HowToConfigure

Outdated

Most of the information in this wiki page is outdated or inapplicable to Hadoop 2.x and will be deleted soon. Refer to the Configuration section of the latest 2.x stable release docs instead.

How To Configure Hadoop - *hadoop-0.15.0 and later*

Hadoop's configuration has been significantly changed in the hadoop-0.15.0 release (http://issues.apache.org/jira/browse/HADOOP-785).

Important changes:

1. Primarily hadoop-site.xml is now override-able by users via job.xml and other additional configuration resources. 2. mapred-default.xml is no longer supported. 3. Cluster-admins should judiciously use the new notion of *final* parameters to ensure critical config parameters aren't override-able (see 1).

Please consult the javadocs for the Configuration class in the hadoop-0.15.0 release for more details. See QuickStart and Hadoop Cluster Setup /Configuration for a description of Hadoop configuration for 0.21.0.

How To Configure Hadoop - *pre hadoop-0.15.0*

Primary XML Files

Hadoop is configured with a set of files. The files are loaded in the order listed in the table below, with the lower files in the table overriding the higher ones:

Filename	Description
hadoop-default.xml	Generic default values
mapred-default.xml	Site specific default values
job.xml	Configuration for a specific map/reduce job
hadoop-site.xml	Site specific value that can not be modified by the job

Look up path

Configuration files are found via Java's Classpath. Only the first instance of each file is used. The $HADOOP_CONF_DIR is added by the bin/hadoop script to the front of the path. When installing Hadoop on a cluster, it is best to use a conf directory outside of the distribution. That allows you to easily update the release on the cluster without changing your configuration by mistake.

hadoop-default.xml

This file has the default values for many of the configuration variables that are used by Hadoop. This file should never be in $HADOOP_CONF_DIR so that the version in the hadoop-*-core.jar is used. (Otherwise, if a variable is added to this file in a new release, you won't have it defined.)

mapred-default.xml

This file should contain the majority of your site's customization of Hadoop. Although this file name is prefixed with mapred, the default settings for the user maps and reduces are controlled by it.

Some useful variables are:

Name	Meaning
dfs.block.size	size in bytes of each data block in DFS
io.sort.factor	number of files input to each level in the merge sort
io.sort.mb	size of buffer to sort the reduce inputs in
io.file.buffer.size	number of bytes used for buffering io files
mapred.reduce.parallel.copies	number of threads fetching map outputs for each reduce
dfs.replication	number of replicas for each DFS block
mapred.child.java.opts	options passed to child task jvms
mapred.min.split.size	minimum number of bytes in a map input split

mapred.output.compress	Should the reduce outputs be compressed?

The rest of the valid property names and their default values can be found in the current docs.

job.xml

This file is never created explicitly by the user. The map/reduce application creates a JobConf, which is serialized when the job is submitted.

hadoop-site.xml

This file overrides any settings in the job.xml and therefore should be very minimal. Usually it just contains the address of the NameNode, the address of the JobTracker, and the port and working directories for the various servers.

Environment Variables

For the most part, you should only need to define $HADOOP_CONF_DIR. Other environment variables are defined in $HADOOP_CONF_DIR/hadoop-env.sh.

Variables in hadoop-env.sh, include:

Name	Meaning
JAVA_HOME	Root of the Java installation
HADOOP_HEAPSIZE	MB of heap for the servers
HADOOP_IDENT_STRING	User name of the cluster
HADOOP_OPTS	Extra arguments to the JVM
HADOOP_HOME	Hadoop release directory
HADOOP_LOG_DIR	Directory for log files
HADOOP_PID_DIR	Directory to store the PID for the servers
HADOOP_ROOT_LOGGER	Logging configuration for hadoop.root.logger. default: "INFO,console"
HADOOP_SECURITY_LOGGER	Logging configuration for hadoop.security.logger. default: "INFO, NullAppender"
HDFS_AUDIT_LOGGER	Logging configuration for hdfs.audit.logger. default: "INFO,NullAppender"

Log4j Configuration

Hadoop logs messages to Log4j by default. Log4j is configured via log4j.properties on the classpath. This file defines both what is logged and where. For applications, the default root logger is "INFO,console", which logs all message at level INFO and above to the console's stderr. Servers log to the "INFO, DRFA", which logs to a file that is rolled daily. Log files are named $HADOOP_LOG_DIR/hadoop-$HADOOP_IDENT_STRING-<server>.log.

For Hadoop developers, it is often convenient to get additional logging from particular classes. If you are working on the TaskTracker, for example, you would likely want
log4j.logger.org.apache.hadoop.mapred.TaskTracker=DEBUG in your log4j.properties.

Audit Logging

In 0.18 and later, one can enable audit logging from the Namenode. By default, events logged to this appender are forwarded to the Namenode log, which will radically increase the number of events emitted from that interface (see example). Audit events are emitted as a set of key=value pairs for the following keys:

Format

key	value								
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="efa17530-ecf0-44e5-97b0-c3a67b5594e4"><ac:plain-text-body><![CDATA[	ugi	<user>,<group>[,<group>]*	]]></ac:plain-text-body></ac:structured-macro>						
ip	<client ip address>								
cmd	(open	create	delete	rename	mkdirs	listStatus	setReplication	setOwner	setPermission)
src	<path>								
dst	(<path>	"null")							
perm	(<user>:<group>:<perm mask>	"null")							

Sample line of audit output:

```
<log4j header> ugi=wsmith,users,staff ip=/192.168.0.10 cmd=mkdirs src=/foo/bar dst=null perm=wsmith:staff:rwxr-xr-x
```

Example logging audit events to rolling log, syslog:

```
# Log at INFO level to DRFAAUDIT, SYSLOG appenders
log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=INFO,DRFAAUDIT,SYSLOG

# Do not forward audit events to parent appenders (i.e. namenode)
log4j.additivity.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=false

# Configure local appender
log4j.appender.DRFAAUDIT=org.apache.log4j.DailyRollingFileAppender
log4j.appender.DRFAAUDIT.File=/var/log/audit.log
log4j.appender.DRFAAUDIT.DatePattern=.yyyy-MM-dd
log4j.appender.DRFAAUDIT.layout=org.apache.log4j.PatternLayout
log4j.appender.DRFAAUDIT.layout.ConversionPattern=%d{ISO8601} %p %c: %m%n

# Configure syslog appender
log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender
log4j.appender.SYSLOG.syslogHost=loghost
log4j.appender.SYSLOG.layout=org.apache.log4j.PatternLayout
log4j.appender.SYSLOG.layout.ConversionPattern=%d{ISO8601} %p %c: %m%n
log4j.appender.SYSLOG.Facility=LOCAL1
```