# Best practices when accessing Big Data … or any other data!

Dr Rosemary Francis
CEO Ellexus: The I/O profiling company

In this talk I will explore best practices when accessing data on local or shared file systems. I will use examples of what can go wrong taken from real customer problems to back up how simple guidelines and good use of available tools can make a massive difference to the performance, reliability, scalability and portability of your code. There are free and commercial tools that can help, but they need to be combined with good coding and good working practices. Data doesn't have to be big to cause a problem, but as data sets grow, the way we access data has never been a more important consideration. Our customers work in scientific and high-performance computing with different trade-offs to make between time-to-market, reliability and performance, but what they all have in common is that they have to care about I/O.

# Ellexus Ltd: The I/O Profiling Company

**Products:** We make tools to help you

- improve application performance,
- protect shared storage, and
- manage application dependencies.

**Customers include:**

TACC TEXAS The University of Texas at Austin diamond

wellcome sanger institute DELL EMC

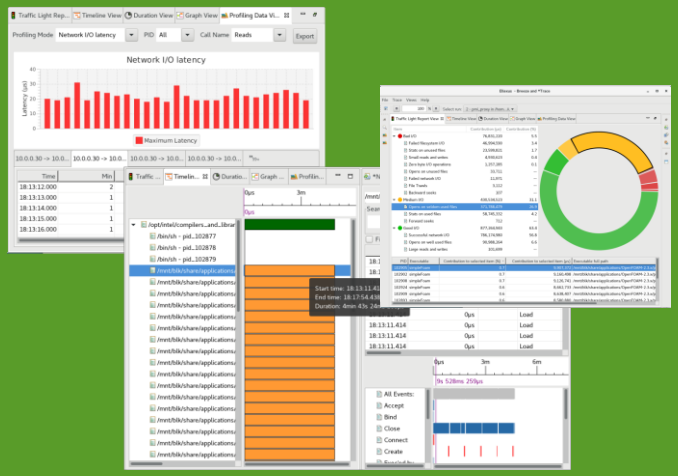
ANSYS arm SYNOPSYS®

# Ellexus enterprise products

## Take control of the way you access your data

breeze

Detailed I/O profiling
Application discovery




**Dependencies**
What do I need to include in my container?

**I/O profiling**
What resources do I need to run it?

**Debug and triage**
Why am I not getting the results I expect?

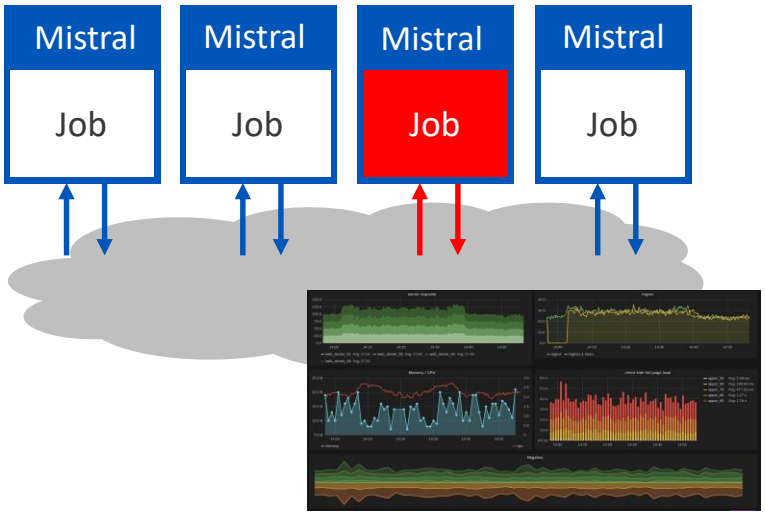
# Ellexus enterprise products

## Take control of the way you access your data

Protect storage from rogue jobs

Find bottlenecks in production

Chargeback and procurement




mistral

Live system telemetry:
I/O monitoring in production

# Why profile I/O

**Detect dependencies** for containerization and migration

**Application correctness** in delivery and deployment

**Understand resources** for sizing and procurement of IT resources or cloud

**Profile I/O** for tuning and optimization

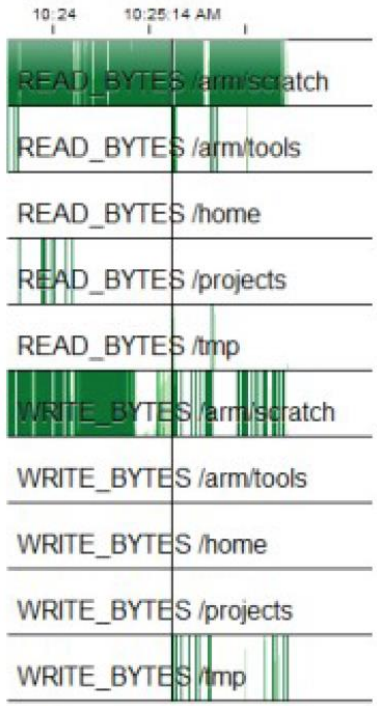
**Monitor I/O** for chargeback and troubleshooting in the field

# The noisy neighbour problem
## Bad performance case study

A small number of jobs can overload shared file systems and cause system bottlenecks.

This software build is overloading shared storage by putting data in the wrong place.

**Example of a rogue job from Arm:**




Temporary data is written to shared storage

Local storage is unused

# Fast, agile and (hybrid) cloud ready

~~Fast, agile and (hybrid) cloud ready~~

# Performance, portability and planning

~~Fast, agile and (hybrid) cloud ready~~

# Performance, portability and planning

**Performance:** Understand I/O patterns and requirements

**Portability:** Understanding application dependencies

**Planning:** System telemetry and monitoring

# Where does bad I/O come from?

Third party tools and libraries

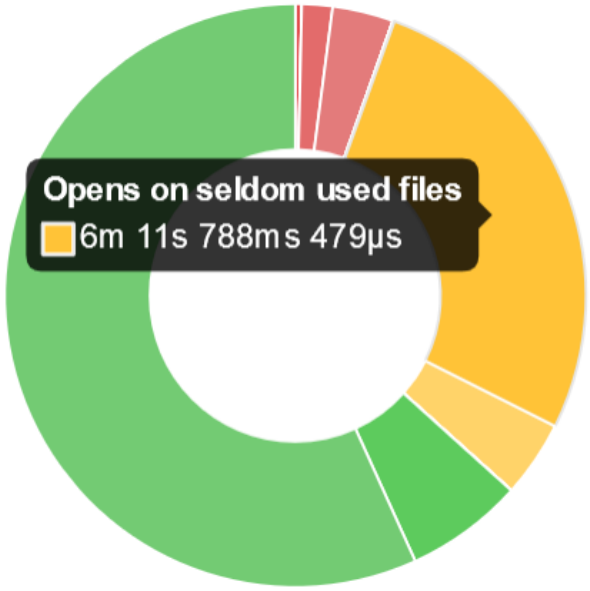
Legacy code

Misunderstandings about IT infrastructure

Changes in your working environment

and occasionally… Bad code and lazy design

# How much time are you wasting doing bad I/O?




- Small reads and writes
- Opens on unused files
- Stats on unused files
- Failed filesystem I/O
- Backward seeks
- File Trawls
- Zero byte I/O operations
- Failed network I/O
- Opens on seldom used files
- Stats on used files
- Forward seeks
- Large reads and writes
- Opens on well used files
- Successful network I/O

