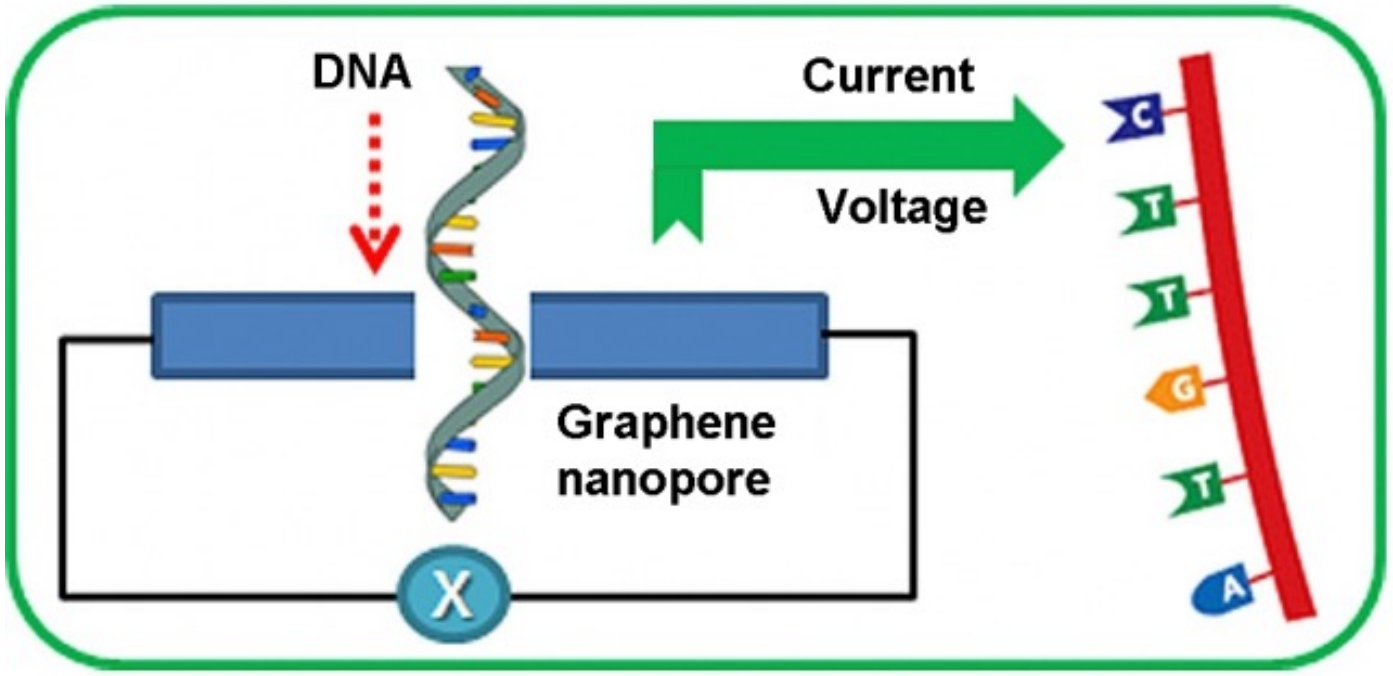
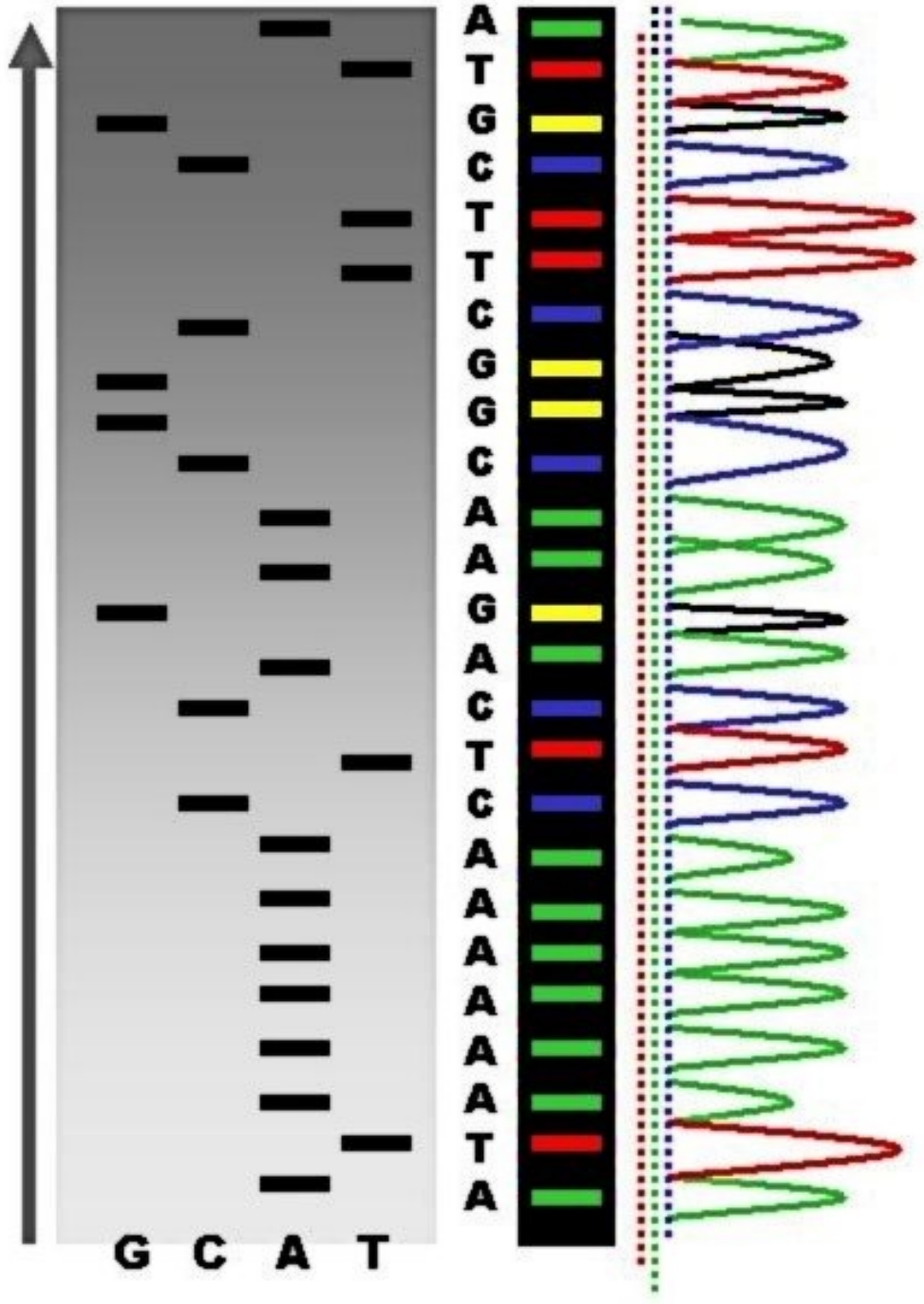
WHO ATE MY IOPS? IDENTIFYING HEAVY HITTERS USING JOBSTATS

James Beal, Principal Systems Administrator
Dave Holland, Principal Systems Administrator
V09 2022/11/30


INTRODUCTION

DNA SEQUENCING





THE PAST

Capillary sequencer (very slow)

- ***2 Mega ($2*10^6$) bases a day.***
- Read length of about 1000 base pairs
- A single human genome is about 6.4 gigabases

