Tez and Cloud Storage

Problem

Hadoop Compatible File System implementations are used to access and process data that lives on Cloud Storage platforms such as Azure (WASB), AWS (S3), and Google Compute (GS).

Unlike the hdfs and viewfs filesystems which are always available, configuring these additional filesystems for Tez is slightly more involved, due to the way the working set of JARs are transferred in Tez.

The primary requirement of course, is the core-site.xml to have the fs implementations called out (for something like S3, it would be the key **fs.s3a.impl**).

Solution (in Tez 0.8)

Specifically for Azure and AWS, the Hadoop versions after 2.7.0 include the HCFS jars as part of the hadoop install. This can be adopted into the build by rebuilding Tez with *mvn package -Paws -Pazure .*

Solution (in Tez 0.5 - 0.7)

There are 3 class loader contexts for Tez, which all need to have these JARs accessible to work correctly.

The first one is the client-side, which is the easiest, since you can confirm that by doing a "hadoop fs -ls s3a://<bucket>/" to make sure that the local installation, keys etc are setup correctly.

The second and third are the ApplicationMaster and TaskAttempt classpath. The correct way to ship a JAR to both these locations is to upload the dependencies for the HCFS to an HDFS location (say, /apps/tez/aux-jars).

That path has to be appended to the t**ez.aux.uris** as **${fs.defaultFS}/apps/tez/aux-jars/** - both the ApplicationMaster and all TaskAttempts will automatically localize all the files from that path, so that you access the cloud storage HCFS without any further changes to your setup.

The same location can be used to side-load JARs into the latter classpath contexts, whether they are for Cloud HCFS, other HCFS (Apache Silk, Tachyon etc), or other data access dependencies (mongodb/elasticsearch/solr).