breeze

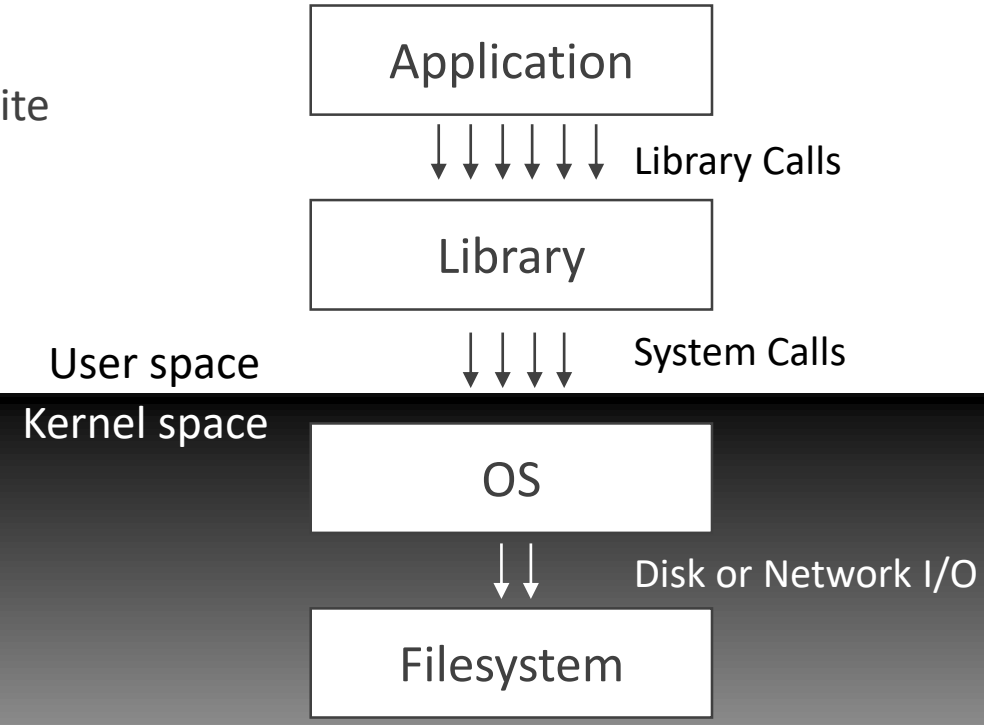
# Small reads and writes are bad for everyone

Libraries may not buffer I/O well

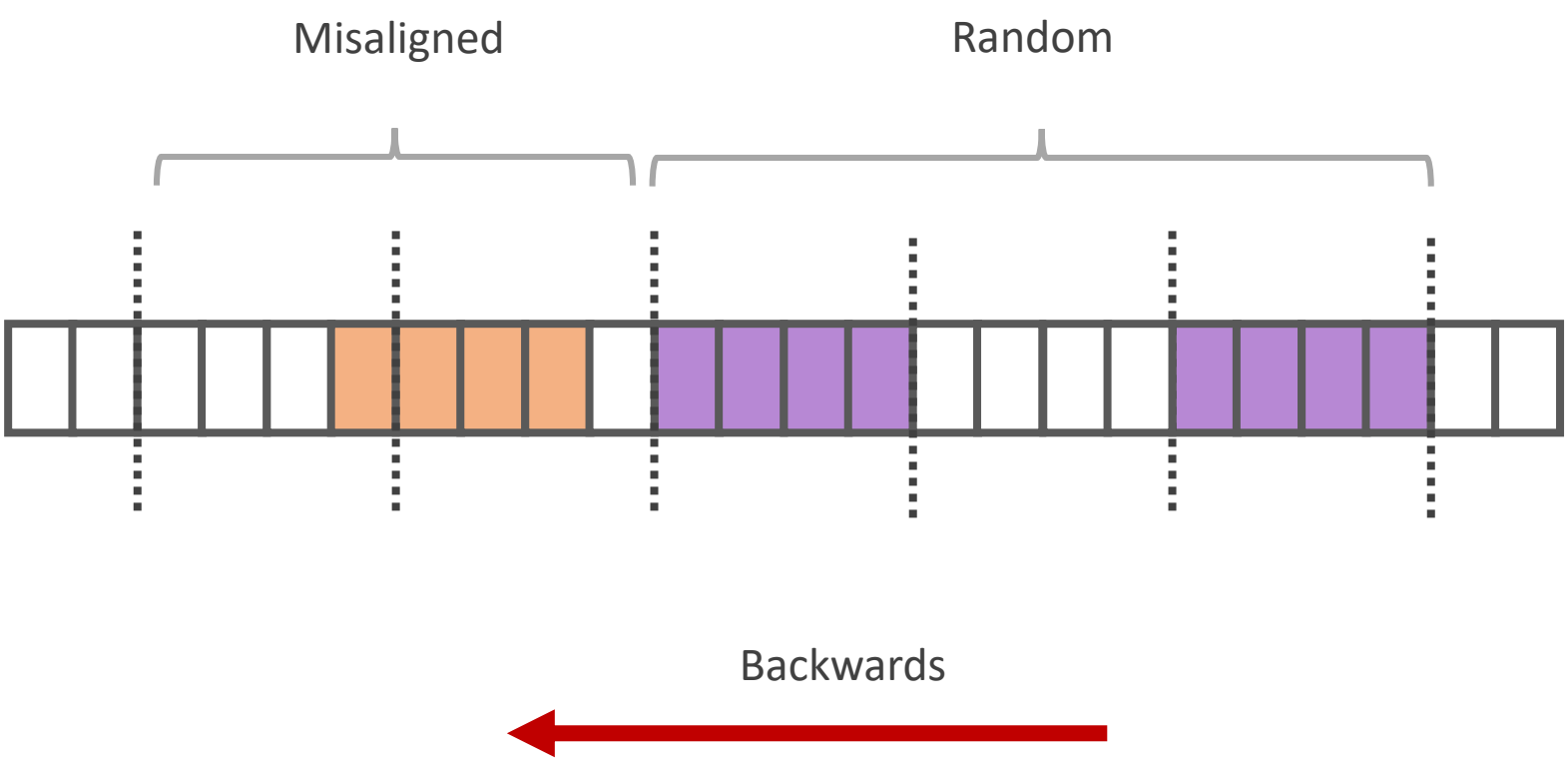
OS may not buffer I/O well

Every read to a shared file system is a write

Spectre and Meltdown!



# Misaligned and random I/O



# Housekeeping and deletes

Delete as you go along – don’t delete everything at the end.

