

Catching rogue jobs before they overload shared storage

Dr Rosemary Francis, CEO and Co-founder



The I/O Profiling Company - Protect. Balance. Optimise.

www.ellexus.com

Who we are

Ellexus is the I/O profiling company.

The company was founded to solve I/O and dependency problems for semiconductor design flows, but we now work across the HPC space in life sciences, cancer research, automotive, oil and gas etc.

Our latest tool Mistral has been developed with ARM to protect shared storage from bad data access patterns on large compute clusters.



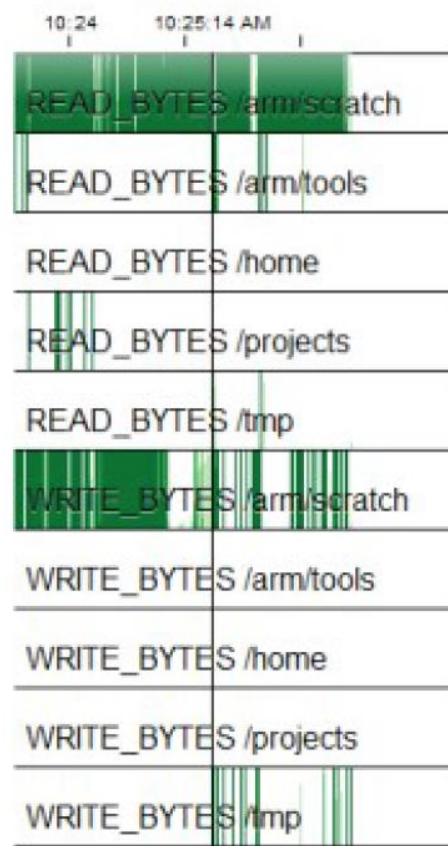
Catching rogue jobs before they overload shared storage
www.ellexus.com

Mistral:

Solving the noisy neighbour problem

In a compute cluster it is possible for a small number of jobs to overload the shared file system. **Example of a rogue job from ARM:**

Lots of data is written to remote storage (/scratch).



But almost nothing is written to local storage (/tmp).



Mistral:

Protecting you against rogue jobs

Hotspots in your cluster can be caused by different types of I/O:

Too much data

A user accidentally submitted 1,000 jobs with the debug flag left on.

They generated too much data and crashed the file system.

Millions of meta data operations

One user wrote:

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

This crashed the meta data server.

Too many small files

One user accidentally created a million zero-length files.

The file system crashed when they were created and then again when they were deleted.



Mistral

Protect your storage from bad I/O patterns and rogue jobs.

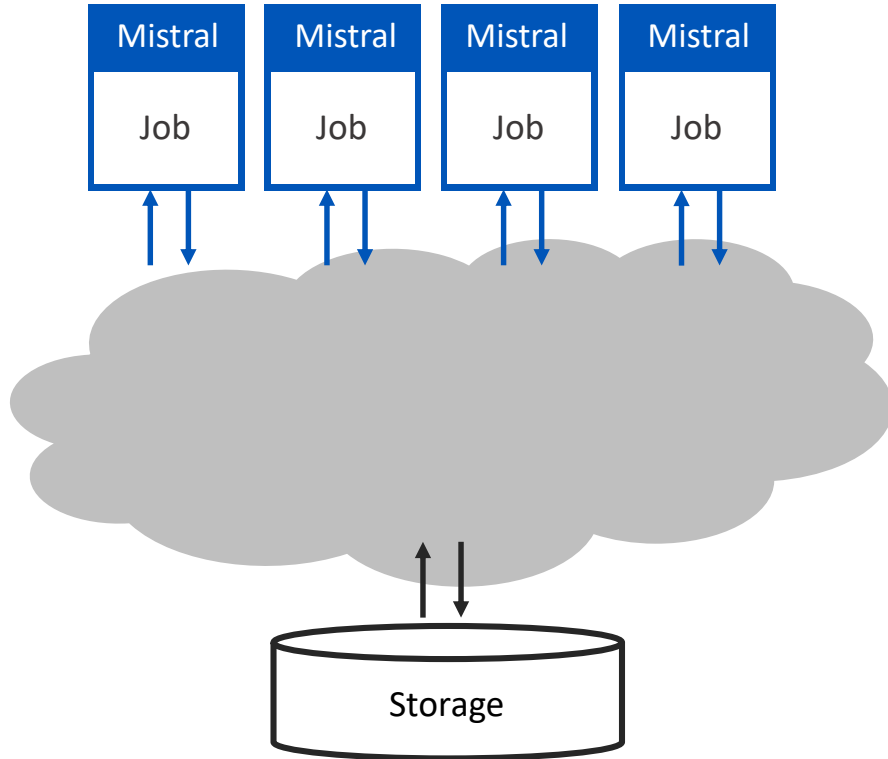
Load **balance** your storage for good quality of service.

