Moab Insight

Administrator Guide 9.1.2

February 2018

Adaptive
C O M P U T I N G

Adaptive Computing Enterprises, Inc.
1712 S. East Bay Blvd., Suite 300
Provo, UT 84606
+1 (801) 717-3700
www.adaptivecomputing.com

Scan to open online help

Contents

Welcome

Welcome to *Moab Insight 9.1.2 Administrator Guide.*

This guide is intended as a reference for system administrators.

In this guide:

Chapter 1: Understanding Insight

This chapter provides an introduction to Moab Insight and information about Insight's features and functions.

1.1 About Insight

Insight is an optional component of the Moab HPC Suite that collects the data that Moab emits on its message queue and stores it in a database. The message queue is efficient, can be encrypted, and tolerates disconnections and restarts on either side.

You *must* install Insight if you want to use Moab Viewpoint.
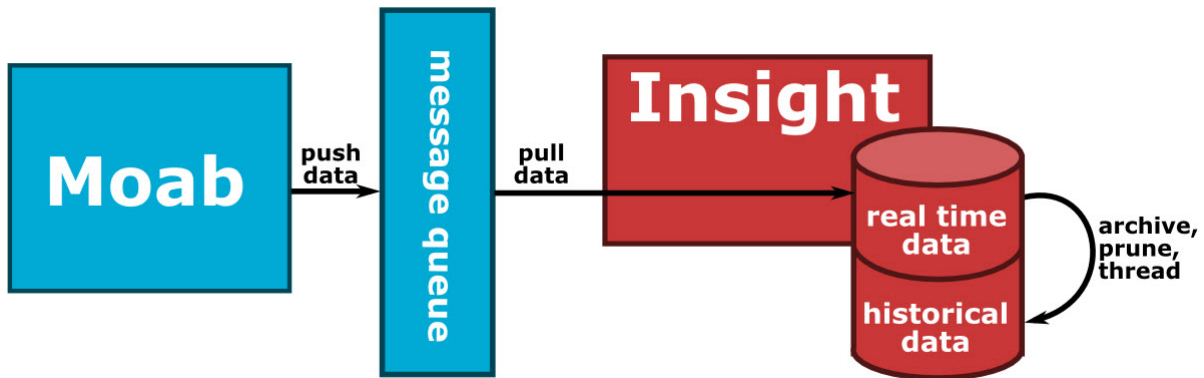
Moab publishes a large, coherent snapshot of its object model to the message queue at the end of each scheduling iteration and event-based updates as they occur; for instance, whenever a job starts or finishes. Insight collects this data and writes it to the database. When Moab and Insight experience periods of disconnection, Moab uses a buffer to temporarily save that data. When Insight has resumed listening, Moab will resend the data. You can configure the size of the buffer to suit the needs of your environment. See 2.2 Configuring Reliable Message Delivery - page 18 for more information.

Insight also takes historical samples of certain aspects of the jobs and nodes stored in Moab's MongoDB database. Moab's Mongo database only stores data about about the current state of the cluster. Thus completed jobs and deprovisioned nodes quickly disappear from Moab's database. By contrast, Insight's MongoDB database retains both current and historical information including idle, running, and completed jobs. Insight also stores data about currently active and deprovisioned nodes.

To keep the database from growing too large, Insight periodically prunes its database to eliminate stale or expired data. New samples are retained with high temporal granularity. As samples age, they are retained with progressively decreasing granularity. By default, data that is older than two years will be deleted to free resources. You can configure the retention duration to better suite the needs of your environment. See 1.3 The Insight Pruning Policy - page 6 for more information.

Related Topics

- Chapter 1: Understanding Insight - page 3

1.2 The Insight Sample Rate

The sample rate is the rate at which Insight samples job, node, and cluster data. This data is used in Viewpoint dashboards, charts, and graphs that show resource utilization over time.

You can make the sample rate as granular as once per minute or as relaxed as once per hour. To maximize the usefulness of samples in statistical analysis and dashboards, ensure that your sample rate divides evenly into 10- and 60-minute intervals. Valid values include 2 minutes, 5 minutes, 10 minutes, 20 minutes, and 30 minutes; but values such as 3 minutes and 15 minutes are invalid.

Insight enables you to tune the sample rate for your environment. See Tuning the Sample Rate - page 5 for more information.

Insight also has a pruning policy that manages how long sample data is retained. See 1.3 The Insight Pruning Policy - page 6 for more information.

