Texas Instruments: Oracle Database VCS Sync Git Implementation

Project Proposal

Akaanksh Raj Kambalimath
Kaushal Bommena
Rameen Housseini
Thanh Nguyen
Zen Park

Contents

1 Abstract

Texas Instruments' centralized version control system is outdated. Git has replaced centralized version control. Git is decentralized, distributed among developers, customizable, and a large community. The benefits are clear because TI's database needs a lot of revisions. We plan to implement Git through JGit (a library in Java) to take revisions provided by the advanced queue in the database. We will do the following: first, we use a "time-controlled while-loop" to take the pre-written list of revisions; second, write the list to local storage; and third, JGit will commit then push changes in the local filesystem. This filesystem will be located in a remotely-connected repository. Furthermore, we will create new repositories as necessary. This way, different teams of developers can collaborate and revert to branch-specific changes without worrying about any mistakes.

2 Executive Summary

To reiterate and develop on the abstract, this project will essentially perform the same functions that TI's existing centralized version control implementation is doing, except our implementation will have the added advantage of using a more developed version control system with more modern support and better features. Right now, there is an extremely expensive closed source solution that is available to enterprise consumers, the name of this service is called Gitora.

Our project will have the source code for the solution available to the sponsor so that they may further develop it and it is much cheaper to commission us through UTD than to have to use this service. However, there is a bug that is preventing different version SQL databases from communicating with one another via the automated internal PL/SQL (Procedural Language extensions to SQL) packages in the database that is responsible for creating the entries in the Git table, which populates our list of revisions that will be written out to the server's storage and pushed to the online repositories.

We will investigate this issue once we have finished implementing the synchronization of revisions and the ability to create new repositories. Once we are done fixing this bug, we can work on getting the specific desired implementation for pushing/pulling changes to the production branch of their databases. The pre-existing Gitora service likely does not offer the automatic package generation nor would they provide the BitBucket high-level production implementation desired by TI.

3 Introduction

For this project, we introduce a distributed version control system via JGit, and Git while replacing their pre-existing central version system solution. This way, developers will have the flexibility of having the entire project or a portion of the project's repository hosted on their own local machines so that they can work online and create their own branches to better organize the introduction of new features and bug fixes. We will refine the PL/SQL package on their database that automatically creates new entries from revisions so they can be committed to the central repository.

There is another solution that covers most of what TI desires called Gitora that is expensive and closed source, which brings the massive downside of not making the solution adaptable to their needs, nor will they

be able to add their own features to the project in the future, not to mention it is cheaper to commission UTD to have their students work on a solution.

Our code is documented well so future developers will be able to easily interact with the code. We will have a timer loop in a Java file running on the central server every 10 seconds, which will retrieve the list of revisions from a particular auto-populated table on the database and create Git objects, write them out to the local storage and commit/push the contents of the local storage git objects to the central server, creating new repositories if they don't exist on the online BitBucket server. Then, we will investigate the pre-existing PL/SQL package on the database to see if we can get it to work with older and different versions of the Oracle SQL database since some of the other databases run on an older version. We will then check to see if the desired implementation works for the higher production level, which would be so that upper-level developers can ensure changes on the development and staging branches will not conflict with the production-level branch. We are confident in our ability to perform the first fundamental part of the project, the second and subsequent parts of the project we are not very sure about since it is too early to tell and we had a late start to the project but we will try our best.

4 Resources

4.1 Project Team

- Texas Instrument Representative (Company Mentor): Melissa Bolcar
- Faculty Advisor: Srimathi Srinivasan
- Core Developer Team: Rameen Housseini, Zen Park, Thanh Ngyuen, Akaanksh Raj Kambalimath, Kaushal Bommena

4.2 Software Used

Virtual Private Networks will be used to establish a connection to Texas Instrument servers. GlobalProtect is a client provided by TI, which is what Rameen will use. OpenConnect is a compatible open-source solution, which is what all the other members of the group will use.

The Apache Netbeans IDE used to write scripts in order to leverage the database to create, clone, and manipulate PL/SQL entities to push into the new Version Control Software. Oracle Database 19c is used as a database to maintain/store PL/SQL entities, powered by Oracle's advanced queue - automatically tracking changes/revisions that have been made to certain Texas Instruments-related projects.

