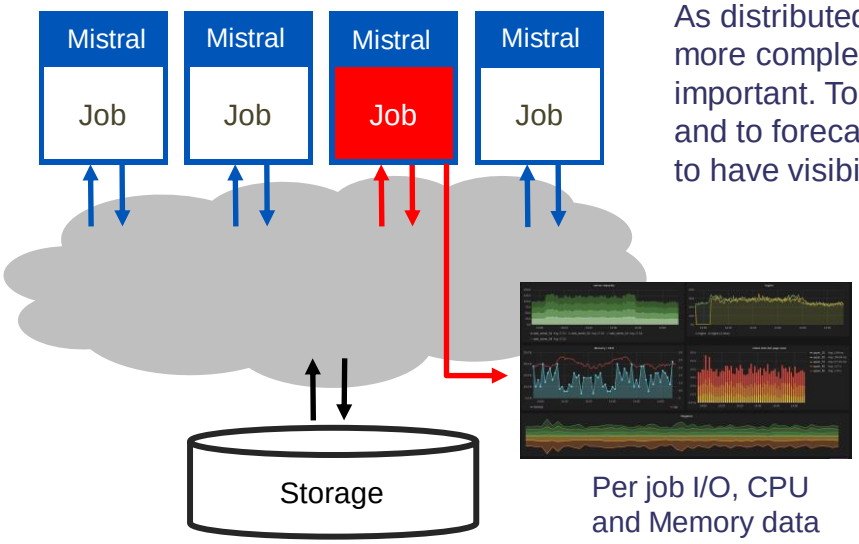
Mistral

I/O monitoring for HPC applications

Lightweight enough to run in production, but flexible enough to ensure that you get the most from on-prem HPC and that you have the information you need to manage hybrid cloud. Mistral is the leading application monitoring tool for HPC and scientific applications. Mistral monitors I/O, CPU and memory, quickly locating rogue jobs, storage bottlenecks and keeping track of what is running on the clusters day-to-day.



Per job I/O, CPU and Memory data

As distributed systems and compute clusters become more complex, the need for monitoring becomes more important. To be able to run the compute efficiently and to forecast and design for the future it is important to have visibility on what is being run today.

By allowing you to control information gathered, you can ensure that you know about the important events without being overwhelmed with too much data. It is a storage agnostic solution for monitoring I/O bandwidth, meta data, I/O performance, CPU and memory.

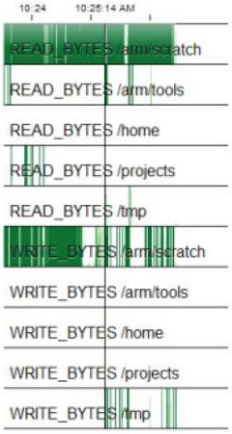
Key facts and capabilities

- **Mistral measures:** CPU, memory and I/O, collecting per-job, per-user and per-host metrics, broken down by job and by file system.
- **Mistral is lightweight:** Run in production or on large test suites to find bad I/O before it becomes a problem.
- **Mistral delivers on operational telemetry and business intelligence**: To ensure that the infrastructure you have is running efficiently and that you can plan for tomorrow.
- **Storage protection**: Detect rogue jobs and bad I/O patterns such as excess meta data, random I/O and small reads and writes.
- **Quality of service**: Throttle I/O for individual jobs to maintain a good quality of service.
- **Storage and scheduler agnostic**: Mistral integrates with the orchestration framework and delivers storage agnostic metrics for each job or workflow.
- **On prem and in the cloud**: Getting the most from the on-prem investment while exploiting hybrid cloud for agility takes knowledge and telemetry delivered by Mistral.

Mistral case studies

We worked with the HPC department at ARM to design Mistral. Olly Stephens, Engineering Systems Architect at ARM, said of the project to develop Mistral with Ellexus:

"We wanted to develop a system that will allow the infrastructure to protect itself somewhat against I/O behaviour that is considered a risk. In particular, we wanted the ability for aggressive use of the storage infrastructure to be automatically detected early and remedial steps taken quickly."



This is an example of a job at Arm that was overloading shared storage. It is supposed to store the temporary files on local storage, but it has been configured to store them on expensive shared storage instead. When run at scale this overloaded the shared file system, but once found was easy to fix.

Improving the storage performance at Diamond Light Source

Diamond Light Source is the UK's national particle accelerator. Each experiment generates vast amounts of data, which must be stored in real time and processed quickly. Experiments cover a range of fields so the team handles a wide range of data rates and compute applications.

Diamond Light Source used Mistral to identify straightforward measures to improve performance and cut down runtime. Profile findings revealed various inefficiencies and inefficient program startups that slowed the system down, impeding the work of different users.

Frederik Ferner, Senior Computer Systems Administrator at Diamond Light Source, said:

"By using Mistral, our team has already made a marked improvement to various applications that we maintain in-house. We have been able to reduce the impact of noisy neighbours, reduce runtime and identify applications with bad I/O.

"We intend to keep using Mistral to profile more applications and improve the overall architecture of our systems for in-house and third-party tools."

Managing I/O in the biggest supercomputers in the world

The Texas Advanced Computing Center (TACC) designs and operates some of the most powerful computing resources in the world. Its new supercomputer, Frontera, is the 5th most powerful supercomputer in the world. As a result, the IT team comes across many I/O-related challenges, as their systems handle many different users and workloads.

