Altair Mistral Dashboard Guide

Version 2025.2.0

Contents

1 Mistral Grafana/ElasticSearch Dashboards

1.1 Mistral Overview Dashboard

1.1.1 Purpose

Show overall system utilisation so that you can understand how jobs are using CPU, Memory and their I/O patterns.

Quickly find jobs with high numbers of I/O operations that could be overloading shared storage. We call these jobs noisy neighbours.

1.1.2 Data

This dashboard offers an overview of job I/O, CPU and Memory utilisation. By default data is shown per job and per file system. It shows file system access and computer resource usage.

- *File system read and write - throughput per mount point:* Measures the amount of data read or written per second.
- *Meta data I/O per mount point:* Shows the number of metadata operations performed per second.
- *Memory and CPU:* Helps in assessing if a job is CPU or memory-bound.
  - *CPU time:* aggregated system and user time for all processes in a job
  - *Memory-RSS, Memory – Process Virtual Size:* aggregated RSS and VM memory utilisation for all processes in a job
  - *Host CPU: aggregated user, system and iowait time across all processes running on the node.*

1.1.3 Interpretation

Noisy neighbour jobs can overload shared file systems with too many I/O operations. Jobs can overload a file system with too many reads, too many writes or too many metadata operations. Typically if a job is doing enough I/O to overload the file system they will be doing a lot more operations than other jobs so will be easy to see in the graphs.

Memory is aggregated across multiple processes so memory that is shared between processes may be double counted. For this reason, memory utilisation should be taken as an upper limit to the amount of memory used for each job.

CPU time is per second so 1s = 100% utilisation of one core. CPU time may be larger than 1s if multiple cores have been allocated to the job.

We show the host CPU, memory and IOWait metrics so that you can understand the overall load on a host. A job may be slow because the host is very busy.

Host *IOWait* Indicates the time the system spent waiting on I/O. The operating system only record iowait when all threads that can run on a given core are blocked on I/O.

1.2 Mistral Job IO Detail Dashboard

1.2.1 Purpose

Show overall system utilisation so that you can understand how jobs are using CPU, Memory and their I/O patterns. This dashboard shows more detail than the Mistral Overview Dashboard and contains some additional metrics.

Quickly find jobs with high numbers of I/O operations that could be overloading shared storage. We call these jobs noisy neighbours. This detailed dashboard may help you find jobs that overload shared storage with more complicated patterns than you can see in the Overview Dashboard.

1.2.2 Data

Meta data I/O – count: shows the number of metadata operations per second. The data is broken down by operation type.

File system seeks: shows the number of seek operations per second.

File system writes and reads: shows the number of write or read operations per second or the amount of data transferred per second. The writes and reads are divided by operation size for small, medium and large I/O sizes. Note that the I/O operations are divided by the size of the operation, not by the size of the file. Mistral does not record the file size.

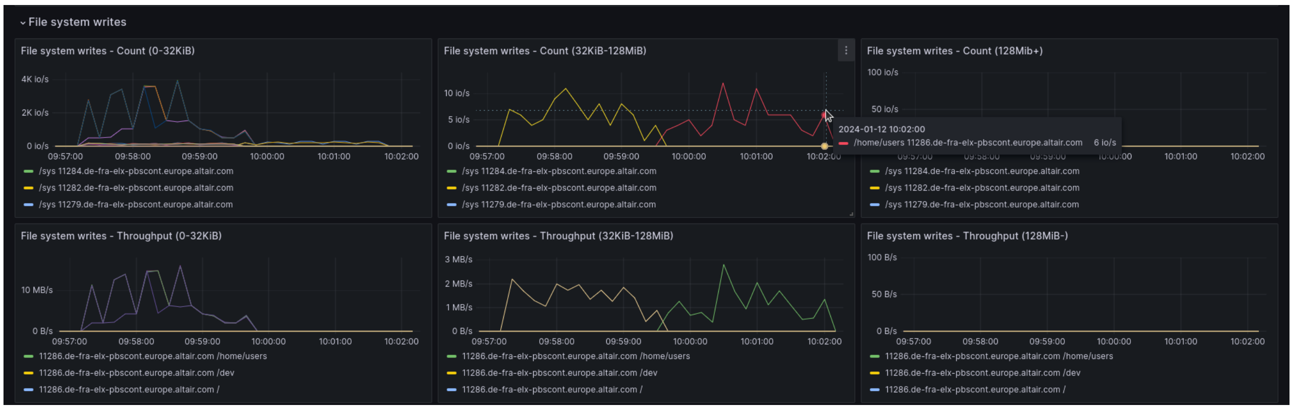
File system writes
File system writes - Count (0-32KiB)
File system writes - Count (32KiB-128MiB)
File system writes - Count (128Mib+)
File system writes - Throughput (0-32KiB)
File system writes - Throughput (32KiB-128MiB)
File system writes - Throughput (128MiB-)
/sys 11284.de-fra-elx-pbscont.europe.altair.com
/sys 11282.de-fra-elx-pbscont.europe.altair.com
/sys 11279.de-fra-elx-pbscont.europe.altair.com
11286.de-fra-elx-pbscont.europe.altair.com /home/users
11286.de-fra-elx-pbscont.europe.altair.com /dev
11286.de-fra-elx-pbscont.europe.altair.com /
2024-01-12 10:02:00
/home/users 11286.de-fra-elx-pbscont.europe.altair.com 6 io/s

Figure 1: Write counts and throughput split by operation size

1.2.3 Interpretation

Some metadata operations are faster than others. Open and Delete operations are more expensive than access operations. Some jobs overload shared file systems by opening a file for every read or write. Some jobs overload shared file systems by performing a lot of deletes at the end of the job.

