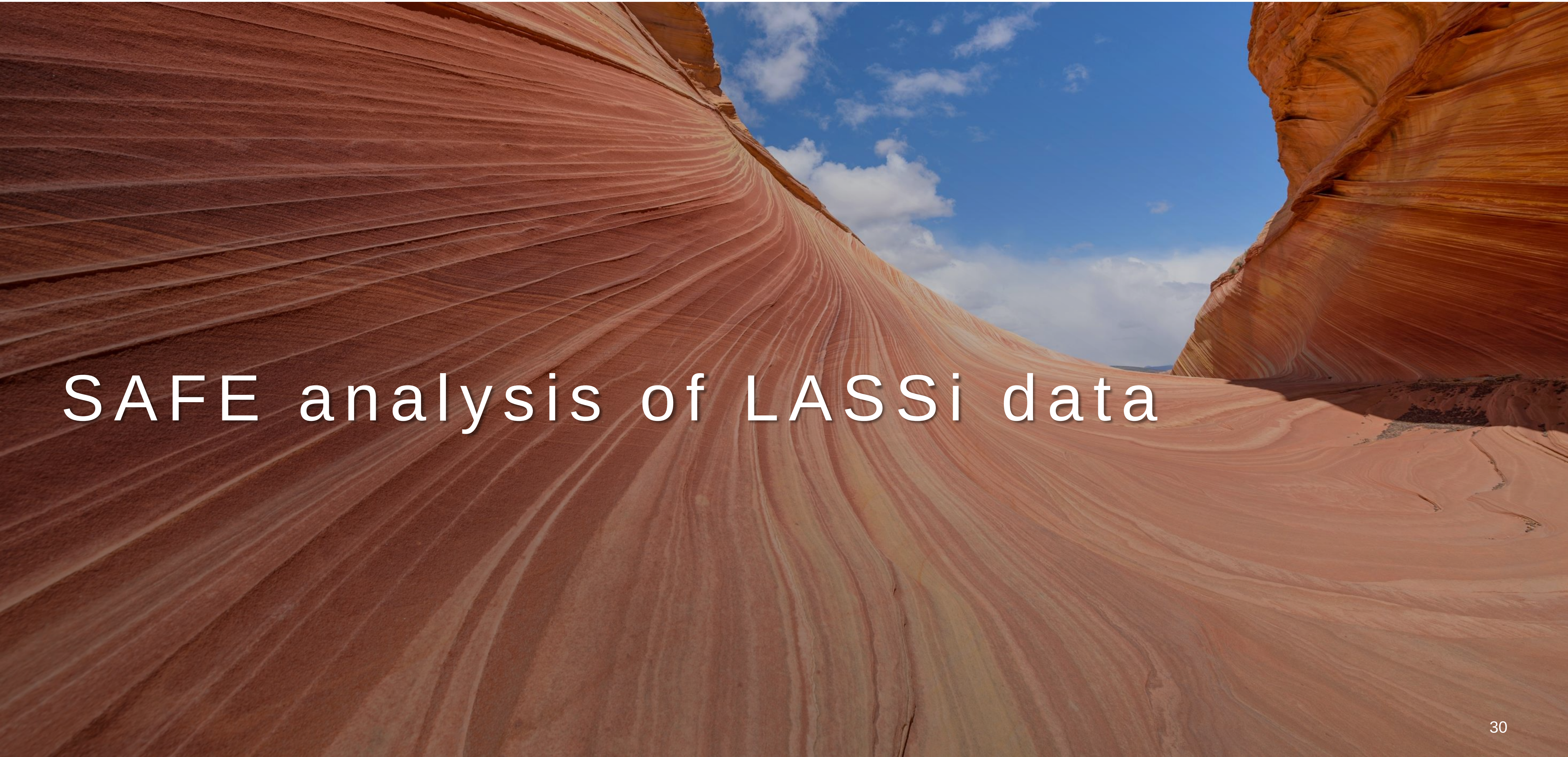
- High metadata usage and bad-quality I/O.
- A cluster of iPIC3D applications perform good-quality I/O.



Risk/quality profile of *incompact*

- Incompact3D is a powerful high-order flow solver.
- Most applications show good-quality I/O.
- A cluster of applications runs away from the axis shows bad-quality I/O.