# Failed I/O and file system trawls

Looking for a file? Don’t stat everything to find it

PATH variables

→ Zero tolerance on bad I/O

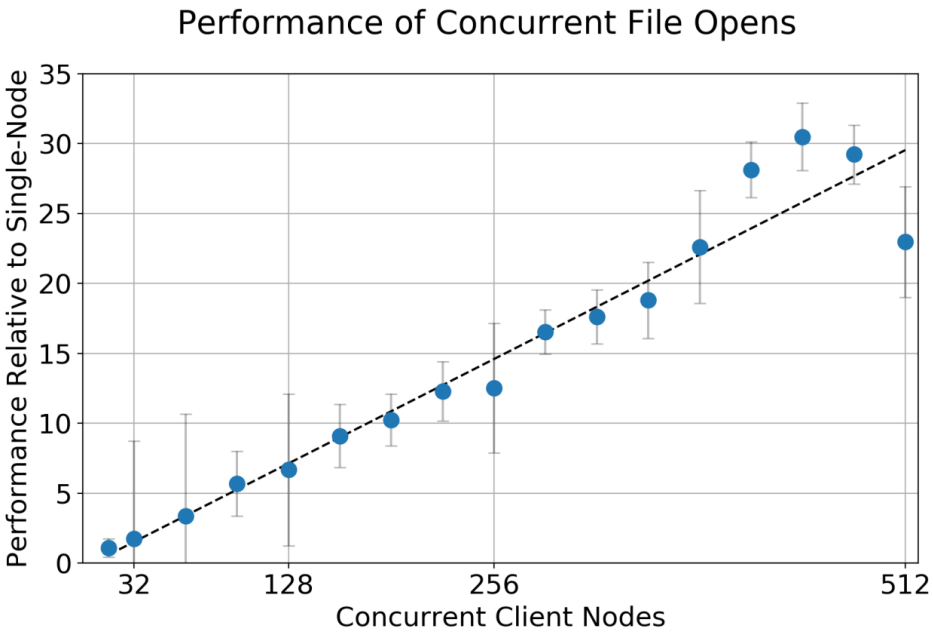
# Opens and closes

Open loops!

```
for(… ) {
      open()
      write()
      close()
}
```

One customer crashed Lustre doing this

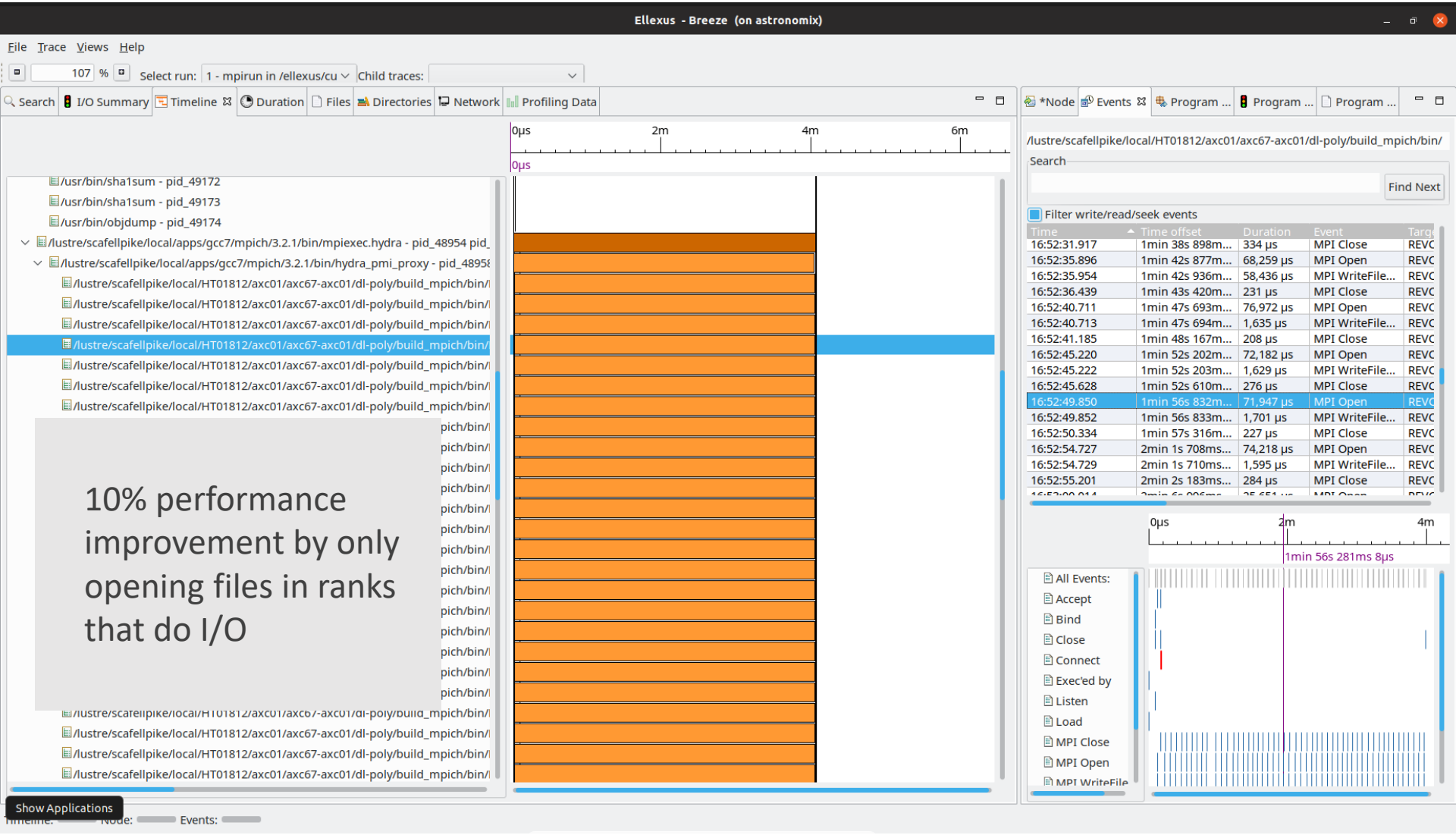
Another got a 700% speed up when they fixed the loop!




