USING HADOOP WITH CEPHFS

The Ceph file system can be used as a drop-in replacement for the Hadoop File System (HDFS). This page describes the installation and configuration process of using Ceph with Hadoop.

DEPENDENCIES

- CephFS Java Interface
- Hadoop CephFS Plugin

Important: Currently requires Hadoop 1.1.X stable series

INSTALLATION

There are three requirements for using CephFS with Hadoop. First, a running Ceph installation is required. The details of setting up a Ceph cluster and the file system are beyond the scope of this document. Please refer to the Ceph documentation for installing Ceph.

The remaining two requirements are a Hadoop installation, and the Ceph file system Java packages, including the Java CephFS Hadoop plugin. The high-level steps are two add the dependencies to the Hadoop installation `CLASSPATH`, and configure Hadoop to use the Ceph file system.

CEPHFS JAVA PACKAGES

- CephFS Hadoop plugin (hadoop-cephfs.jar)

Adding these dependencies to a Hadoop installation will depend on your particular deployment. In general the dependencies must be present on each node in the system that will be part of the Hadoop cluster, and must be in the `CLASSPATH` searched for by Hadoop. Typically approaches are to place the additional `jar` files into the `hadoop/lib` directory, or to edit the `HADOOP_CLASSPATH` variable in `hadoop-env.sh`.

The native Ceph file system client must be installed on each participating node in the Hadoop cluster.

HADOOP CONFIGURATION

This section describes the Hadoop configuration options used to control Ceph. These options are intended to be set in the Hadoop configuration file *conf/core-site.xml*.

Property	Value	Notes
fs.default.name	Ceph URI	ceph://[monaddr:port]/
ceph.conf.file	Local path to ceph.conf	/etc/ceph/ceph.conf
ceph.conf.options	Comma separated list of Ceph configuration key/value pairs	opt1=val1,opt2=val2
ceph.root.dir	Mount root directory	Default value: /
ceph.mon.address	Monitor address	host:port
ceph.auth.id	Ceph user id	Example: admin
ceph.auth.keyfile	Ceph key file	
ceph.auth.keyring	Ceph keyring file	
ceph.object.size	Default file object size in bytes	Default value (64MB): 67108864
ceph.data.pools	List of Ceph data pools for storing file.	Default value: default Ceph pool.
ceph.localize.reads	Allow reading from file replica objects	Default value: true

SUPPORT FOR PER-FILE CUSTOM REPLICATION

The Hadoop file system interface allows users to specify a custom replication factor (e.g. 3 copies of each block) when creating a file. However, object replication factors in the Ceph file system are controlled on a per-pool basis, and by default a Ceph file system will contain only a single pre-configured pool. Thus, in order to support per-file replication with Hadoop over Ceph, additional storage pools with non-default replications factors must be created, and Hadoop must be configured to choose from

these additional pools.

Additional data pools can be specified using the `ceph.data.pools` configuration option. The value of the option is a comma separated list of pool names. The default Ceph pool will be used automatically if this configuration option is omitted or the value is empty. For example, the following configuration setting will consider the pools `pool1`, `pool2`, and `pool5` when selecting a target pool to store a file.

```
<property>
  <name>ceph.data.pools</name>
  <value>pool1,pool2,pool5</value>
</property>
```

Hadoop will not create pools automatically. In order to create a new pool with a specific replication factor use the `ceph osd pool create` command, and then set the `size` property on the pool using the `ceph osd pool set` command. For more information on creating and configuring pools see the RADOS Pool documentation.

Once a pool has been created and configured the metadata service must be told that the new pool may be used to store file data. A pool is be made available for storing file system data using the `ceph fs add_data_pool` command.

First, create the pool. In this example we create the `hadoop1` pool with replication factor 1.

```
ceph osd pool create hadoop1 100
ceph osd pool set hadoop1 size 1
```

Next, determine the pool id. This can be done by examining the output of the `ceph osd dump` command. For example, we can look for the newly created `hadoop1` pool.

```
ceph osd dump | grep hadoop1
```

The output should resemble:

```
pool 3 'hadoop1' rep size 1 min_size 1 crush_rule 0...
```

where 3 is the pool id. Next we will use the pool id reference to register the pool as a data pool for storing file system data.

```
ceph fs add_data_pool cephfs 3
```

The final step is to configure Hadoop to consider this data pool when selecting the target pool for new files.

```
<property>
        <name>ceph.data.pools</name>
        <value>hadoop1</value>
</property>
```

POOL SELECTION RULES

The following rules describe how Hadoop chooses a pool given a desired replication factor and the set of pools specified using the `ceph.data.pools` configuration option.

1. When no custom pools are specified the default Ceph data pool is used.
2. A custom pool with the same replication factor as the default Ceph data pool will override the default.
3. A pool with a replication factor that matches the desired replication will be chosen if it exists.
4. Otherwise, a pool with at least the desired replication factor will be chosen, or the maximum possible.

DEBUGGING POOL SELECTION

Hadoop will produce log file entry when it cannot determine the replication factor of a pool (e.g. it is not configured as a data pool). The log message will appear as follows:

```
Error looking up replication of pool: <pool name>
```

Hadoop will also produce a log entry when it wasn’t able to select an exact match for replication. This log entry will appear as follows:

```
selectDataPool path=<path> pool:repl=<name>:<value> wanted=<value>
```