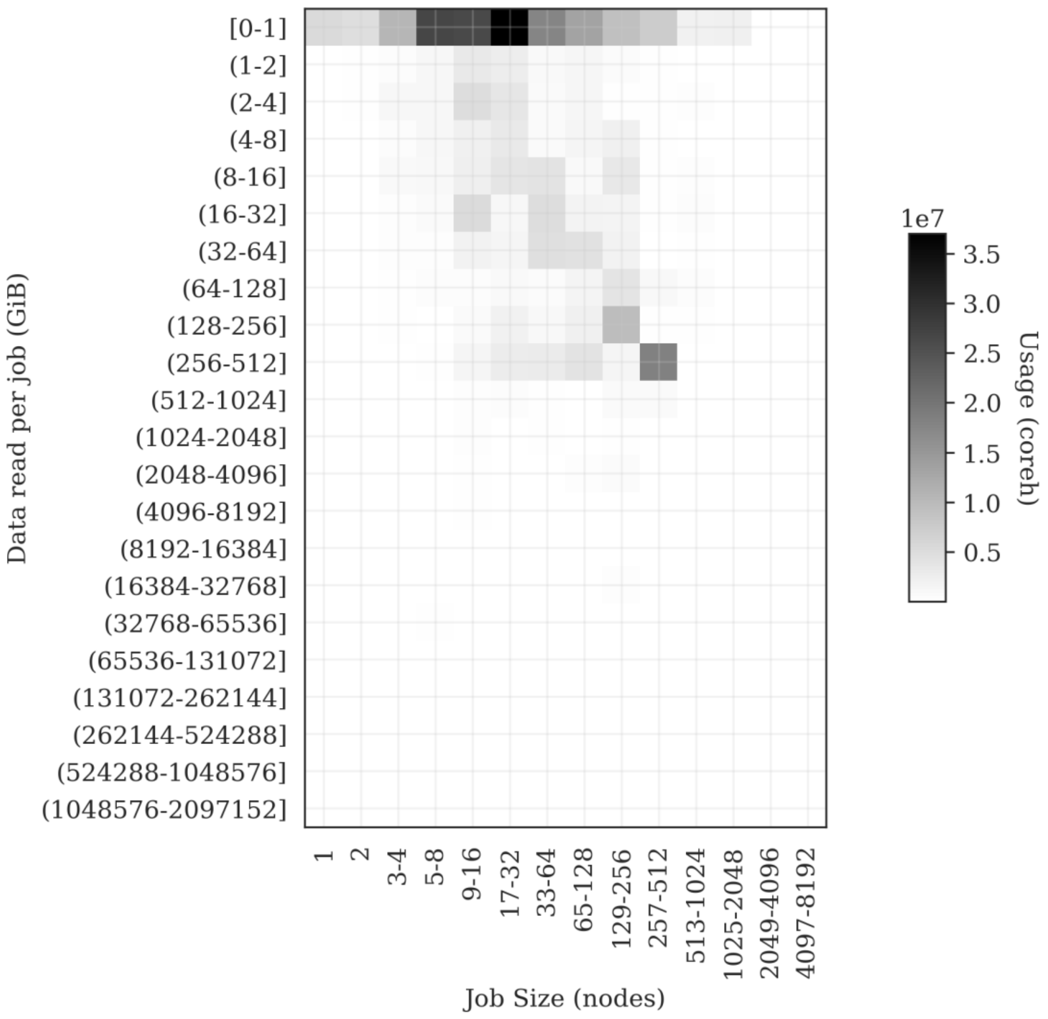


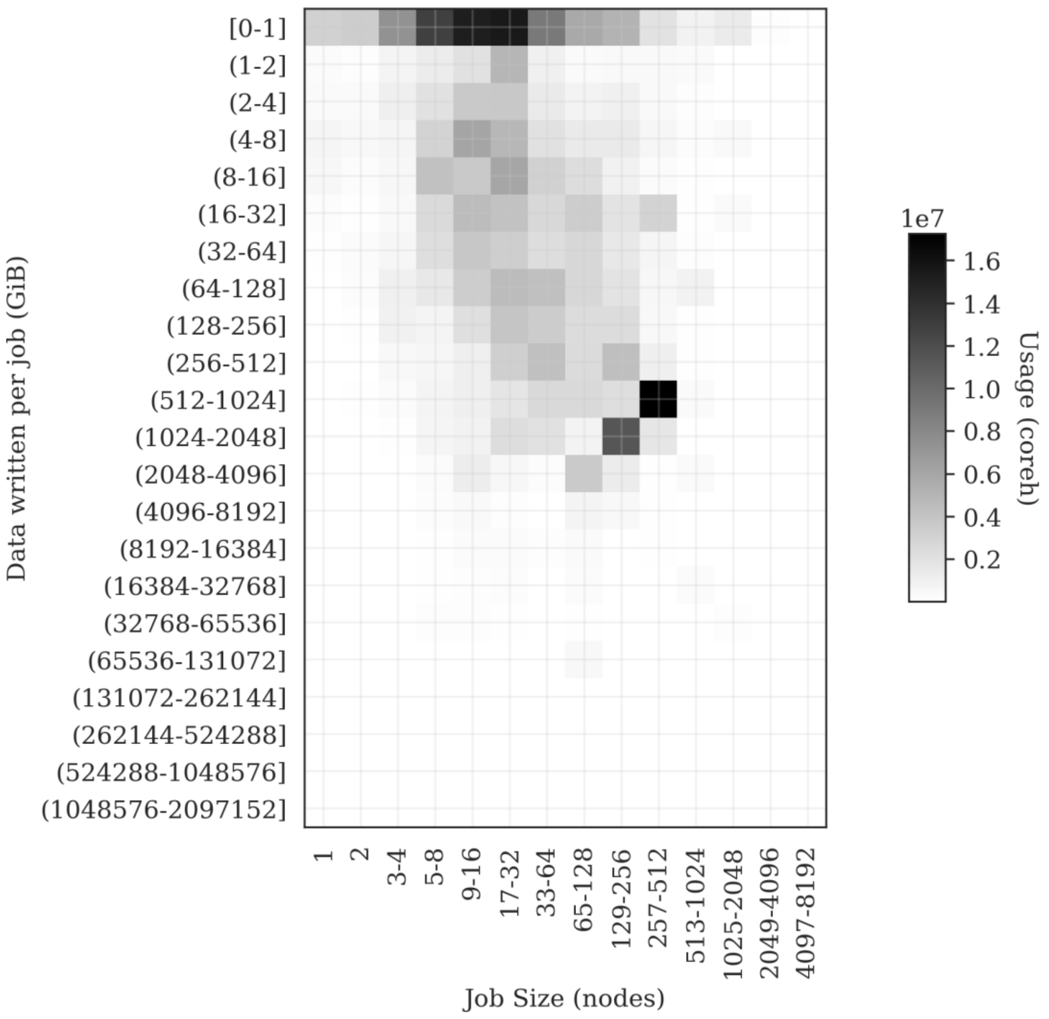
SAFE analysis of LASSi data

Notes

- Analysis period: July - December 2018; Initial analysis – lots still to do!
- All jobs that ran for more than 5 minutes included
- LASSi samples I/O from all jobs once every 3 minutes
- Only covers accesses to Lustre file system
- Only data amounts/rates reported – analysis of I/O ops to follow
- Research areas identified by project membership