Access or stat operations are usually fast, but some applications perform a stat before every open and some applications stat many locations in a PATH variable before finding a file to open. Some applications stat every file in a large directory. So although stats are fast, if the application performs thousands of stats per second this can put a high load on the file system.

We show seek operations in this dashboard so you can see how random the data access is. Shared file systems prefer to stream data so random access is more expensive.

The dashboard shows writes and reads divided into small, medium and large I/O operation size. Small reads and writes can put heady loads on shared file systems, especially when the accesses are random so high numbers of seeks and high numbers of small reads would put a larger load on the file system.

Large reads and writes are less common, but also can put a high load on the file system.

1.3 Mistral Job I/O Summary Dashboard

1.3.1 Purpose

The Job I/O Summary Dashboard shows data for jobs that have completed. Data is shown aggregated over the runtime of the job instead of over time as in other dashboards. The tables show the jobs with the most I/O.

Identify which jobs have used the most resources or made bad use of I/O calls. These are the jobs that would benefit from further profiling and optimisation.

For details on what constitutes bad I/O, see the full Mistral Manual section Rules for bad I/O.

1.3.2 Data

- *Job resource usage*: Table of job metrics for runtime, CPU and memory
- *IO Time*: shows the time spent in good, medium and bad I/O. There are four charts showing job data aggregated for all mount points, data split per job and per mount point, and for each of these the total time spent in I/O as well as the time spend in good and bad I/O as a percentage of the total time spent.
  - IO Time by Job
  - Percentage IO Time by Job
  - IO Time by job per mount point
  - Percentage IO Time by Job by mountpoint

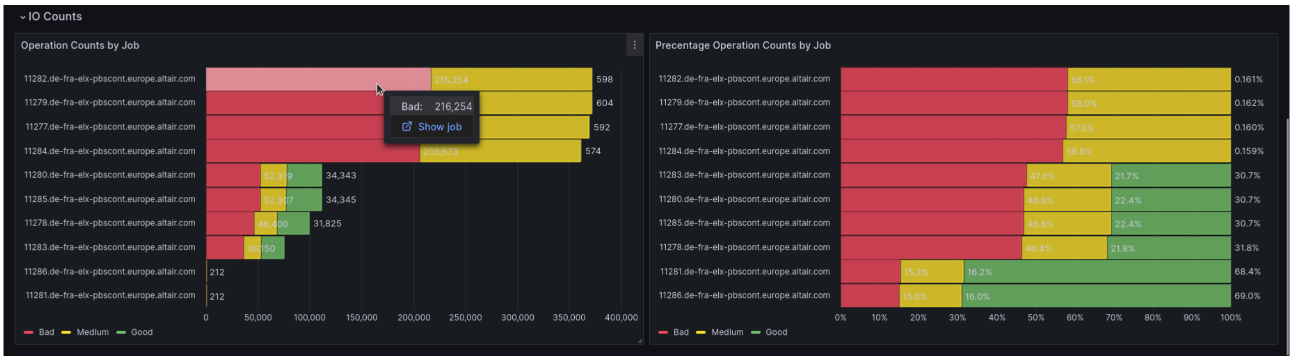

Figure 2: IO operations split into good, medium and bad categories per job reported as both counts and percentages

1.3.3 Interpretation

The first table shows the max CPU usage in terms of the number of cores and it shows the max memory utilisation. This shows what resources the job needs in order to run. It does not mean that the job will utilise those resources continuously throughout the job. Memory is aggregated across all processes so if processes have shared memory, then the memory utilisation may be higher than the amount actually used.

Jobs can overload shared storage with bad I/O. In general, bad I/O on local storage has a lower impact. You can see the bad I/O per mount point to see the amount of bad I/O that a job does against a shared file system.

To understand the impact of bad I/O you should look at both the amount of time spent in bad I/O and the percentage of bad I/O for the job. A job may be short and perform and a small amount of bad I/O overall, but if that job is the primary application run on the HPC cluster then a high percentage of bad I/O could overwhelm the shared storage overall. Similarly, a job might not be run often, but if it does a lot of bad I/O then just one or two instances of that job could overwhelm the shared storage.

So, to understand the impact of bad I/O you must look at the jobs that Mistral has selected as having the most bad I/O and decide if they are affecting shared storage or just local disk. You must then look at the application that has run and compare that with you HPC cluster metrics on how often that application has run.

