NEXTGenIO: Resource Requirement Specification for Novel Data-aware and Workflow-enabled HPC Job Schedulers

Manos Farsarakis

@efarsarakis

e.farsarakis@epcc.ed.ac.uk

EPCC, The University of Edinburgh

Hi, I'm Manos!


THE UNIVERSITY of EDINBURGH
Emmanouil (Manos) Farsarakis
Applications Consultant
Tel +44 (0)131 651 7832
Fax +44 (0)131 650 6555
farsarakis@epcc.ed.ac.uk
www.epcc.ed.ac.uk.
epcc
The University of Edinburgh
James Clerk Maxwell Building
Peter Guthrie Tait Road
Edinburgh EH9 3FD

NEXTGenIO summary

Project

- Design, develop, and exploit HPC and HPDA system with NVRAM in compute nodes
- 36 month duration
- €8.1 million
- Approx. 50% committed to hardware development
- http://www.nextgenio.eu/
- This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 671951.

Partners

epcc

intel

FUJITSU
Fujitsu Technology Solutions

BSC

TECHNISCHE UNIVERSITÄT DRESDEN

allinea

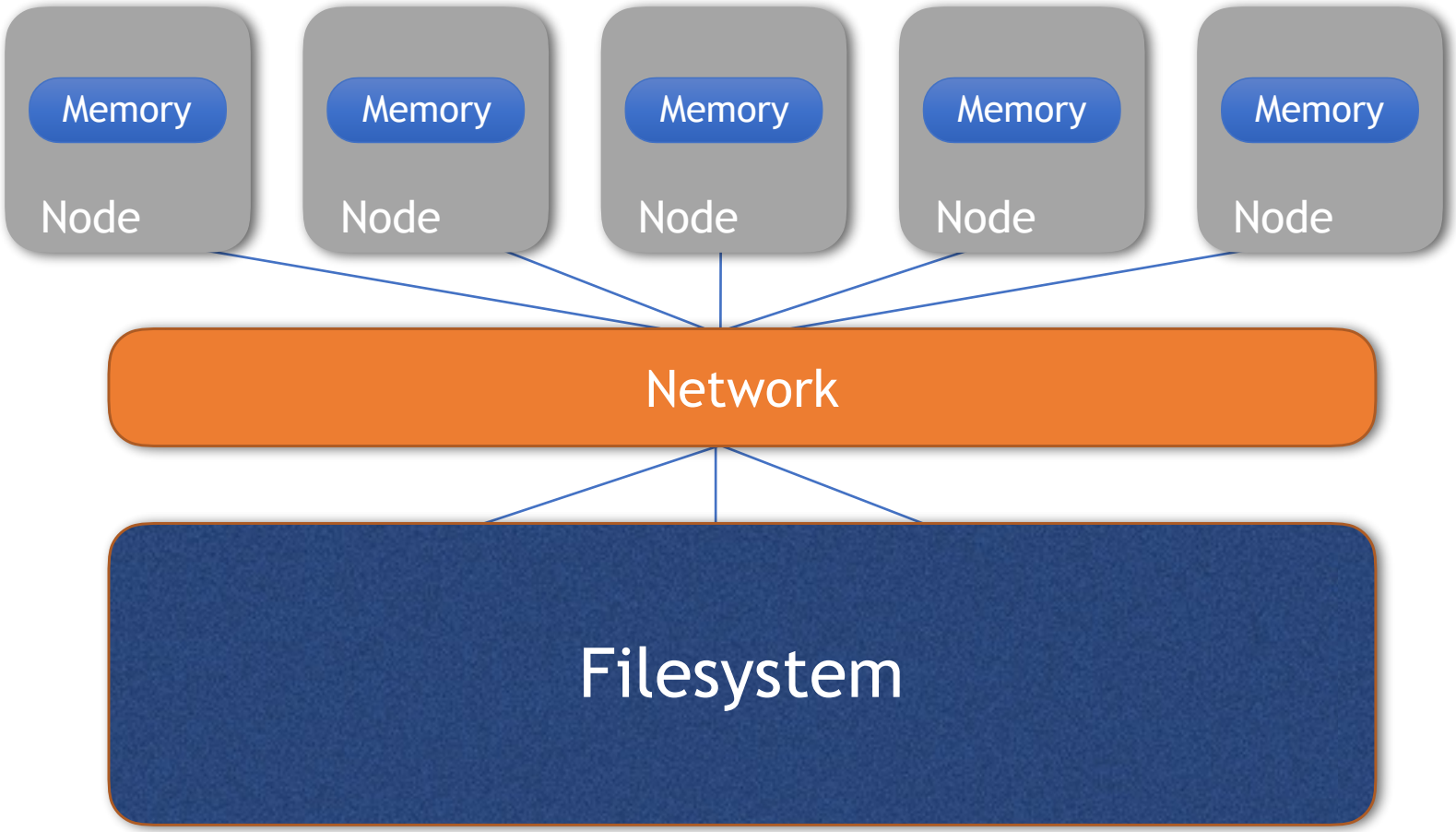
ECMWF

ARCTUR unverse

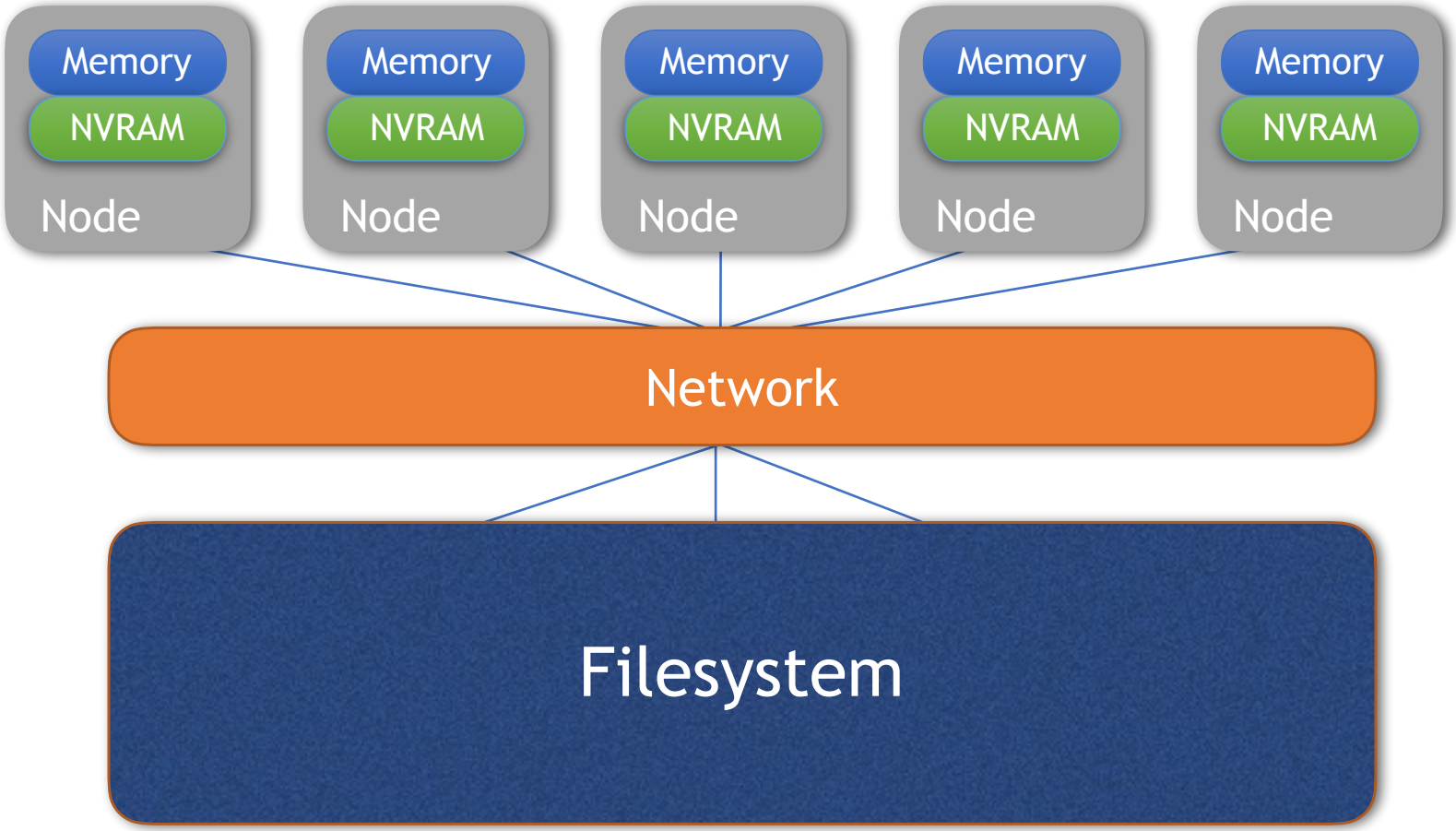
Our objectives

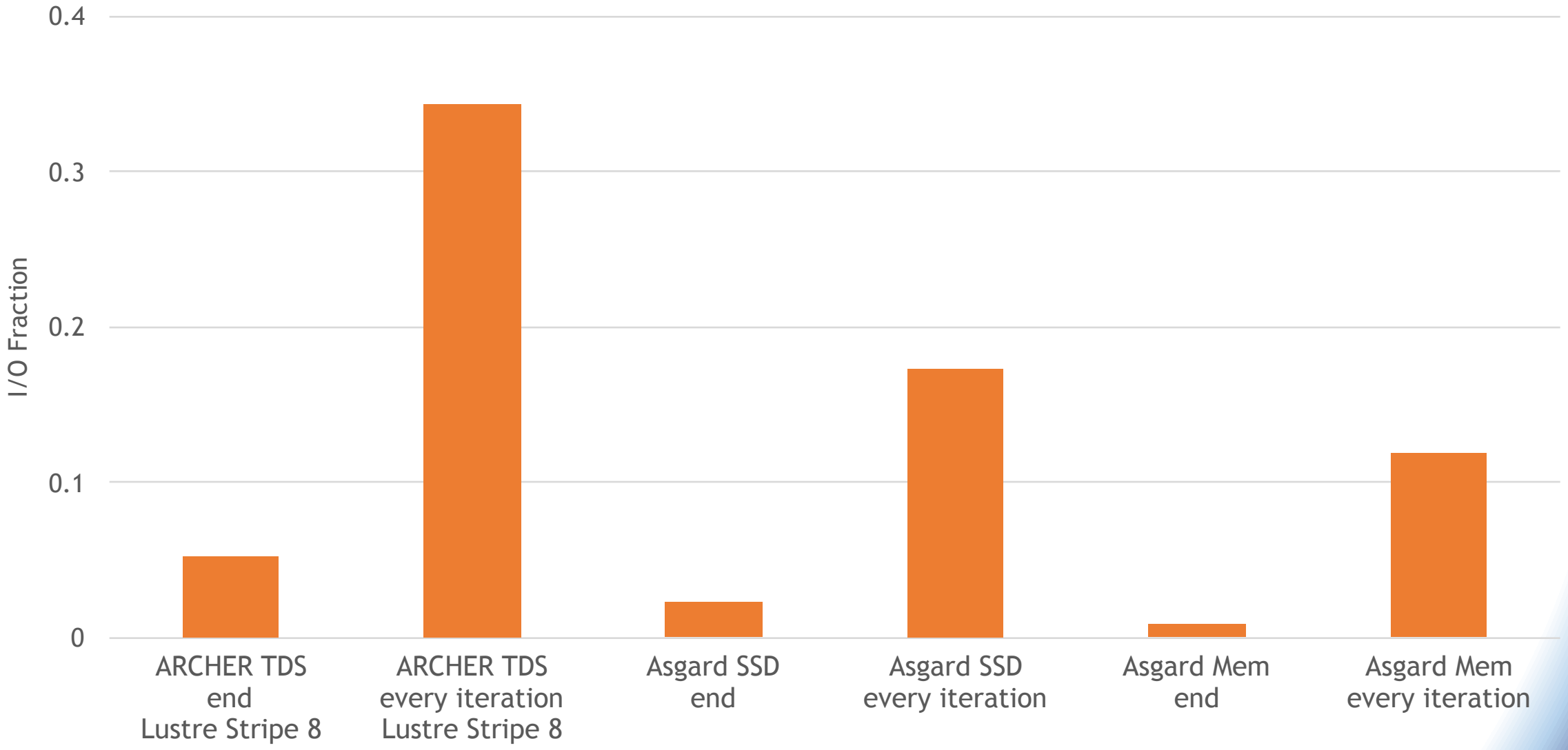
- Hardware platform prototype
  - Demonstrating the prototype's broad applicability for both HPC and data centric applications
- Exascale I/O investigation
  - Understanding how best to exploit NVRAM
- Systemware development:
  - Producing the necessary software to enable Exascale application execution on the hardware platform
- Application co-design
  - Understanding individual application I/O profiles and typical I/O workloads on shared systems running multiple different applications