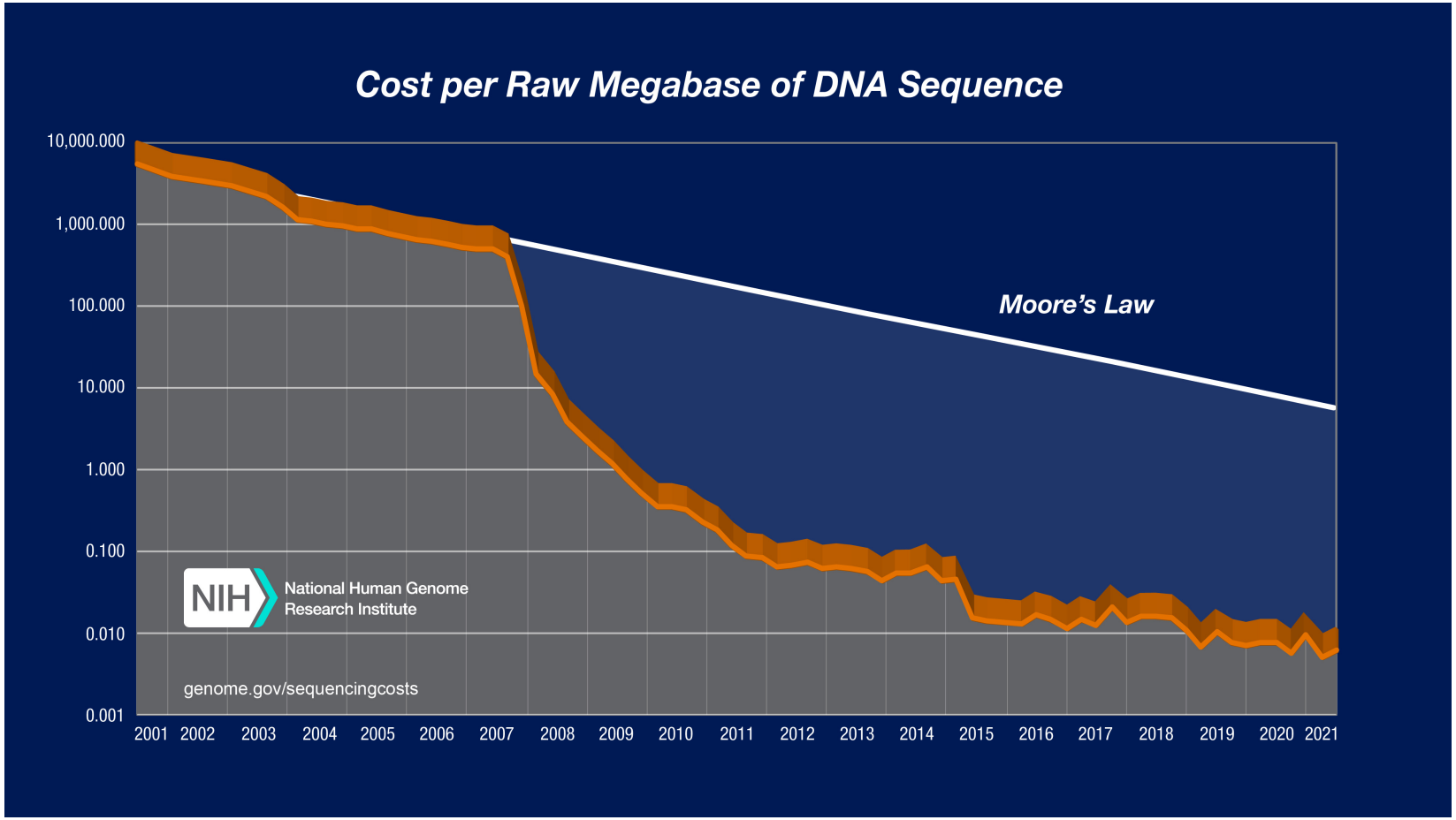
THE PRESENT

SBS (sequencing by synthesis, Novaseq X).

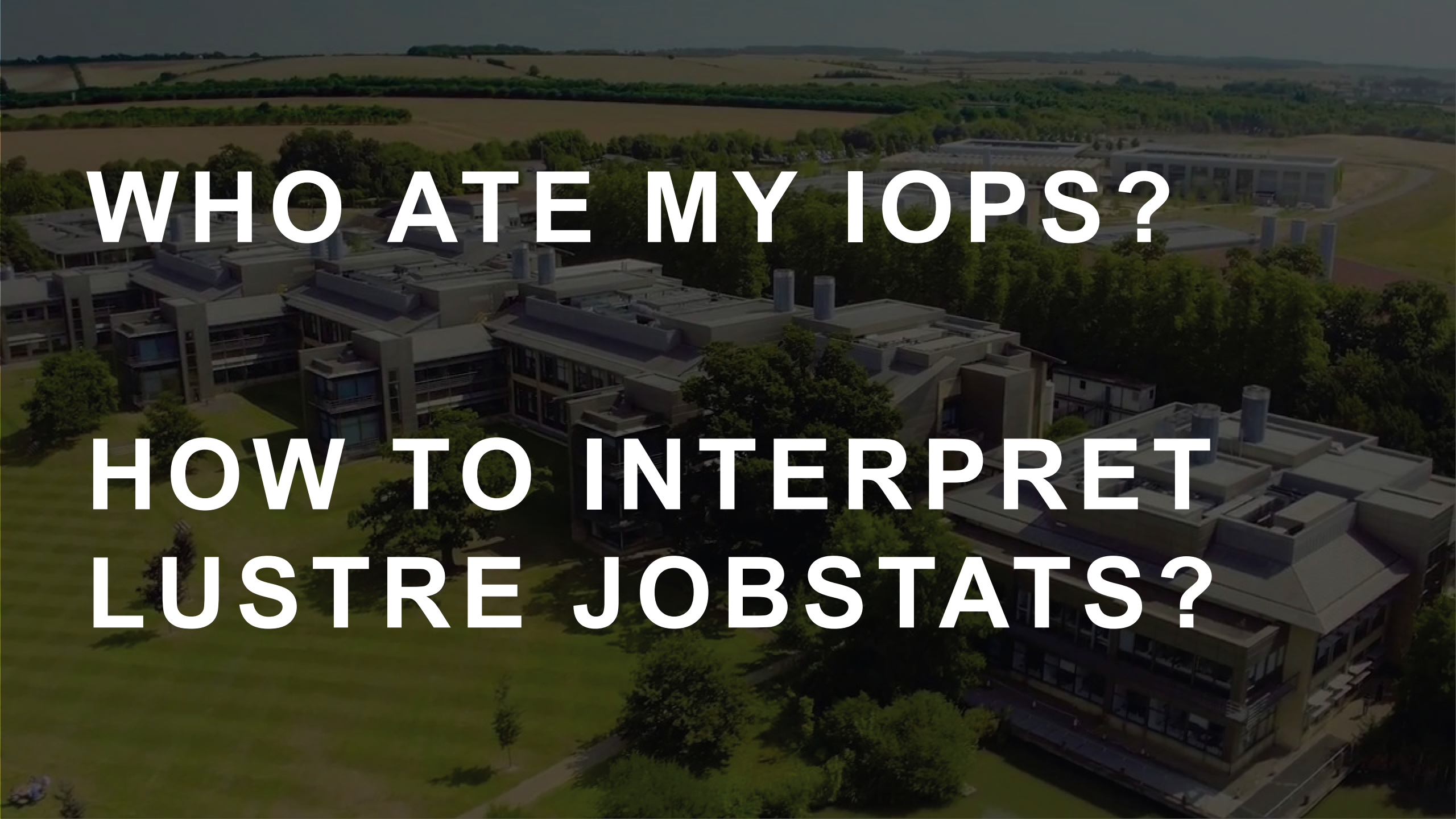
- ***8 Terabases ($8*10^{12}$) a day per sequencer.***
- 2.5 bytes a base or 20TB per sequencer per day working space.
- 150 base pair read length
- We have 20 of the previous generation currently in production.

THE FUTURE

Cost of sequencing is falling rapidly, HUGE impact on compute and storage!



https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data


WHO ATE MY IOPS?
HOW TO INTERPRET LUSTRE JOBSTATS?

