1. Apache Ignite In-Memory Accelerator For Apache Hadoop

Ignite 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

Ignite 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 Ignite 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 Apache Ignite Installation

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

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

If Ignite 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 Ignite Management Console, called Visor, and observe started nodes. To startup Visor, you should execute the following script:

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

3. Configuration

To configure Ignite nodes you can change configuration files at `config` directory of Ignite

installation. Those are conventional Spring files. Please refer to shipped configuration files and Ignite javadocs for more details.

3.1 Distributed File System Configuration

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

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

```
igfs://igfs_name@host_name
```

Where `igfs_name` is IgniteFS instance name, `host_name` is any host running Ignite node with that IgniteFS instance configured. For more details please refer to IgniteFS documentation.

3.2 Apache Hadoop Client Configuration

To run Apache Hadoop jobs with Ignite 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 Ignite templates or you can find these templates at `config/hadoop/core-site.ignite.xml` and `config/hadoop/mapred-site.ignite.xml` respectively and perform the needed configuration manually.

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

4. Running Apache Hadoop Job With Ignite In-Memory Accelerator

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

When all the configuration is complete and Ignite nodes are started, running Apache Hadoop

job will be the same as with conventional Apache Hadoop distribution except that all Ignite 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 IGFS 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

Ignite comes with CLI (command) based DevOps Managements Console, called Visor, delivering advance set of management and monitoring capabilities.

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

```
`bin/ignitevisorcmd.sh`
```

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