MINECRAFT SAVE TEST FRAMEWORK

CHRISTOPHER ALEXANDER, KUANG-HSUAN LIN AND VEEN OUNG

Microsoft

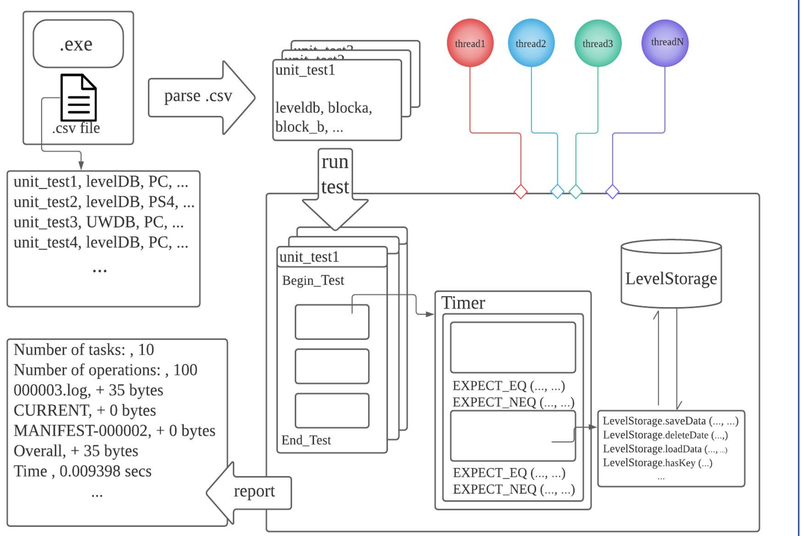
MOTIVATION

- Minecraft Bedrock edition is currently utilizing “LevelDB” to save worlds data. This works well, however Minecraft is pushing the technology to the limits.
- Thus, we develop a testing framework in C++ that can be used to test and evaluate key/value store implementations.
- Allow future Minecraft developers to easily experiment with “LevelDB” replacements.

REQUIREMENT

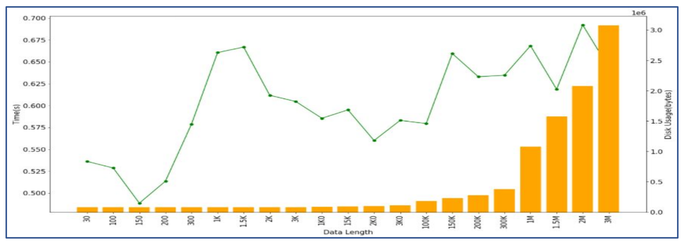
- Visual Studio 2019
- C++20
- CMake 3.19.0-rc3
- Python 3.6.2

ARCHITECTURE

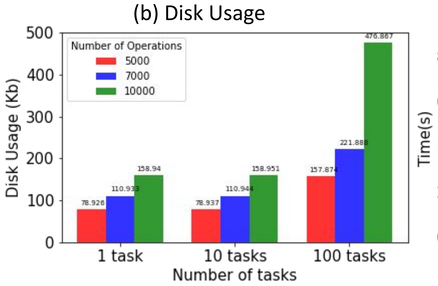


USER OUTPUT DELIVERABLES

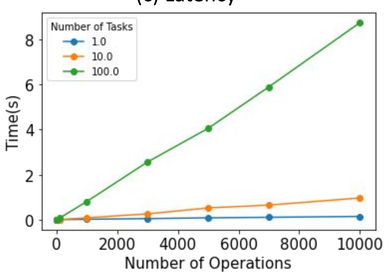
(a) Data Length



(b) Disk Usage

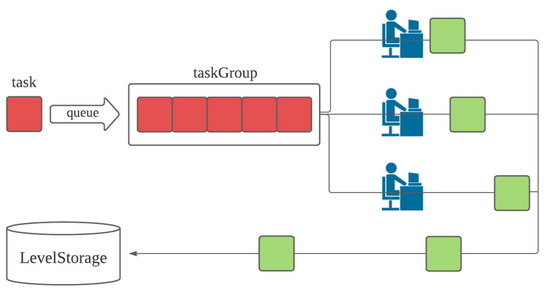


(c) Latency



TESTS

- Perform testing on Minecraft storage system’s interface granularly and as a whole.
- In each test, tasks (database operations) will be queued into taskGroup and configured to run in single/multi-threaded environments.
- Focus on getting read/write latency and disk usage



PROPERTIES

- Flexibility
  - StorageType: LevelStorage (LevelDB) / MockUpStorage(std::map)
  - SizeOfData: 1 byte / 100000 bytes
- Foolproof
  - Easy to build, clients need no programming experiences
- Quantization
  - Output to a csv file for evaluation
  - Visualize the read/write latency with Python script
- Interactivity
  - Command line interface

```
Usage: <target path> -i <source path> -o <result path>
Options:
        -i,--input      Specify the input file name (e.g. testcase.csv)
        -o,--output     Specify the output file name (e.g. metric.csv)
        -h,--help       Show this help message

Example: ./SAVE_TEST_FRAMEWORK -i case1.csv -o metric.csv
```

CONCLUSION

- Testing and evaluating “LevelDB” and other key/value storage implementations’ robustness and performance has been made possible through our framework.
- The framework has helped identify bugs in Minecraft’s codebase along the development process.

FUTURE WORK

- A C# implementation of a simple front end solution, allowing a user to pick an implementation to test, and visually show the results.

REFERENCE

- A. Ravishankar, “How to Write a Minimal Unit Testing Framework in C++”, Medium, [Accessed: 15-Mar-2021].

ELECTRICAL & COMPUTER ENGINEERING

UNIVERSITY of WASHINGTON

ADVISOR: RANIA HUSSEIN, PAYMAN ARABSHAHI, SHRUTI MISRA

INDUSTRY MENTOR: ROBERT HUNT

SPONSOR: MICROSOFT