All jobs


Data read per job (GiB)
[0-1]
(1-2]
(2-4]
(4-8]
(8-16]
(16-32]
(32-64]
(64-128]
(128-256]
(256-512]
(512-1024]
(1024-2048]
(2048-4096]
(4096-8192]
(8192-16384]
(16384-32768]
(32768-65536]
(65536-131072]
(131072-262144]
(262144-524288]
(524288-1048576]
(1048576-2097152]
1
2
3-4
5-8
9-16
17-32
33-64
65-128
129-256
257-512
513-1024
1025-2048
2049-4096
4097-8192
Job Size (nodes)
1e7
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Usage (coreh)

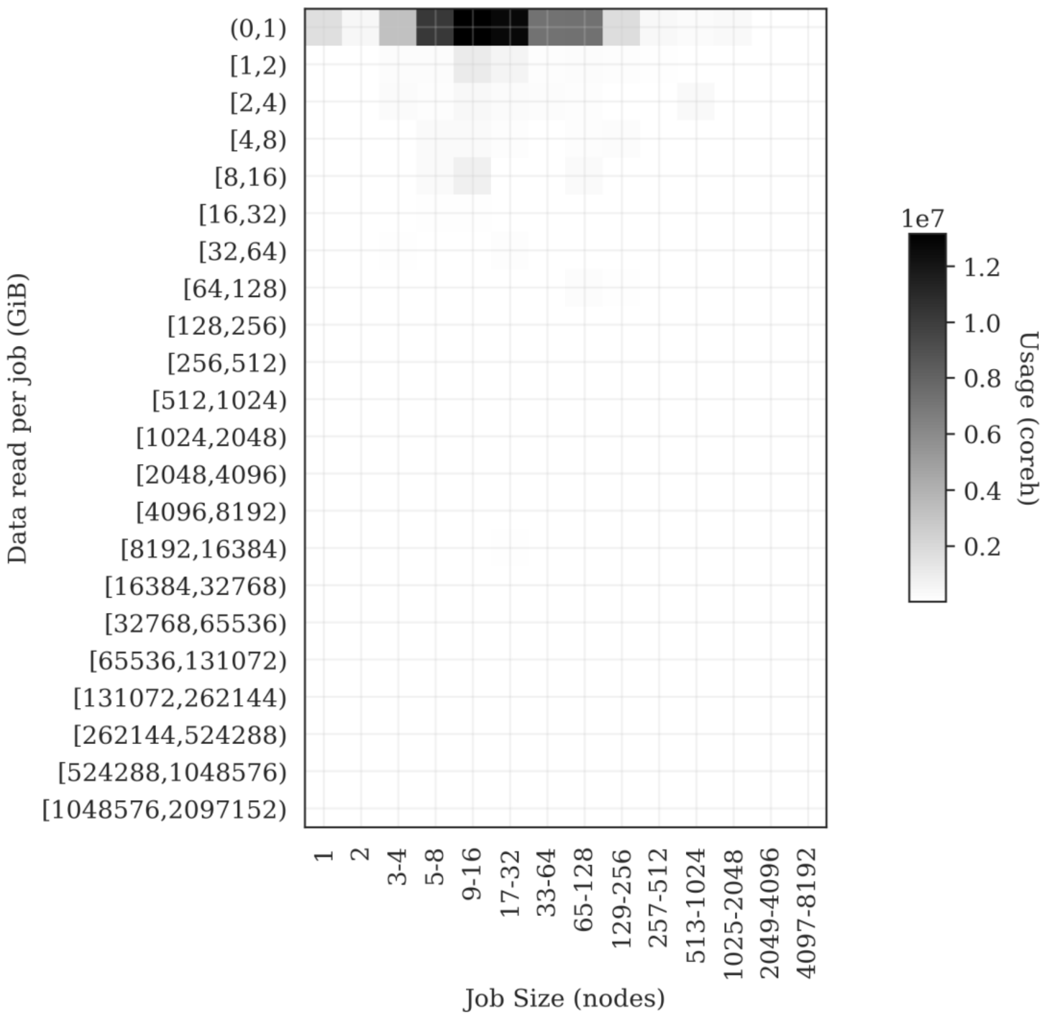

Data written per job (GiB)
[0-1]
(1-2]
(2-4]
(4-8]
(8-16]
(16-32]
(32-64]
(64-128]
(128-256]
(256-512]
(512-1024]
(1024-2048]
(2048-4096]
(4096-8192]
(8192-16384]
(16384-32768]
(32768-65536]
(65536-131072]
(131072-262144]
(262144-524288]
(524288-1048576]
(1048576-2097152]
1
2
3-4
5-8
9-16
17-32
33-64
65-128
129-256
257-512
513-1024
1025-2048
2049-4096
4097-8192
Job Size (nodes)
1e7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
Usage (coreh)

