title: Hadoop Monitoring Integration description: Sematext's Hadoop server monitoring provides overview of the common Hadoop components within your cluster such as NameNode, DataNode, CPU, memory, JVM and more. It also includes MRv1 version specific reports such as JobTracker, JobTracker Queues, TaskTracker as well as YARN specific ResourceManager, ResourceManager Queues, and NodeManager report

Overview

Sematext Monitoring supports monitoring of both **MRv1** (0.22 and earlier, 1.0, 1.1) and **YARN** (0.23, 2.*) based Hadoop versions. Since the architecture is different, Sematext Monitoring uses different application types for them and different reports are available.

Common reports for all Hadoop types:

- Overview
- NameNode
- DataNode
- CPU & Mem
- Disk
- Network
- JVM
- GC

In addition to that, MRv1 versions also get the following reports:

- JobTracker
- JobTracker Queues
- TaskTracker

While reports specific for YARN versions are:

- ResourceManager
- ResourceManager Queues
- NodeManager

In some cases, some reports will be empty because particular Hadoop version doesn't expose some metrics over JMX. For instance, 0.20, 0.21, 0.22 **MRv1** versions of Hadoop will not have data in **JobTracker**, **JobTracker Queues** and **TaskTracker** reports (while 1.0 and 1.1 will have all reports populated). **NOTE**: regardless of this, you can monitor JVM stats of **JobTracker** and **TaskTracker** processes under JVM report for all **MRv1** versions (0.20, 0.21, 0.22 included). Also, since **SecondaryNameNode** doesn't expose specific metrics, it doesn't have a special report, but it can also be monitored under JVM report (for instance, you can create an alert to notify you when its heap size reaches some limit or drops to 0, meaning the process likely died).

All **YARN** versions (0.23, 2.*) will display all available reports and we expect all new Hadoop versions to continue behaving like that.

YARN versions don't have separate reports for the following components (since they don't expose specific metrics):

- **HistoryServer**
- **WebAppProxy**

However, you can still monitor these processes under JVM report, in the same way as **SecondaryNameNode** can be monitored in **MRv1** setups. You can also define any alerts which are based on JVM metrics which should be good enough for most situations.

Integration

- Instructions: https://apps.sematext.com/ui/howto/Hadoop-YARN/overview

Metrics

Metric Name Key *(Type) (Unit)*	Description
data node bytes read**hadoop.dn.io.read** *(long counter)*	
data node bytes written**hadoop.dn.io.write** *(long counter)*	
data node reads from local client**hadoop.dn.io.read.local** *(long counter)*	
data node reads from remote client**hadoop.dn.io.read.remote** *(long counter)*	
data node writes from local client**hadoop.dn.io.write.local** *(long counter)*	
data node writes from remote client**hadoop.dn.io.write.remote** *(long counter)*	
data node heartbeats avg time**hadoop.dn.io.write.heartbeats.time** *(double gauge) (ms)*	
data node heartbeats ops**hadoop.dn.io.write.heartbeats** *(long counter)*	
block checksum op avg time**hadoop.dn.blocks.op.checksum.time** *(double gauge) (ms)*	
block checksum num ops**hadoop.dn.blocks.op.checksum** *(long counter)*	
block report op avg time**hadoop.dn.blocks.op.reports.time** *(double gauge) (ms)*	
block report ops**hadoop.dn.blocks.op.reports** *(long counter)*	
copy block op avg time**hadoop.dn.blocks.op.copies.time** *(double gauge) (ms)*	
copy block ops**hadoop.dn.blocks.op.copies** *(long counter)*	
read block op avg time**hadoop.dn.blocks.op.reads.time** *(double gauge) (ms)*	

