Operating Systems and Distributed Systems

Fall 2022

Project 3 & 4

The third and fourth projects are open design projects on blockchain. You can have a max of 3 students per team, and you will design and implement a blockchain from scratch. The larger project has two parts with two separate deadlines, and we refer them as Project 3 and 4, respectively, but you should consider it as a single project.

## Part I (Project 3)

In the first part of the project, you will need to implement the bitcoin protocol as we introduced in the lecture. You only need to implement the core parts of it, including

* A client program that can submit transactions to the miners;
* Miner local function:
  + The offline data structure of a single block (you do not have to implement the Merkel tree);
  + The offline chain with hash pointers to the previous block;
  + Offline verify the validity of the chain;
* Distributed transactions
  + A transaction dissemination mechanism to broadcast transactions to miners;
  + A PoW function with adjustable difficulty;
  + A mining algorithm to allow miners to find the solution to the PoW puzzle;
  + Broadcasting blocks to other miners;
  + Longest chain rule to choose the right fork if there is one;
* You need to be able to run at least 5 miners and let them execute the protocol above and generate a chain with at least 100 blocks without error.

You can omit things like persistent storage (i.e. writing chains to disk), and execution of the smart contract, etc. As a simplification, specifically for broadcasting, you do NOT need to implement node joining and leaving, just if all nodes hold a static list of all other nodes (other nodes can fail, but if they fail, just ignore them).

You can use any open-source libraries in your project. You can continue to use the servers you set up for Project 2.

You will need to demo this implementation to the TAs and report the baseline performance (you can define the performance metric, but you will need to document it clearly).

**Things to submit**

1. A link to your repository and giving TAs access permissions.
2. A live demo to the TAs, TAs will arrange a time with you for the demo.
3. As before, a Makefile that can build your blockchain agents, and a deploy target that can copy your compiled binaries to the nodes in your node list.
4. A written report (not too long, 3-4 pages should be enough), describing your design, assumptions, and key evaluation results. Also document the omissions / changes / additional assumptions you made, that are different from the original blockchain system.
5. A written proposal (1 page) for your improvement plan (for part II / Project 4), no longer than a single page. You should describe what improvement you propose, what metric we can use to evaluate your improvements, and briefly describe how you plan to do it (no need for details). See the next part for suggestions on improvements, but you can choose your own improvements instead of following these suggestions.

**Deadline**

Sunday, Dec 18th @11:59. I.e., you have 4 weeks to do this part. However, you should really start early as you might have all kinds of problems on the libraries, machines, etc.

## Part II (Project 4)

For the second part of this project, you need to propose several improvements to your baseline. The minimal number of the improvements proposed equals the number of team members.

The improvement can be anywhere from the consensus protocol, networking, local data structure, programming interface or incentive mechanism design, or even designing a better user interaction with the blockchain. Or you can choose to add back features such as Merkel tree or node joining/ leaving protocols you omit in Part I.

For feature improvements, you should be able to demo the feature to the TAs. For performance improvements, you should have figures / reports showing that there are actual improvements.

You will need to present a demo of your implementation and performance numbers at the end of the semester (after the final exam).

The project will be graded for the novelty of the improvement design ideas, the complexity of the implementation, design document and report writing quality, as well as code quality and actual quality of the *improvements* in your evaluation (relative improvement over the baseline in Part I).

**Things to submit**

1. A link to your repository and giving TAs access permissions;
2. A demo to the TAs, TAs will arrange a time with you for the demo;
3. TAs will NOT try to build and run this project, so it is up to you if you want to make a good Makefile or not (I suggest that you do it for your own convenience).
4. Revise / extend your written report and proposal from Part I (not too long, 8 pages including the part from Part I should be enough). In this part, you should add materials to describe how you achieve your improvement, and any additional assumptions you add. Also, you need to show your key improvement results. You should also analyze how you achieved the improvements.
5. Note that the minimal number of the improvements proposed equals the number of team members.

**Deadline**

The final demo will be in Week 17 (second exam week), depending on your exam schedule. Reports are due on the last day of the semester (Friday, 1/13 at midnight).