How to Contribute

Be involved!

Contributors should join the Giraph mailing lists. In particular:

- The user list (to help others)
- The commit list (to see changes as they are made)
- The dev list (to join discussions of changes) – This is the best place to understand where we are headed.

Please keep discussions on the lists about Giraph so that everyone benefits. Emailing individual committers with questions about specific Giraph issues is discouraged. See http://people.apache.org/~hossman/#private_q.

Technical process

1) Checkout the Apache Giraph trunk

```bash
$ git clone git://git.apache.org/giraph.git
```

(committers' repo)

```bash
$ git clone https://git-wip-us.apache.org/repos/asf/giraph.git
```

2) Create a JIRA on https://issues.apache.org/jira/browse/GIRAPH

3) Work on your change. When it is working and you are ready to submit, make sure that 'mvn clean install' succeeds. 'mvn clean install' will insure that all the local unit tests pass, the license headers are there, and basic checks are done to insure code conventions are met. You should also verify that the tests work on a single node Hadoop instance. You can get instructions from http://ssc.io/running-giraphs-unit-tests-in-pseudo-distributed-mode/.

4) Create a patch with the appropriate JIRA number (where XXX is the JIRA number)

```bash
$ svn diff > GIRAPH-XXX.patch
```

5) Attach the patch to the JIRA by going to the JIRA, then clicking on "More actions" -> "Attach Files".

6) Make sure to edit the JIRA to select the Patch Available check box.

7) (Optional) Some folks find it easier to review code on reviewboard. You can go to https://reviews.apache.org/ and click "New Review Request", select the giraph-git repository, set the base repository to "git://git.apache.org/giraph.git", and enter your diff from step 4). After you submit it, fill in the necessary information and then in the bugs section put "GIRAPH-XXX", which matches your JIRA number. This will automatically post it to the JIRA when you’re done submitting it.

8) Wait for a committer to review your code. If there are changes to be made, go to step 3). Otherwise your code will be committed.

ReviewBoard notes

If you choose or are requested to open a reviewboard for your code, please note that the base path is /trunk.