Metric Name Key *(Type) (Unit)*	Description
read block ops**hadoop.dn.blocks.op.reads** *(long counter)*	
replace block op avg time**hadoop.dn.blocks.op.replaces.time** *(double gauge) (ms)*	
replace block ops**hadoop.dn.blocks.op.replaces** *(long counter)*	
write block op avg time**hadoop.dn.blocks.op.writes.time** *(double gauge) (ms)*	
write block ops**hadoop.dn.blocks.op.writes** *(long counter)*	
blocks read**hadoop.dn.blocks.read** *(long counter)*	
blocks removed**hadoop.dn.blocks.removed** *(long counter)*	
blocks replicated**hadoop.dn.blocks.replicated** *(long counter)*	
blocks verified**hadoop.dn.blocks.verified** *(long counter)*	
blocks written**hadoop.dn.blocks.write** *(long counter)*	
jobtracker heartbeats**hadoop.jt.heartbeats** *(long counter)*	
running maps**hadoop.jt.maps.running** *(long gauge)*	
running reduces**hadoop.jt.reduces.running** *(long gauge)*	
waiting maps**hadoop.jt.maps.waiting** *(long gauge)*	
waiting reduces**hadoop.jt.reduces.waiting** *(long gauge)*	
blacklisted maps**hadoop.jt.maps.blacklisted** *(long counter)*	
blacklisted reduces**hadoop.jt.reduces.blacklisted** *(long counter)*	
trackers**hadoop.jt.reduces.trackers** *(long counter)*	
blacklisted trackers**hadoop.jt.reduces.trackers.blacklisted** *(long counter)*	
decommissioned trackers**hadoop.jt.reduces.trackers.decommissioned** *(long counter)*	
graylisted trackers**hadoop.jt.reduces.trackers.graylisted** *(long counter)*	
reduce slots**hadoop.jt.slots.reduce** *(long gauge)*	
map slots**hadoop.jt.slots.map** *(long gauge)*	
occupied map slots**hadoop.jt.slots.map.occupied** *(long gauge)*	
occupied reduce slots**hadoop.jt.slots.reduce.occupied** *(long gauge)*	
jobs completed**hadoop.jt.jobs.completed** *(long counter)*	
jobs failed**hadoop.jt.jobs.failed** *(long counter)*	
jobs killled**hadoop.jt.jobs.killed** *(long counter)*	
jobs preparing**hadoop.jt.jobs.preparing** *(long gauge)*	
jobs running**hadoop.jt.jobs.running** *(long gauge)*	
jobs submitted**hadoop.jt.jobs.submitted** *(long counter)*	
maps completed**hadoop.jt.maps.completed** *(long counter)*	
maps failed**hadoop.jt.maps.failed** *(long counter)*	
maps killled**hadoop.jt.maps.killed** *(long counter)*	
maps launched**hadoop.jt.maps.launched** *(long counter)*	

Metric Name Key *(Type) (Unit)*	Description
reduces completed**hadoop.jt.reduces.completed** *(long counter)*	
reduces failed**hadoop.jt.reduces.failed** *(long counter)*	
reduces killled**hadoop.jt.reduces.killed** *(long counter)*	
reduces launched**hadoop.jt.reduces.launched** *(long counter)*	
map slots**hadoop.jt.maps.slots** *(long gauge)*	
reduce slots**hadoop.jt.reduces.slots** *(long gauge)*	
waiting maps**hadoop.jt.waiting.maps** *(long gauge)*	
waiting reduces**hadoop.jt.waiting.reduces** *(long gauge)*	
running 0**hadoop.jt.running.0** *(long gauge)*	
running 60**hadoop.jt.running.60** *(long gauge)*	
running 300**hadoop.jt.running.300** *(long gauge)*	
running 1440**hadoop.jt.running.1440** *(long gauge)*	
live nodes**hadoop.nn.nodes.live** *(long gauge)*	
dead nodes**hadoop.nn.nodes.dead** *(long gauge)*	
decom nodes**hadoop.nn.nodes.decom** *(long gauge)*	
blocks total**hadoop.nn.blocks** *(long gauge)*	
corrupt blocks**hadoop.nn.blocks.corrupt** *(long gauge)*	
excess blocks**hadoop.nn.blocks.excess** *(long gauge)*	
missing blocks**hadoop.nn.blocks.missing** *(long gauge)*	
blocks pending deletion**hadoop.nn.blocks.pending.deletion** *(long gauge)*	
blocks pending replication**hadoop.nn.blocks.pending.replication** *(long gauge)*	
scheduled replication blocks**hadoop.nn.blocks.scheduled.replication** *(long gauge)*	
under replicated blocks**hadoop.nn.blocks.underreplicated** *(long gauge)*	
capacity remaining**hadoop.nn.capacity.remaining** *(long gauge)*	
capacity total**hadoop.nn.capacity** *(long gauge)*	
capacity used**hadoop.nn.capacity.used** *(long gauge)*	
total files**hadoop.nn.files** *(long gauge)*	
create file ops**hadoop.nn.files.ops.create** *(long counter)*	
get listing ops**hadoop.nn.files.ops.listing** *(long counter)*	
delete file ops**hadoop.nn.files.ops.delete** *(long counter)*	
file info ops**hadoop.nn.files.ops.info** *(long counter)*	
created files**hadoop.nn.files.created** *(long counter)*	
appended files**hadoop.nn.files.appended** *(long counter)*	
renamed files**hadoop.nn.files.renamed** *(long counter)*	
deleted files**hadoop.nn.files.deleted** *(long counter)*	
num allocated containers**hadoop.nm.containers.allocated** *(long gauge)*	