Tuning the Sample Rate

You can uncomment and edit the `archiver.schedule` parameter in the Insight configuration file (`/opt/insght/etc/config.groovy`) to tune the Insight sample rate for your environment. The default is:

```
//archiver.schedule = "0 */1 * * * *" // Every minute
```

Specifying the archiver.schedule is similar to configuring a Cron job. There are six space-separated fields representing second, minute, hour, day, month, and weekday. You can give month and weekday names as the first three letters of the English names. For example:

`0 0 * * * *` – the top of every hour of every day

`*/10 * * * * *` – every ten seconds

`0 0 8-10 * * *` – 8:00, 9:00 and 10:00 every day

`0 0/30 8-10 * * *` – 8:00, 8:30, 9:00, 9:30 and 10:00 everyday

`0 0 9-17 * * MON-FRI` – on the hour from 9:00 to 5:00 on weekdays

`0 0 0 25 12 ?` – every Christmas at midnight

> ⓘ Configuring a higher sampling rate than the rate at which Moab conducts scheduling cycles may have little benefit. Moab Workload Manager emits most data to Insight only after it polls the resource managers and schedules jobs. Thus, Moab emits a great deal of the data only after the scheduling cycle completes. See Moab Workload Manager's RMPOLLINTERVAL parameter for more information.
>
> Configuring Insight to sample more frequently than Moab schedules may result in wasted processing time; not more granular data.

1.3 The Insight Pruning Policy

The Insight pruner controls how large the samples collections (such as job_samples, node_samples, system_samples) in the Insight Mongo database grow and how far back in time they stretch. Pruning is designed so that recent data is retained at a very high granularity while older data is retained at a lower granularity. By default, after two years data expires and is removed to free up space. Pruning helps prevent the Insight Mongo database from getting so large that it consumes all resources on the host.

> ⓘ Pruning is performed by the Insight MongoDB. Insight does not have to be running in order for pruning to occur.

The following rules govern temporal granularity in the samples collections.

- Retain minute granularity for the last 60 minutes

- Retain 10 minute granularity for the last 24 hours

- Retain hour granularity for the last seven days

- Retain 6 hour granularity for the last month

- Retain day granularity for up to two years

For the last hour, Insight will retain per minute samples. For data older than one hour, the finest granularity that will be retained is one sample per 10 minutes. Still older data is governed by the rules above.

During the last hour, the samples are retained with one-per-minute granularity; during the last day, with one-per-ten-minutes granularity; during the last week, with one-per-hour granularity; during the last month, with one-per-six-hours granularity; and finally, indefinitely, with one-per-day granularity.

The default Insight pruning policy is complex and states the following: keep minute-by-minute samples back to the beginning of the previous hour; keep 10-minute samples back to the beginning of the previous day; keep hourly samples back to the beginning of the previous week; keep 6-hour samples back to the beginning of the previous day; and keep daily samples forever.

The following is an example of the sample dates that could be present if the local time was 2018-10-28 19:45:00 MST.

```
... // For the last hour one sample per minute is retained
"sampleDate" : "2018-10-28 19:45:00 MST"
"sampleDate" : "2018-10-28 19:44:00 MST"
"sampleDate" : "2018-10-28 19:43:00 MST"
"sampleDate" : "2018-10-28 19:42:00 MST"
"sampleDate" : "2018-10-28 19:41:00 MST"
... // For the last day one sample per 10 minutes is retained
"sampleDate" : "2018-10-28 19:40:00 MST"
"sampleDate" : "2018-10-28 19:30:00 MST"
"sampleDate" : "2018-10-28 19:20:00 MST"
"sampleDate" : "2018-10-28 19:10:00 MST"
... // For the last week one sample per hour is retained
```

```
"sampleDate" : "2018-10-27 18:00:00 MST"
"sampleDate" : "2018-10-27 17:00:00 MST"
"sampleDate" : "2018-10-27 16:00:00 MST"
"sampleDate" : "2018-10-27 15:00:00 MST"
"sampleDate" : "2018-10-27 14:00:00 MST"
... // For the last month one sample per 6 hours is retained
"sampleDate" : "2018-10-27 12:00:00 MST"
"sampleDate" : "2018-10-27 06:00:00 MST"
"sampleDate" : "2018-10-27 00:00:00 MST"
"sampleDate" : "2018-10-26 18:00:00 MST"
"sampleDate" : "2018-10-26 12:00:00 MST"
"sampleDate" : "2018-10-26 06:00:00 MST"
"sampleDate" : "2018-10-26 00:00:00 MST"
... // For the last two years one sample per day is retained
"sampleDate" : "2018-09-28 00:00:00 MST"
"sampleDate" : "2018-09-27 00:00:00 MST"
"sampleDate" : "2018-09-26 00:00:00 MST"
"sampleDate" : "2018-09-25 00:00:00 MST"
```

Related Topics

- Chapter 1: Understanding Insight - page 3

1.4   Configuration Values

The `/opt/insight/etc/config.groovy` file contains the configuration values.