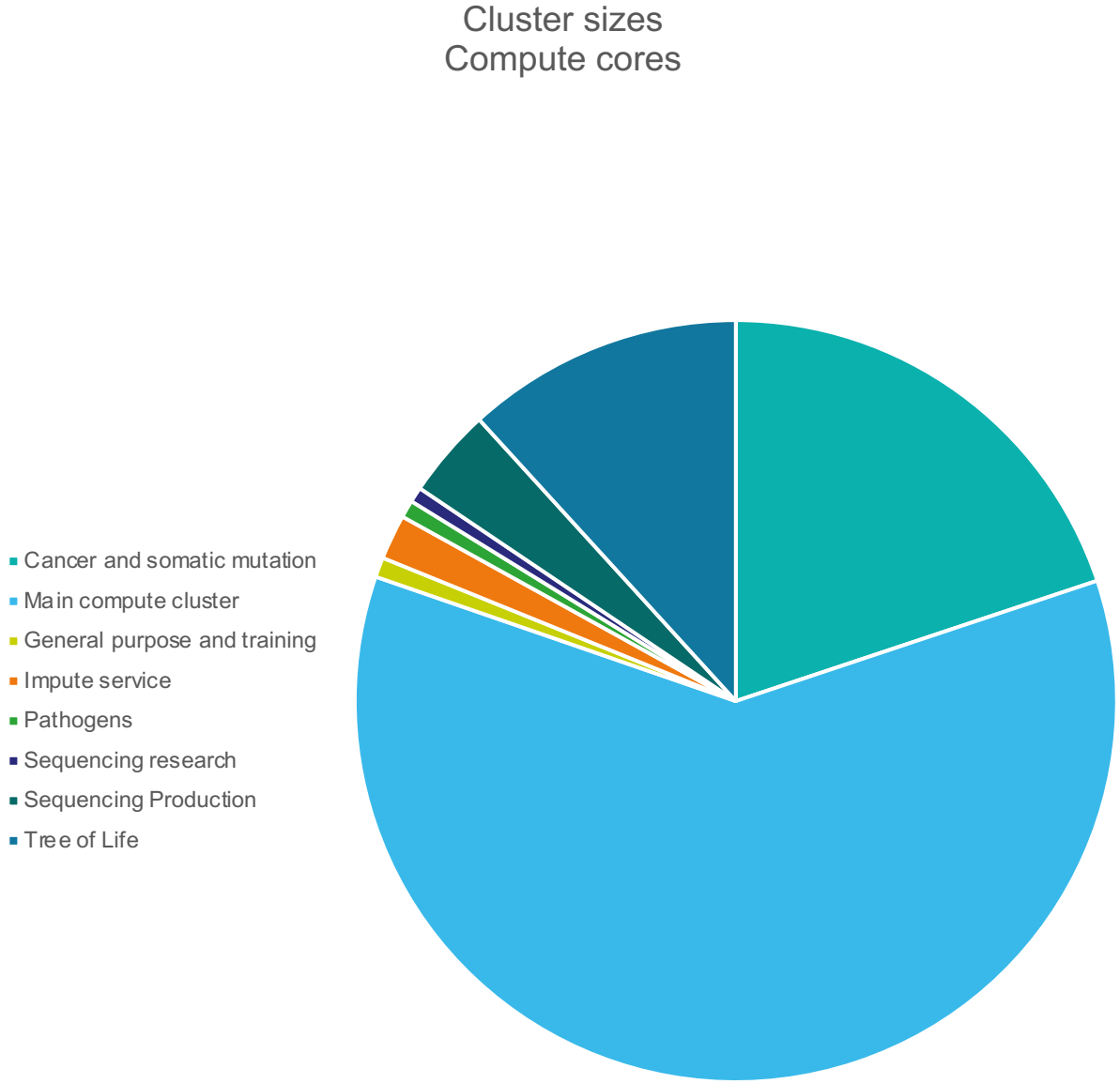
REQUIREMENTS

- User name
- Group name
- LSF project
- Compute cluster
- A pony

COMPUTE CLUSTERS

- A number of very similar configuration LSF clusters of different sizes. Separate clusters are used to ringfence resources.
- All* lustre filesystems are mounted everywhere.
- Lustre filesystems are not generally dedicated to a single group.

```
jb23@seqfarm3-head1:~$ df -H /lustre/*
Filesystem                                  Size  Used Avail Use% Mounted on
172.17.220.18@tcp:172.17.220.19@tcp:/lus17  3.1P  2.2P  886T  71% /lustre/scratch117
172.17.220.28@tcp:172.17.220.29@tcp:/lus18  3.1P  1.8P  1.3P  59% /lustre/scratch118
172.28.64.102@tcp:172.28.64.103@tcp:/lus19  2.5P  2.2P  254T  90% /lustre/scratch119
10.177.252.5@tcp:10.177.252.4@tcp:/lus20    3.1P  1.2P  1.9P  38% /lustre/scratch120
10.160.32.5@tcp:10.160.32.4@tcp:/lus23      5.3P  2.8P  2.5P  53% /lustre/scratch123
10.160.36.4@tcp:10.160.36.5@tcp:/lus24      5.3P  1.8P  3.5P  34% /lustre/scratch124
10.160.40.4@tcp:10.160.40.5@tcp:/lus25      5.3P  229T  5.1P   5% /lustre/scratch125
10.160.42.4@tcp:10.160.42.5@tcp:/lus26      5.3P  327T  5.0P   7% /lustre/scratch126
```



JOBSTATS

- https://doc.lustre.org/lustre_manual.xhtml#jobstats
- A method of tagging I/O with additional metadata.
- Introduced in Lustre 2.3 (in 2011)
- A 32 character tag.
- The default tag is procname_uid
  - executable_name.uid
- A filesystem or a compute node can be configured to read a environment variable as the tag.

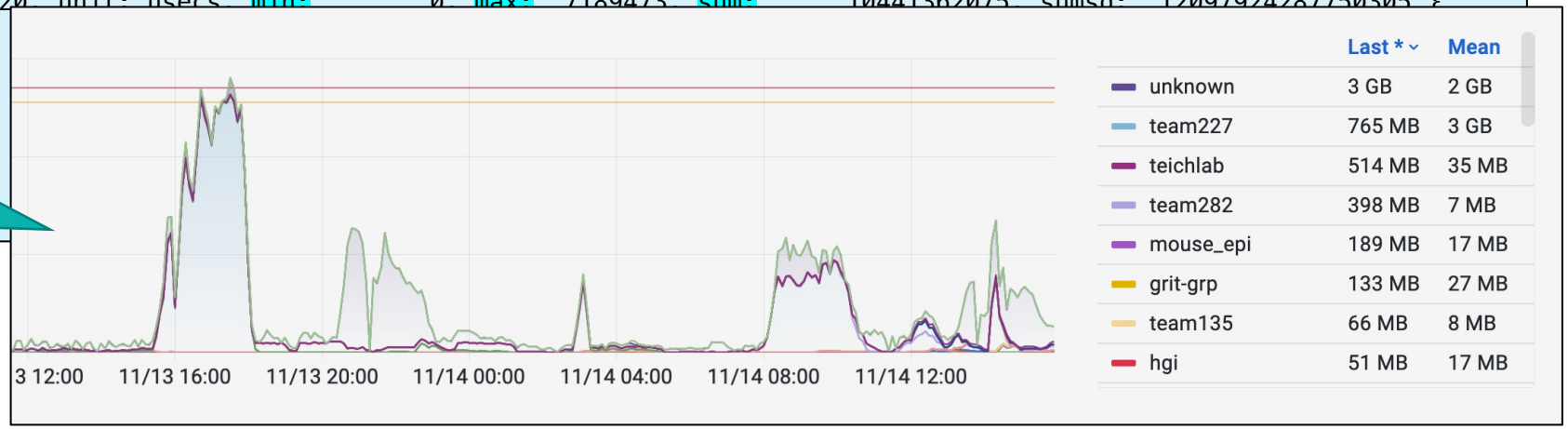
RAW JOBSTATS OUTPUT

All the information we need is in “jobstats” but is difficult to interpret

```
# lctl get_param obdfilter.lus24-OST0000.job_stats
obdfilter.lus24-OST0000.job_stats=
job_stats:
- job_id:          SA1CUzMGBwYH,analysis-cgp
  snapshot_time:   1667904951
  read_bytes:      { samples:   297356696, unit: bytes, min:     4096, max: 16777216, sum: 1762446621732864, sumsq: 9915053801461514240 }
  write_bytes:     { samples:    20518685, unit: bytes, min:        1, max: 16777216, sum:   98655040843302, sumsq: 4627393946935432142 }
  read:            { samples:   297356696, unit: usecs, min:        1, max: 12945252, sum:   11870681362787, sumsq: 3234937531575968175 }
  write:           { samples:    20518685, unit: usecs, min:        1, max: 10280151, sum:      98226898749, sumsq:  50535912682848539 }
  getattr:         { samples:           0, unit: usecs, min:        0, max:        0, sum:                 0, sumsq:                    0 }
  setattr:         { samples:        5697, unit: usecs, min:        4, max:  6119237, sum:        151065923, sumsq:    327770191069603 }
  punch:           { samples:     1458531, unit: usecs, min:        5, max: 31397444, sum:      15753447595, sumsq:  36605999152526109 }
  sync:            { samples:     6517520, unit: usecs, min:        0, max:  7189473, sum:      10441362075, sumsq:  12097924287750305 }
  destroy:         { samples:
  create:          { samples:
  statfs:          { samples:
  get_info:        { samples:
  set_info:        { samples:
  quotactl:
  prealloc:
```

Jobstats command

Let’s export to Grafana, so it is easy to read




EXAMPLE 1

General / Lustre per user stats

now-2d to now-4m

filesystem lus23 type Enter variable value

byfarm
bygroup
bylsf_project
byusername

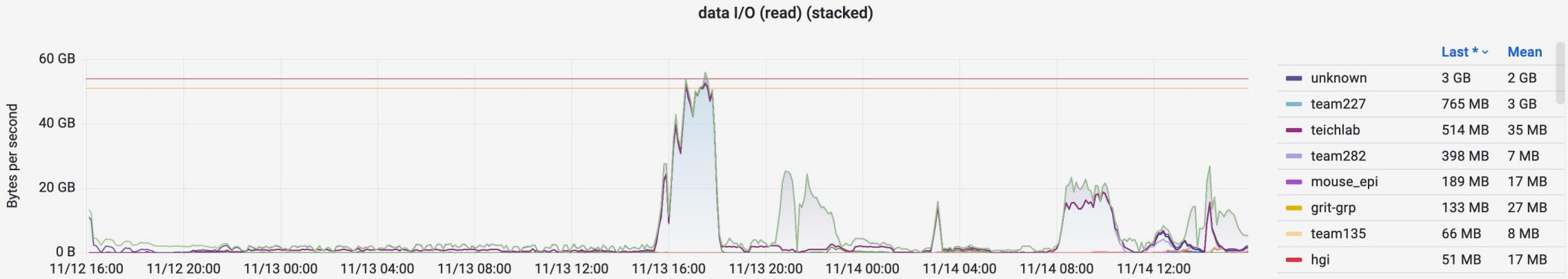
Select filesystem

asional u ata "spikes" are due to a Lustre server bug and should be disregarded.