Single user load rarely causes a problem at TACC and usually iops, not bandwidth, is the limiter. However, meta data can get swamped and file count is often a problem: 3 billion files is a push for Lustre.

The TACC team is using Mistral to monitor meta data, specifically to catch users who delete a lot of files at once. Over time the team will add further aspects to monitor.

Mistral installation

Technical overview

Mistral logs data in a scalable database so a graphing dashboard such as Grafana or Splunk can be used to view the information. These give you live updates so you can find rogue jobs quickly or look back over historical data.

Both Splunk and Grafana let you filter on job, host, project or user. This makes it easy to do chargeback or to find system bottlenecks.

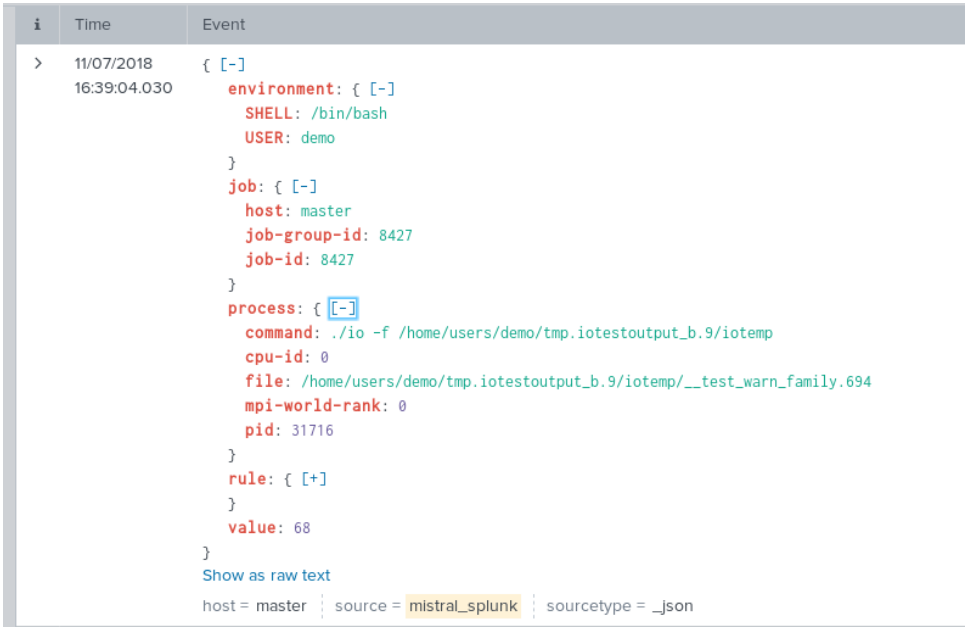
You can also set custom alerts that tell you when a measure has deviated too far from the mean so you can quickly spot a problem.

Mistral shows:

- Reads, writes and meta data
- Random vs streaming I/O
- Throughput and IOPS
- I/O sizes
- I/O performance
- Burst vs streaming profile

It's easy to add in custom metrics to measure machine health for a full system overview.

Although the graphical overview gives you a good summary of the system, to triage a bottleneck once it is found it is better to look at the additional information available in the database. Mistral logs the process that caused the I/O to spike and the last file it accessed giving you *what happened* as well as the *why it happened*.



Mistral installation

Technical overview

Mistral is easy to configure using the template setting that comes with the tool.

Intregration with the job scheduler

Mistral can be launched on a single application when you start it, but to run Mistral in production or on a large number of jobs it's easiest to use the job scheduler or launch configuration to automatically set up the Mistral environment.

```
JOB_STARTER= /share/tools/ellexus/mistral_latest/launch.sh; %USRCMD
```

What goes in the launch script?

The launch script sets up the licence, the database configuration and the Mistral I/O contract.

```
# The contract controls what Mistral will measure
export MISTRAL_CONTRACT_MONITOR_GLOBAL=${MISTRAL_INSTALL}/global.contract


# Tell Mistral to send output to elastic search via an output plug-in.
export MISTRAL_PLUGIN_CONFIG=${MISTRAL_INSTALL}/elastic_plugin.conf


source ${MISTRAL_INSTALL}/mistral
```

Data can be sent to any database using a configurable plugin. Mistral is shipped with plugins for Elasticsearch, InfluxDB, MySQL and Splunk. The following shows the database settings for an Elasticsearch plugin that pushed data every 5s.

```
PLUGIN,OUTPUT
PLUGIN_PATH,${MISTRAL_INSTALL}/mistral_elasticsearch
INTERVAL,5
PLUGIN_OPTION,--index=mistral
PLUGIN_OPTION,--host=10.0.0.100
PLUGIN_OPTION,--port=9200
PLUGIN_OPTION,--username=mistralelastic
PLUGIN_OPTION,--password= mistral
PLUGIN_OPTION,--error=${HOME}/mistral_elastic.log
END
```

How does the Mistral contract control what is measured?

The Mistral I/O contract is a flexible way of controlling what Mistral measures. To profile an application in detail you can set all available rules to trigger as soon as one I/O operation is performed. To monitor a whole cluster it is better to set fewer rules with higher thresholds so that only the data you need is collected. For the full list of what can be measured please refer to the user manual.

```
#data is collected every 5s
2,monitortimeframe,5s

#log the number of open calls that exceed 500 to /home
home-count-open,/home,open,all,count,500

#log the mean latency of read calls from /usr that are less than 4KB
#if the mean latency exceeds 50us
usr-lat-read_small,/usr,read,-4kB,mean-latency,50us
```