Metric Name Key *(Type) (Unit)*	Description
allocated GB**hadoop.nm.allocated.gb** *(long gauge) (GB)*	
available GB**hadoop.nm.available.gb** *(long gauge) (GB)*	
containers completed**hadoop.nm.containers.completed** *(long counter)*	
containers failed**hadoop.nm.containers.failed** *(long counter)*	
containers inited**hadoop.nm.containers.initiating** *(long gauge)*	
containers killed**hadoop.nm.containers.killed** *(long counter)*	
containers launched**hadoop.nm.containers.launched** *(long counter)*	
containers running**hadoop.nm.containers.running** *(long gauge)*	
shuffle connections**hadoop.nm.shuffle.connections** *(long counter)*	
shuffle output size**hadoop.nm.shuffle.output.bytes** *(long counter) (bytes)*	
shuffle outputs failed**hadoop.nm.shuffle.output.failed** *(long counter)*	
shuffle outputs ok**hadoop.nm.shuffle.output.ok** *(long counter)*	
active applications**hadoop.rm.apps.active** *(long gauge)*	
active users**hadoop.rm.users.active** *(long gauge)*	
agg containers allocated**hadoop.rm.agg.containers.alloc** *(long counter)*	
containers released**hadoop.rm.containers.released** *(long counter)*	
containers allocated**hadoop.rm.containers.alloc** *(long gauge)*	
allocated MB**hadoop.rm.memory.alloc.mb** *(long gauge) (MB)*	
applications completed**hadoop.rm.apps.completed** *(long counter)*	
applications failed**hadoop.rm.apps.failed** *(long counter)*	
applications killed**hadoop.rm.apps.killed** *(long counter)*	
applications pending**hadoop.rm.apps.pending** *(long gauge)*	
applications running**hadoop.rm.apps.running** *(long gauge)*	
applications submitted**hadoop.rm.apps.submitted** *(long counter)*	
available MB**hadoop.rm.memory.available.mb** *(long gauge) (MB)*	
containers pending**hadoop.rm.containers.pending** *(long gauge)*	
pending MB**hadoop.rm.memory.pending.mb** *(long gauge) (MB)*	
containers reserved**hadoop.rm.containers.reserved** *(long gauge)*	

Metric Name Key *(Type) (Unit)*	Description
reserved MB**hadoop.rm.memory.reserved.mb** *(long gauge) (MB)*	
running 0**hadoop.rm.running.0** *(long gauge)*	
running 60**hadoop.rm.running.60** *(long gauge)*	
running 300**hadoop.rm.running.300** *(long gauge)*	
running 1440**hadoop.rm.running.1440** *(long gauge)*	
active NMs**hadoop.rm.nm.active** *(long gauge)*	
decom NMs**hadoop.rm.nm.active.decom** *(long gauge)*	
lost NMs**hadoop.rm.nm.active.lost** *(long gauge)*	
rebooted NMs**hadoop.rm.nm.active.rebooted** *(long gauge)*	
unhealthy NMs**hadoop.rm.nm.active.unhealthy** *(long gauge)*	
map task slots**hadoop.tt.maps.slots** *(long gauge)*	
maps running**hadoop.tt.maps.running** *(long gauge)*	
reduce task slots**hadoop.tt.reduces.slots** *(long gauge)*	
reduces running**hadoop.tt.reduces.running** *(long gauge)*	
tasks completed**hadoop.tt.tasks.completed** *(long counter)*	
tasks failed ping**hadoop.tt.tasks.failed.ping** *(long counter)*	
tasks failed timeout**hadoop.tt.tasks.failed.timeout** *(long counter)*	