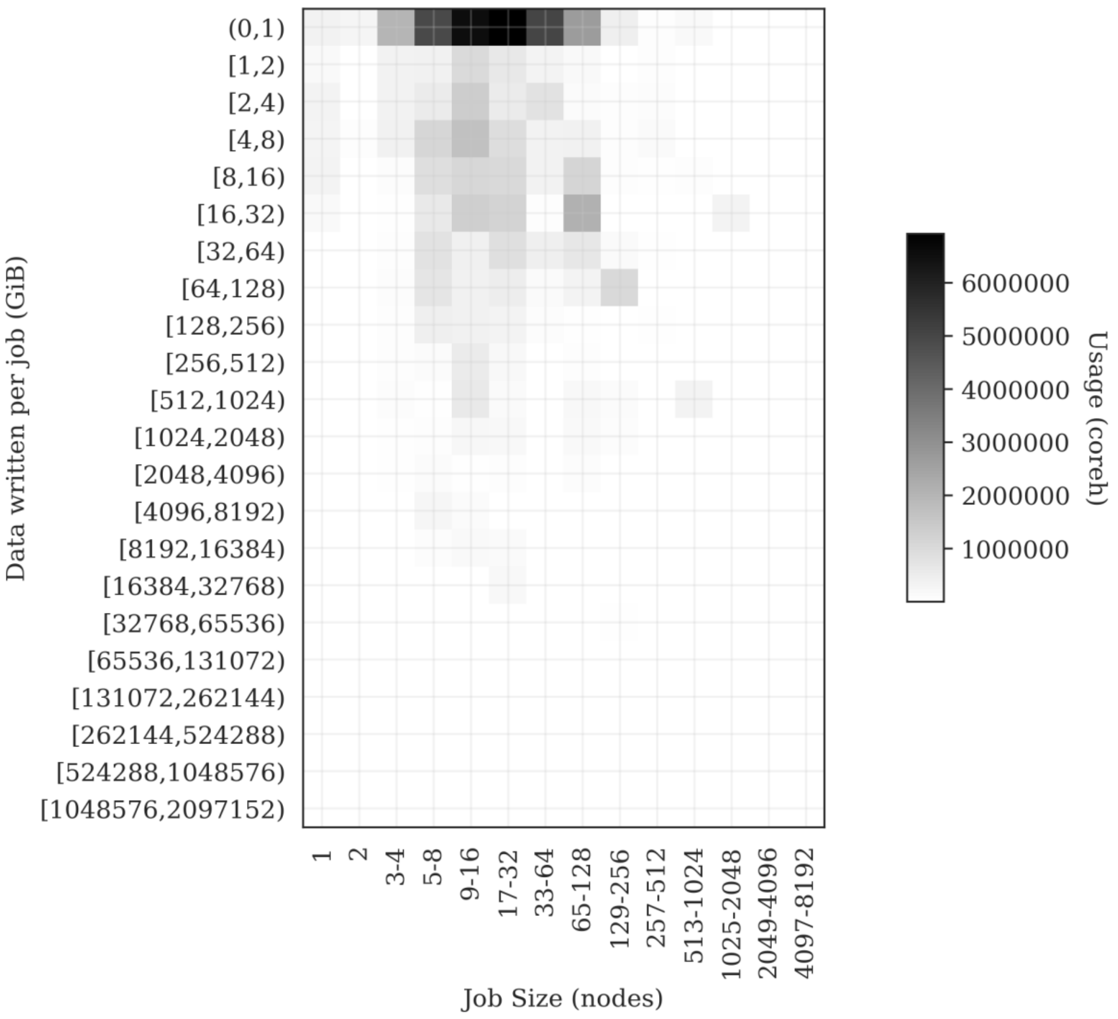
All jobs

Data per job (GiB)	% Usage (Read)	%Usage (Write)
(0, 4)	59.8%	34.8%
[4, 32)	14.7%	21.5%
[32, 256)	13.4%	17.8%
[256, 2048)	11.1%	21.4%
[2048,)	1.0%	4.5%

- Large amount of use associated with small amount of data
- Jobs that use larger amounts of I/O generally write twice as much data as they read
- No very strong link between job size and amount of I/O activity
- Much more data read and written than the size of file system would allow – data is transient
- This is an overlay of different I/O use modes