Old System

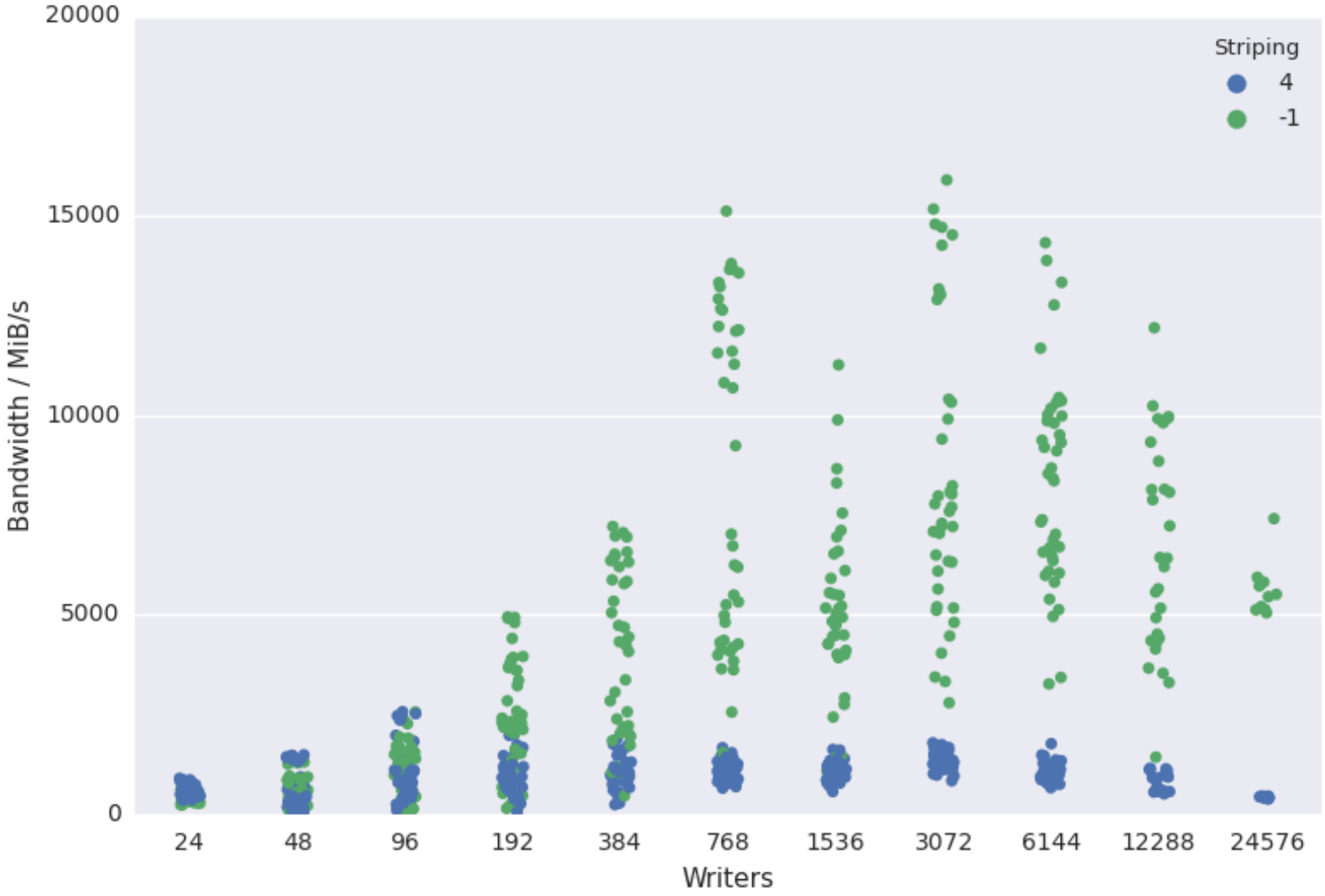

Memory
Node
Memory
Node
Memory
Node
Memory
Node
Memory
Node
Network
Filesystem

New System (1)


Memory
NVRAM
Node
Memory
NVRAM
Node
Memory
NVRAM
Node
Memory
NVRAM
Node
Memory
NVRAM
Node
Network
Filesystem

