
# 1. GridGain In-Memory Accelerator For Apache Hadoop

GridGain's In-Memory Accelerator For Apache Hadoop is designed to deliver uncompromised performance for existing Apache Hadoop 2.2 or above applications with zero code change as well as simplicity of installation and configuration across all the supported platforms.

# 2. Installation

GridGain distribution comes in a ZIP file that simply needs to be unzipped. The Accelerator requires Apache Hadoop of version 2.2 or above to be already installed on the system either using Apache Bigtop packages or manually (manual installation just means that Apache Hadoop binary distribution must be unpacked somewhere on the system). In case of manual installation `HADOOP_HOME` environment variable must point to the installation directory of Apache Hadoop.

> **NOTE:** You do not need any Apache Hadoop processes to be started, you only need to deploy the Apache Hadoop distribution on your system. Nevertheless you can run Apache Hadoop jobs with GridGain's Accelerator over HDFS, in this case up and running HDFS infrastructure will be needed.

The Accelerator comes with command line setup tool `bin/setup-hadoop.sh` (`bin/setup-hadoop.bat` on Windows) which will guide you through all the needed setup steps (note that the setup tool will require write permissions to the Apache Hadoop installation directory).

Installation requirements:

1. Windows, Linux, or MacOS environment.
2. Java 7 or 8 (latest update is advisable).
3. Point `JAVA_HOME` environment variable to your JDK or JRE installation.

4. Apache Hadoop 2.2 or above installed.
5. Point `HADOOP_HOME` environment variable to the installation directory of Apache Hadoop.
6. Run `bin/setup-hadoop.{sh|bat}` setup script and follow instructions.

> **NOTE:** On Windows platform Apache Hadoop client requires `JAVA_HOME` path to not contain space characters. Java installed to `C:\\Program Files\` will not work, install JRE to correct location and point `JAVA_HOME` there.

### 2.1 Check GridGain Installation

After setup script successfully completed, you can execute the GridGain startup script. The following command will startup GridGain node with default configuration using multicast node discovery.

```
bin/ggstart.{sh|bat}
```

If GridGain was installed successfully, the output from above commands should produce no exceptions or errors. Note that you may see some other warnings during startup, but this is OK as they are meant to inform that certain functionality is turned on or off by default.

You can execute the above commands multiple times on the same machine and make sure that nodes discover each other. Here is an example of log printout when 2 nodes join topology:

```
... Topology snapshot [nodes=2, CPUs=8, hash=0xD551B245]
```

You can also start GridGain Management Console, called Visor, and observe started nodes show up on Visor Dashboard. To startup Visor, you should execute the following script:

```
bin/ggvisorcmd.{sh|bat} (Command-line, available in open source)
bin/ggvisorui.{sh|bat} (GUI mode, enterprise version only)
```

# 3. Configuration

To configure GridGain nodes you can change configuration files at `config` directory of GridGain installation. Those are conventional Spring files. Please refer to shipped configuration files and GridGain javadocs for more details.

### 3.1 Distributed File System Configuration

GridGain has it's own distributed in-memory file system called GGFS. Hadoop jobs can use it instead of HDFS to achieve maximum performance and scalability. Setting up GGFS is much simpler than HDFS, it requires just few tweaks of GridGain node configuration and does not require starting any additional processes. Default configuration shipped with the Accelerator contains one configured instance named "ggfs" which can be used as reference.

Generally URI for GGFS which will be used by Apache Hadoop looks like:

```
ggfs://ggfs_name@host_name
```

Where `ggfs_name` is GGFS instance name, `host_name` is any host running GridGain node with that GGFS instance configured. For more details please refer to GGFS documentation.

### 3.2 Apache Hadoop Client Configuration

To run Apache Hadoop jobs with GridGain cluster you need to configure `core-site.xml` and `mapred-site.xml` at `$HADOOP_HOME/etc/hadoop` directory the same way as it is done in templates shipped with the Accelerator. The setup tool `bin/setup-hadoop.{sh|bat}` will ask you to replace those files with GridGain's templates or you can find these templates at `docs/core-site.gridgain.xml` and `docs/mapred-site.gridgain.xml` respectively and perform the needed configuration manually.

Apache Hadoop client will need to have GridGain jar files in classpath, the setup tool will care of that as well.

## 4. Running Apache Hadoop Job With GridGain's In-Memory Accelerator

To run Apache Hadoop job with GridGain cluster you have to start one or multiple GridGain nodes and make sure they successfully discovered each other.

When all the configuration is complete and GridGain nodes are started, running Apache Hadoop job will be the same as with conventional Apache Hadoop distribution except that all GridGain nodes are equal and any of them can be treated as Job Tracker and DFS Name Node.