Time taken to open a file across a large number of compute nodes

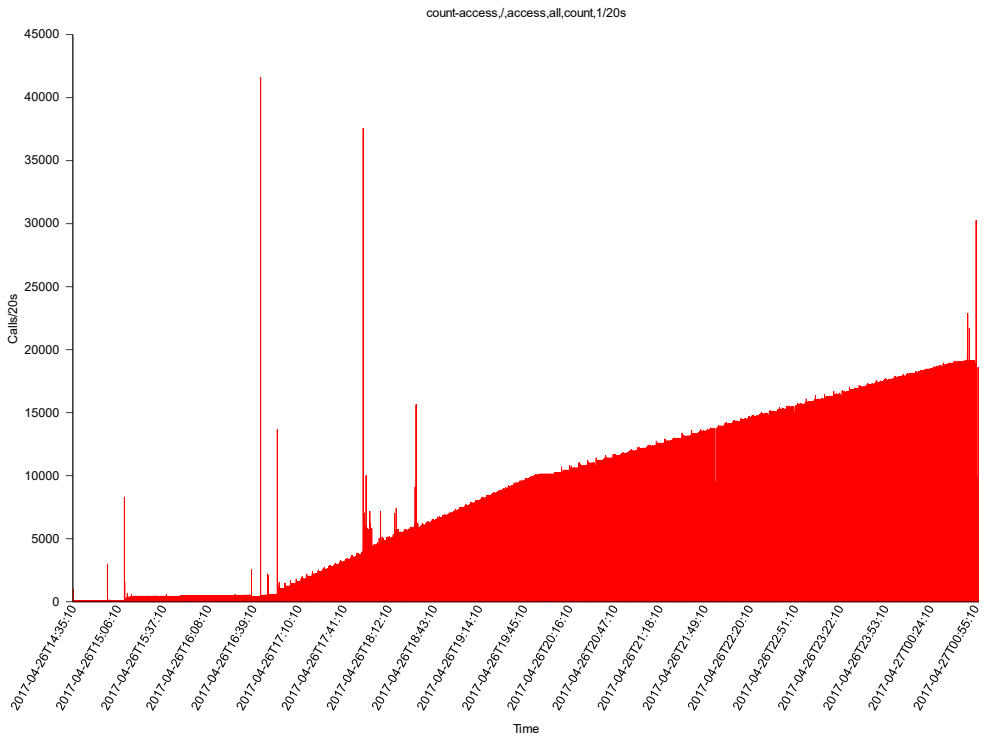
Credit: Glenn Lockwood, Next Platform

# Opens and closes

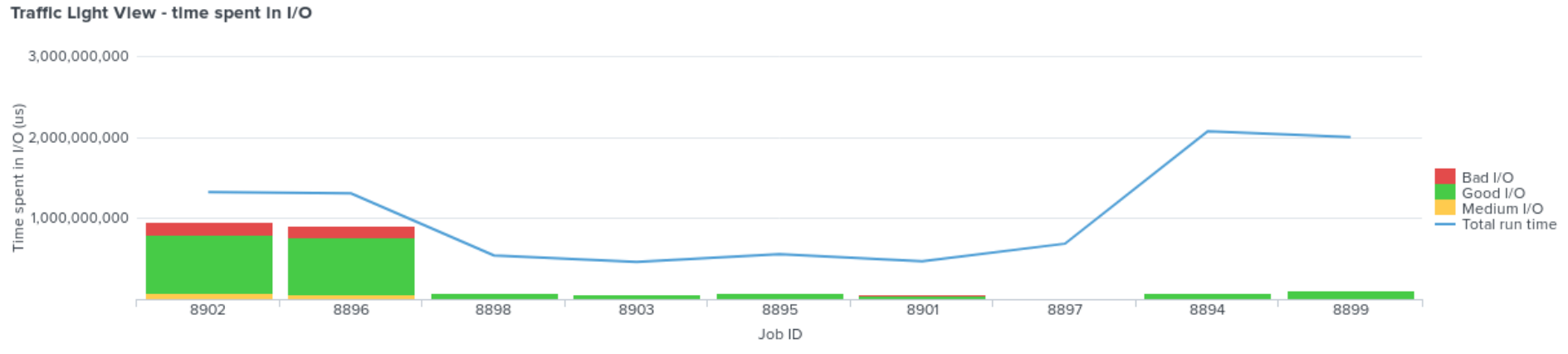


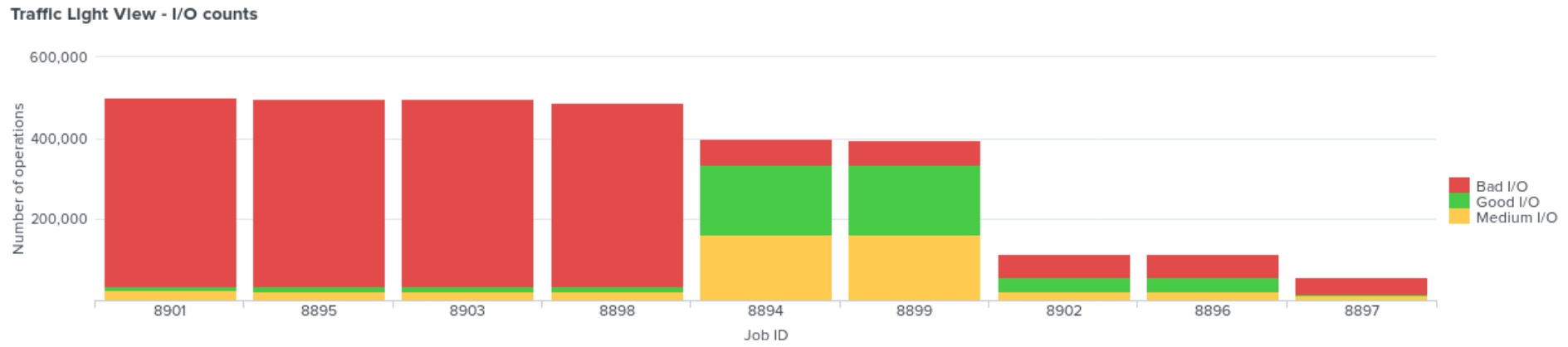
# Meta-data accesses
# Case study: genome pipeline