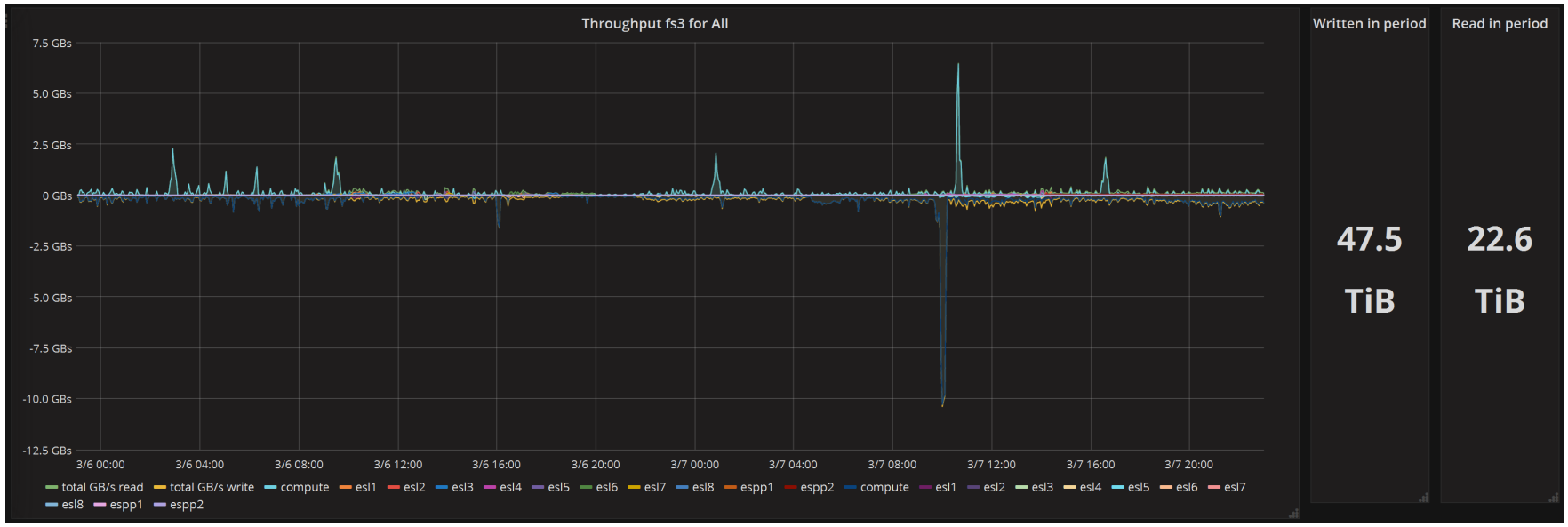

I/O Fraction
0.4
0.3
0.2
0.1
0
I/O Fraction
ARCHER TDS end Lustre Stripe 8
ARCHER TDS every iteration Lustre Stripe 8
Asgard SSD end
Asgard SSD every iteration
Asgard Mem end
Asgard Mem every iteration

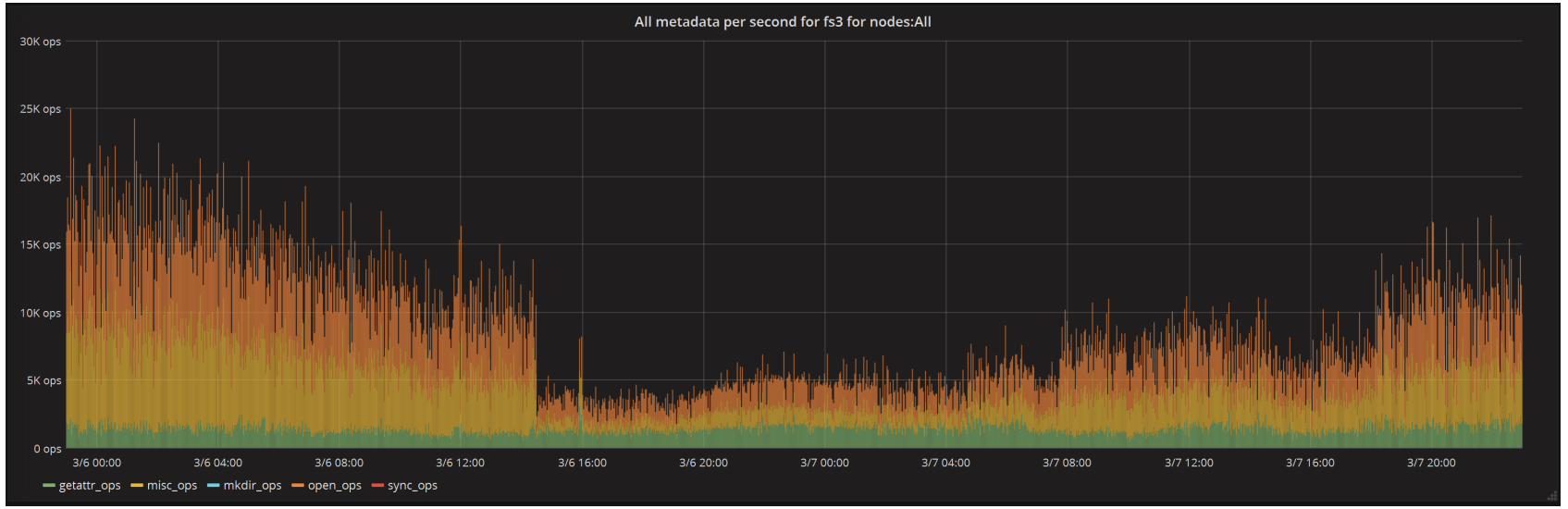
I/O Performance



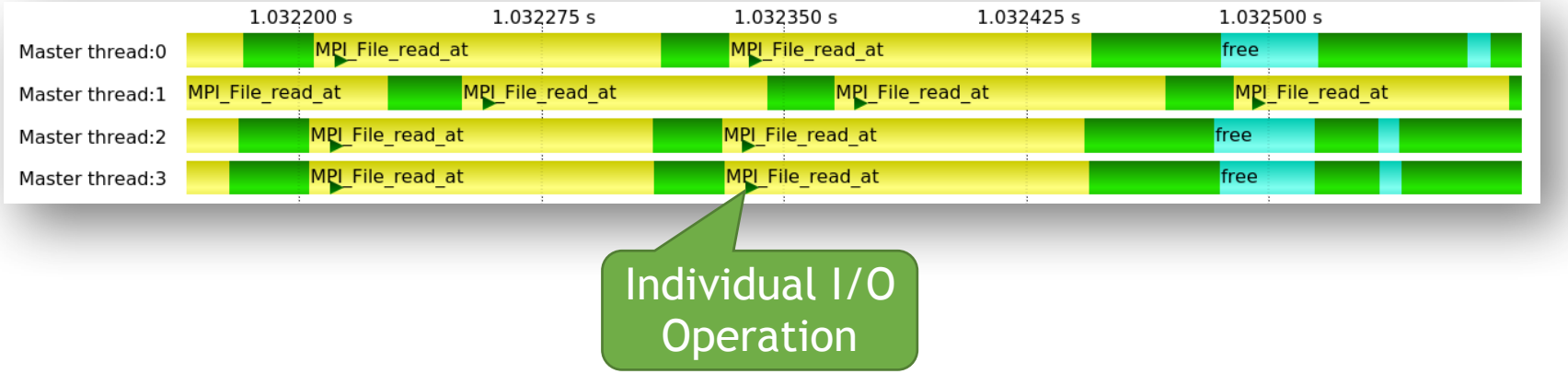
- https://www.archer.ac.uk/documentation/white-papers/parallelIO-benchmarking/ARCHER-Parallel-IO-1.0.pdf