The following table identifies and describes the Insight-specific settings that you can change.

Property	Type	Default	Description
archiver.jobs.expireAfterSeconds	Integer	63244800	The number of seconds a completed job will remain in the Insight Mongo database before being pruned (732 days).
archiver.jobSamples.expireAfterSeconds	Integer	63244800	The number of seconds a job sample will remain in the Insight Mongo database before being pruned (732 days).
archiver.jobStateJournal.expireAfterSeconds	Integer	63244800	The number of seconds a job state journal record will remain in the Insight Mongo database before being pruned (732 days).
archiver.nodeHistory.expireAfterSeconds	Integer	63244800	A node that has not been updated in this number of seconds will be pruned from `node_history` collection in the Insight Mongo database (732 days).
archiver.nodeLifeCycleJournal.expireAfterSeconds	Integer	63244800	A node life cycle record older than this number of seconds will be pruned from the Insight Mongo database (732 days).
archiver.nodes.expireAfterSeconds	Integer	63244800	A node that has not been updated in this number of seconds will be pruned from the nodes collection in the Insight Mongo database (732 days).
archiver.nodesAuxiliary.expireAfterSeconds	Integer	63244800	A node auxiliary record that has not been updated in this number of seconds will be pruned from the Insight Mongo database (732 days).
archiver.nodeSamples.expireAfterSeconds	Integer	63244800	The number of seconds a node sample will remain in the Insight Mongo database before being pruned (732 days).

Property	Type	Default	Description
archiver-.reservations.expireAfterSeconds	Integer	63244800	A reservation that has not been completed in this number of seconds will be pruned from the Insight Mongo database.
archiver-.resourceJobTimeline.expireAfterSeconds	Integer	63244800	The number of seconds resource job timeline data will remain in the Insight Mongo database before being pruned
archiver-.systemSamples.expireAfterSeconds	Integer	63244800	The number of seconds a system sample will remain in the Insight Mongo database before being pruned.
health.stale.threshold.seconds	Integer	300	Insight tables are considered stale if they have not been updated within this many seconds.
insight.port	Integer	5568	Port on which Insight listens for control messages (start, stop, sync, health).
insight.skip.database.migration	Boolean	false	If true, Insight will skip migrating data written by older versions of Insight to the current database schema.
kafka.enabled	Boolean	false	Specifies whether reporting messages should be sent to Kafka (for processing by the Reporting Framework).
kafka.bootstrap.servers	String	loc-alhost:909-2	A comma separated list of `host:port` pairs to use for establishing the initial connection to Kafka. (e.g. `"host1:port1,host2:port2"`)
kafka.batch.size	Integer	200	The default size (in bytes) of a batch of messages. If the number of bytes of messages waiting to be sent to Kafka does not exceed `kafka.batch.size`, then Insight will wait up to `kafka.linger.ms` milliseconds for the batch to fill up before sending. A batch size of zero will disable batching.

Property	Type	Default	Description
kafka.linger.ms	Integer	1	How long to wait for a batch to fill up (meet the default batch size) before sending the batch to Kafka.
kafka.buffer.memory	Long	33554432	The total bytes of memory that can be used to buffer records waiting to be sent to Kafka.
kafka.max.block.ms	Long	60000	How long the Kafka thread will block before throwing an exception, if the buffer is full and there are messages waiting to be sent.
kafka.topic	String	insight	The Kafka topic that Insight will publish messages to.
moab.dataPort	Integer	5574	Port on which Insight listens for incoming messages from Moab.
moab.mongo.databaseName	String	moab	The MongoDB database name to use for Moab.
moab.mongo.host	String	127.0.0.1	The MongoDB host to use for Moab.
moab.mongo.jobSyncSchedule	String	"0 0 0 * * *"	A cron expression that configures how often Insight should sync jobs with Moab's Mongo database.

Property	Type	Default	Description
moab.mongo.password	String	secret4	(Optional) The password to use when connecting to MongoDB for Moab. 🛈 The following characters must be escaped in strings in the `/opt/insight/etc/config.groovy` and `/opt/mws/etc/mws-config.groovy` files (such as when used in a password): \ (backslash), " (double quote), ' (single quote), $ (dollar sign). Example: `mongo.password="my\$cool\$password"`. It is recommended that you avoid using these characters.
moab.mongo.port	Integer	27017	The MongoDB port to use for Moab.
moab.mongo.username	String	insight_user	(Optional) The username to use when connecting to MongoDB for Moab.
mongo.databaseName	String	insight	The MongoDB database name to use for Insight.
mongo.host	String	127.0.0.1	The MongoDB host to use for Insight.