Stat calls are used to track application progress

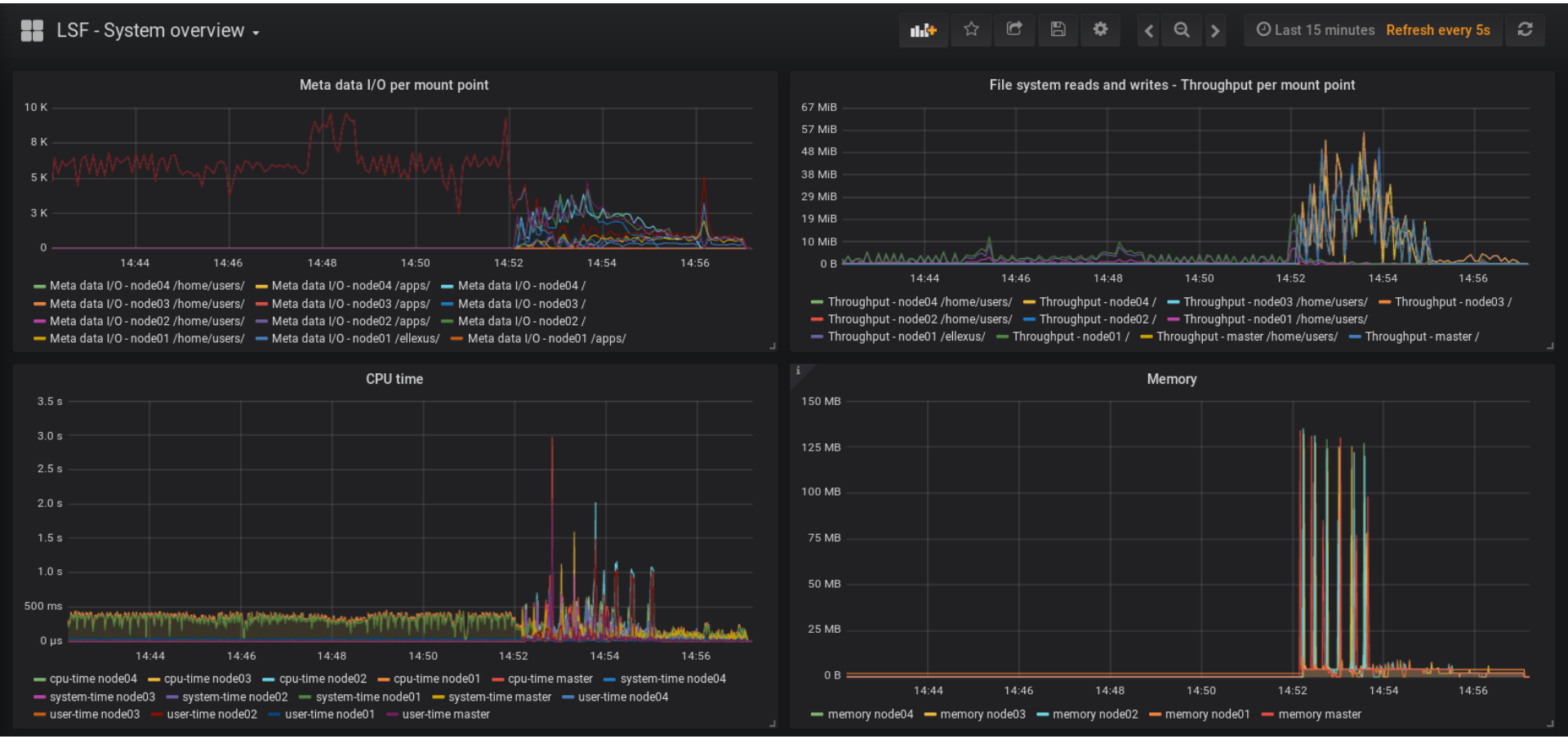


# How much time are you wasting doing bad I/O?



# What is normal?



# Tuning cancer pipelines at the Sanger Institute

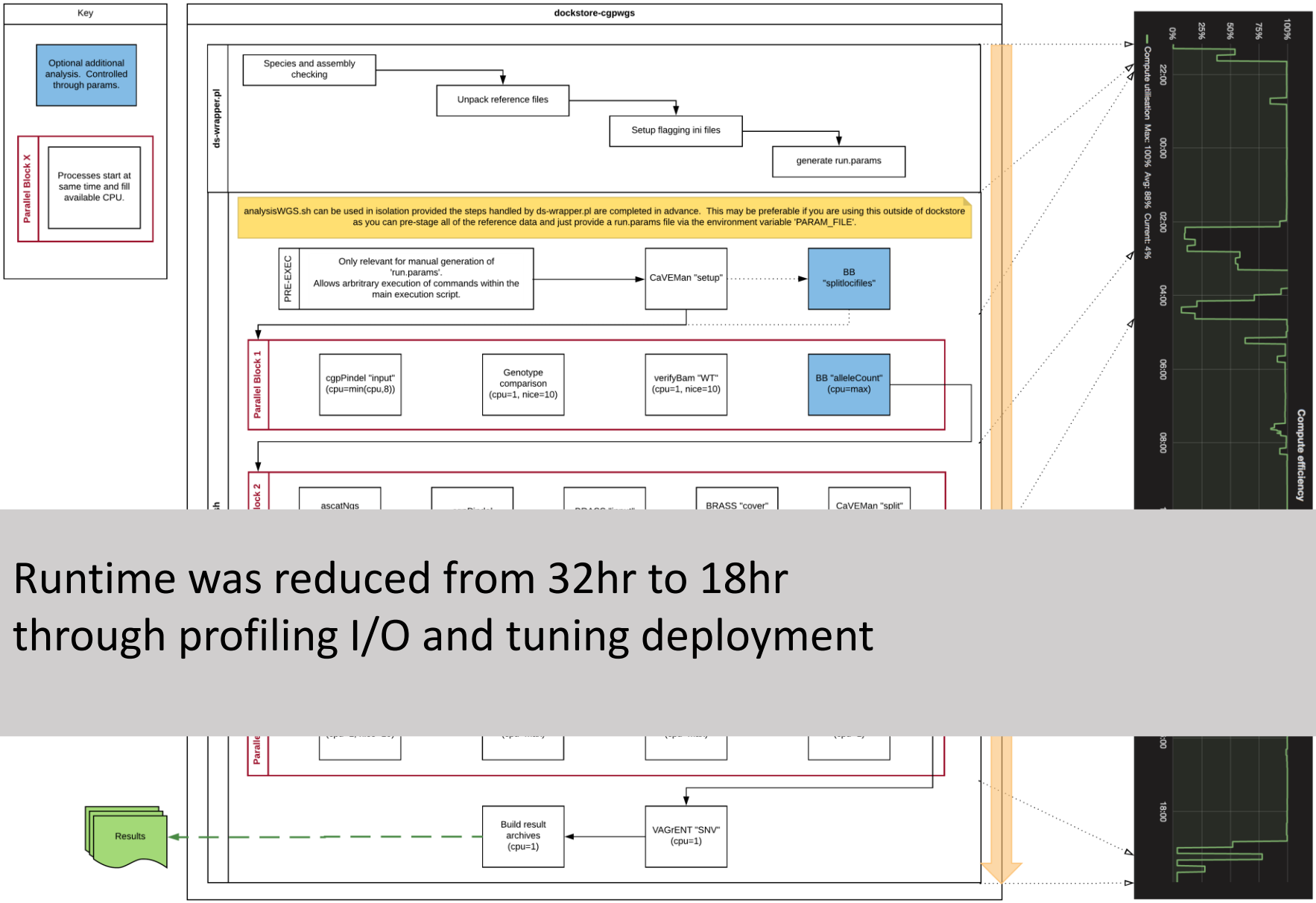
The Pancancer project: 2000 whole genomes at multiple HPC sites

→ Containerised pipelines for portability

→ I/O tuned with Ellexus tools

→ Storage now needs to be sized correctly

# Tuning cancer pipelines at the Sanger Institute

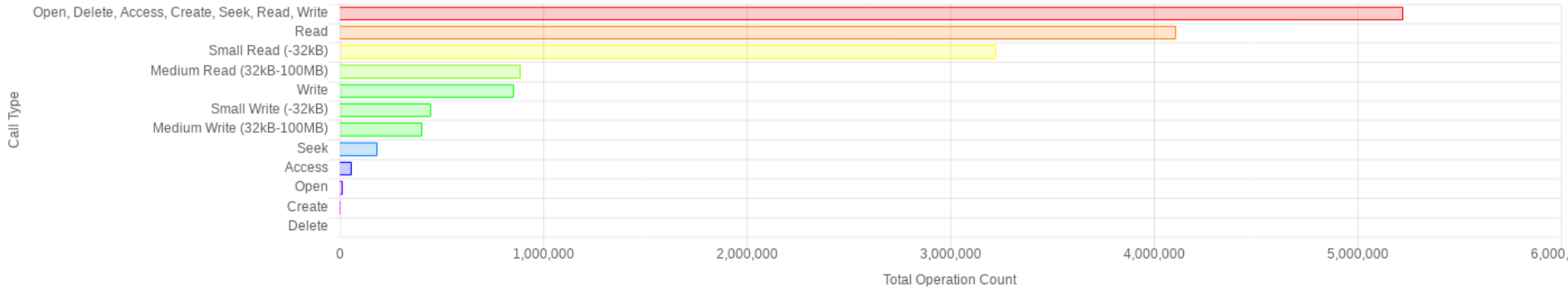



Runtime was reduced from 32hr to 18hr through profiling I/O and tuning deployment