1.4 Mistral Network Dashboard

1.4.1 Purpose

This dashboard offers an overview of job network activity. I/O is grouped by destination IP addresses.

This is disabled by default but can be enabled by setting the `network.Enabled` option in your Mistral configuration file to on.

1.5 Mistral Performance Dashboard

1.5.1 Purpose

Shows the performance of the file system. With values for the duration and efficiency of metadata and read/write I/O operations.

By highlighting the maximum durations of these operations, it guides administrators and users in pinpointing specific areas where optimization can lead to significant performance improvements.

1.5.2 Data

Max call duration for metadata I/O: the max I/O time for individual meta data operations. This is not the same a time spent in I/O for all meta data operations.

Max call duration for writes (small, medium and large): the max I/O time for individual write operations. This is not the same a time spent in I/O for all write operations.

Max call duration for reads (small, medium and large): the max I/O time for individual read operations. This is not the same a time spent in I/O for all read operations.

1.5.3 Interpretation

We show the max operation time to indicate the worst case performance of the file system. The operation time is the time it takes for the I/O call to return so it includes any local I/O overhead, any network overhead and the overhead of the shared file system itself.

Some file systems show very high I/O operation times when they are under heavy load or when they are performing backup and synchronisation operations. Operations that normally take just a few ms can take longer than a second and this can badly affection application performance. You can look for such operations here.

By default Mistral samples I/O performance so Mistral will not capture every slow I/O operation. If the application does not do much I/O then you may be missing performance data because Mistral will only sample I/O when the number of operations hits a threshold. See…

2 Understanding Mistral Profile Data

2.1 Operation Types

2.1.1 Read vs Write

Determine the balance between read and write operations. Some applications are read-heavy, while others are write-heavy. Often jobs read a lot at the start and only write at the end. If the applications are to be run at scale and are I/O bound then it might be a advantageous to stagger the start times so that jobs do not all read and write at the same time. This information can be used to either balance how jobs are scheduled or tune file system settings.

2.1.2 Metadata

Assess the frequency and impact of metadata operations (open, access, create, delete, fschange, connect and accept) on the overall I/O performance. It may be simple to reduce the number of these operations by for instance reducing the number of paths in the PATH variable.

2.2 I/O Patterns

2.2.1 Sequential vs Random I/O

Identify if an application performs more sequential or random I/O. Sequential I/O typically results in better throughput (due to fewer cache misses) whilst random I/O can lead to performance degradation. The number of seek operations relative to the number of read/writes is a good measure for this.

2.2.2 Stat calls vs Opens

Some applications stat many locations in a PATH variable before finding the right file and opening it. If there are many more stat() or access() calls for each open() call then this might be the case. This can be improved by removing paths that are not needed in the PATH variable.

2.2.3 Opens vs Reads or Writes

Some applications will open a file for every read or write. This may be necessary if the application reads from or writes to many small files. If the application is reading from or writing to a large file and opening it each time this puts a high load on shared file system and will affect system performance. It might be advantageous to cache that file on local storage to reduce the load on shared file systems.

2.3 Read/Write Operation Size

2.3.1 Small read or write operations

Applications may perform many small I/O operations. This may be caused by reading from or writing to many small files, or by reading from or writing to a larger file with small operations. Both of these patterns increase the load on the shared file system.

2.3.2 Large read or write operations

Very large read or write operations (e.g. writing everything in memory in one operation to file) can be high latency and can put a high load on the shared file system. They should be avoided.

2.4 Resource Utilization

2.4.1 CPU and Memory usage

Combining the I/O usage and CPU usage can show bottlenecks in a job. If high I/O usage coincides with high CPU usage, this could indicate a need for I/O optimization.

2.4.2 Network Bandwidth

For distributed systems it is important to understand how I/O operations impact network bandwidth.

2.5 Time spent in I/O

The time spent in I/O can be an indicator of how an application is going to be affected by file system performance. A job that spends more time in I/O may be more I/O bound than a job with very little time spent in I/O. It is not a certain indicator however because the job with very little time spent in I/O may be doing I/O in one thread that is on the critical path so increasing I/O latency could still have a big affect on the job run time.

A job that spends a lot of time in I/O may be I/O bound, CPU bound and memory bound at the same time. Each I/O operation usually takes some CPU and memory resources to issue. So even if the application is using less than one CPU core it may be able to issue more I/O operations more quickly if it were to run on a faster core. So if you are trying to speed up an I/O intensive application you should experiment with different types of storage as well as different CPU and memory resources.

2.5.1 I/O wait time

I/O wait time is measured per host so will only be recorded when all the threads that can run on a core are in I/O wait stage. This is not recorded per job so if the host is spending time in I/O wait time it is an indicator that at least one job on that host is spending time in I/O wait. If the job uses the host exclusively then a high I/O wait time indicates that the job is I/O bound.

2.6 I/O performance

The performance of the file system will affect the performance of an application. The time taken to return from I/O operations is measured by Mistral. If the I/O operations are taking a long time this might be because a shared file system is under heavy load or it might be because there is congestion in the network.

Varying I/O latency can explain varying job run times.