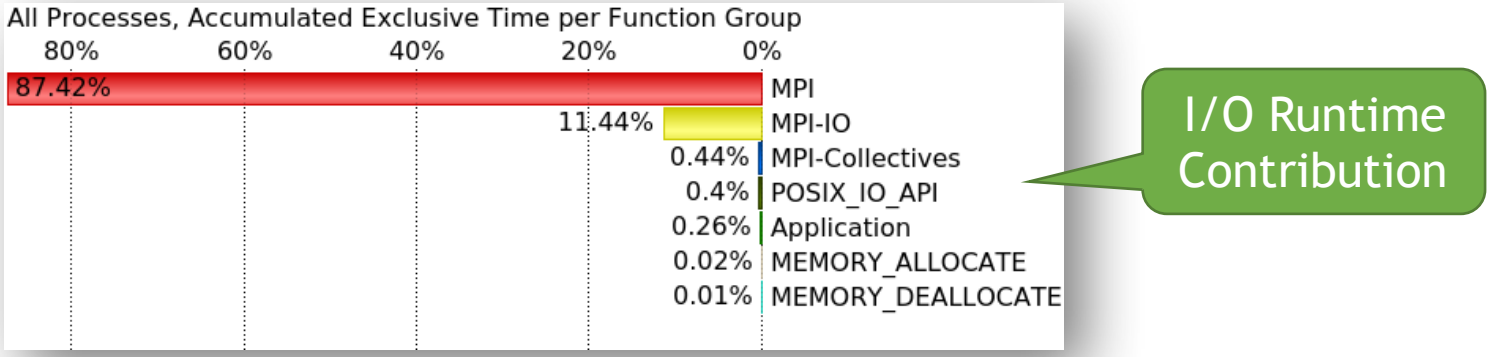
I/O


Throughput fs3 for All
7.5 GBs
5.0 GBs
2.5 GBs
0 GBs
-2.5 GBs
-5.0 GBs
-7.5 GBs
-10.0 GBs
-12.5 GBs
3/6 00:00
3/6 04:00
3/6 08:00
3/6 12:00
3/6 16:00
3/6 20:00
3/7 00:00
3/7 04:00
3/7 08:00
3/7 12:00
3/7 16:00
3/7 20:00
total GB/s read
total GB/s write
compute
esl1
esl2
esl3
esl4
esl5
esl6
esl7
esl8
espp1
espp2
compute
esl1
esl2
esl3
esl4
esl5
esl6
esl7
esl8
espp1
espp2
Written in period
47.5
TiB
Read in period
22.6
TiB

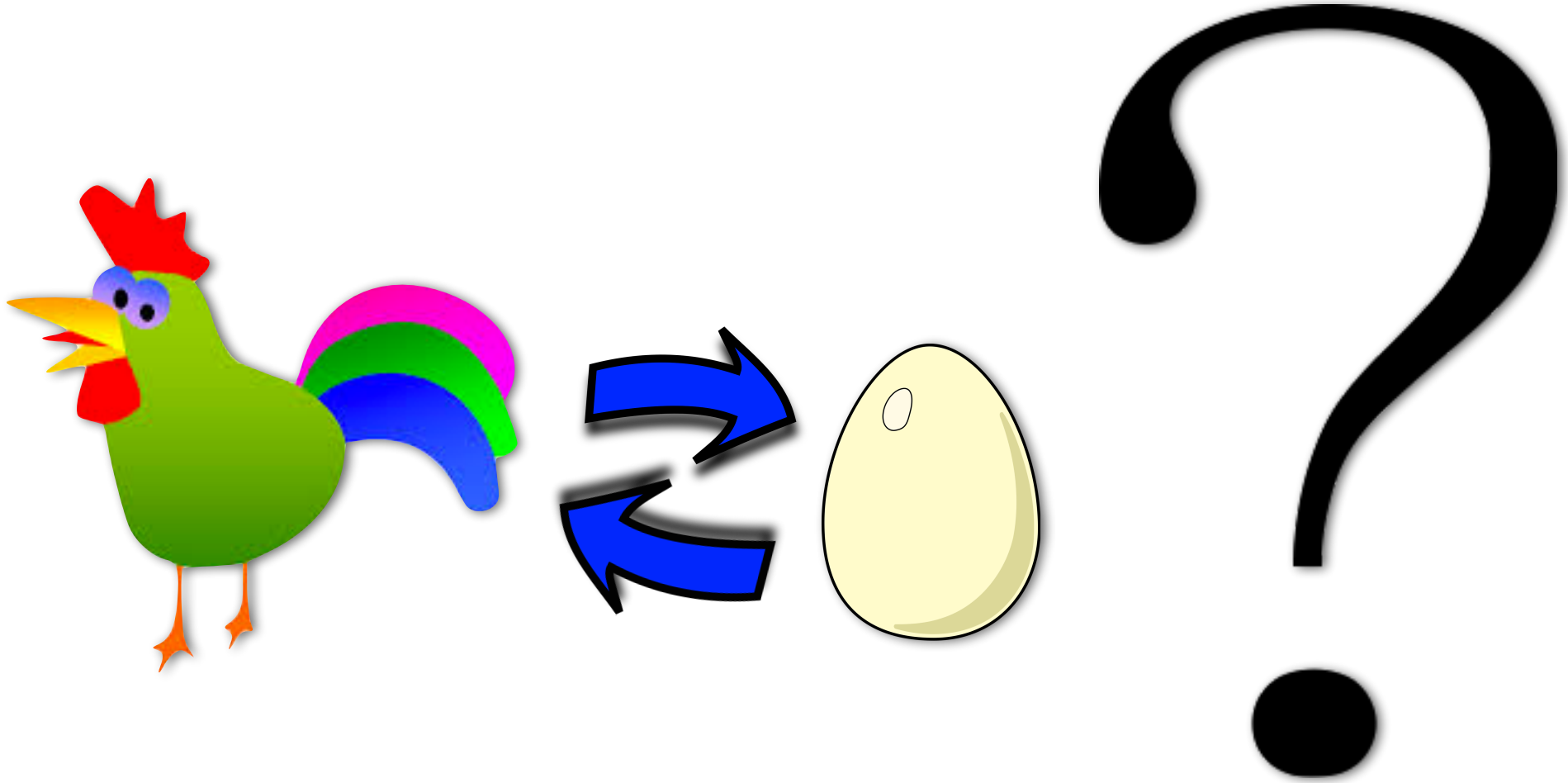

All metadata per second for fs3 for nodes:All
30K ops
25K ops
20K ops
15K ops
10K ops
5K ops
0 ops
3/6 00:00
3/6 04:00
3/6 08:00
3/6 12:00
3/6 16:00
3/6 20:00
3/7 00:00
3/7 04:00
3/7 08:00
3/7 12:00
3/7 16:00
3/7 20:00
getattr_ops
misc_ops
mkdir_ops
open_ops
sync_ops