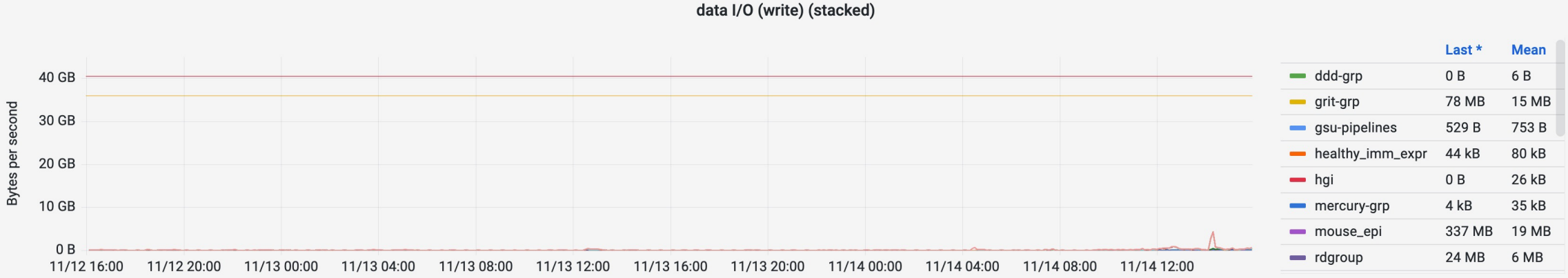
overall I/O shown ternal.sanger.ac.uk/d/000000203/lustre-stats?orgId=1

idicates 90% of te ty. The yellow line indicates 85%. Around these levels, the filesystem will feel slow to interactive users.

Read/Write



	Last *	Mean
unknown	3 GB	2 GB
team227	765 MB	3 GB
teichlab	514 MB	35 MB
team282	398 MB	7 MB
mouse_epi	189 MB	17 MB
grit-grp	133 MB	27 MB
team135	66 MB	8 MB
hgi	51 MB	17 MB



	Last *	Mean
ddd-grp	0 B	6 B
grit-grp	78 MB	15 MB
gsu-pipelines	529 B	753 B
healthy_imm_expr	44 kB	80 kB
hgi	0 B	26 kB
mercury-grp	4 kB	35 kB
mouse_epi	337 MB	19 MB
rdgroup	24 MB	6 MB

Stats

General / Lustre per user stats

now-2d to now-4m

filesystem lus23 type Enter variable value

byfarm
bygroup
bylsf_project
byusername

Select how to aggregate data

NB occasional u … at … stre server bug and should be disregarded.

Compare with overall I/O shown … iterna … d=1

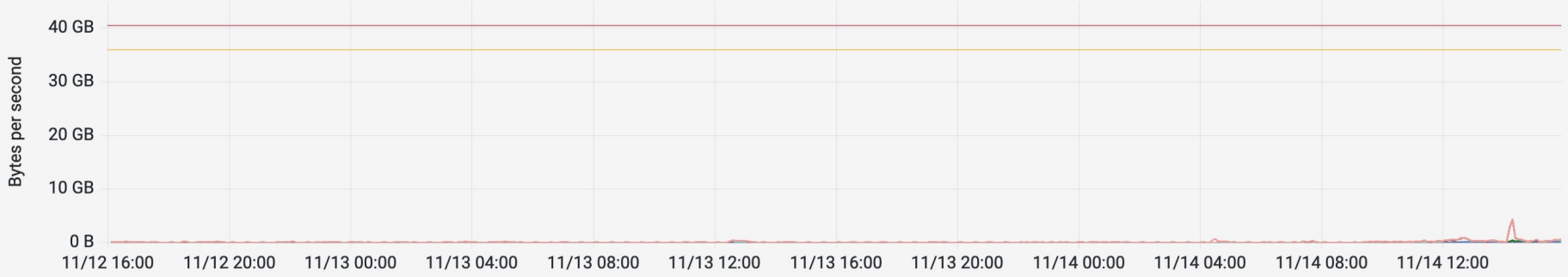
The red line indicates 90% of te … ty. Th … ls, the filesystem will feel slow to interactive users.

Read/Write

data I/O (read) (stacked)

	Last *	Mean
unknown	3 GB	2 GB
team227	765 MB	3 GB
teichlab	514 MB	35 MB
team282	398 MB	7 MB
mouse_epi	189 MB	17 MB
grit-grp	133 MB	27 MB
team135	66 MB	8 MB
hgi	51 MB	17 MB

data I/O (write) (stacked)



	Last *	Mean
ddd-grp	0 B	6 B
grit-grp	78 MB	15 MB
gsu-pipelines	529 B	753 B
healthy_imm_expr	44 kB	80 kB
hgi	0 B	26 kB
mercury-grp	4 kB	35 kB
mouse_epi	337 MB	19 MB
rdgroup	24 MB	6 MB

Stats

General / Lustre per user stats

now-2d to now-4m

filesystem lus23 type Enter variable value

byfarm
bygroup
bylsf_project
byusername

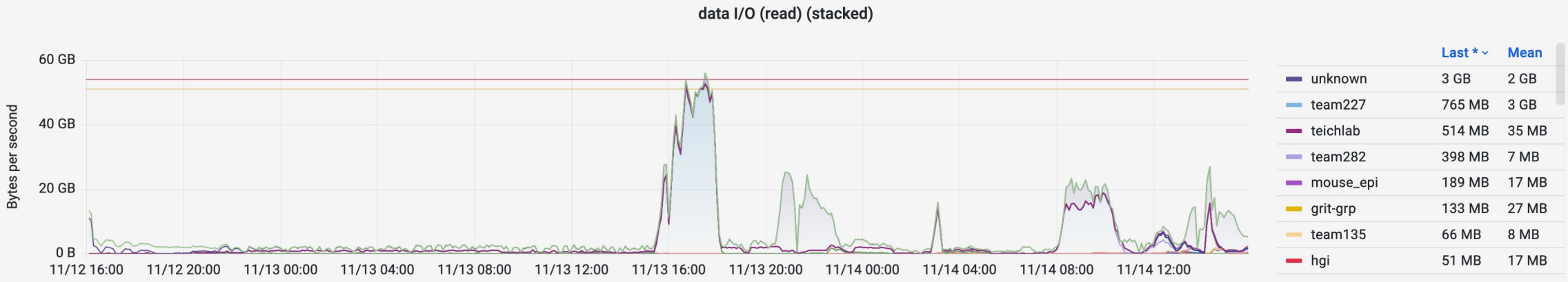
Observable window

NB occasional u ata "spikes" are due to a Lustre server bug sregarded.

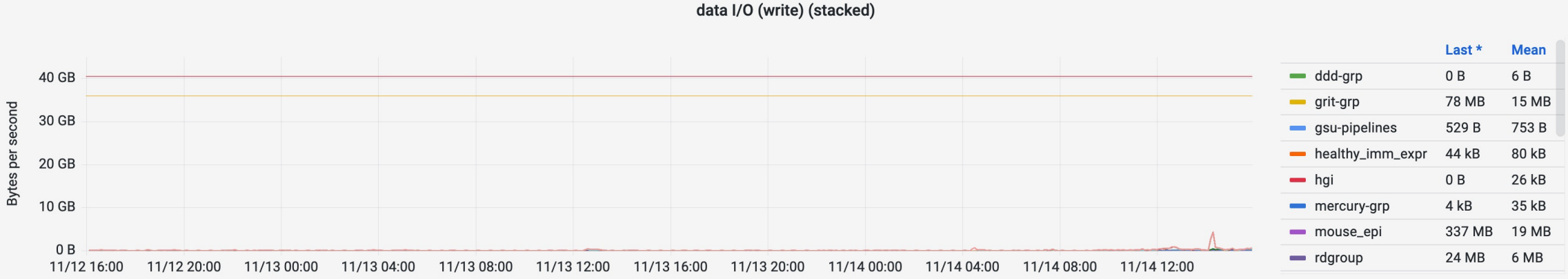
Compare with overall I/O shown ternal.sanger.ac.uk/d/000000203/lustre-stats?orgId=1

The red line indicates 90% of te ty. The yellow line indicates 85%. Around these levels, the filesystem will feel slow

Read/Write



	Last *	Mean
unknown	3 GB	2 GB
team227	765 MB	3 GB
teichlab	514 MB	35 MB
team282	398 MB	7 MB
mouse_epi	189 MB	17 MB
grit-grp	133 MB	27 MB
team135	66 MB	8 MB
hgi	51 MB	17 MB



	Last *	Mean
ddd-grp	0 B	6 B
grit-grp	78 MB	15 MB
gsu-pipelines	529 B	753 B
healthy_imm_expr	44 kB	80 kB
hgi	0 B	26 kB
mercury-grp	4 kB	35 kB
mouse_epi	337 MB	19 MB
rdgroup	24 MB	6 MB

Stats

General / Lustre per user stats

now-2d to now-4m

filesystem lus23 type Enter variable value

byfarm
bygroup
bylsf_project
byusername

NB occasional u ata "spikes" are due to a Lustre server bug and should be disregarded.