To run "Word Count" example you can load some text files to GGFS using standard Apache Hadoop tools:

```
cd $HADOOP_HOME/bin

./hadoop fs -mkdir /input

./hadoop fs -copyFromLocal $HADOOP_HOME/README.txt /input/WORD_COUNT_ME.
txt
```

Run the job:

```
./hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/*-mapreduce-examples-*.
jar wordcount /input /output
```

Check results:

```
./hadoop fs -ls /output

./hadoop fs -cat /output/part-r-00000
```

A job can be ran on multiple nodes on localhost or in cluster environment the same way. The only changes needed to switch Apache Hadoop client to a cluster are to fix host in default DFS URI in `core-site.xml` and host in job tracker address in `mapred-site.xml`.

# 5. Management & Monitoring with Visor

GridGain comes with GUI and CLI (command) based DevOps Managements Consoles delivering advance set of management and monitoring capabilities. Visor GUI is based on a standalone Java application and CLI version is built on top of Scala REPL providing fully scriptable and customizable DevOps environment.

To start Visor GUI console (enterprise edition only), execute the following command:

```
`bin/ggvisorui.sh`
```

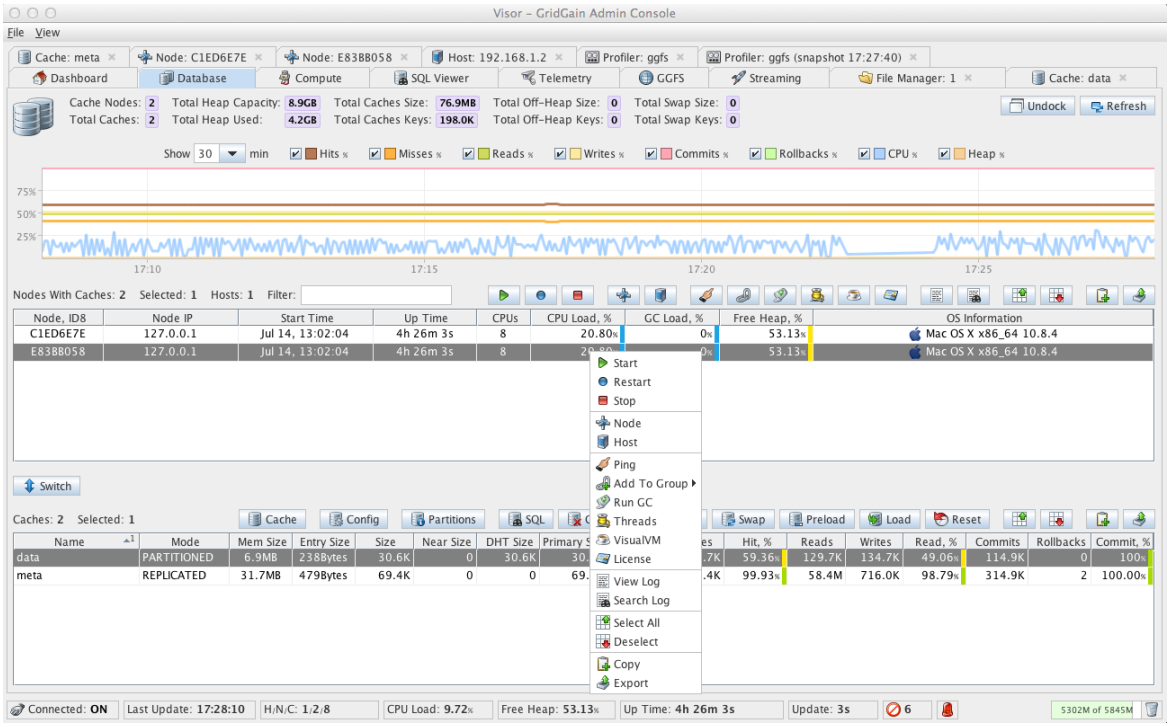
To start Visor in console mode you should execute the following command:

```
`bin/ggvisorcmd.sh`
```