Materials science



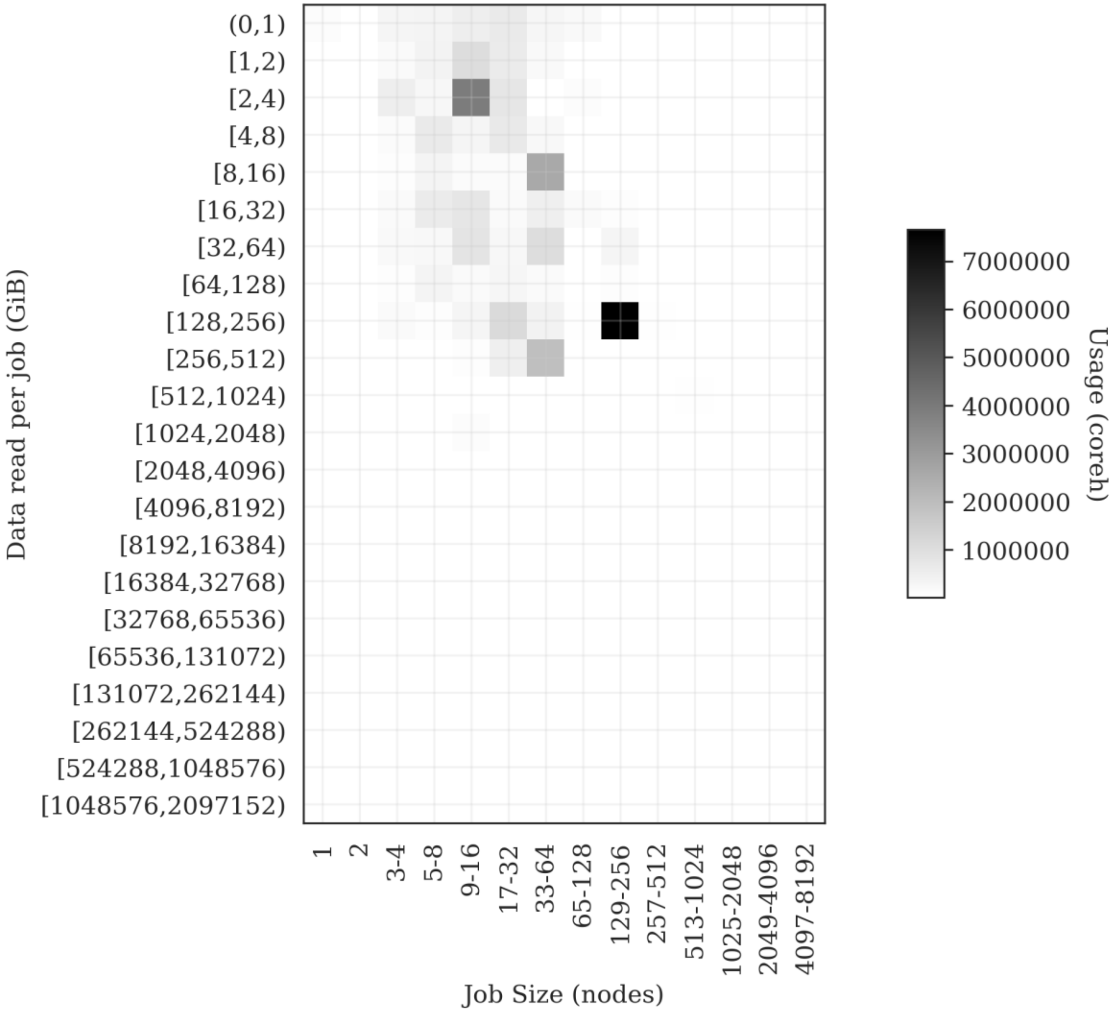


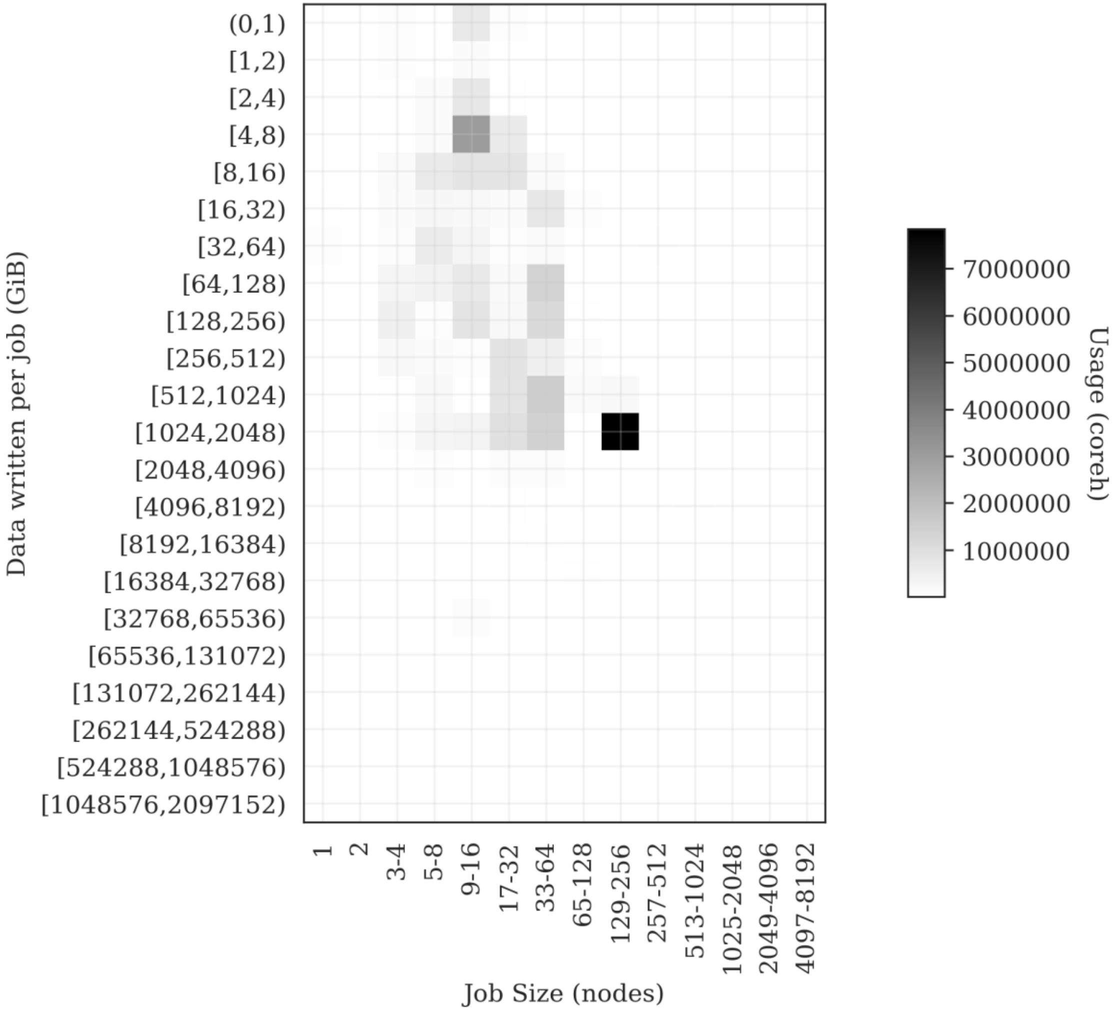
Material science

Data per job (GiB)	% Usage (Read)	%Usage (Write)
(0, 4)	**94.3%**	**55.4%**
[4, 32)	4.2%	**25.0%**
[32, 256)	1.1%	12.3%
[256, 2048)	0.4%	5.1%
[2048,)	0.2%	2.2%

- Use dominated by periodic electronic structure codes, such as VASP, CASTEP, CP2K, Quantum Espresso…
- This I/O pattern can be understood as:
  - Small input data: often just a description of the initial atomic coordinates, basis set specification and a small number of calculation parameters.
  - Small output data: including properties of the modelled system such as energy, final atomic coordinates and descriptions of the wave function.
- Significant usage (37.6%) for jobs that write larger amounts of data ([4, 256) GiB).