Compare with overall I/O shown ternal.sanger.ac.uk/d/000000203/lustre-stats?orgId=1

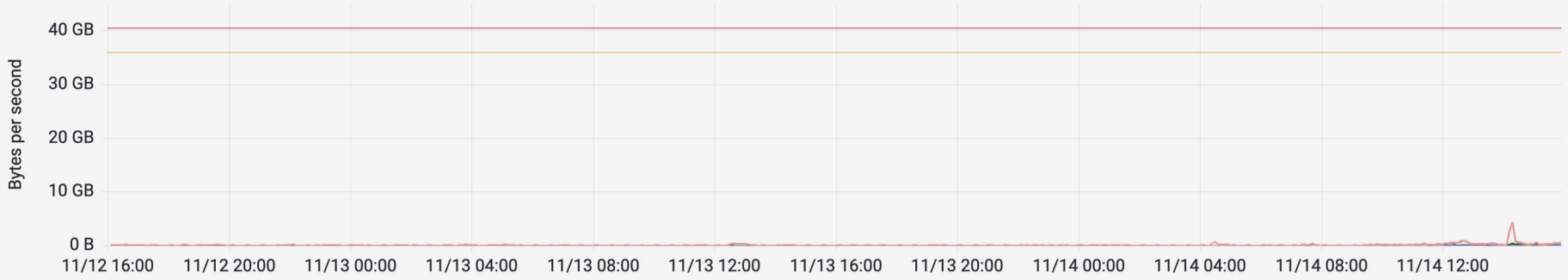
The red line indicates 90% of te ty. The yellow line indicates 85%. Around these levels, the filesystem will feel slow to interactive users.

Read/Write

data I/O (read) (stacked)

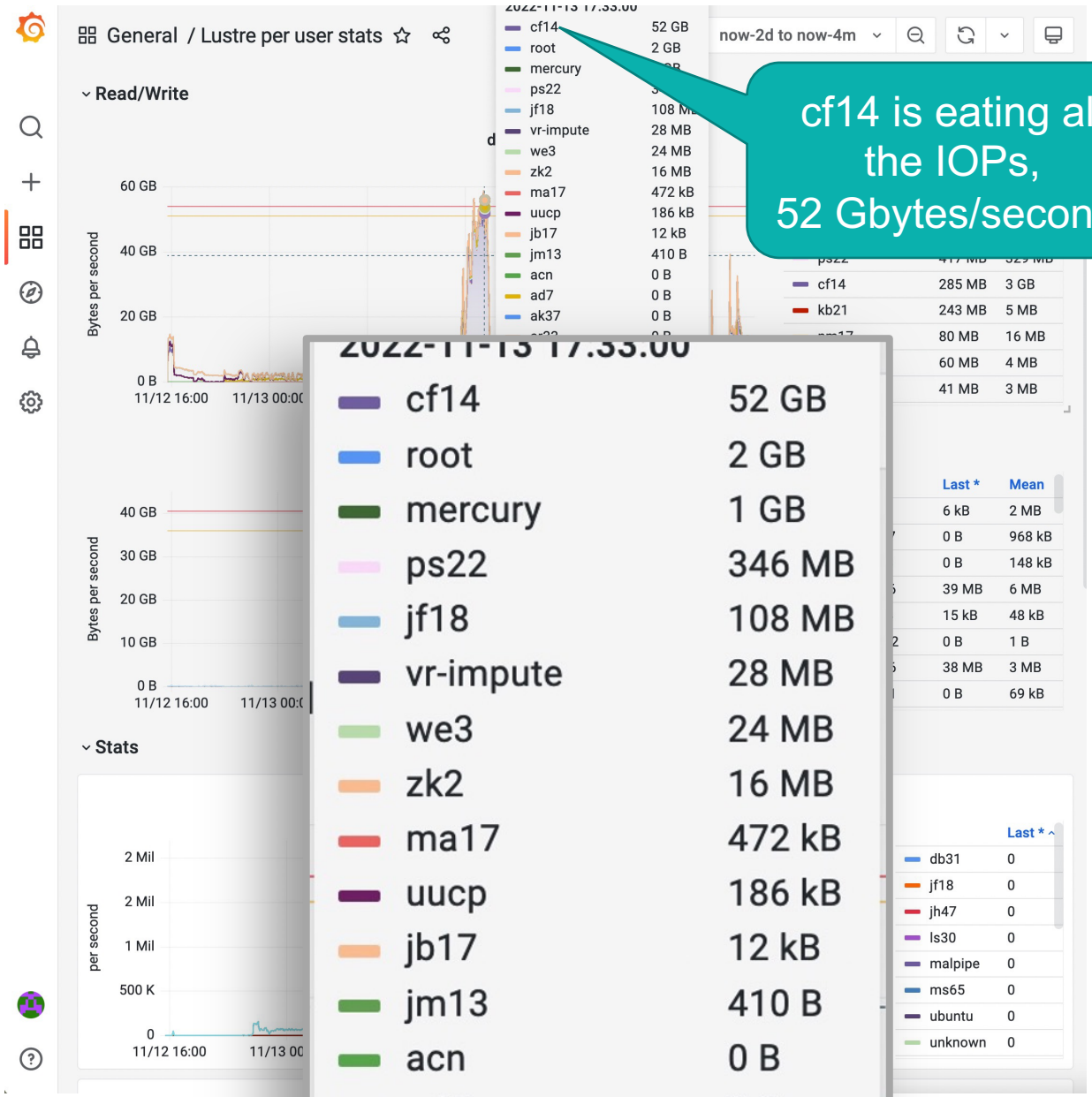
	Last *	Mean
unknown	3 GB	2 GB
team227	765 MB	3 GB
teichlab	514 MB	35 MB
team282	398 MB	7 MB
mouse_epi	189 MB	17 MB
grit-grp	133 MB	27 MB
team135	66 MB	8 MB
hgi	51 MB	17 MB

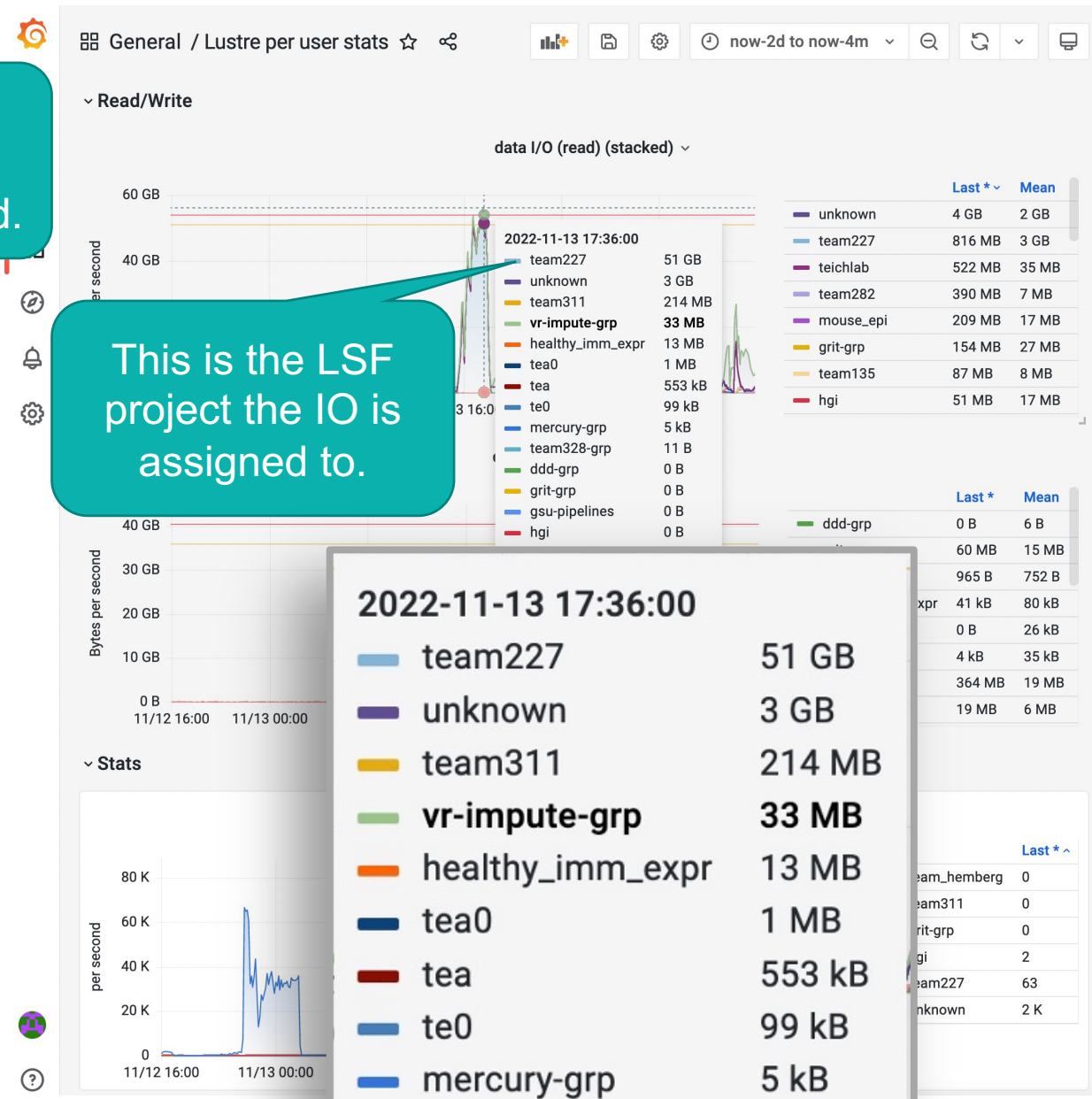
data I/O (write) (stacked)