Bitbucket will be used, as it is a Git-based source code repository for hosting our project. Project Management is facilitated through the Jira software work management tool. Used to track project progress and issue stories for developers in an agile work environment.

4.3 Machines

All developers will use their own machines with the project being hosted locally and the final product committed to Texas Instrument servers. Since TI's VPN prevents internet connection to regular websites, Kaushal

Bommena, Zen Park, Thanh Nguyen and Akaanksh Raj Kambalimath are using Linux virtual machines connected to the TI VPN to work on the project.

4.4 Communication

Our team internally is using Discord to facilitate communication and internal team meetings. We are using GroupMe and Email to communicate with the company representative. Lastly, we are using Cisco WebEx to hold meetings with our team, the faculty mentor and company representative.

5 Key Roles

- Point of Contact: Zen Park
- Meeting Organizer: Zen Park
- Meeting Note-Taker: Rotating per-meeting
- Java Developers: Rameen Housseini, Zen Park, Thanh Ngyuen, Akaanksh Raj Kambalimath, Kaushal Bommena
- PL/SQL Developers: Rameen Housseini, Zen Park, Thanh Ngyuen, Akaanksh Raj Kambalimath, Kaushal Bommena

6 Communication Plan

We communicate through Discord because of its ease of use. On Fridays from 3 to 4 pm, we meet with our faculty advisor and sponsor weekly via Cisco Webex. We have GroupMe in case we need to communicate with our company sponsor quickly and have a group email from TI so we can more effectively communicate to our University mentor and company sponsor in case email is required. Webex is the only VoIP client allowed while under the company VPN, but those of us that have the project set up on a VM will be able to use Discord on their host machine while working on the project. We all have each others' phone numbers so if their attention is urgently needed they can be reached that way. Meeting minutes are shared via Discord and Discord will be used for assignments/project development when possible. Email and GroupMe may be used if the company sponsor needs to share files with the rest of the group.

7 Risk Analysis/Contingency Plan

7.1 Security

Ensuring a secure connection approach is critical for our clients at Texas Instruments. Leakage of data and resources and unauthorized access to the project are possibilities we wish to avoid. In order to do so, we enable multiple factors of authentication including the use of authentication applications such as entrust as well as connecting securely to private servers provided by the client. We estimate that compatibility issues between our local machines and the security software being used may occur. As a result, we plan to

keep an open line of communication with our company mentor as well as the TI local help desk to resolve issues as and when they arise.

7.2 Illness/Developer Unavailability

The risk is if someone might get ill or have a personal issue that they cannot contribute to the project anymore. To counter this potential risk, we as a team can review what has been covered in the weekly meeting with the client to give them updates for their absences. We would also encourage our team members to document their reasons to let the team know in advance so that we can allocate time and human resources to maintain the flow of the project. If the person drops the class for permanent absences due to personal excuses. We would contact them to discuss keeping a record of what they have contributed so far for their credits though they are no longer in the team. Then we would inform the client about this issue as soon as possible, and assure them that the rest of the team can still carry on the project.

7.3 Out-of-Date Technology

New versions of technology are constantly released to help with efficient use. Editing an entire instruction with small changes is difficult because it takes time to research. When facing that problem, our first approach is to confirm with the client to inform them so that they can allocate time to update the instructions as soon as possible.

Problem-solving is part of our skills so we have tested with closely compatible versions to deliver the same outcomes as the out-of-date tools. Besides, through communication, we can obtain approval from the client to use software with self familiarity and experience to minimize the risk of incompatibility.

7.4 Steep Learning Curve/Lack of Technical Knowledge

The scope of the VCS project would prove to be a great platform for growth in terms of technical development. We recognize that through this opportunity, we would run into situations that are outside our current understanding and capabilities. Educating ourselves on new scripting languages, APIs, various other libraries, and so on are learning curves we expect to face and overcome in short periods of time in a hands-on approach. Utilizing our off-campus time to capture as much fundamental understanding to dedicate our knowledge and skills to the product of the company is our utmost priority.

8 Costs

TI has provided us with all tools necessary: an IDE, database tools for software development, Jira for issue and progress tracking, and Confluence as a knowledge base. Furthermore, our team is working with several new technologies, like Spring Framework and Oracle Database Packages, in which we may need training. We estimate that tutorials and documentation freely available online will suffice, however, we may need paid training for some advanced topics. We will also need computer part upgrades to handle the large influx of big data to avoid crashing. We have encountered the blue screen of death several times.