Property	Type	Default	Description
mongo.password	String	secret4	(Optional) The password to use when connecting to MongoDB for Insight. ⓘ The following characters must be escaped in strings in the `/opt/insight/etc/config.groovy` and `/opt/mws/etc/mws-config.groovy` files (such as when used in a password): \ (backslash), " (double quote), ' (single quote), $ (dollar sign). Example: `mongo.password="my\$cool\$password"`. It is recommended that you avoid using these characters.
mongo.port	Integer	27017	The MongoDB port to use for Insight.
mongo.replicaSet	String	-----	The MongoDB replica set servers to use (for example, `"moab1:27017,moab2:27017"`); note that `mongo.host` must be set to null to use this option.
mongo.ssl	Boolean	false	Specifies whether the driver should use an SSL connection to MongoDB.
mongo.username	String	insight_user	(Optional) The username to use when connecting to MongoDB for Insight.
moab.reliabilityPort	Integer	5575	Port used for reliable message delivery between Moab and Insight.
messageQueue.secretKey	String	_	Base64-encoded AES key used to encrypt messages sent from Moab to Insight.
worker.count	Integer	20	The number of worker threads Insight will use to process messages from Moab.

Related Topics

Chapter 2: Configuring and Troubleshooting Insight

This chapter provides procedures for customizing and troubleshooting Insight.

2.1 Tuning Insight for Your System

This topic provides information to tune your configuration to keep pace with Moab Workload Manager.

2.1.1 Available Hardware

The number of threads Insight uses to process messages from Moab can be adjusted by changing the `worker.count` parameter in `/opt/insight/etc/config.groovy`.

If the Insight Server Host has many processors you may want to consider increasing the number of threads. For example, if the Insight host has 16 cores you may consider configuring 4 threads per core.

```
// 16 cores * 4 threads per core = 64 cores
worker.count = 64
```

> ⓘ Once you have changed this value you will need to restart Insight for the changes to take effect.

2.1.2 Java Runtime Environment

Insightshould be run on Oracle® Java® Runtime Environment (JRE) version 8.

Normally Insight will fail to start and will log an error to /opt/insight/log/insight.log if run on an unsupported JRE. The following is a sample error message:

```
java.lang.Exception: Unsupported Java Virtual Machine (OpenJDK 64-Bit Server VM): see
the Requirements section of the Insight Installation Guide.
```

This verification check prevents you from accidentally using a JRE which is known to have problems with Insight. However, there may be times when you want to force Insight to run on an unsupported JRE. This might happen if a supported JRE is not available for your platform. In these cases you tell Insight to skip the JRE verification check by setting the insight.skip.jre.verification parameter in the /opt/insight/etc/config.groovy configuration to "true". For example:

```
insight.skip.jre.verification = true
```

> ⓘ By default, Insight sets its `JAVA_HOME` environment variable to `/usr/java/latest`. To use a different JRE, edit `/opt/insight/etc/insight.conf` and change the value of `JAVA_HOME`.

2.1.3 Tuning RMPOLLINTERVAL for High-Throughput Systems

By default, each time a new job starts Moab polls its resource managers to make sure it has the most up-to-date information and sends a series of messages to Insight. In high-throughput systems, there may be jobs starting every few seconds. This causes a high volume of messages to be sent to Insight. On some systems, Insight may have difficulty processing messages as fast as Moab sends them. This can lead to the web interface having outdated data. In such cases it may be necessary to tune Moab's `RMPOLLINTERVAL` parameter to allow Insight to run optimally.

As described in the `RMPOLLINTERVAL` documentation, this parameter controls how much time elapses between when Moab polls its resource managers. It defaults to:

```
RMPOLLINTERVAL 30
```

This means that the minimum poll interval is 0 seconds and the maximum is 30 seconds. As explained above, having a minimum of 0 seconds results in a new poll every time a job starts. If no jobs start within 30 seconds of the last poll, Moab will initiate a new poll. However, if new jobs are starting every second and Insight is getting behind (Viewpoint is showing outdated data), then it may be wise to specify a minimum that is greater than 0. For example, we can specify a minimum interval of 10 seconds as follows:

```
RMPOLLINTERVAL 10,30
```

This means that even if new jobs start, Moab will not poll the resource manager until at least 10 seconds have passed since the last poll. This prevents Moab from overwhelming Insight with extra—possibly unnecessary—messages each time a job starts. However, it means that it may be up to 10 seconds before Insight (and thus Viewpoint) learns of changes. Therefore, for high-throughput systems, some tuning of this parameter may be required to achieve a balance that causes Moab to send messages frequently enough that Insight's database remains up-to-date, but not so many messages that Insight cannot process them fast enough.