I/O


1.032200 s
1.032275 s
1.032350 s
1.032425 s
1.032500 s
Master thread:0
Master thread:1
Master thread:2
Master thread:3
MPI_File_read_at
free
Individual I/O Operation


All Processes, Accumulated Exclusive Time per Function Group
80%
60%
40%
20%
0%
87.42%
MPI
11.44%
MPI-IO
0.44%
MPI-Collectives
0.4%
POSIX_IO_API
0.26%
Application
0.02%
MEMORY_ALLOCATE
0.01%
MEMORY_DEALLOCATE
I/O Runtime Contribution

Age old question…



Questions for you

- How do YOU do I/O?
- How much I/O do YOU do?

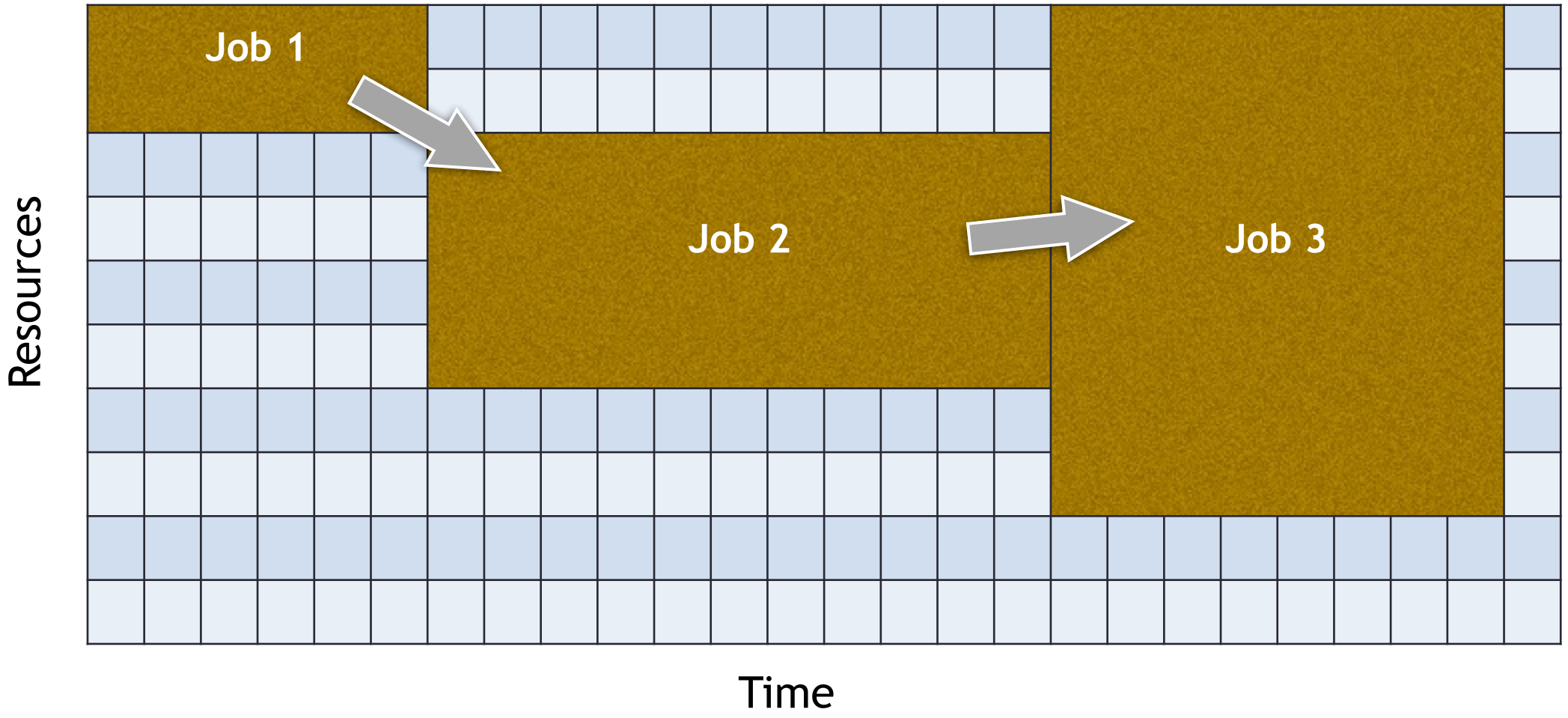
But more importantly…

- How do you WANT to do I/O?
- How much I/O would you WANT to do?