9 Timetable

9.1 Phase 1: 3/25/2022

Create Git commits from revision items from the table with correct file changes, dates, and authors, set up JUnit with the current project, and write JUnit tests for existing and newly implemented code.

9.2 Phase 2: 4/8/2022

Create a new remote BitBucket repository when a new project is found in the table, write JUnit tests for newly implemented code.

9.3 Phase 3: 4/18/2022

Fix a bug in the Oracle Database Package that prevents different versions from interacting with each other in the VCS Sync database.

9.4 Phase 4: 4/29/2022

Update PL/SQL code to provide any additional data required for the application, incorporate new data into our application, and write JUnit tests for newly implemented code.

9.5 Stretch Goals

Migrate existing CVS implementation to work with Git, write JUnit tests for newly implemented code.

10 Evaluation

We plan to keep track of our progress utilizing Agile workflows in terms of software development. By maintaining stand-up meetings on a regular basis and creating stories of variable estimates and importance, we are able to assign work in sprints and reach deliverables in a more structured and organized manner.

Assigning stories the values of the Fibonacci sequence such as 1,2,3,5,8,13, ... is one way we plan on going about estimating stories based on effort, cost, and time. Throughout the meeting, we communicate what we have done, the obstacles we have faced, potential solutions to the roadblocks, and the overall completion of the story.

Building off from the answers above, when we use stories, we will use “cards” (virtually) that have acceptance criteria for each item and we plan on moving in increments from start (fundamental and prior aspects of the Git control system desired) to finish.

The acceptance criteria will be straightforward from what we already know. For example, the different possible ways we implement the JGit API will not have a significant impact on the speed of the project, and no speed requirement has been requested from TI as of yet.

The acceptance criteria will be based mainly on the functionality of the story items such as setting up a Java function where a revision object in a list can be converted to a JGit object that can be pushed and committed, this acceptance criteria is straight-forward and does not have room for interpretation, we anticipate all stories will follow this format. We will also compare our Git implementation with their previous CVS implementation to ensure no loss of functionality.

More specifically, we will use Jira's scrum board interface to have our storyboard and keep track of our progress. The latter will be kept track of via an Excel spreadsheet. The Jira interface has four columns which are "to do," "in progress," "in review," and "done" where within those columns, there are individual cards describing tasks that need to be done and individual people that claim those cards. All of this is contained within an "epic".

11 Contact Information

11.1 Rameen Housseini - rxh170130@utdallas.edu

Senior student at the University of Texas at Dallas, expected graduation by the end of Spring 2022 with BS in Computer Science, working as a part-time contracted programmer at 8folds mfg LLC.

11.2 Zen Park - zen.park@utdallas.edu

A linguist with a big interest in the computing world. Hobbies include winning competitive games, listening to classical music, or most likely studying your mother tongue. Can currently speak five languages with sheer failure yet proud of making an effort. Graduates in Summer 2022.

11.3 Thanh Ngyuen - thanh.nguyen2@utdallas.edu

Pursuing Masters of Computer Science, Intelligent Systems Track. Senior at the University of Texas at Dallas majoring in Computer Science. Expected to graduate in Spring 2023. Working part-time as a social media analyst at Genpact LLC.

11.4 Akaanksh Raj Kambalimath - ark@utdallas.edu

Pursuing Masters of Computer Science, Cybersecurity Track graduating Spring 2023, and Senior at the University of Texas at Dallas majoring in Computer Science graduating Spring 2022. Working part-time as a freelance software developer for several production studios and Esports organizations.

11.5 Kaushal Bommena - kvb200000@utdallas.edu

Senior at the University of Texas at Dallas majoring in Computer Science. Expected Graduation by Spring 2022. Working part-time as a software technical consultant at Credera Enterprises Company LLC.

12 Sources

- A Guide to JGit
- Centralized vs Distributed Version Control: Which One Should We Choose?
- Gitora – Source Control for Oracle

13 Signatures

Rameen Housseini (Team Member)	Zen Park (Team Member)
Thanh Ngyuen (Team Member)	Akaanksh Raj Kambalimath (Team Member)
Kaushal Bommena (Team Member)	
Melissa Bolcar (Company Mentor)	Srimathi Srinivasan (Faculty Advisor)