Important Hadoop Daemon Properties

- Hadoop has a many number of configuration properties
- For any real world cluster, certain properties are essential.
- Such properties could be set in the Hadoop site files; namely
  - *core-site.xml*
  - *hdfs-site.xml*
  - *mapred-site.xml*

Important HDFS Daemon Proerties

Property Name	Type	Default Value	Description
fs.default.name	URI	file:///	The default filesystem. The URI defines the hostname and port that the namenode's RPC sever runs on. The default port is 8020.
dfs.name.dir	Comma-separated directory names	${hadoop.tmp.dir}/dfs/name	The list of directories where the namenode stores its persistent metadata. The namenode stores a copy of the metadata in each directory in the list.
dfs.data.dir	Comma-separated directory names	${hadoop.tmp.dir}/dfs/data	A list of directories where the datanode stores blocks. Each block is stored in only one of these directories.
fs.checkpoint.dir	Comma-separated directory names	${hadoop.tmp.dir}/dfs/names econdary	A list of directories where the secondary namenode stores checkpoints. It stores a copy of the checkpoint in each directory in the list.

Important MapReduce Daemons Properties

Property Name	Type	Default Value	Description
mapred.job.tracker	Hostname and port	local	The hostname and port that the jobtracker's RPC server runs on.
mapred.local.dir	Comma-separated directory names	${hadoop.tmp.dir}/mapred/local	A list of directories where MapReduce stores intermediate data for jobs. The data is cleared out when the job ends.
mapred.system.dir	URI	${hadoop.tmp.dir}/mapred/system	The directory relative to fs.default.name where shared files are stored during a job run
mapred.tasktracker.map.tasks.maximum	int	2	The number of map tasks that may be run on a tasktracker at any one time.
mapred.tasktracker.reduce.tasks.maximum	int	2	The number of reduce tasks that may be run on a tasktracker at any one time.

Important MapReduce Daemons Properties(Contd…)

Property Name	Type	Default Value	Description
mapred.child.java.opts	String	-Xmx200m	The JVM options used to launch the tasktracker child process that runs map and reduce tasks.
mapreduce.map.java.opts	String	-Xmx200m	The JVM options used for the child process that runs map tasks.
mapreduce.reduce.java.opts	String	-Xmx200m	The JVM options used for the child process that runs reduce tasks.

A typical Core-Site.XML

```
<?xml version="1.0"?> <!-- core-site.xml -->
  <configuration>
      <property>
      <name>fs.default.name</name>
      <value>hdfs://namenode/</value>
      <final>true</final>
      </property>
  </configuration>
```

A typical hdfs-site.xml

```
<?xml version="1.0"?> <!-- hdfs-site.xml -->
<configuration>
<property>
<name>dfs.name.dir</name>
 <value>/disk1/hdfs/name,/remote/hdfs/name</value> <final>true</final>
</property>

<property> <name>dfs.data.dir</name>
 <value>/disk1/hdfs/data,/disk2/hdfs/data</value> <final>true</final>
 </property>

 <property> <name>fs.checkpoint.dir</name>
 <value>/disk1/hdfs/namesecondary,/disk2/hdfs/namesecondary</value>
 <final>true</final>
 </property>
 </configuration>
```

A typical mapred-site.xml configuration file

```
<?xml version="1.0"?>
<!-- mapred-site.xml -->
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>jobtracker:8021</value>
<final>true</final>
</property>
 <property>
<name>mapred.local.dir</name>
<value>/disk1/mapred/local,/disk2/mapred/local</value>
<final>true</final>
</property>
 <property>
<name>mapred.system.dir</name>
<value>/tmp/hadoop/mapred/system</value>
<final>true</final>
</property>
```

A typical mapred-site.xml configuration file

```
<property>
<name>mapred.tasktracker.map.tasks.maximum</name>
<value>7</value>
<final>true</final>
</property>

<property>
<name>mapred.tasktracker.reduce.tasks.maximum</name>
<value>7</value>
<final>true</final>
</property>

<property>
<name>mapred.child.java.opts</name>
<value>-Xmx400m</value>
<!-- Not marked as final so jobs can include JVM debugging options -->
</property>
</configuration>
```

Hadoop Daemon Address and Ports

- Hadoop daemons generally run both an RPC server for communication between daemons and an HTTP server to provide web pages for human consumption
- Each server is configured by setting the network address and port number to listen on
- By specifying the network address as 0.0.0.0, Hadoop will bind to all addresses on the machine.
- port number of 0 instructs the server to start on a free port

RPC server properties

Property name	Default value	Description
fs.default.name	file:///	When set to an HDFS URI, this property determines the namenode's RPC server address and port. The default port is 8020 if not specified.
dfs.datanode.ipc.address	0.0.0.0:50020	The datanode's RPC server address and port.
mapred.job.tracker	local	When set to a hostname and port, this property specifies the jobtracker's RPC server address and port. A commonly used port is 8021.
mapred.task.tracker.report.address	127.0.0.1:0	The tasktracker's RPC server address and port. This is used by the tasktracker's child JVM

HTTP Server Properties

Property name	Default value	Description
mapred.job.tracker.http.address	0.0.0.0:50030	The jobtracker's HTTP server address and port
mapred.task.tracker.http.address	0.0.0.0:50060	The tasktracker's HTTP server address and port
dfs.http.address	0.0.0.0:50070	The namenode's HTTP server address and port
dfs.datanode.http.address	0.0.0.0:50075	The datanode's HTTP server address and port
dfs.secondary.http.address	0.0.0.0:50090	The secondary namenode's HTTP server address and port

Other Hadoop Properties

S.No	Property	File Specified
1	Buffer Size - Hadoop uses a buffer size of 4 KB (4,096 bytes) for its I/O operations. Now a days it can be extended upto 128 KB	Set this using the io.file.buffer.size property in *core-site.xml*.
2	HDFS Block Size-The HDFS block size is 64 MB by default, but many clusters use 128 MB (134,217,728 bytes) or even 256 MB (268,435,456 bytes) to ease memory pressure on the namenode and to give mappers more data to work on.	Set this using the dfs.block.size property in *hdfs-site.xml*.
3	Trash-Hadoop filesystems have a trash facility, in which deleted files are not actually deleted, but rather are moved to a trash folder, where they remain for a minimum period before being permanently deleted by the system.	The minimum period in minutes that a file will remain in the trash is set using the fs.trash.interval configuration property in *core-site.xml*. By default, the trash interval is zero, which disables trash.