HadoopShims

Shims for various Hadoop Versions

One of the main issues when dealing with Hadoop is that the various different versions could have potential minor incompatibilities and to a greater extent, the absence/presence of a variety of features. For example, when using Tez, the main issue till date has been the handling of YARN Timeline Server. Hadoop 2.4 introduced YARN Timeline support. However, Security/ACLs support was only introduced in Hadoop 2.6 and required different APIs to be invoked.

In Tez, to work with different Hadoop versions, one must do the following:

- Set "hadoop.version" in the top-level pom.xml to match the hadoop version being compiled against.
- Enable the hadoop profile based on your hadoop version:
    - hadoop26 is the default profile (active by default) and works with hadoop versions 2.6.x and 2.7.x
    - hadoop24 works with versions 2.4.x and 2.5.x and requires hadoop26 profile to be disabled
    - hadoop22 works with versions 2.2.x and 2.3.x and requires hadoop26 profile to be disabled
    - hadoop28 profile will work with version 2.8.x.
- Please refer to BUILDING.txt at the top of the source tree for the most up-to-date information.

For the most part, the features that are version dependent are usually plugins and they can be cleanly separated into different modules. The profiles above largely decide which modules will be built or not built based on the hadoop version. However, they are some features that require APIs that are a bit more invasive in nature and a plugin approach does not really work well. A couple of examples of this are:

- HDFS-9184: Logging HDFS operation's caller context into audit logs
- YARN-2448: RM should expose the resource types considered during scheduling when AMs register

To make use of this APIs, we have introduced a shim layer as part of TEZ-2924. The basic shim interface defines the various APIs whose implementations are version-specific and the version-specific shim implementation provides the runtime implementation to match the version of Hadoop. The Hadoop shims are discovered via ServiceLoaders (no configuration knobs required). The Tez framework will cycle through all providers that it finds on its classpath, query each one to see if it supports the current version of Hadoop and pick the first Shim that does.

To implement a distro-specific shim:

- Implement/Extend the HadoopShim class
- Implement/Extend the HadoopShimProvider class
- Create a jar for the implementations with the required provider-configuration file under META-INF/services.
- Add this jar to the tez tarball defined via "tez.lib.uris" or via "tez.aux.uris"
- As mentioned earlier, if there are other shims that match the Hadoop version, your specific shim may not be picked up at runtime. To avoid this issue, you may need to remove the "hadoop-shim-2.*" jars from the tez tarball as needed.

For more questions/clarifications, please send questions to dev@tez.apache.org