Optimise your applications for the storage and your infrastructure for your jobs.



Mistral:

How does it work?



Mistral sits at the application level.

The technology wraps every job and intercepts or controls all I/O calls.

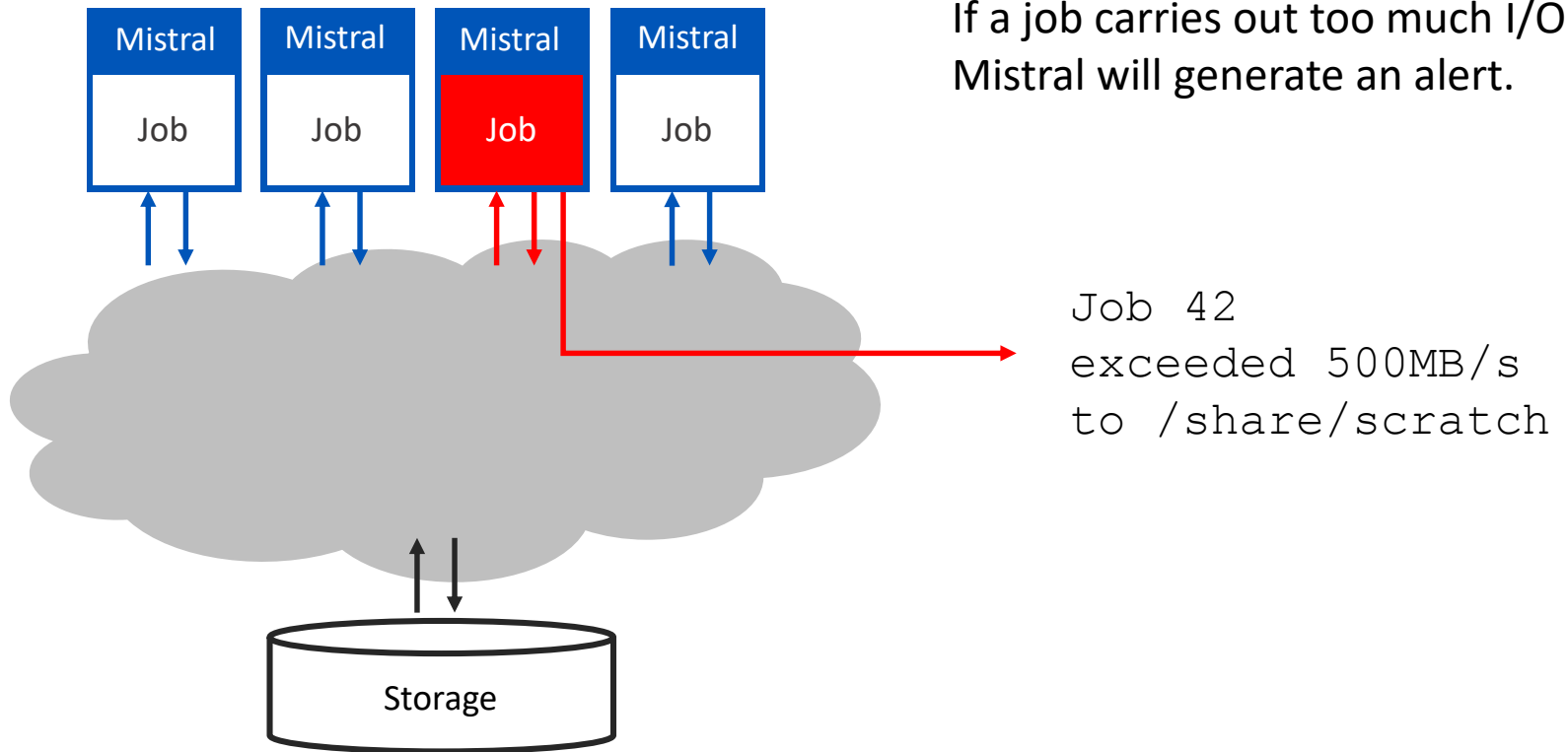
Mistral is launched with the job directly or via the scheduler. This can be invisible to users.

```
> mistral mycommand.sh
```



