

DevelopingOnTrunkAfter279Merge

MR-279 branch is merged into mapreduce trunk and this changes things a bit for developing on mapreduce.

You can get all the help that is needed from (1) the README at <http://svn.apache.org/repos/asf/hadoop/common/trunk/hadoop-mapreduce-project/hadoop-yarn/README> and (2) the INSTALL file at <http://svn.apache.org/repos/asf/hadoop/common/trunk/hadoop-mapreduce-project/INSTALL>. Reproducing some of those contents here for the short-term lookup.

Checking out source code

svn checkout <http://svn.apache.org/repos/asf/hadoop/common/trunk/hadoop-mapreduce>

Directory structure

trunk/

- - hadoop-mapreduce (was mapreduce before)

trunk/hadoop-mapreduce - Classic code. JT/TT reside here

- - build.xml - src

trunk/hadoop-mapreduce/ - New code related to yarn reside here.

- - assembly - pom.xml - hadoop-mr-client - hadoop-yarn - Yarn APIs, libraries, and server code
 - - hadoop-yarn-api
 - - hadoop-yarn-common – hadoop-yarn-server - Server code, [ResourceManager](#), [NodeManager](#),

server libraries and tests.

- — hadoop-yarn-server-common
- — hadoop-yarn-server-nodemanager
- — hadoop-yarn-server-resourcemanager
- — hadoop-yarn-server-tests
- - hadoop-mr-client - [MapReduce](#) server and client code
 - - hadoop-mapreduce-client-app
 - - hadoop-mapreduce-client-core
 - - hadoop-mapreduce-client-jobclient
 - - hadoop-mapreduce-client-common
 - - hadoop-mapreduce-client-hs
 - - hadoop-mapreduce-client-shuffle

Building

Building yarn code and install into the local maven cache.

- - mvn clean install - In case you want to skip the tests run: mvn clean install -DskipTests

Building classic code once yarn code is built.

- - ant veryclean jar jar-test -Dresolvers=internal

Eclipse

- 1) For hacking on the new yarn+MR code in eclipse, you should run

"mvn eclipse:eclipse" and then import the checked out source root as a maven project.

- 2) For developing on classic JT/TT code, running "ant eclipse" and

importing as java project should continue to work.

FAQ

1) Build fails with "[ERROR] Failed to execute goal org.codehaus.mojo:make-maven-plugin:1.0-beta-1:autoreconf (autoreconf) on project hadoop-yarn-server-nodemanager: autoreconf command returned an exit value != 0. Aborting build; see debug output for more information. -> [Help 1]"

This means that you don't have the autotool chain necessary for building the native code. You will need to build the native code to build [LinuxContainerExecutor](#) needed for running the cluster with security enabled.

If you are not interested in running it with security enabled, you can skip building the native code by passing "-P-cbuild".

2) Build fails with "[ERROR] Failed to execute goal org.codehaus.mojo:exec-maven-plugin:1.2:exec (generate-sources) on project hadoop-yarn-api: Command execution failed. Process exited with an error: 1 (Exit value: 1) -> [Help 1]"

This means that you don't have protoc installed on your machine. Installing protoc (and adding a non-standard installation directory to your LD_LIBRARY_PATH) should get it working.

3) Getting errors while compiling protobuf 2.4 For compiling YARN, You need to have protobuf 2.4.0a or higher (Download from <http://code.google.com/p/protobuf/downloads/list>). Install protobuf 2.4.0a or higher (Download from <http://code.google.com/p/protobuf/downloads/list>)

- install the protoc executable (configure, make, make install)
- install the maven artifact (cd java; mvn install) Installing protoc requires gcc 4.1.x or higher. If the make step fails with (Valid until a fix is released for protobuf 2.4.0a)
./google/protobuf/descriptor.h:1152: error:
`google::protobuf::internal::Mutex*google::protobuf::DescriptorPool::mutex_'
is private
Replace descriptor.cc with <http://protobuf.googlecode.com/svn-history/r380/trunk/src/google/protobuf/descriptor.cc>

Hope that helps. If you run into issues, please send an email or create a JIRA issue.

Building on Linux

Linux distributions may install protocol buffers via their repositories. This can save all the installation problems, or it can cause extra problems

Look in your package manager for any "protoc" or "protocol buffers" compiler and library; check the version.

If the version is 2.4.0 or later, select these and install them.

If the version is below that, do not install them, and uninstall them if they are present, then follow the installation instructions above.

Problem: Protoc compiler fails on Linux if you see an error like

```
protoc: error while loading shared libraries: libprotobuf.so.7: cannot open shared object file: No such file or directory
```

It may be that the installation already has an older copy of protocol buffers installed, and this is getting in the way of the newly installed version

Test: run `protoc --version` to see what version is picked up. Here is the example of a valid version

```
$ protoc --version libprotoc 2.4.1  
}}}
```

If a version older than 2.4.0 appears, you need to uninstall protoc and possibly libprotoc using your platform's package management tools.