

Git Cache Maintenance - GSoC 2022

Meeting notes, see [project plan](#) for details

[Class Diagram](#)

[Flowchart Diagram](#) (Needs to be updated)

Jul 15, 2022

Attending

- Hrushikesh Rao
- Rishabh
- Mark Waite

Agenda

- Action items
 - Updating the project page details (done and merged)
 - UI specific doubts:
 - Figure out how to make the help tooltip work on the UI - [PR](#)
- Presentation for July 21, 2022 (midpoint)
 - 15 minutes
 - Demonstration
 - Load some caches on his computer, intentionally clutter the repository
 - Unpack all the objects from the pack files
 - Run the maintenance task to see the log
 - Completed git maintenance implementation for wide range of CLI git versions
 - 2.30+ provides a git maintenance command
 - Prior to 2.30, fake it with direct calls to git
 - Some cases simply don't have feature (e.g. commit-graph)
 - Approach / structure / high level diagram to explain the concepts
 - Versions that we support and why that matters to users
 - Challenges encountered?
 - Handling private repositories
 - Handling cache conflicts
 - Maintenance not implemented in git, but we need it
 - What's in the next phase?
 - Displaying progress, results, etc.

- Test drive the presentation next Wednesday during our regular session
- Test failures on ci.jenkins.io
 - Needs more code review, use the failing tests to better understand the code
- Maintaining caches that have private repositories (credentials)
 - Part of phase 2 of the project
- Cache cleanup from the HashSet
 - No process that removes cache entries from the HashSet
- Testing the plugin implementation

Jul 12, 2022

Attending

- Hrushikesh Rao
- Rishabh
- Mark Waite

Agenda

- Action items
 - Updating the project page details (done and merged)
 - UI specific doubts:
 - Figure out how to make the help tooltip work on the UI - [PR](#)
- Testing the plugin implementation - 30 minutes
 - Operations
 - Prefetch
 - Public repositories
 - Commit graph
 - Public repositories
 - Private repositories
 - Garbage collection
 - Public repositories
 - Private repositories
 - Loose objects
 - Public repositories
 - Private repositories
 - Incremental repack
 - Public repositories
 - Private repositories
 - Things to watch
 - System load - how is the overall system performing
 - Execution time of the maintenance operation
 - Impact of maintenance on other jobs
 - Start a git plugin test job

- Start a git client plugin test job
- Scan multibranch repositories
- Start a maintenance task
- Scan multibranch repositories while maintenance is running
 - Monitor the logs
- Cache locks during maintenance
 - Attempt a multibranch scan during a long garbage collection
 - If the repository is already locked, does the maintenance correctly wait for lock on that cache?
 - Are locks released on a cache when maintenance task finishes work with that cache?
- Presentation for July 21, 2022 (midpoint) - 30 minutes
- Maintaining caches that have private repositories (credentials)
- Cache cleanup from the HashSet
 - No process that removes cache entries from the HashSet

Jun 14, 2022

Attending

- Hrushikesh Rao
- Mark Waite

Agenda

- Action items
 - Updating the project page details (done and merged)
 - UI specific doubts:
 - Figure out how to make the help tooltip work on the UI - [PR](#)
-

Jun 8, 2022

Attending

- Hrushikesh Rao
- Mark Waite
- Rishabh Budhouliya

Agenda

- Action item
 - Updating the project page details
 - UI specific doubts:
 - Figure out how to make the help tooltip work on the UI - [PR](#)
 - Horizontally align buttons in UI
 - Create a design document for the project

- How to manage maintenance tasks - discussing the Build Discarder Plugin strategy
 - Dividing the process management into two steps:
 - Step 1: Take user's input to schedule tasks
 - Step 2: Build intelligence by taking system metrics as an input to manage/schedule processes
 - Difference in duration of the sub-process is a heuristic to used to gauge utilization
- Discussing 2 strategies to implement the project
 - Global Build Discarder plugin approach
 - Limits scheduling frequency to an hourly fashion
 - Parameterized Plugin using Cron Syntax approach
 - Provides more customizability for the admin to schedule tasks
- Cron Syntax would afford the flexibility for users to select a schedule when the Jenkins system is idle hence avoiding the liability of this feature to overload the system by background tasks which consume considerable amount of computation power (for example: git gc)
- We should only do caches that are maintained by the Jenkins core itself on the controller and not job workspaces
- Sanity check at the implementation level to make sure the user defined schedule is being run by this feature.
- Safeguards at the UI level to check sanity of the scheduling