Related Topics

2.2 Configuring Reliable Message Delivery

Moab and the Insight daemon gracefully handle disconnects or restarts. If you restart Moab, you do not have to restart Insight. To handle instances of disconnect, or downtime, Moab stores all data it attempts to send to Insight in memory and on disk. Once Moab generates enough data to meet the maximum storage size, it begins to delete the oldest data and make room for the new data. The size of the data on the disk is two times the maximum storage size. By default, the maximum storage size is 1 GB with 2 GB max on disk. You can customize the storage size for your unique environment.

To configure reliable message delivery

1. Open the Moab configuration file on the Moab head node and set the INSIGHTSTORESIZE and INSIGHTSTOREDIR configuration parameters. See Appendix A: Moab Parameters in Moab Workload Manager for parameter information.

```
[moab]$ vi /opt/etc/moab
...
INSIGHTSTORESIZE 2048 # 2 GB store size with 4 GB on disk
INSIGHTSTOREDIR /tmp/insight_store
...
```

   If INSIGHTSTOREDIR is a relative path, the Moab home directory is prepended to it. It uses the given path if it is an absolute path.

2. Restart Moab in order for the new configuration parameters to take effect.

Related Topics

- Chapter 2: Configuring and Troubleshooting Insight - page 15

2.3   Troubleshooting

This topic contains information on troubleshooting and resolving issues.

- Remove Node or Job from Insight
- Sync Insight Database with Moab Database

2.3.1   Remove Node or Job from Insight

To remove a node or job from the Insight MongoDB database use:

```
/opt/insight/utils/delete-objects-from-db.py
```

> **ⓘ** Run `delete-objects-from-db.py -h` for more usage information.

2.3.2   Sync Insight Database with Moab Database

> **ⓘ** As of 9.1.2, Insight automatically syncs with the Moab database once per day. To modify this schedule, see the documentation on the moab.mongo.jobSyncSchedule configuration value.

> **ⓘ** This procedure lets you force a sync of the Insight databases with the Moab database. This can be useful if Insight has been down for a long time and you want to currently running jobs show up in Insight/Viewpoint. Another use case is when a job that completed hours ago shows up in Viewpoint as still being active.

To temporarily stop all traffic to Insight and sync all jobs in the Insight database with the jobs in the Moab database use:

```
/opt/insight/utils/sync.py
```

If an error like the following is printed to the console, follow the steps shown below:

```
Did not receive response from socket within the timeout, the other side of the socket
is probably not running
```

1. Make sure Insight is running.

   - Red Hat 6-based systems

     ```
     [root]# service insight status
     insight (pid 7637) is running...
     ```

- Red Hat 7- or SUSE-12-based systems

```
[root]# systemctl status insight
insight (pid 7637) is running...
```

2. Make sure there is a process listening on the Insight control port (`5568` by default) for all interfaces, including the loopback interface. This can be determined by running `netstat` and making sure the response has `0.0.0.0:5568`.

```
[root]# netstat -ant | grep 5568 | grep -i listen
Bad
tcp    0      0 192.168.0.4:5568     0.0.0.0:*      LISTEN  #listening only on external
interface (e.g. eth0)
tcp    0      0 127.0.0.1:5568       0.0.0.0:*      LISTEN  #listening only on loopback
interface (e.g. lo)
Good
tcp    0      0 0.0.0.0:5568         0.0.0.0:*      LISTEN  #listening on all
interfaces
```

3. If Insight is not listening on all interfaces, stop Insight.

- Red Hat 6-based systems

```
[root]# service insight stop
```

- Red Hat 7- or SUSE-12-based systems

```
[root]# systemctl stop insight
```

4. Edit `/opt/insight/etc/config.groovy` and make sure the `insight.host` property is set to "*".

```
insight.host = "*"
```

5. Start Insight.

- Red Hat 6-based systems

```
[root]# service insight start
```

- Red Hat 7- or SUSE-12-based systems

```
[root]# systemctl start insight
```

6. Verify using `netstat` that Insight is now listening on all interfaces as shown above, then retry running `sync.py`.

Related Topics

- Chapter 2: Configuring and Troubleshooting Insight - page 15