ProjectSplit

After the Hadoop 0.20 branch was created, Hadoop was split in 3 sub-projects within the website, Jira, and mailing lists: Common, HDFS, and MapReduce.

A change to that project split has been proposed on general@. Here are questions and answers raised in that discussion.

1. What is being proposed?

- Move `common`, `mapreduce` and `hdfs` directories to be sibling directories under `trunk`, `branches`, and `tags`.
- **CURRENT SVN REPO LAYOUT:** {{hadoop / {common, mapreduce, hdfs} / { trunk, branches, tags }}}
- **PROPOSED SVN REPO LAYOUT:** `hadoop / { trunk, branches/*, tags/* } / {common, mapreduce, hdfs}`
- **Additional changes:**
- remove the old `hadoop / {pig, hive, and zookeeper}` directories
- move `hadoop / {common, mapreduce, hdfs} / site` to `hadoop / site / {common, mapreduce, hdfs}`
  - move `hadoop / hdfs / branches / HDFS-*` to `hadoop / branches / HDFS-*`
  - move `hadoop / mapreduce / branches / MAPREDUCE-*` to `hadoop / branches / MAPREDUCE-*`
- This effort is done as HADOOP-7106

2. Why? Don't we want to separate these 3 projects further and release them separately?

- We're a long way from releasing these 3 projects independently. Given that, they should be branched and released as a unit. This SVN structure enforces that and provides a more natural place to keep any top level build and pkg scripts that operate across all 3 projects. The proposed change would allow you can make checkouts, branches and tags with a single command.

3. Is this undoing the project split?

- No. The proposal is NOT a full undo of the project split. It does not put common, hdfs, and mapreduce back together into a single source tree. This proposal might be better described as a tweak or a bug fix to the existing project split.

4. How do we avoid introducing dependencies between the projects?

- Automated project builds (via Hudson) will continue to build and mvn deploy these 3 projects separately. A new step, however, may be introduced to automatically package the projects into a single releasable artifact.
- Code reviewers will need to continue to look for undesirable dependencies across projects.
- A tool (like JDepend) could be used to enforce certain dependency rules. Takers?

5. The committer list for each of the sub projects today is different. How do we reconcile them?

- We keep the status quo. Today all Hadoop committers technically have permission to commit to all 3 project trees but we rely on the honor system that committers will only commit to the projects for which they have permission. This will not change under the current proposal.

6. I'm a git user. Will this screw up my git history?

- A script will be provided to make the git history look sane. [Todd: provide link to relevant Jira]

7. As a developer, what specific steps do I need to take to re-base my workspace?

- FILL IN

8. As a release engineer, what specific steps do I need to take to update my continuous integration servers?

- FILL IN

9. How will contributing be affected?

- FILL IN
- RELATED: When we have a patch that is mainly HDFS or MR focused but will need changes across projects, can we just put up one patch in HDFS/MR or do we still need to open a parallel common JIRA?

10. How will committing be affected?

- FILL IN

11. Will the project be mavenized at the same time?

- No. That is a separate issue needing separate discussion.

12. What documentation needs udpate due to this change?

- HowToContribute
- http://hadoop.apache.org/common/version_control.html
- http://hadoop.apache.org/hdfs/version_control.html
- http://hadoop.apache.org/mapreduce/version_control.html
- ADD ANY OTHERS YOU FIND