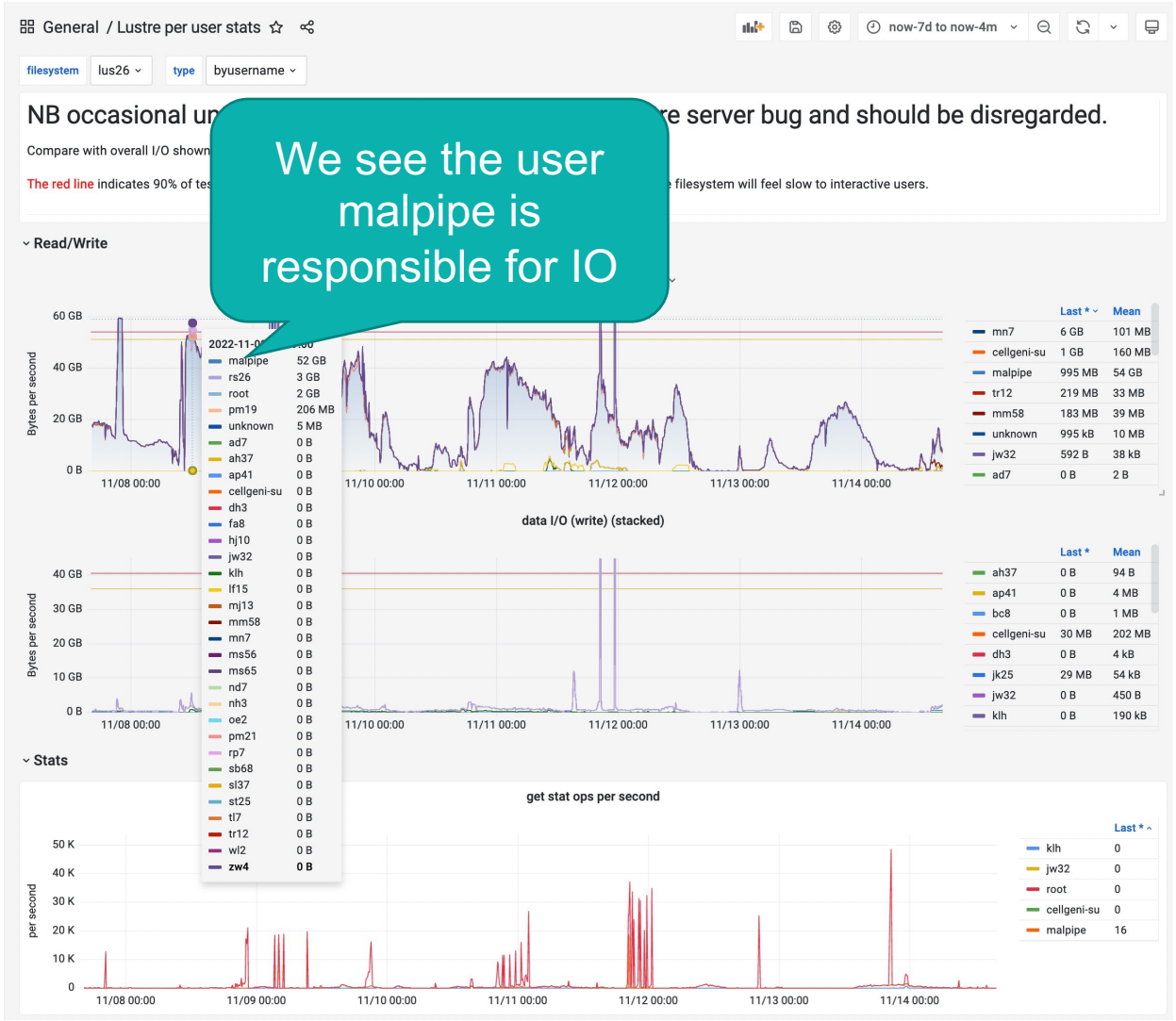
	Last *	Mean
ddd-grp	0 B	6 B
grit-grp	78 MB	15 MB
gsu-pipelines	529 B	753 B
healthy_imm_expr	44 kB	80 kB
hgi	0 B	26 kB
mercury-grp	4 kB	35 kB
mouse_epi	337 MB	19 MB
rdgroup	24 MB	6 MB

Stats


General / Lustre per user stats
Read/Write
now-2d to now-4m
cf14 is eating all the IOPs, 52 Gbytes/second.
2022-11-13 17:33:00
cf14 52 GB
root 2 GB
mercury 1 GB
ps22 346 MB
jf18 108 MB
vr-impute 28 MB
we3 24 MB
zk2 16 MB
ma17 472 kB
uucp 186 kB
jb17 12 kB
jm13 410 B
acn 0 B
ad7 0 B
ak37 0 B
ar33 0 B
Bytes per second
Stats
per second


General / Lustre per user stats
Read/Write
now-2d to now-4m
data I/O (read) (stacked)
This is the LSF project the IO is assigned to.
2022-11-13 17:36:00
team227 51 GB
unknown 3 GB
team311 214 MB
vr-impute-grp 33 MB
healthy_imm_expr 13 MB
tea0 1 MB
tea 553 kB
te0 99 kB
mercury-grp 5 kB
team328-grp 11 B
ddd-grp 0 B
grit-grp 0 B
Bytes per second
Stats
per second


EXAMPLE 2


General / Lustre per user stats
now-7d to now-4m
filesystem
lus26
type
byusername
NB occasional un
re server bug and should be disregarded.
Compare with overall I/O shown
The red line indicates 90% of tes
filesystem will feel slow to interactive users.
We see the user malpipe is responsible for IO
Read/Write
Bytes per second
60 GB
40 GB
20 GB
0 B
11/08 00:00
11/10 00:00
11/11 00:00
11/12 00:00
11/13 00:00
11/14 00:00
malpipe 52 GB
rs26 3 GB
root 2 GB
pm19 206 MB
unknown 5 MB
ad7 0 B
ah37 0 B
ap41 0 B
cellgeni-su 0 B
dh3 0 B
fa8 0 B
hj10 0 B
jw32 0 B
klh 0 B
lf15 0 B
mj13 0 B
mm58 0 B
mn7 0 B
ms56 0 B
ms65 0 B
nd7 0 B
nh3 0 B
oe2 0 B
pm21 0 B
rp7 0 B
sb68 0 B
sl37 0 B
st25 0 B
tl7 0 B
tr12 0 B
wl2 0 B
zw4 0 B
Last * Mean
mn7 6 GB 101 MB
cellgeni-su 1 GB 160 MB
malpipe 995 MB 54 GB
tr12 219 MB 33 MB
mm58 183 MB 39 MB
unknown 995 kB 10 MB
jw32 592 B 38 kB
ad7 0 B 2 B
data I/O (write) (stacked)
40 GB
30 GB
20 GB
10 GB
0 B
ah37 0 B 94 B
ap41 0 B 4 MB
bc8 0 B 1 MB
cellgeni-su 30 MB 202 MB
dh3 0 B 4 kB
jk25 29 MB 54 kB
jw32 0 B 450 B
klh 0 B 190 kB
Stats
get stat ops per second
per second
50 K
40 K
30 K
20 K
10 K
0
11/09 00:00
klh 0
jw32 0
root 0
cellgeni-su 0
malpipe 16