Jun 1, 2022

Attending

- Hrushikesh Rao
- Rishabh Budhouliya

Agenda

- Action item
 - Updating the project page details
 - UI specific doubts:
 - Figure out how to make the help tooltip work on the UI - [PR](#)
 - Horizontally align buttons in UI
 - Create a design document for the project
 - Descriptor pattern explanation
- Questions and answers
 - Working on UI draft PR
 - Divide it into UI work and backend work
 - Add user input for controlling the lifecycle of the tasks
 - Display list of tasks
 - Display progress/history of tasks run
 - Filter repo by name/size
 - Backend
 - Start taking a look at Build Discarder code base
 - How would you start and manage a separate process in the background?

- Implement a queue based mechanism to run tasks
-

Git Plugin -> context -> a single repo

Class: GitSCMSource a single repo

Code -> context -> all repos within the jenkins system

May 27, 2022

Attending

- Mark Waite
- Hrushikesh Rao
- Rishabh Budhouliya

Agenda

- Action item
 - Upload the recording of previous meeting
- Questions and answers
 - Architectural concerns - how should we proceed?
 - Cache maintenance happens in the background (runs on a separate thread async in the Jenkins controller, running a separate git process to do the work)
 - Use a queue to keep a relatively few maintenance tasks running concurrently (set some maximum number of tasks running)
 - Control resource consumption with the queue
 - If the user mistakenly scheduled too many maintenance tasks at the same time, do not overload the controller
 - We don't know the duration of the maintenance task (git gc of the Linux kernel takes much longer than fetch of a small repository)
 - Dequeue an element, perform the tasks in that element, then take the next maintenance tasks
 - Edge case
 - Garbage collection maintenance task is scheduled, will take 60 minutes
 - Prefetch maintenance task is scheduled, will wait in the queue
 - Do we need to handle duplicates that are in queue
 - A queued Jenkins Freestyle build with its parameters will be replaced (not executed) if a new build is scheduled with the same parameters
 - Discard duplicates rather than add them to queue
 - Don't schedule a second gc of a repo when a gc is already scheduled or in progress

- Maintenance task uniquely identified by (queue contains only unique tasks)
 - Repository it is processing
 - Task it is performing
 - Is there a way to list all the caches on a controller?
 - Find a cache for a repository - AbstractGitSCMSource
 - No obvious method to iterate all caches, add one
 - If you need a wider sample of Jenkins configuration, see
 - <https://github.com/MarkEWaite/docker-lfs/tree/lfs-with-plugins>
 - Jenkins controller with several interesting jobs and job types
 - Build it with “docker_build.py”
 - Start it with “docker_run.py”
 - Architecture of the SCM system
 - [Writing an SCM plugin](#)
 - [SCM consumer guide](#)
 - [SCM implementation guide](#)
 - Maintenance tasks are “above” the concept of a Jenkins job rather than within the concept of a Jenkins job
 - Configured at a global level
 - Should we safeguard the user from themselves so that they cannot schedule jobs to run too frequently?
 - A large Jenkins controller (ci.jenkins.io has 2000 multibranch pipelines, each with 5-50 jobs), if I ask to schedule hourly, 10000 maintenance may still not complete
 - Should we graph queue length over time so admins can see if they are scheduling? See the “load statistics” graph
 - Add the maintenance task to the queue
 - Do we iterate over tasks (gc, prefetch, commit-graph, ...) and process all repositories or do we iterate over repositories (needs much more complicated user interface)
 - Prefer to perform a task on all repositories, keeps UI simple, avoids prompting the user to choose which repositories should be scheduled
 - Could extend the user interface to allow exclusion by filtering
 - Start with a global configuration page
 - See the sample UI that is in the proposal
 - Implement the sample UI using the design library
 - Next session in a week, Mark will miss the session, Rishabh send the URL, record the session