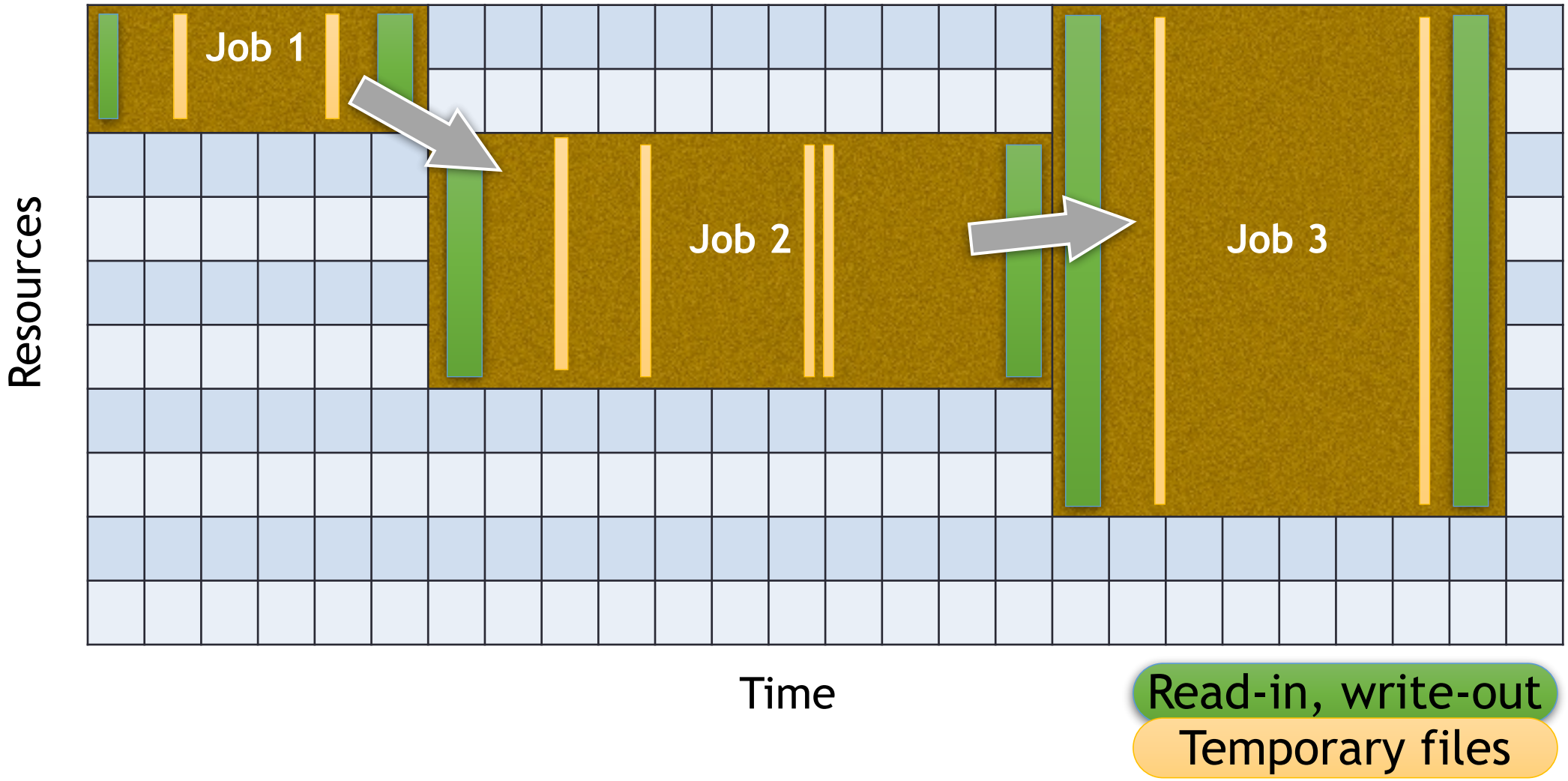
Types of things we are thinking about…

- Often read, never write files
- Frequently used files
- Temporary runtime files
- Disaster recovery files
- Workflows (which often include the above topics with renewed importance)