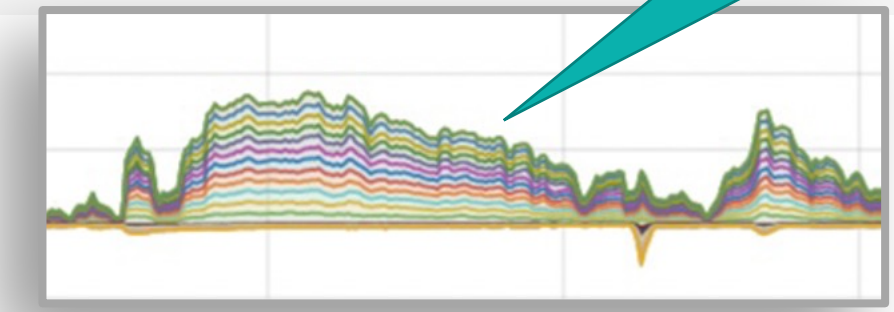

General / Lustre per user stats
now-7d to now-4m
filesystem
lus26
type
bylsf_project
NB occasional unexpected data "spikes" are due to a Lustre server bug and should be disregarded.
Compare with overall I/O shown h
The red line indicates 90% of teste
system will feel slow to interactive users.
Pipeline software masks environment variable.
Read/Write
Bytes per second
60 GB
40 GB
20 GB
0 B
11/08 00:00
11/11 00:00
11/12 00:00
11/13 00:00
11/14 00:00
unknown 54 GB
team219-grp 3 GB
cellgeni 37 MB
gsu-pipelines 65 kB
ce0 0 B
d 0 B
ddd-grp 0 B
ddd0 0 B
ddd-grp 0 B
ddd 0 B
ddd0 0 B
hgi 0 B
op0 0 B
op0n_target_OTAR20 0 B
ope0 0 B
open_target_OTAR20 0 B
t 0 B
te0 0 B
te0m298 0 B
tea 0 B
tea0 0 B
team 0 B
team112 0 B
team152 0 B
team154-vwork 0 B
team282 0 B
team283 0 B
team298 0 B
team78-grp 0 B
tei0 0 B
teichlab 0 B
Last * Mean
ddd-grp 6 GB 96 MB
unknown 3 GB 17 GB
cellgeni 911 MB 159 MB
open_target_OTAR20 344 MB 33 MB
gsu-pipelines 44 kB 73 MB
ce0 0 B 338 B
d 0 B 59 B
ddd0-grp 0 B 55 B
data I/O (write) (stacked)
40 GB
30 GB
20 GB
10 GB
0 B
cellgeni 0 B 10 MB
ddd-grp 177 MB 9 MB
gsu-pipelines 21 kB 5 MB
open_target_OTAR20 298 MB 2 MB
team112 0 B 67 kB
team219-grp 0 B 8 MB
team283 0 B 9 kB
team298 0 B 9 MB
Stats
get stat ops per second
per second
30 K
20 K
10 K
0
11/08 00:00
11/09 00:00
11/10 00:00
11/11 00:00
11/12 00:00
11/13 00:00
11/14 00:00
team112 0
gsu-pipelines 0
unknown 19

Switching to filesystem stats

malpipe identified using the filesystem

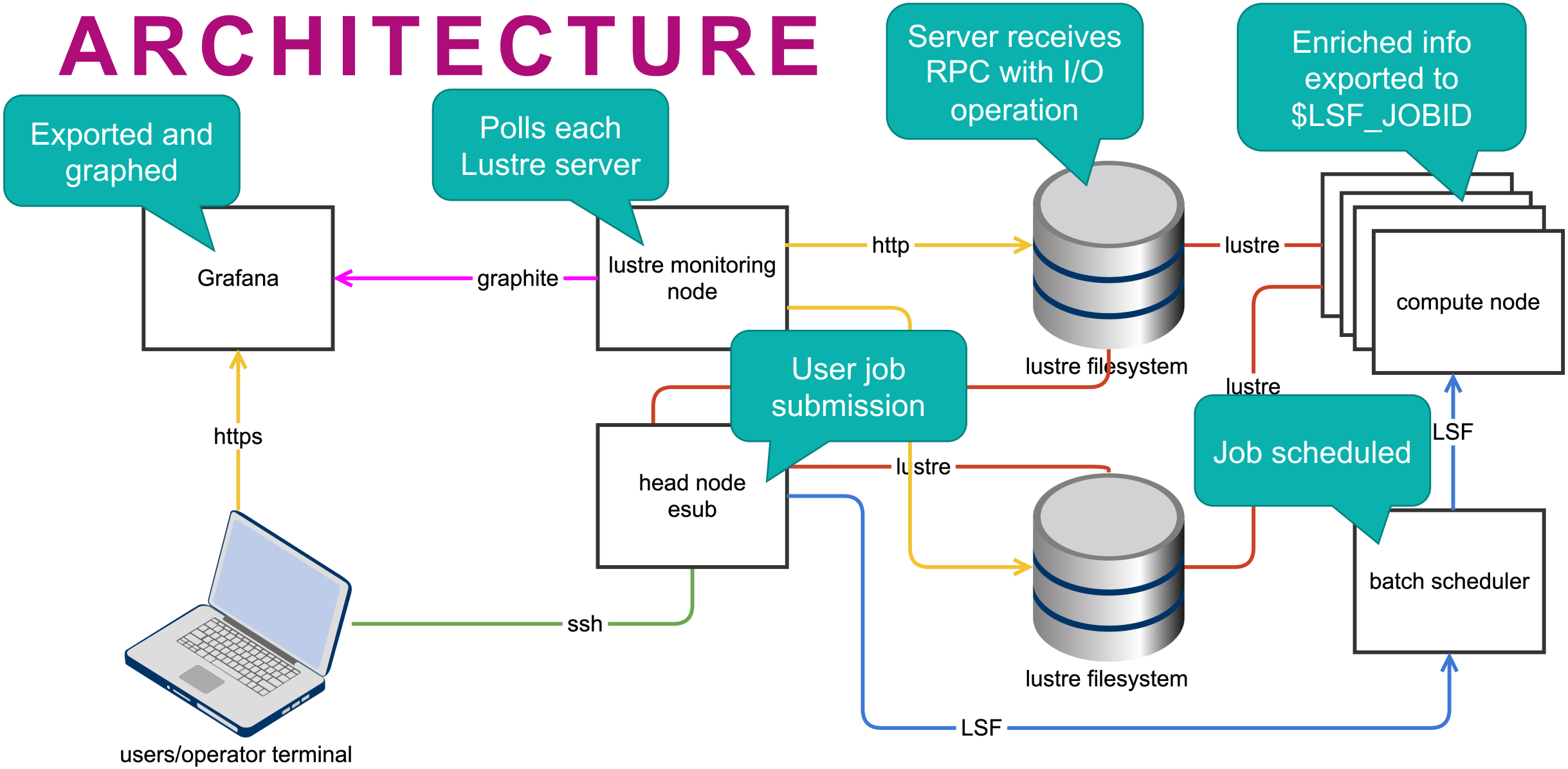
The filesystem is busy

The io is balanced over the OSTs




HOW IT WORKS

ARCHITECTURE



EXPORT JOBSTATS TO GRAPHITE

Parse jobstat output, including enriched information on user, group and more

1. Simple web app for Lustre servers – Return JSON structure with jobstat info

```
def do_GET(self):
      data={}
      self._set_headers()
      osts=map( lambda x: x.replace("/proc/fs/lustre/obdfilter/","",glob.glob("/proc/fs/lustre/obdfilter/lus*"))
      mdts=map( lambda x: x.replace("/proc/fs/lustre/mdt/","",glob.glob("/proc/fs/lustre/mdt/lus*"))
      for ost in osts:
        data[ost]=run_lctl('lctl get_param obdfilter.{0}.job_stats'.format(ost).split())
      for mdt in mdts:
        data[mdt]=run_lctl('lctl get_param mdt.{0}.job_stats'.format(mdt).split())
      self.wfile.write(json.dumps(data))
```

2. Parse the jobstat output to create our data structure

```
fix_jobname = re.compile(r'^- job_id:(\s*)(\S(?:.*\S)?)\s*$')

def process_line(line):
  # Quote the job_id
  line = fix_jobname.sub(r'- job_id:\1"\2"',line)
  # Ensure valid yaml with a space
  return line.replace("max:", "max: ").replace("min:", "min: ").replace("sum:", "sum: ").replace("samples:", "samples: ").replace("sumsq:", "
sumsq: ")

def run_lctl(cmd):
  p = Popen(cmd, stdout=PIPE)
  # skip the first two headers lines
  # https://stackoverflow.com/questions/15571137/yaml-scanner-scannererror-while-scanning-a-directive
  data_stream = "\n".join(map(process_line,p.communicate()[0].split("\n")[2:]))
  return  yaml.load(data_stream)
```

3. Encode & Decode the user, group, and other extra info into the Job ID

```
if var_has_contents("LSB_SUB_MODIFY_ENVFILE"):
  with open(os.environ["LSB_SUB_MODIFY_ENVFILE"], "a") as modfile:
    header = "SA1"
```

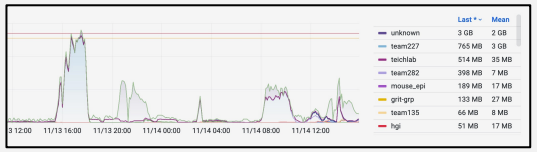
4. Decode the Job ID to recover user, group, and other extra info

```
def decode_jobname(name,store):
  store['project']="unknown"
  store['uid']="unknown"
  store['pgid']="unknown"
```

5. Example JSON output, including enriched data – import directly into Grafana!

```
curl -s  http://lus23-oss1-mgmt.internal.sanger.ac.uk:8080  | jq .
{
  "lus23-OST0001": [
    {
      "write_bytes": {
        "max": 16777216,
        "sum": 1277816778962,
        "samples": 413738,
        "unit": "bytes",
        "min": 4
      },
      "pgid": "15152",
      "job_id": "SA1TwDkwOzA7,team135",
      "uid": "14784",
…
      "project": "team135",
      "gid": "15152",
      "farm": "tol",
      "read_bytes": {
        "max": 16777216,
        "sum": 51752874618880,
        "samples": 22381014,
        "unit": "bytes",
        "min": 4096
      }
    ]
}
```