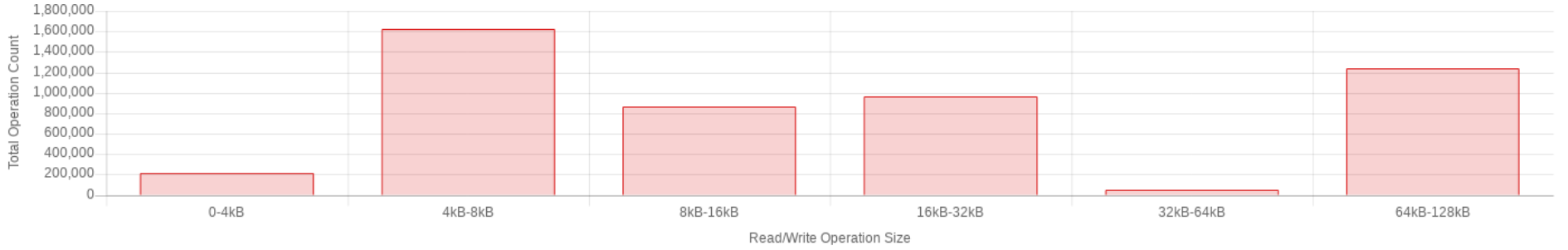
# Profiling the cancer pipeline

## AWS m5.xlarge 4vCPU 16GB

Number of I/O operations() by type




Size of read and write operations()



# Storage comparison

| | Time* | | Cost per month | |
|---|---|---|---|---|
| GP2 | 52m 23s | 100% | 174.11 | 100% |
| Magnetic EBS | 1h 01m 44s | 118% | 174.43 | 100% |
| Provisioned 100 IOPS | 1h 42m 01s | 195% | 184.61 | 106% |
| Throughput optimised HDD | 1h 19m 32s | 152% | 189.01 | 109% |
| 150GB NVMe | 51m 27s | 98% | 191.79 | 110% |
| Provisioned 500 IOPS | 54m 22s | 104% | 215.01 | 123% |