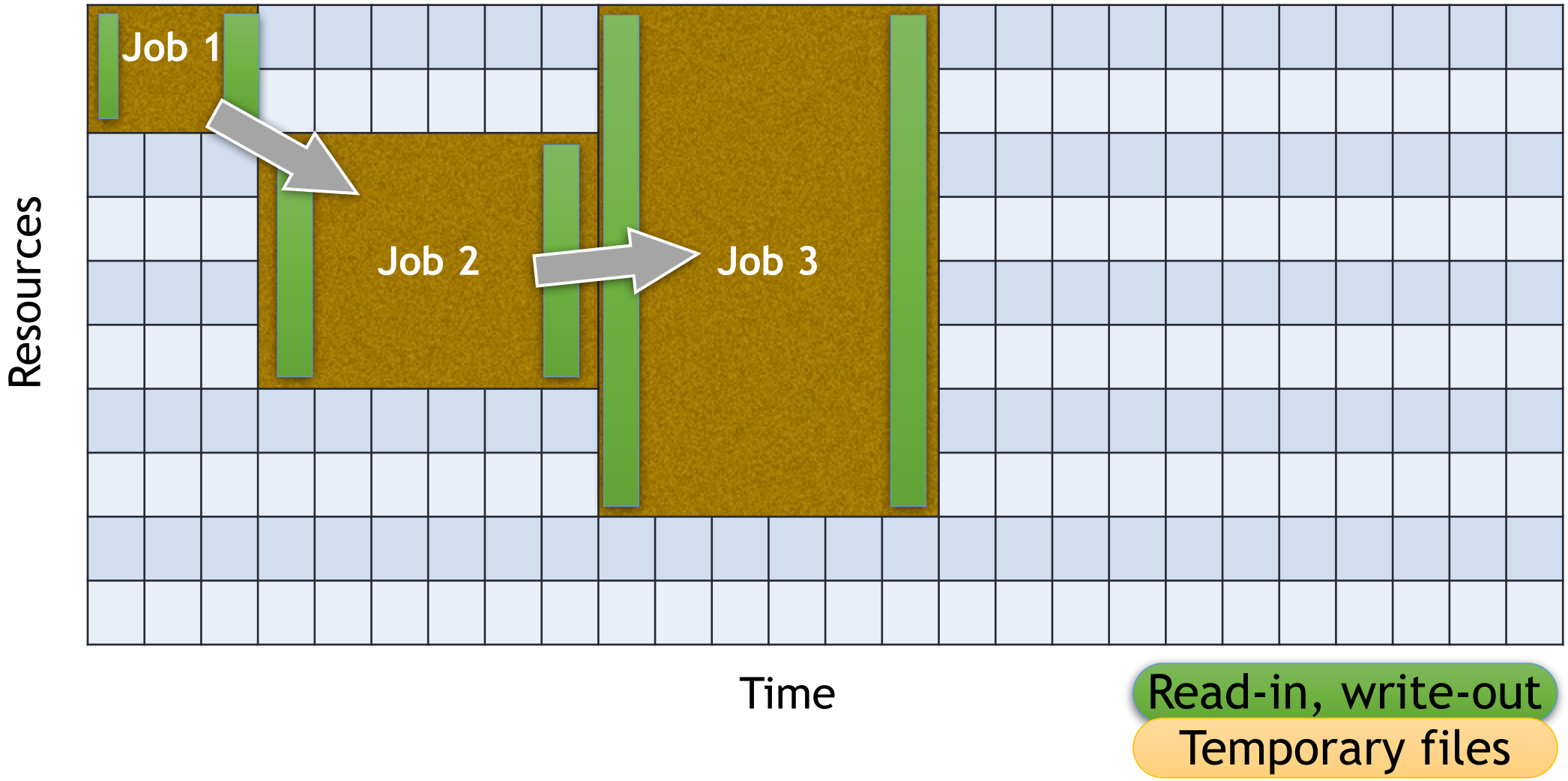
Workflows



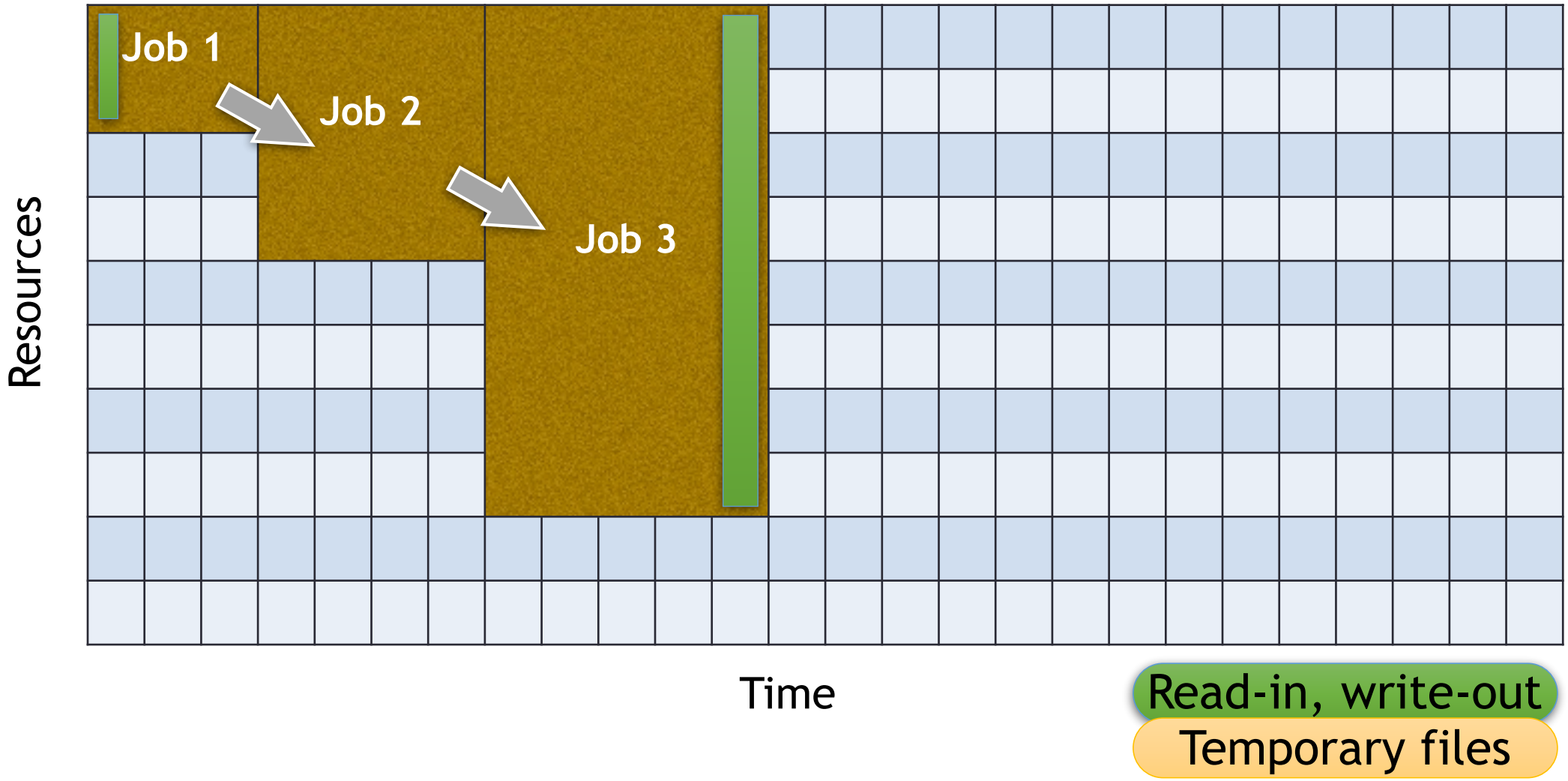
Workflows


Job 1
Job 2
Job 3
Resources
Time
Read-in, write-out
Temporary files

Workflows: Data Aware (1)



Workflows: Data Aware (2)



The Problem:

Data Aware Scheduler needs information about the data!

The Solution:

```
### JOB RESOURCE REQUIREMENT EXAMPLE FILE###
### STANDARD JOB VOCABULARY ###
#JOB_NAME: JOB2    ### ( OF 3 JOB WORKFLOW)
#QUEUE: STANDARD
#WALLTIME: 00:30:00
#NUM_NODES: 2
#PROCS_PER_NODE: 3
#DEPENDENT_ON: JOB
### AUGMENTED JOB RESOURCE REQUIREMENTS SPEC ###
### NEW VOCABULARY ###
#HPS: [
 [../JOB1/ON_NODE_FILES/checkpoint%[0-9]+ %.out, 10MB, IN],
 [../JOB2/READ_ONLY/initial_conditions% [0-9]+ %.dat, 2MB,IN],
 [../JOB2/SHARED/param_dictionary.dat, 50MB, READ_ONLY],
 [../JOB2/LOCAL/savedParams%[0-9]+ %.out, 5MB, LOCAL],
 [../JOB2/OUTPUT/checkpoint%[0-9]+%.out, 19MB,OUT]
        ]
#PM_FILE: "/PATH/TO/JOB2/mappings.dat"
aprun -n 6 -N 3 ./myApp --input=/PATH/TO/JOB1/ON_NODE_FILES
```

Summary

- NEXTGenIO developing a ***full*** hardware and software solution
- Data-Aware-Scheduler development has shown us that current job descriptions are not enough
- We have introduced JRRS as a means to bridge this gap
- Development is in initial stages: We welcome input!