Mistral:

How does it work?

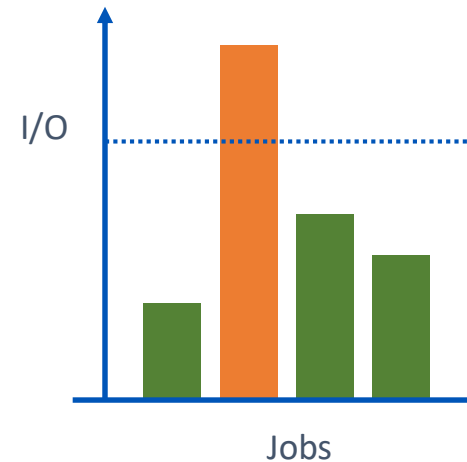


Mistral:

Advanced systems management

Use I/O data from Mistral to:

- Identify I/O hotspots and rogue jobs
- Locate bad I/O patterns
- Educate users

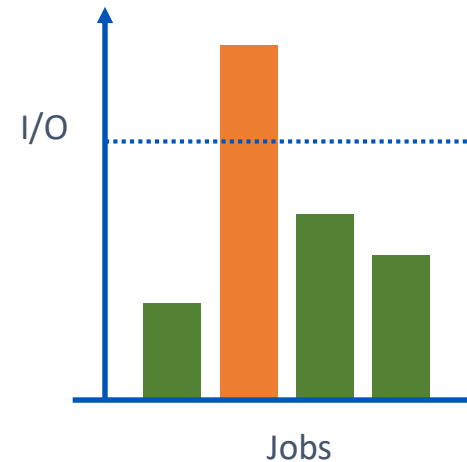


Mistral:

Advanced systems management

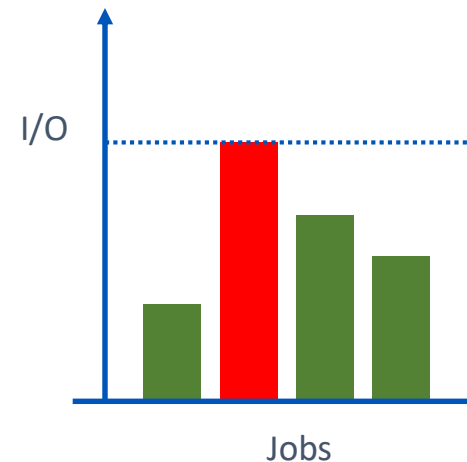
Use I/O data from Mistral to:

- Identify I/O hotspots and rogue jobs
- Locate bad I/O patterns
- Educate users



Use Mistral I/O throttling controls to:

- Improve quality of service
- Load balance shared storage

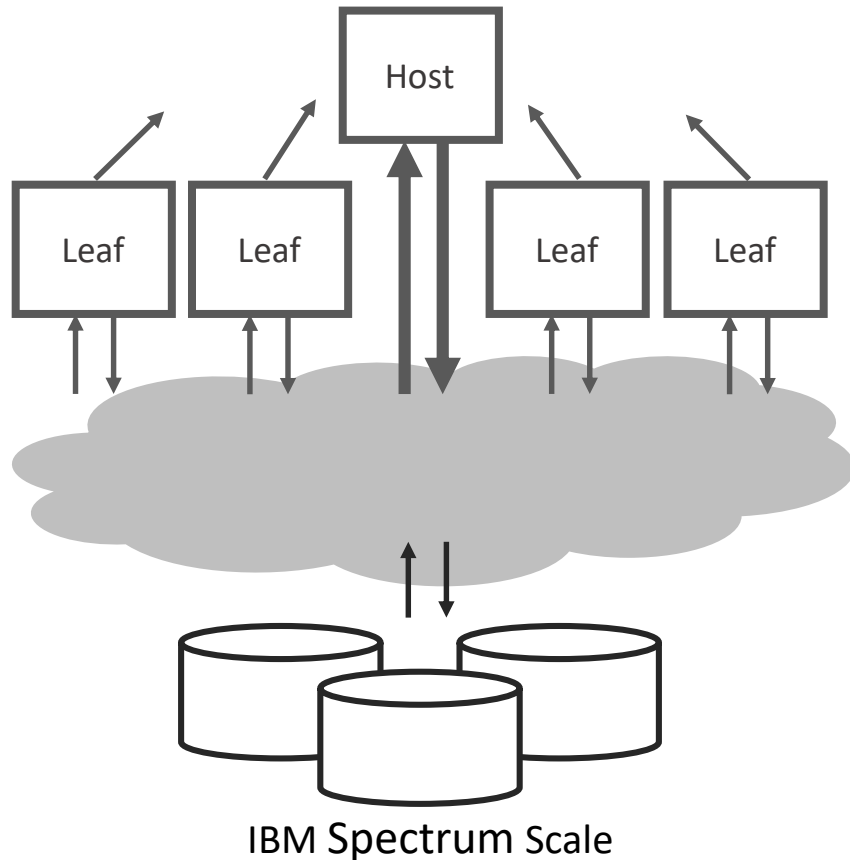


Customer case study: Catching rogue jobs

... stories from a Silicon Valley software startup



Customer case study: system architecture



Host-Leaf architecture

Host compute nodes:

- Native Spectrum Scale
- 4 Gigabit/s Infiniband
- 32 Slots

Leaf compute nodes:

- Send data over RCP to host nodes
- Clustered NFS

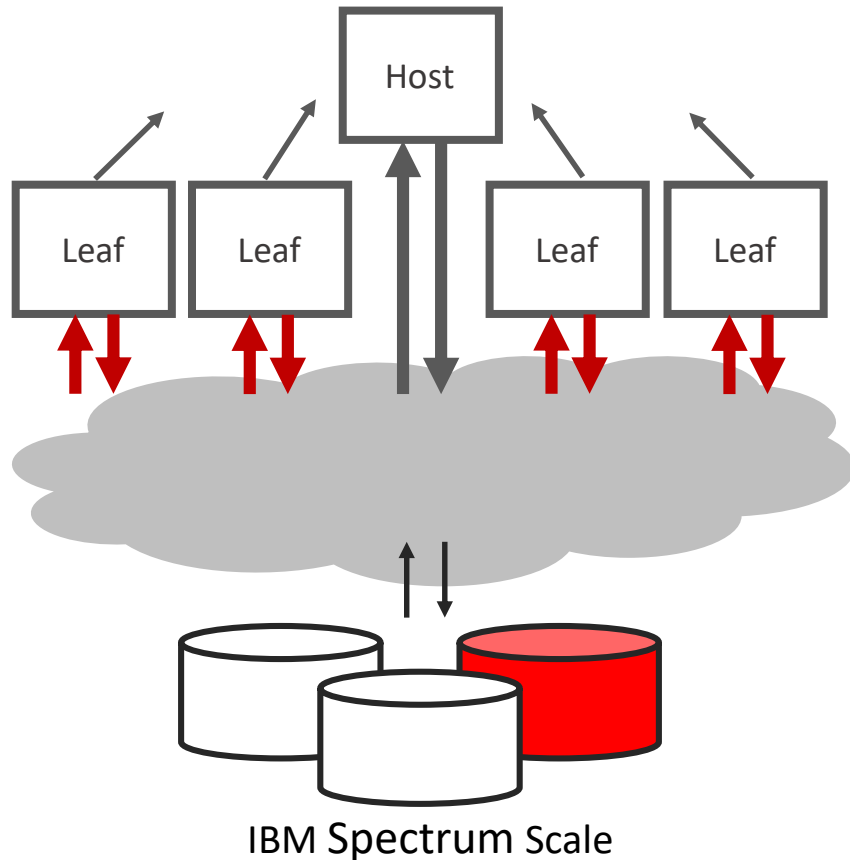
Storage:

- 8 Spectrum Scale servers



Customer case study

Problem 1: Bad working practices



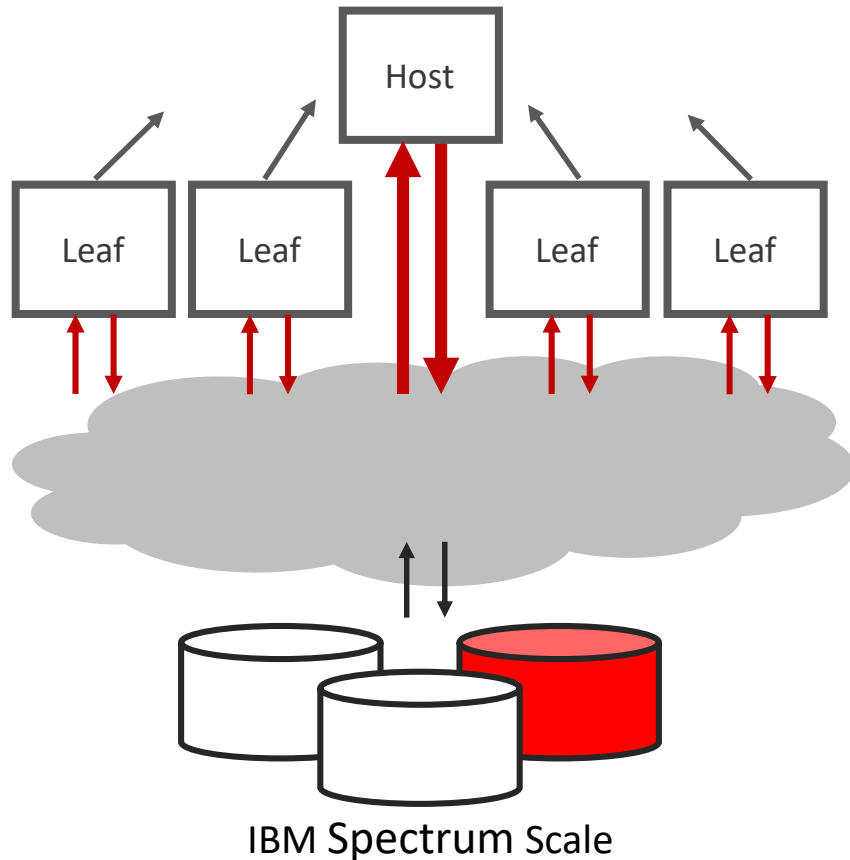
Jobs sometimes write from the leaf nodes instead of from the host.

If they hammer a particular mount point that can overload one of the filers.



