COMSE6156 Topics in Software Engineering Progress Report

George DiNicola (gd2581)

Overview

My final project is an extension of the research I did for my midterm paper. I've been implementing the design presented in TDRB [1], which added internal tamper-proof functionality to a centralized relational database by storing hashed records on a blockchain. Since TDRB middleware does not have any source code published, I chose to implement it based on the design guidelines provided in the TDRB middleware paper [1]. The design specifications from TDRB [1] are not very detailed, so it is unlikely that the application I've been working on is exactly like the one created for the study. The goal of the project is to evaluate the middleware in new ways (addressed in the research questions section).

So far, I have written the middleware application (in Python 3.6) that converts relational tables into the data structures suggested by the authors to store on the blockchain, as well as retrieves those records from a configured MySQL database. I've also created the MySQL database as well as some scripts for inserting test records. The "tamper detection module" was implemented using the high-level algorithms presented in the study (except for the blockchain query and Redis cache query portions).

Currently, I am working on using Hyperledger Fabric to store the data structures that will be used to detect tampering. This part has taken the longest and required a lot of research to use. I currently have the Hyperledger Fabric working for an example blockchain and understand how to set all the configurations and write code that calls the Hyperledger Fabric the way I need to use it. I am very much hoping to get back on track when finished with this portion.

The remaining tasks are to finish the module that will perform blockchain operations on the custom data structures, set up and configure the Redis cache database, integrate the middleware with the blockchain, integrate the middleware with Redis, and evaluations of the design (which will involve scripting automated tests and seeding more test data).

Value to the User Community

The user community for which my project is targeted is database administrators, database developers, security professionals, software engineers, and system administrators. The value my project will provide to my targeted user community is the following:

- Understand the impact of query response time efficiency when running a blockchain-based middleware application for different peer node configurations (e.g., number of peer nodes)
- Help engineers identify the maximum number of blockchain nodes that can be used so they can use as many nodes as possible without violating SLAs (i.e., if the average query time for an application needs to be below a certain threshold to handle the app's needs).
- Understand if a blockchain-based middleware solution is appropriate for practical/commercial usage based on the response times of different node configurations

Research Questions

Below are the research questions I am addressing:

- How much is query response time affected by the number of blockchain node configurations (number of Peer nodes, number of CA nodes)
  - The TDRB study addressed this, however they only compared using 4 peer nodes, 2 CA nodes, and 1 Orderer node vs. 10 peer nodes, 5 CA nodes, and 1 Orderer node.
- How well does the TDRB middleware application scale when more Peer nodes (nodes that will hold copies of the ledger) are added to the network?
- What is the storage overhead of using the middleware application (specifically the storage added by the Hyperledger Fabric blockchain) vs only using MySQL?

Demo

I intend to give the elevator pitch explaining what technologies were used to create the middleware and its tamper-proof benefits. For the demo, I intend to (ahead of time) screen record my usage of the application and execute query executions for the following scenarios:

1. Commit a record to the blockchain using the middleware and check the results that should also indicate no tampering occurred
2. Query an untampered record using the middleware
3. Tamper with a record inside of the MySQL database and query the same table using the middleware to show the results containing the tamper information
4. Show graphs for query response time vs. number of peer nodes in the blockchain network

Code Delivery

I intend to deliver the code using Github.

My Github repository: https://github.com/GeorgeDiNicola/TDRB-Middleware-Extension

My Kanban board for my project (in GitHub): https://github.com/GeorgeDiNicola/TDRB-Middleware-Extension/projects/1

References

[1] Lian J, Wang S, Xie Y. Tdrb: An efficient tamper-proof detection middleware for relational database based on blockchain technology. IEEE Access. 2021 Apr 28;9:66707-22.