

# Idea: Git plugin performance improvement

## Project Goals:

- Improve Jenkins git plugin performance by fixing known issues in performance critical areas.
- Implement performance comparisons between the command line git and JGit implementations using the Java Microbenchmark Harness.
- Improve the performance of git operations that use the command line git implementation by selectively replacing CLI implementations with JGit when benchmarks show a significant performance gain.

## Fix Known Issues - Avoid Redundant Fetch

The Jenkins git plugin copies remote git repositories into Jenkins workspaces on agents as part of a 'checkout'. The checkout creates an empty git repository in the workspace, configures it with 'git config', and populates it with 'git fetch'. Unfortunately, the most commonly used path through the code will call 'git fetch' twice.

The second call to 'git fetch' is useless when it is using the same arguments as the first call. It wastes server time, network bandwidth, and job time. With large repositories, that waste of time may be a minute or more.

The project idea to avoid the redundant fetch is:

1. Implement at least one automated test to detect the redundant fetch
2. Implement changes in the git plugin and the git client plugin to avoid the second call to 'git fetch' when using the same arguments as the first call to 'git fetch'.

## Performance Comparisons

1. Compare [CliGitAPIImpl](#) and link: [JGitAPIImpl](#) using the [Java Microbenchmark Harness](#) by writing tests that compare the performance of the same operations using two different implementations
2. Present the results of those comparisons.
3. Use the results of the comparisons and common use cases to prioritize operations for performance improvement.

## Performance Improvements

1. Confirm that existing automated tests adequately verify the existing implementations. If they do not adequately verify implementation behaviors, write additional automated tests to verify the existing implementations
2. Replace a command line git implementation with a JGit implementation in cases where benchmarks show JGit is faster
3. Confirm that benchmark tests show the performance improvement.
4. Confirm that automated tests confirm no loss of functionality.

## Quickstart

The project idea is expected to require changes in both the git client plugin and the git plugin.

Install a git client plugin development environment by following the [contributing instructions](#). Compile the plugin, run its automated tests, and confirm that the automated tests are passing. Enable coverage reporting and review the coverage report.

Install a git plugin development environment by following the [contributing instructions](#). Compile the plugin, run its automated tests, and confirm that the automated tests are passing. Enable coverage reporting and review the coverage report.

Consider implementing a fix for one of the [newbie friendly issues](#).

## Links

- [Jenkins GSoC mailing list](#) discussion of redundant fetch removal idea
- [Jenkins GSoC mailing list](#) discussion of JMH benchmark idea
- [JENKINS-49757](#) - Git plugin calls fetch twice per checkout
- [JENKINS-56404](#) - Remove redundant fetch

## Skills to study/improve

- Basic knowledge of Jenkins (as a user)
- Basic knowledge of Git (as a user)
- Basic knowledge of Java programming language

The project will benefit many of the 250 000+ worldwide installations of the git plugin

## Project Metadata

**Created on:** 2020

**Goal:** Improve git plugin performance

**Champion:** Mark Waite

**Champion Github Id and link:** <https://github.com/MarkEWaite>

**Champion Jenkins JIRA/LDAP id:** markewaite

**Champion Time Zone:** UTC-7

**Champion Role:** I am making this proposal as a **mentor**

**Project Category:** plugin

**SIG/Subproject:** Platform

**Project Gitter chat room:** <https://gitter.im/jenkinsci/git-plugin>

## Potential Mentors

The potential mentors are:

- 1) Mark Waite - confirmed as mentor
- 2) Francisco Fernandez - mentor confirmation pending

3) Mentors could come from the other git plugin developers (Rene Schiebe, Jacob Keller, ...)