Customer case study

Problem 2: Debug flag left on

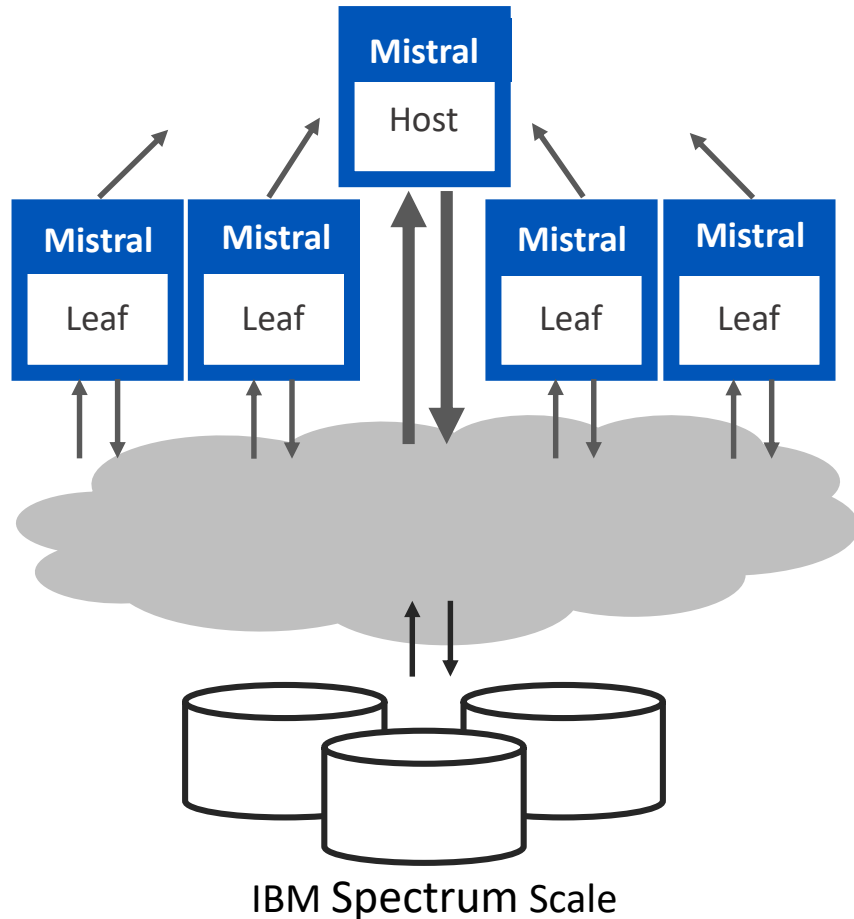


Job is debugged on one core then
handed over to run at scale
– debug flag left on accidentally.

This happens weekly.



Customer Case Study: Mistral deployment



Production jobs are wrapped in Mistral via an LSF job starter.

Mistral generates an alert if:

- Leaf nodes write to the file system
- Host nodes write too much data

Two use cases:

- My house is on fire
- My house was on fire, but I missed it

Integration with RTM makes this possible.



Example contracts

Example contract:

monitoringtimeframe, 60s

NAME	PATH/MOUNT POINT	IO TYPE	LIMIT
M1,	/gpfs/,	read,	500MB
M2,	/gpfs/,	write	500MB
M3,	/gpfs/,	create+delete,	2000

Example alert:

TIME	RULE	VIOLATION	PID	PROGRAM	FILE	JOB ID
17:30:05	M1	643MB/60s,	2784,	/tools/bin,	/gpfs/tools/config,	12
17:30:05	M2	505MB/60s,	2784,	/tools/bin,	/gpfs/data/file01,	12
17:30:06	M3	7405/60s,	2784,	/tools/bin,	/gpfs/data/set05,	12



Mistral RTM integration: Goals

To provide LSF administrators with a way to profile applications prior to them being onboarded into the cluster.

To provide at-a-glance monitoring exceptions delivered by the Mistral product.

To integrate with RTM's Fault Management framework so administrators can be notified when there is an event in progress.

To provide users with the ability to review the storage profile of their applications and workflows to search for solutions to performance problems.



Table showing:

- Contract name
- Contract type
- Mistral I/O rule
- I/O type
- Measurement type
- I/O size range
- Event type
- Threshold
- Observed I/O



Mistral RTM integration: Who is using the storage?



Console | Graphs | JobIQ | CLog | Thold | Syslog | **Grid** | License | Nectar | Admin

admin (Administrator) | Log out | Help

Grid -> Mistral Events | RTM is monitoring 10 hosts on 1 cluster.