Climate modelling



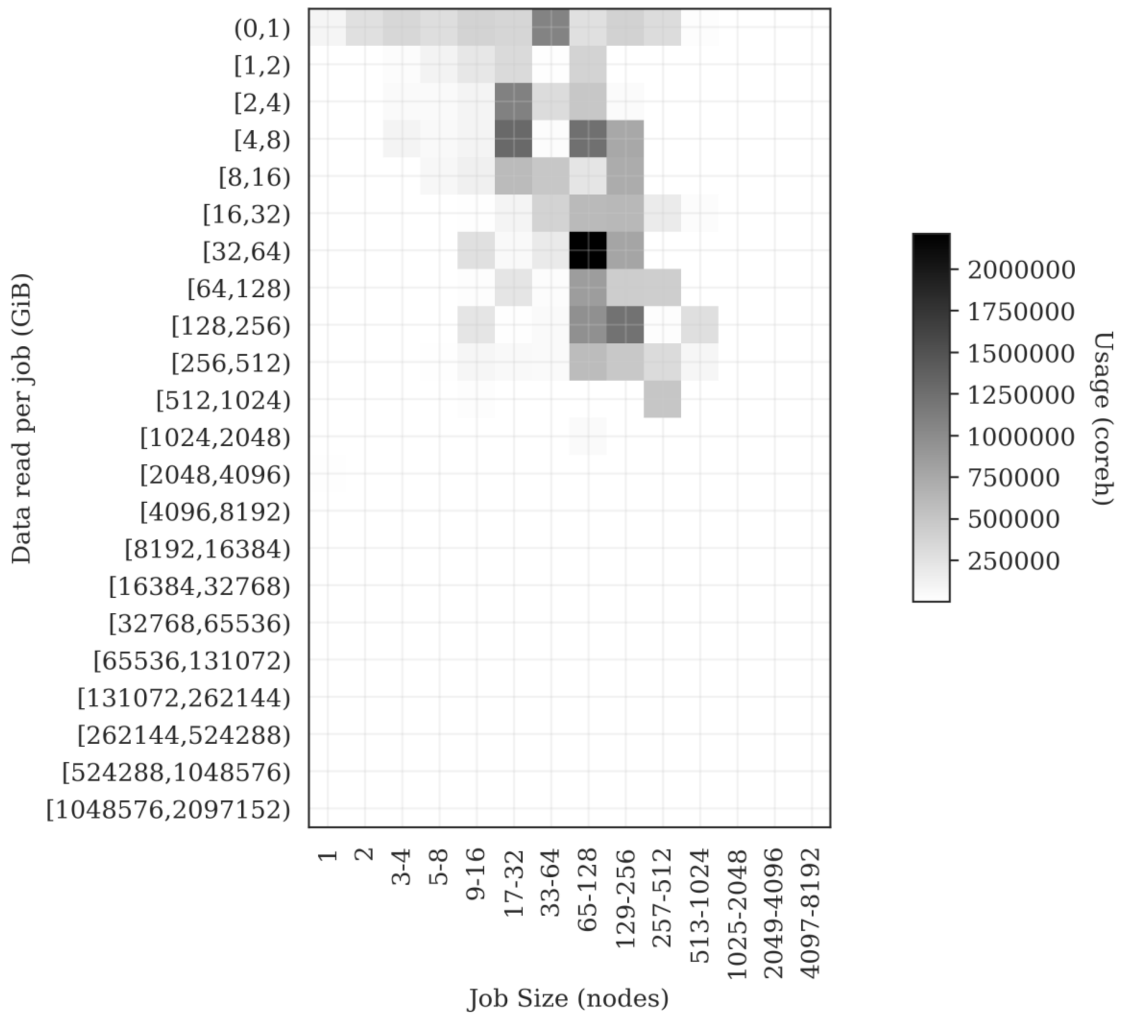


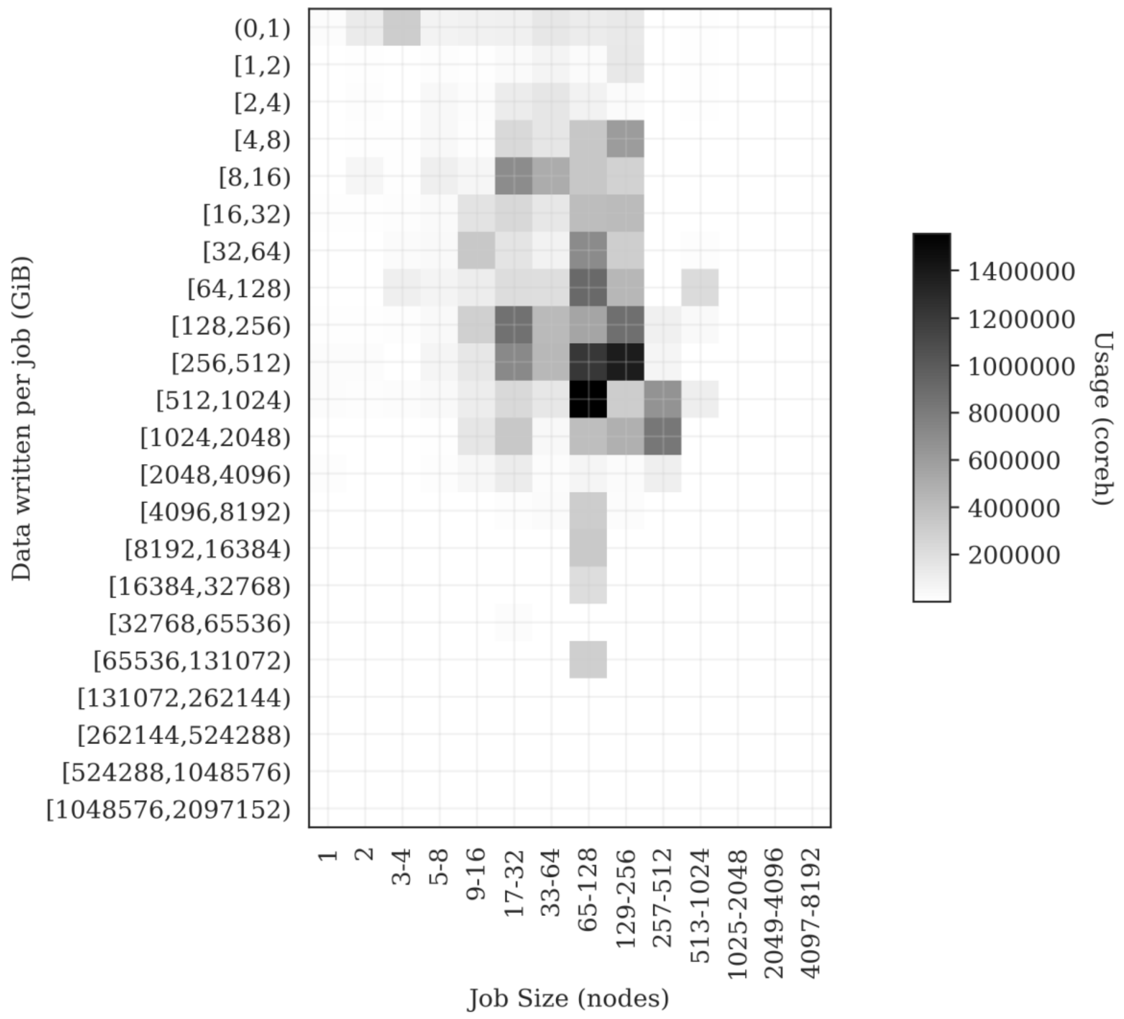
Climate modelling

Data per job (GiB)	% Usage (Read)	%Usage (Write)
(0, 4)	**30.0%**	6.3%
[4, 32)	**22.4%**	**24.0%**
[32, 256)	**39.8%**	**21.1%**
[256, 2048)	7.8%	**46.4%**
[2048,)	0.0%	2.2%

- Applications such as Met Office UM, WRF, NEMO, MITgcm.
- The climate modelling community typically read and write large amounts of data.
- Small range of scales (in terms of length and timescale).
- This pattern can be understood as:
    - Most jobs read in large amounts of observational data and model description data.
    - Most jobs write out time-series trajectories of the model configuration and computed properties for several snapshots throughout the model run. These trajectories are archived and used for further analysis.

CFD


Data read per job (GiB)
(0,1)
[1,2)
[2,4)
[4,8)
[8,16)
[16,32)
[32,64)
[64,128)
[128,256)
[256,512)
[512,1024)
[1024,2048)
[2048,4096)
[4096,8192)
[8192,16384)
[16384,32768)
[32768,65536)
[65536,131072)
[131072,262144)
[262144,524288)
[524288,1048576)
[1048576,2097152)
1
2
3-4
5-8
9-16
17-32
33-64
65-128
129-256
257-512
513-1024
1025-2048
2049-4096
4097-8192
Job Size (nodes)
2000000
1750000
1500000
1250000
1000000
750000
500000
250000
Usage (coreh)