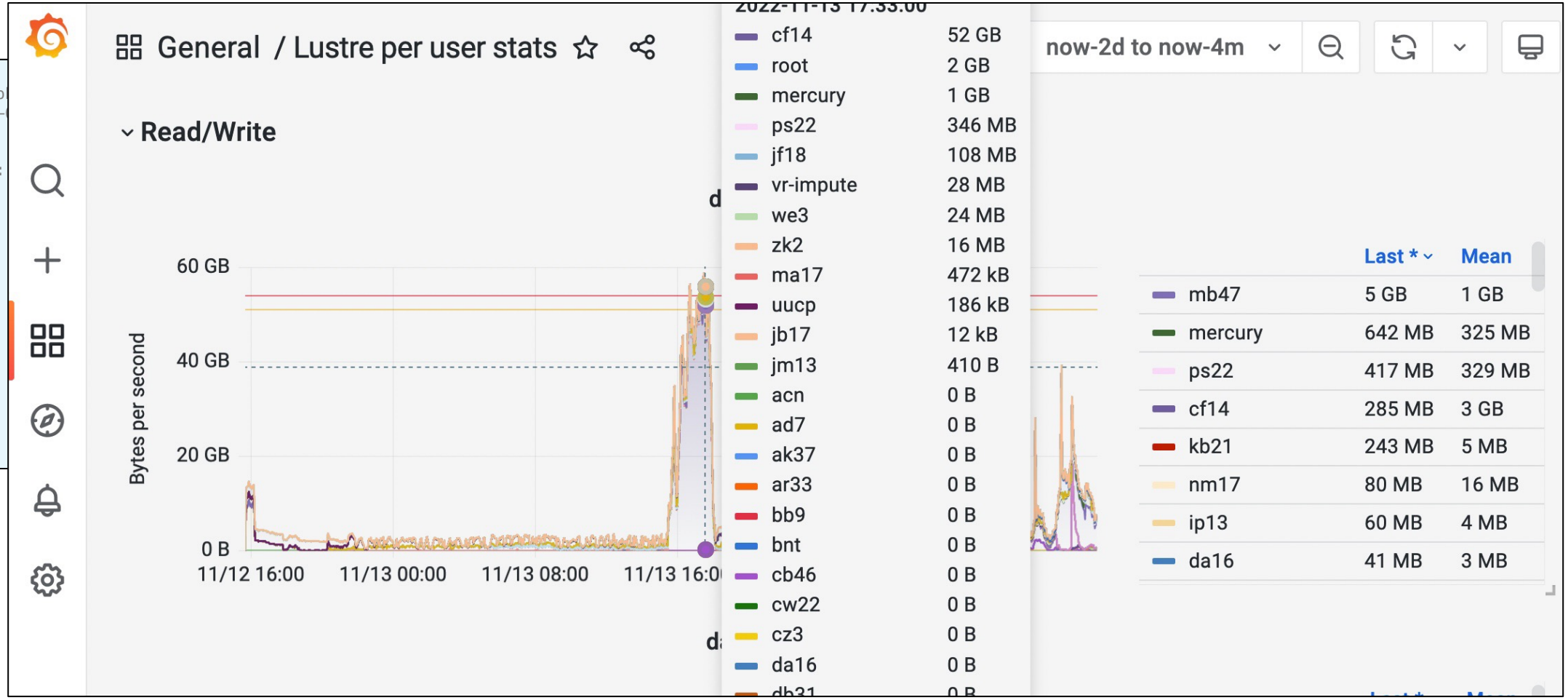
Enriched info exported to Graphite



EXPORT JOBSTATS TO GRAPHITE

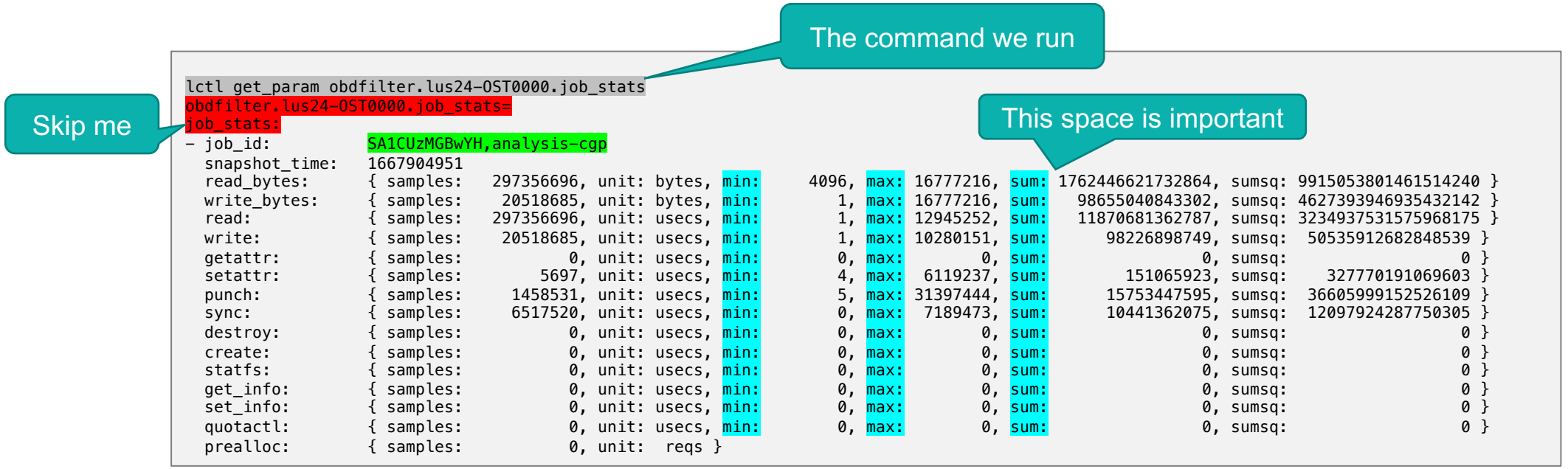
Parse jobstat output, including enriched information on user, group and more

```
lctl get_param o
obdfilter.lus24-
job_stats:
- job_id:
  snapshot_time:
  read_bytes:
  write_bytes:
  read:
  write:
  getattr:
  setattr:
  punch:
  sync:
  destroy:
  create:
  statfs:
  get_info:
  set_info:
  quotactl:
  prealloc:
```



JOBSTATS: CARE NEEDED

- Raw data is fetched in grey with lctl get_param
- **LU-16110**, **LU-13857** – jobs_stats not valid yaml
- **LU-15870** – job_stats is sometimes corrupt
- **LU-15826** – job_id not quoted



JOBSTATS: CONFUSION

- We see corruption in the job id, LU-15870, LU-16251
- We see kworker recorded as the executable with root.
- We see thread-pool with no uid.

Two IO operations wrote to the same slot. Meaning the data is corrupt.

Not a user process

Not a user process

```
- job_id:          kworker/u518:6eamtrynka
  snapshot_time:   1668009798
  read_bytes:      { samples:           1, unit: bytes, min:  663552, max:  663552, sum:          663552 }
  write_bytes:     { samples:           0, unit: bytes, min:       0, max:       0, sum:               0 }

- job_id:          kworker/u130:0
  snapshot_time:   1668009796
  read_bytes:      { samples:       21459, unit: bytes, min:    4096, max: 2297856, sum:       160104448 }
  write_bytes:     { samples:           0, unit: bytes, min:       0, max:       0, sum:               0 }

- job_id:          thread-pool-14113912
  snapshot_time:   1668009775
  read_bytes:      { samples:           2, unit: bytes, min:   30864, max:  983040, sum:         1019904 }
  write_bytes:     { samples:           0, unit: bytes, min:       0, max:       0, sum:               0 }

[root@lus23-oss1 ~]# lctl get_param obdfilter.lus23-OST0000.job_stats | grep job_id | grep thread-pool | wc
-l
126
[root@lus23-oss1 ~]# lctl get_param obdfilter.lus23-OST0000.job_stats | grep job_id | grep kworker | wc -l
155
[root@lus23-oss1 ~]# lctl get_param obdfilter.lus23-OST0000.job_stats | grep job_id | wc -l
330
```

SECURE LUSTRE

- A method of delivering lustre filesystems to OpenStack projects using
  - Sub directory mounts
  - And user id mappings
- https://hpc-news.sanger.ac.uk/2021/05/20/update-on-secure-lustre-lustre-users-group-may-2020/
- Jobid is inserted on the client, the uid inserted may and often does not exist on the outside
- The nodemap the request is received on is unavailable in the jobstats data
- Our current aggregation program appears to drop this traffic.

CONCLUSIONS

- Small change on Lustre server
- Not perfect
- Still useful
- Released under Apache 2 license

- https://github.com/wtsi-ssg/lustre-jobstats

FOR THE FUTURE

- We need users to have a “fair share” for I/O
- Is NRS the answer to that question?

ACKNOWLEDGMENTS

- ISG (Photo is missing Anna, Bruno, Fabio and Sai) (And includes Brett who is no longer with us).

THANK YOU

@wellcomegenomecampus

@wellcomegenome

Wellcome Genome Campus

Wellcome Genome Campus