⇒ The Provisioned IOPS SSDs performed very badly

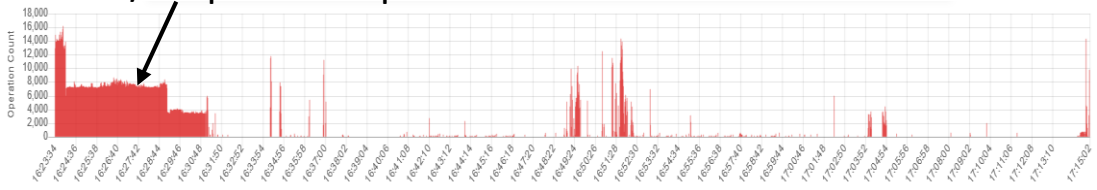
⇒ AWS default option, GP2 is the best

⇒ NVMe was only 2% faster for a 10% price increase

# I/O Operations() over time

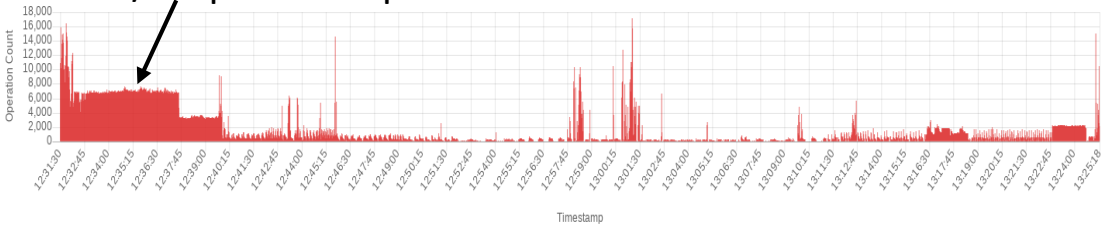
NVMe SSD

~7k I/O operations per second



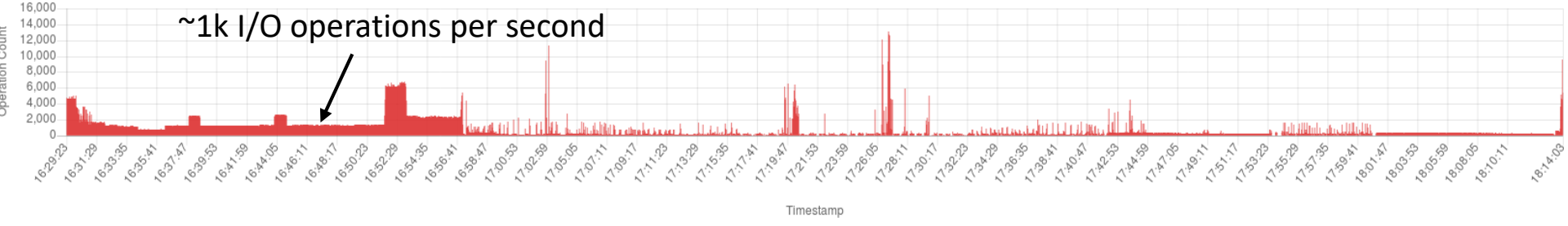

GP2 SSD

~7k I/O operations per second



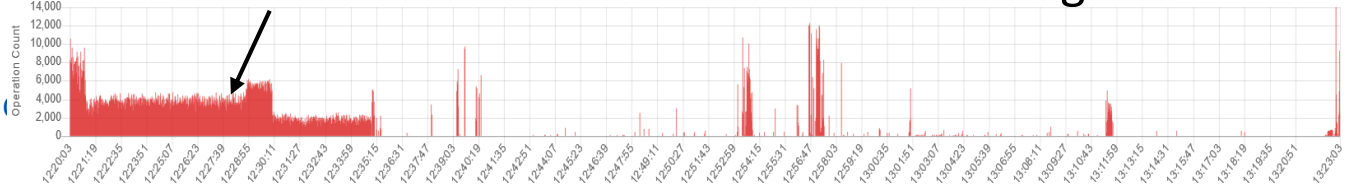

Provisioned 100IOPS SSD

~1k I/O operations per second



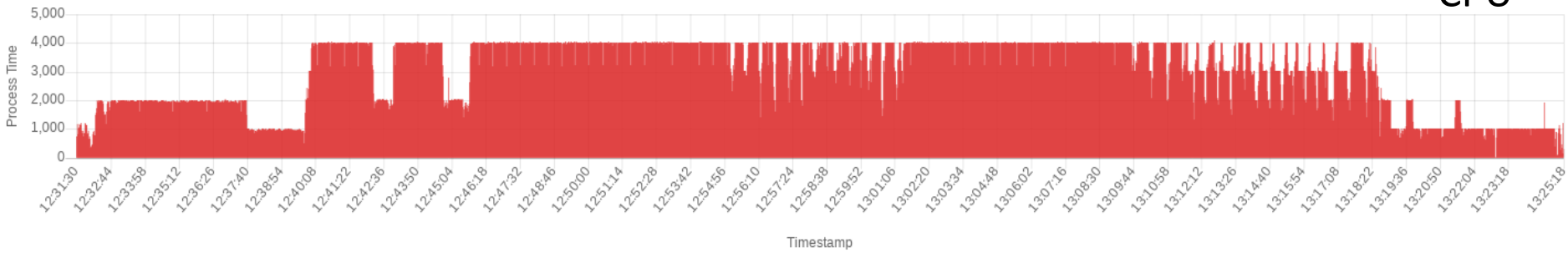

~4k I/O operations per second

Magnetic HDD

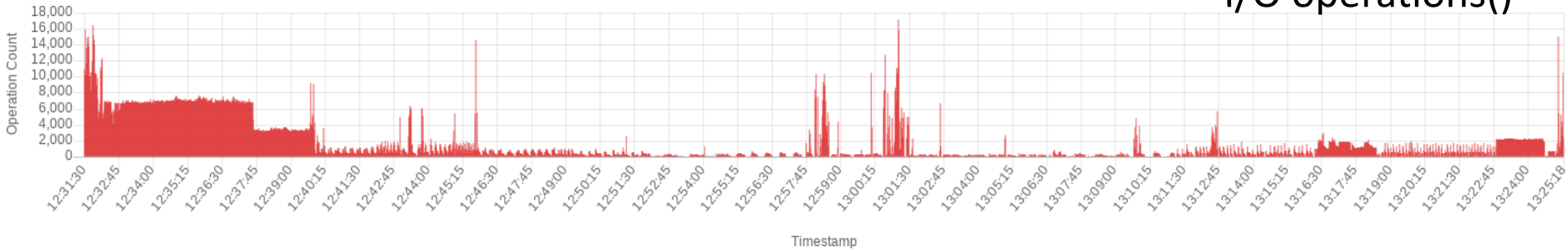


# CPU and I/O Profile (on GP2 SSD)

CPU

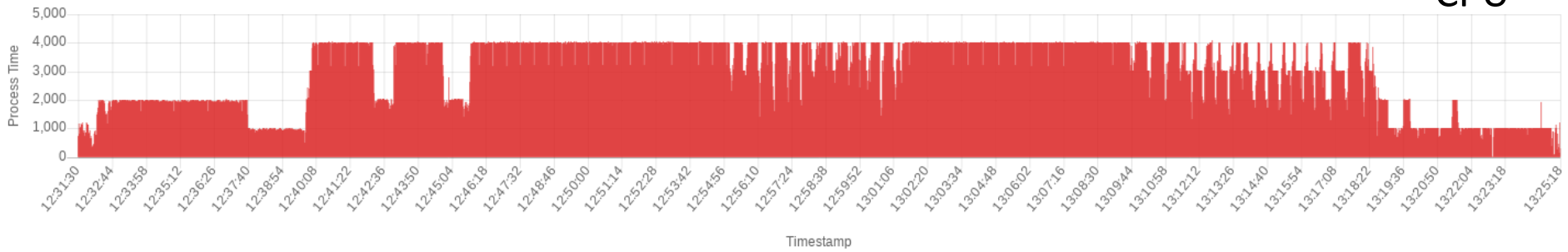



I/O operations()

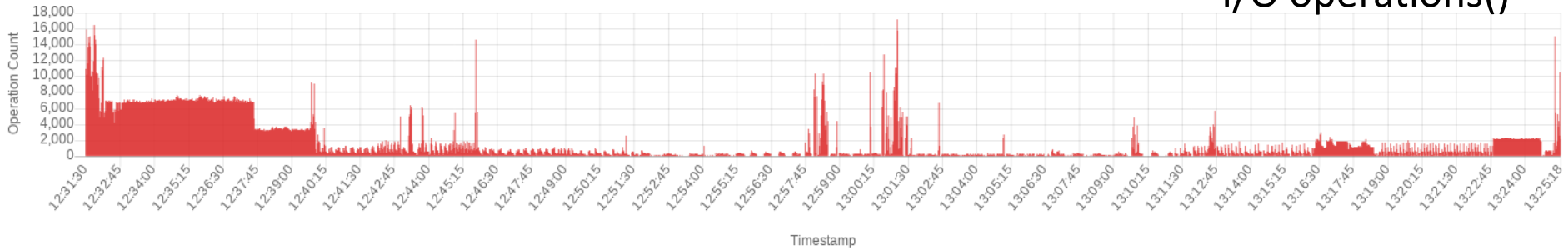


# CPU and I/O Profile (on GP2 SSD)

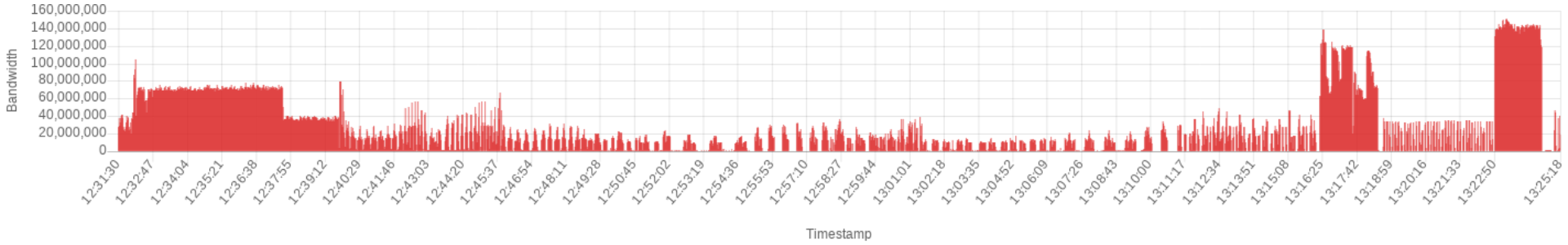
CPU




I/O operations()