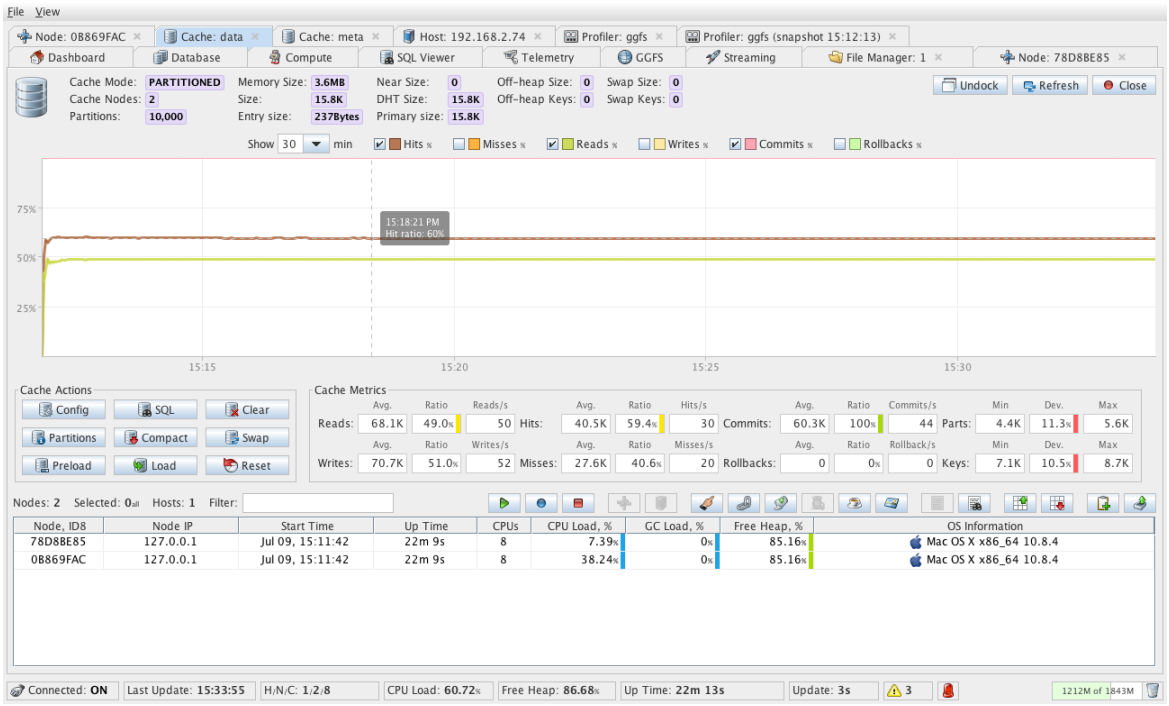
On Windows, run the same commands with `.bat` extension.

> **NOTE:** Visor GUI console has a much richer set of functionality over Visor command-based console. You should always prefer Visor GUI console whenever possible.

Here is an example of `Visor Data Grid Tab` which provides overall view on data grid.




Here is an example of `Visor Cache Tab` which provides view on individual cache.


File View
Node: 08869FAC
Cache: data
Cache: meta
Host: 192.168.2.74
Profiler: ggfs
Profiler: ggfs (snapshot 15:12:13)
Dashboard
Database
Compute
SQL Viewer
Telemetry
GGFS
Streaming
File Manager: 1
Node: 78D8BE85
Cache Mode: PARTITIONED
Cache Nodes: 2
Partitions: 10,000
Memory Size: 3.6MB
Size: 15.8K
Entry size: 237Bytes
Near Size: 0
DHT Size: 15.8K
Primary size: 15.8K
Off-heap Size: 0
Off-heap Keys: 0
Swap Size: 0
Swap Keys: 0
Undock
Refresh
Close
Show 30 min
Hits
Misses
Reads
Writes
Commits
Rollbacks
75%
50%
25%
15:18:21 PM
Hit ratio: 60%
15:15
15:20
15:25
15:30
Cache Actions
Config
SQL
Clear
Partitions
Compact
Swap
Preload
Load
Reset
Cache Metrics
Reads: 68.1K 49.0% 50
Hits: 40.5K 59.4% 30
Commits: 60.3K 100% 44
Parts: 4.4K 11.3% 5.6K
Writes: 70.7K 51.0% 52
Misses: 27.6K 40.6% 20
Rollbacks: 0 0% 0
Keys: 7.1K 10.5% 8.7K
Nodes: 2 Selected: 0 Hosts: 1 Filter:
Node, ID8 | Node IP | Start Time | Up Time | CPUs | CPU Load, % | GC Load, % | Free Heap, % | OS Information
78D8BE85 | 127.0.0.1 | Jul 09, 15:11:42 | 22m 9s | 8 | 7.39% | 0% | 85.16% | Mac OS X x86_64 10.8.4
0B869FAC | 127.0.0.1 | Jul 09, 15:11:42 | 22m 9s | 8 | 38.24% | 0% | 85.16% | Mac OS X x86_64 10.8.4
Connected: ON
Last Update: 15:33:55
H/N/C: 1/2/8
CPU Load: 60.72%
Free Heap: 86.68%
Up Time: 22m 13s
Update: 3s
3
1212M of 1843M