May 24, 2022

Attending

- Mark Waite
- Hrushikesh Rao
- Rishabh Budhouliya

Agenda

- Goals for community bonding
 - Assure that Hrushi is ready to start the coding by Jun 13
 - Comfortable with mentors
 - Comfortable with code review, test driven development, documented, etc.
 - Write a failing test
 - Write the code that fixes the failing test
 - Refactor “mercilessly” to simplify the code
 - Commit (and optionally push so that CI evaluates)
 - Tests expose edge cases in ways that thinking does not
 - Your feature release is more than designing and merging a PR
 - Tests that validate useful scenarios
 - Expectations that you and mentors have of the code
 - Tests assert the correctness of the code
 - Tests help us check multiple platforms (Windows, Linux, BSD, macOS) and multiple CLI git versions (1.8 to 2.36)
 - Git plugin is installed on 285 000 controllers
 - Many of those controllers are critical to the business they support
 - If we break it, they will shout
 - People expect behavioral compatibility from version to version for their use cases
 - Confident sharing updates during GSoC office hours
 - Complete the [checklist from org admins](#)
 - Hrushi - Update project details on jenkins.io
 - Mark - Review Hrushi’s pull requests, discuss ways to ease code review
 - All - agree on communication channels ([Gitter?](#), [Community?](#), other?)
 - <https://gitter.im/jenkinsci/git-plugin> is our preferred chat channel
 - More permanent communications on community.jenkins.io
 - Replacement for mailing lists
 - Easier place to post “blog-like” items
 - Project blog posts written to www.jenkins.io
 - Asciiidoc and content management with GitHub
 - All - scheduling (exams, vacations, etc.)
 - Plan for presentations
 - Hrushi - Phase 1
 - Full Implementation of the Git Maintenance task. This includes support for legacy versions of git which doesn’t support git maintenance tasks by default. (No JGIT)
 - The tests for maintenance tasks would be written when developed using TDD.

- Documentation for the Git Maintenance Task.
 - Blog post to accompany the recording
 - Hrushi - Phase 2 / end of project
 - Display Execution Status for maintenance tasks in the configuration page.
 - A way to exclude repositories for the maintenance tasks.
 - Update the ReadMe of the GitPlugin by adding instructions on how to use git maintenance.
 - // JGit has many missing features. Need to discuss its implementation with mentors.
 - Internationalization
 - Blog post to accompany the recording
 - Scheduling
 - [GSOC timeline](#)
 - May 20 - Jun 12 - Community bonding
 - Jun 13 - Jul 25 - Coding phase 1
 - Jul 25 - Sep 4 - Coding phase 2
 - Sep 5 - Sep 12 - Final week - submit final work and evaluation
 - Sep 19 - mentor final evaluation
 - Meeting dates and times
 - Which days and times work best for all involved?
 - India Standard Times
 - 8:30 - 10:30 AM for Hrushi
 - 10:00 PM or later ok for Hrushi
 - 8:30 - 9:30 AM ok for Rishabh
 - 10:00 PM or later ok for Rishabh
 - 8:30 AM Tuesday, Wednesday, or Friday for Mark
 - Wednesday 8:30 IST for our weekly
 - Known vacation or other time off
 - Mark
 - May 30 - Jun 3 - out of office, but may be able to attend a meeting
 - Rishabh host the meeting next week
 - Jun 7 - Jun 10 - cdCon, Mark attend from the conference
 - Jul 4 - Jul 8 - grandchildren are visiting, unavailable
 - Rishabh
 - Hrushikesh
 - Aug 1 - Aug 12 - End semester Exams (completely unavailable)
- Status report on open pull requests - Mark & Hrushi
 - JUnit 3 -> JUnit 4 migration in git client plugin [PR-824](#)
 - Easy merge, more challenging to confirm no test methods lost
 - Organization name in git plugin [PR-1228](#)
 - Needs interactive testing, deeper checks

- Browser guesser improvements in git plugin [PR-1251](#)
 - Includes a breaking change, pending
- Other topics
 - Configuring development environment is already done (PRs to git client and git)
 - Architecture and class diagram (if it is helpful)
 - Mark is more concerned at the parts that he doesn't understand
 - How to do a lightweight process on the Jenkins controller that isn't a Jenkins job
 - Global build discarder
 - Should we ask the author of Global build discarder to talk to us about the problems, techniques used, etc.