Read bandwidth

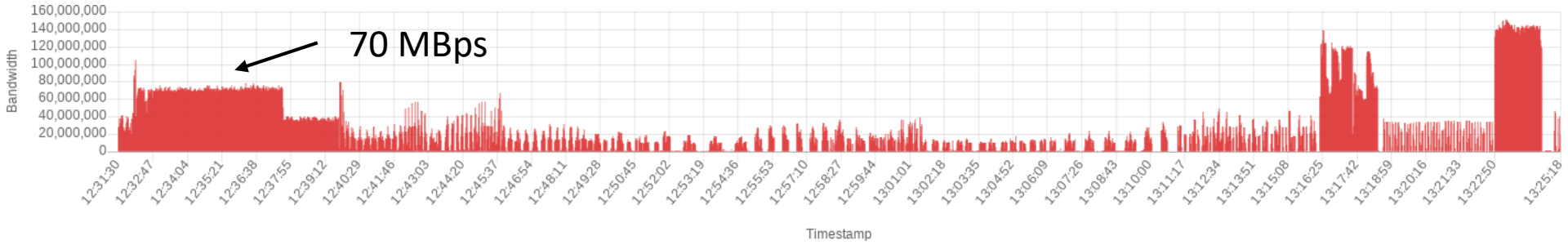


# More CPU and less memory: m5.xlarge vs c5.xlarge
## (still on GP2 SSD default storage)

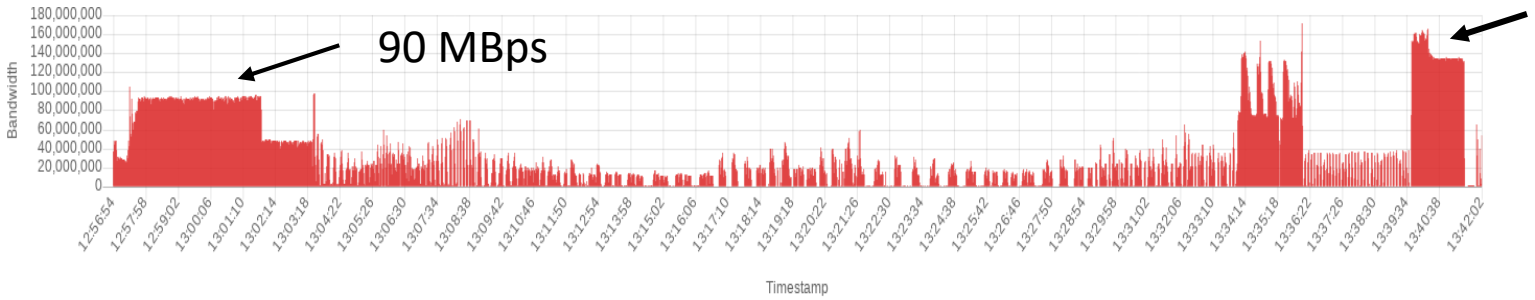
| **M5.xlarge** | **c5.xlarge** |
| --- | --- |
| 4 vCPU | 4 vCPU |
| 16GB | 8GB |
| Runtime: 53min | Runtime: 44min |
| Cost: $0.21 | Cost: $0.16 |

# Read bandwidth: m5.xlarge vs c5.xlarge

Read bandwidth for mx.large 16GB




Read bandwidth for cx.large 8GB




I/O limited by running out of AWS burst credits at the end

# How long did this work take?

Tuning the pipeline took a lot of effort

… but runtime went from 32hr to 18hr

Sizing the storage and compute correctly took three days

… and we saved >10% of cloud costs for the project

“Improving run time often doesn't require extensive rewrites. Knowing where to look is key.”

Keiran Raine, Cancer researcher, Sanger Institute

# Lessons learnt

⇒ Containers make it easy to deploy applications

⇒ Optimising I/O can save you money

⇒ Need to understand all variables to find bottlenecks:

- ⇒ CPU
- ⇒ memory
- ⇒ I/O patterns
- ⇒ I/O performance

# Ellexus best practices for good I/O

Dependencies as the number one check
- Wrong libraries
- Wrong config files
- Changes need to be checked by a human

Regression testing for I/O behaviour

Zero tolerance on bad I/O

Test in production - Often problems are in set up scripts

Tuning and optimisation - if you have time

# Free and open source I/O tools

Strace
- System call tracing

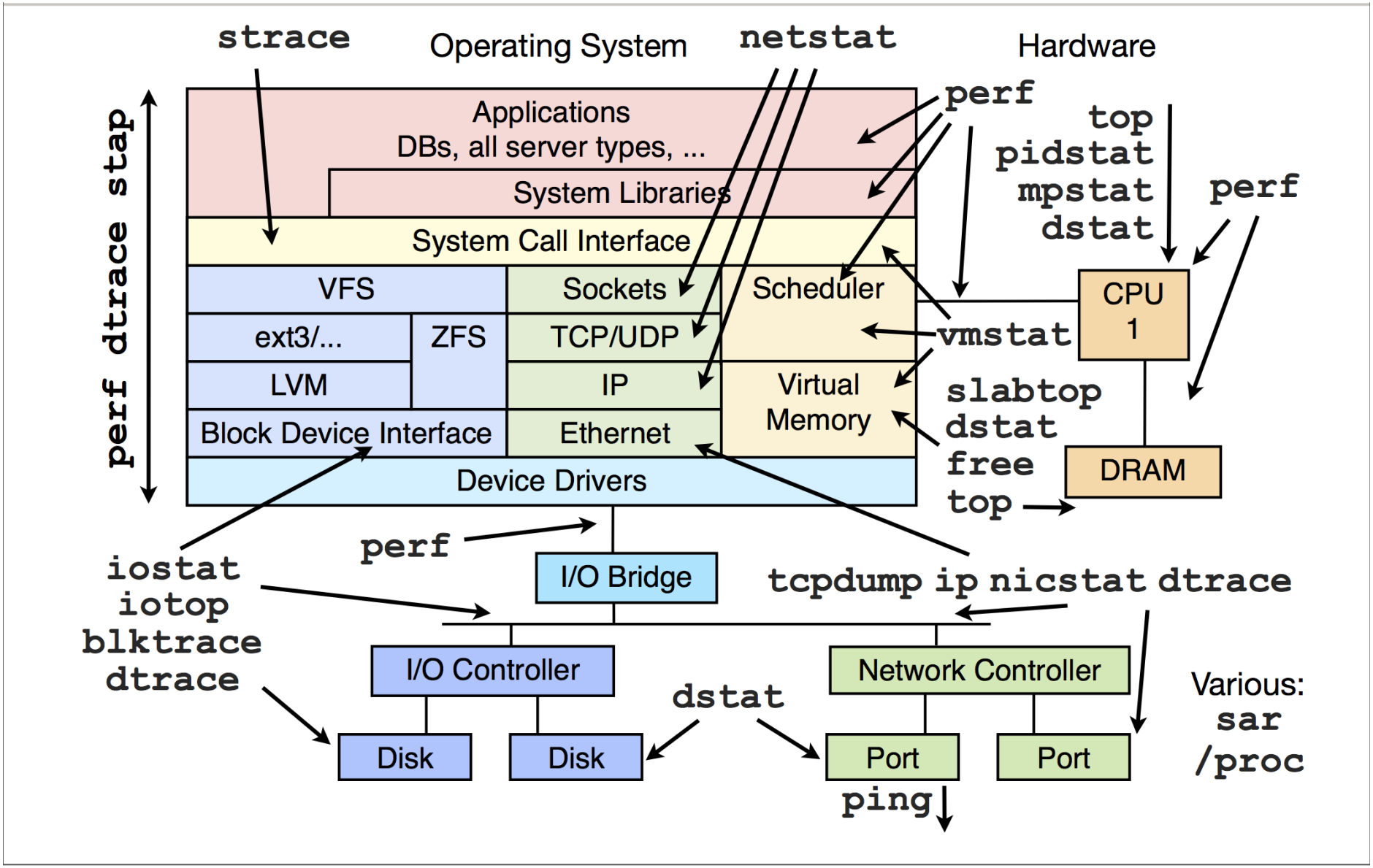
Darshan
- For MPI I/O profiling

iotop
- Like top, but for disk I/O

XALT
- Basic dependency tracking on distributed systems

Perf
- Low level I/O information

# Brendan Gregg's guide to free and open source I/O tools



# Contact us!

Our tools are trusted by research organisations, financial institutions, semiconductor companies and software vendors around the globe.

- Take control of your I/O on-premise and in the cloud
- Whole-system monitoring with APM solutions
- Detailed dependency analysis and bottleneck resolution

Proven to improve performance, increase up time and keep your customers happy.

Dr Rosemary Francis
CEO and director of technology
rosemary@ellexus.com