Current Mistral Events

Cluster: All Queue: All Project: All Rollup Data Go Clear

Search: Rule: All Records: 30 Refresh: 1 Minute

Presets: Last Day From: 2017-03-05 14:36 To: 2017-03-06 14:36 2 Hours

Showing Rows 1 to 30 of 24763 [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21...]

Cluster Name	Job ID	Host	Queue	Project	User	Rule	PID	Observed	Time	Logged Time
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	65.00 MB	1.00 Sec	2017-03-06 12:07:57
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	72.00 MB	1.00 Sec	2017-03-06 12:07:56
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31983	123.64 MB	1.00 Sec	2017-03-06 12:07:56
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	159.00 MB	1.00 Sec	2017-03-06 12:07:55
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	65.00 MB	1.00 Sec	2017-03-06 12:07:54
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31983	137.00 MB	1.00 Sec	2017-03-06 12:07:54
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	147.00 MB	1.00 Sec	2017-03-06 12:07:53
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31983	42.00 MB	1.00 Sec	2017-03-06 12:07:53
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:52
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:52
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:51
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:51
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:50
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:50
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:49
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:49
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:48
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:48
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:47
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:46
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:46
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:46
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:45
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:45
Motown	155471[995]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:44
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:44
Motown	155471[998]	platidpxn04	default	default		Higher_reads_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:43
Motown	155471[995]	platidpxn04	default	default		Higher_writes_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:42
Motown	155471[995]	platidpxn04	default	default		Higher_writes_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:42
Motown	155471[998]	platidpxn04	default	default		Higher_writes_gpfsfpo	31873	42.00 MB	1.00 Sec	2017-03-06 12:07:42

Showing Rows 1 to 30 of 24763 [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21...]

Command: dd bs=1MB count=998 if=/platgpfsgpfsfpo/user/ladams/155471/998/155471.998.ddout of=/dev/null
File: /platgpfsgpfsfpo/user/ladams/155471/998/155471.998.ddout

Table showing:

- Cluster name
- Job ID
- Host Queue
- Project
- User
- Mistral I/O rule
- PID
- I/O observed (MB)
- Time frame
- Log time



Customer case study:

Protecting the file system from R&D jobs

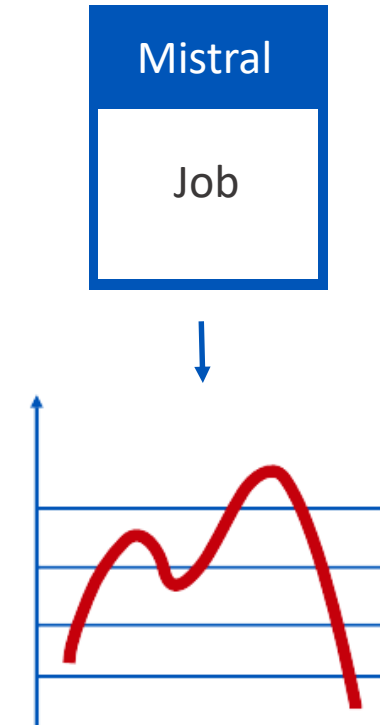
R&D jobs often seg fault and this exposes Mistral.

Solution

1. Guess which job is causing the problem the old way
2. Check problem jobs by profiling I/O in detail

Check for:

- Too much I/O
- Small reads and writes
- Peaks in I/O bandwidth
- Creation or deletion of small files
- Meta data operations: `open()`, `stat()`
- Random access patterns



Make every user an I/O expert



Summary

Profile profile profile.

Protect your storage by monitoring the whole cluster for rogue jobs.
Prevent bad I/O patterns by running health checks on new work flows.

Optimise applications for shared storage and optimise the compute and storage infrastructure for the jobs.



Thank you

Please get in touch for more information.

Dr Rosemary Francis
CEO and Co-founder

Ellexus Ltd

St John's Innovation Centre, Cowley Road, Cambridge CB4 0WS, UK
info@ellexus.com
01223 123456



Mistral: wide-scale I/O profiling and load balancing for shared storage
www.ellexus.com