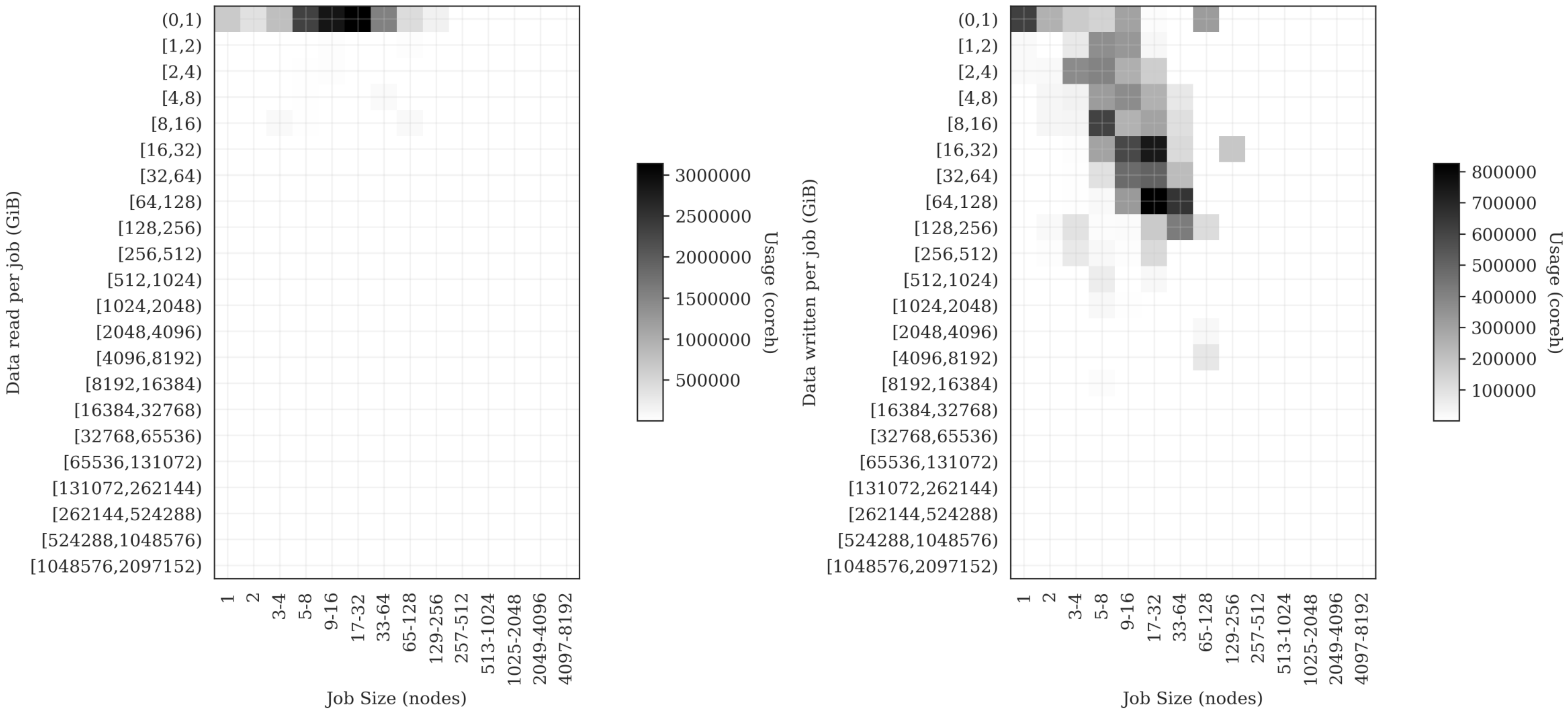

Data written per job (GiB)
(0,1)
[1,2)
[2,4)
[4,8)
[8,16)
[16,32)
[32,64)
[64,128)
[128,256)
[256,512)
[512,1024)
[1024,2048)
[2048,4096)
[4096,8192)
[8192,16384)
[16384,32768)
[32768,65536)
[65536,131072)
[131072,262144)
[262144,524288)
[524288,1048576)
[1048576,2097152)
1
2
3-4
5-8
9-16
17-32
33-64
65-128
129-256
257-512
513-1024
1025-2048
2049-4096
4097-8192
Job Size (nodes)
1400000
1200000
1000000
800000
600000
400000
200000
Usage (coreh)

CFD

Data per job (GiB)	% Usage (Read)	%Usage (Write)
(0, 4)	**27.6%**	7.7%
[4, 32)	**30.7%**	**19.5%**
[32, 256)	**32.8%**	**28.4%**
[256, 2048)	8.5%	**37.9%**
[2048,)	0.4%	8.5%

- Applications such as SBLI, OpenFOAM, Nektar++, and HYDRA.
- CFD community usage shows a similar high-level profile to that for the climate modelling community; however, there is a larger difference in the distribution of usage.
- Jobs for both communities use grid-based modelling approaches.
- Need to read in large model descriptions and write out time-series trajectories with large amounts of data.
- CFD models can range in size from the tiny (e.g. flow in small blood vessels) to the very large (e.g. models of full offshore wind farms).

Biomolecular modelling



Biomolecular modelling

Data per job (GiB)	% Usage (Read)	%Usage (Write)
(0, 4)	**97.9%**	**30.5%**
[4, 32)	2.1%	**34.4%**
[32, 256)	0.0%	**32.6%**
[256, 2048)	0.0%	2.8%
[2048,)	0.0%	0.9%

- Applications such as GROMACS, NAMD, and Amber.
- Jobs in this community read in small amounts of data but write out larger amounts of data.
- The amount of data written is roughly correlated with job size.
- Most jobs produce trajectories with the model system details saved at many snapshots throughout the job to be used for further analysis after the job has finished.

Summary

LASSi Summary

- Designed to help HPC support staff triage and resolve issues of application slowdown due to contention in a shared filesystem.
- Metrics-based analysis in which risk and ops metrics correlate to the quantity and quality of an application's I/O.
- Can be used to:
  - Study the I/O profile of applications.
  - Understand common I/O usage of application groups.
  - Locate the reasons for slowdown of similar jobs.
  - Study filesystem usage in general.
- An application-centric non-invasive approach based on metrics is valuable in understanding application I/O behaviour in a shared filesystem.

SAFE Summary

- Imported per-job aggregate LASSi I/O statistics into SAFE.
  - Allows us to link I/O use statistics to other service aspects.
- Identified and understood 4 broad I/O use modes associated with different research communities:
  - Materials science: read small, write small.
  - Climate modelling: read large, write large; low diversity.
  - CFD: read large, write large; high diversity.
  - Biomolecular modelling: read small, write medium.
- Used identified I/O patterns to qualitatively understand future I/O requirements.

Next steps

- Further development
  - Automate detection of application slowdown and cause
  - Develop a model for application run time and I/O
  - Real time health status and warnings
- Continued / refined monitoring
  - Broaden user contact concerning jobs that may cause slowdown
  - Development of I/O scorechart that can be used for ARCHER resource requests in progress
- Continue analysis
  - Investigate I/O for communities with high I/O requirements but low total use
  - Integrate analysis of I/O operations into SAFE
- Collaboration with other sites that have shown interest

EOF

Acknowledgements:

archer

EPSRC
Engineering and Physical Sciences
Research Council

NERC
SCIENCE OF THE ENVIRONMENT

Stephen Booth (EPCC)

- Paper available here: arXiv:1906.03891
- Further info: Juan Herrera

Thanks for your attention

Questions?