COMSE6156 Topics in Software Engineering Project Proposal

George DiNicola (gd2581)

Topic Introduction and Project Details

My final project is an extension of the research I did for my midterm paper. I intend to implement the design proposed in the TDRB middleware paper [1]. I intend to use the same technology as the authors for setup and deployment: Redis for the cache database, Hyperledger Fabric as the Blockchain, Ubuntu 18.01 LTS as the operating systems, Docker as the containerization technology that will serve as a network between the component, and Amazon Web Services as the cloud provider. Although setup will be similar for the above, I am choosing to write the middleware application in Python, rather than Node.js (what the authors used). Finally, I will use MySQL (database the authors used) as the underlying relational database, however I intend to also try PostGreSQL since it is another open-source SQL-based RDBMS and offers a lot more features than its very lightweight MySQL counterpart.

To evaluate the implementation described above, I intend to write tests the authors did not include in their study. These metrics, based on evaluations of similar technology I found in other studies, include throughput vs transaction per proposal, throughput vs number of nodes in the network, latency vs. number of nodes in the network, mean latency vs. volume of user requests, amount of data vs. cost (in USD) of using Hyperledger Fabric, transaction Arrival Rate (tps) vs throughput, and transaction Arrival Rate (tps) vs latency. Although it is unlikely that all of these tests are necessary, it serves as a good direction for evaluating the implementation.

Finally, I intend to redefine the threat model used in the study. The authors of TDRB [1] analyzed the tamper detection accuracy only *after* the database (and in the other portion of the study, the cache) was tampered with. I think it would be more meaningful to analyze the tamper detection accuracy in a test environment where the tampering is happening concurrently with the detection performed in the query results, or at least between the executions. Specifically, the authors issued the 400 statements for testing detection accuracy of each operation after the data base was tampered with, but I intend to do it concurrently, as well as in-between (in the event that their design does not handle concurrent tampering).

~~Specifically, I want to implement the on-chain and off-chain data partitioning functionality presented in [2] for the ChainSQL open-source middleware client application developed by the authors of [1]. ChainSQL was designed to commit *every* transaction to the Ripple blockchain. This design choice was intentional to enable fast disaster recovery – when every transaction is committed, the entire database system can be recreated from the unmodifiable ledger. While disaster recovery functionality is useful, traditional relational database management systems have accomplished this for years using other methods like database snapshots, database distribution, etc. I propose adding the ability to specify to the ChainSQL client which data should be stored on the blockchain and off the blockchain, allowing database designers more granularity in their choice for determining which data would be more appropriate for the blockchain based on its security needs (i.e., if the sensitivity of the data requires elevated internal tamper-proof resistance).~~

~~Unfortunately, the authors of [2] did not publish their design implementation to any open-source repositories, which presents possible difficulty integrating their design into ChainSQL. I~~

~~believe I can use their design specifications to add a module to ChainSQL that could handle on-chain and off-chain mapping before committing to the ripple blockchain, as well as retrieving on-chain objects from the blockchain. Potential difficulty may arise when determining the granularity of the new functionality, for example entire tables as on-chain or off-chain data vs. portions of tables partitioned into on-chain or off-chain data. While accomplishing both types of functionalities would be ideal, I intend to start with entire tables as on-chain and off-chain, then try to add the functionality for partitioning tables themselves. I plan to empirically evaluate my extension of ChainSQL by measuring query response times for database transactions like I did for my midterm assignment and comparing them to the response times for the same data before the extension. Specifically, I will be measuring the query response times for CREATE, UPDATE, INSERT, DELETE, and DROP database transactions.~~

~~I initially wanted to integrate the concept of a "centralized" ledger that would allow data removal from the blockchain into ChainSQL like in [3], but unfortunately Alibaba's website (they host the public beta for LedgerDB) for gaining access to the technology has been down every time I've checked. It could have been interesting to add this functionality to ChainSQL so that it could comply to data regulatory requirements.~~

Relevance to the Course

My project choice was influenced by topics I learned from reading research papers for the midterm paper. I was previously unaware this topic existed, which I believe directly translates to what we are trying to do in this class: regularly read research to keep up with current trends and see connections between proposed designs that can empower you to design a better system. This topic I found combines my favorite subareas of computer science, which are databases, software engineering, distributed systems, and security, while additionally allowing me to extend the use of these ideas to create something new and interesting. These are all fields I've gained a lot more exposure to at Columbia, which makes it an ideal mix of topics to pursue past the fundamental level.

Motivation

I have been interested in and worked with database management systems for several years. As a Database Engineer and assistant DBA, I had administrative privileges (along with several other members) and always wondered about database security for administrative users. In addition, blockchain is an interesting topic and has even gained popularity among non-technical individuals in the last few years. I think the underlying concepts used in blockchain are very powerful, and I had a financial background when I was an early undergraduate student, so I find the use of a technology used to power cryptocurrency adapted for database security exciting and novel. Finally, internal tampering is extremely difficult to detect in an organization, especially one that handles large volumes of data. I believe further investigation of this topic could add benefit to society by potentially adding more trust to the validity of consumer data held in database systems. Finally, as I mentioned in my midterm paper, the 2021 *Cyber Security Insiders* Insider Threat Report found that 98% of the organizations surveyed believe they are vulnerable to insider attacks, a large increase from only 68% of those surveyed in 2020. In addition, 82% of

the organizations surveyed indicated that they find it is difficult to determine the damage of an insider attack, whereas the previous year was only 58% [4].

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. https://ieeexplore.ieee.org/document/9417201

~~**[1]** Muzammal M, Qu Q, Nasrulin B. Renovating blockchain with distributed databases: An open source system. Future generation computer systems. 2019 Jan 1;90:105-17. https://doi.org/10.1016/j.future.2018.07.042.~~

[2] Marinho SC, Costa Filho JS, Moreira LO, Machado JC. Using a hybrid approach to data management in relational database and blockchain: A case study on the E-health domain. In2020 IEEE International Conference on Software Architecture Companion (ICSA-C) 2020 Mar 16 (pp. 114-121). IEEE. https://ieeexplore.ieee.org/abstract/document/9095697

[3] Yang X, Zhang Y, Wang S, Yu B, Li F, Li Y, Yan W. LedgerDB: A centralized ledger database for universal audit and verification. Proceedings of the VLDB Endowment. 2020 Aug 1;13(12):3138-51. https://dl.acm.org/doi/10.14778/3415478.3415540

[4] 2021 Insider Threat report [Internet]. Cybersecurity Insiders. 2021. Available from: https://www.cybersecurity-insiders.com/portfolio/2020-insider-threat-report-gurucul/