CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: Generative Testing of SSD using TLA+ Model-simulation.	
Sponsor Information: 	**Chris Ortiz**, Senior Technologist SSD Validation Western Digital Corp., Flash Business Unit Chris.ortiz@wdc.com **John Lee**, Senior Director SSD Validation Western Digital Corp., Flash Business Unit John.lee@wdc.com

Project Overview:

Solid-State Drives are fast storage devices that became part of our daily lives from our personal laptops, desktops, gaming console, enterprise and up to the cloud data center where important photo memories, private personal data, work documents, game play mode, etc. are saved and retrieved back. To make sure it works right SSD were being validated extensively like in Western Digital's SSD Validation to make sure it meets or exceeds the product requirements.

Test plans were created for those validations, but the combination of possible test sequences can be limited with the understanding of the test engineers on probable combination of test sequences. We understand that bugs follow Murphy's Law that if it is possible to happen it will happen it is just a matter of when. It is a matter of Probability versus Possibility. We aim to improve our coverage of our testing to cover as much as possible without the unconscious bias from test engineers. We learned that Formal Methods is one of the way industries use to cover such possibilities, but Formal Verification of such Formal Specification can only model-check conceptual Design and not actual product. We like to combine our testing with Formal Methods in a way that by using model-simulation it will drive our testing to try different path of test sequences allowable by the Formal Specification written in TLA+. This is Generative Testing which we got inspiration from "The Evolution of Testing Methodology at AWS: From Status Quo to Formal Methods with TLA+."

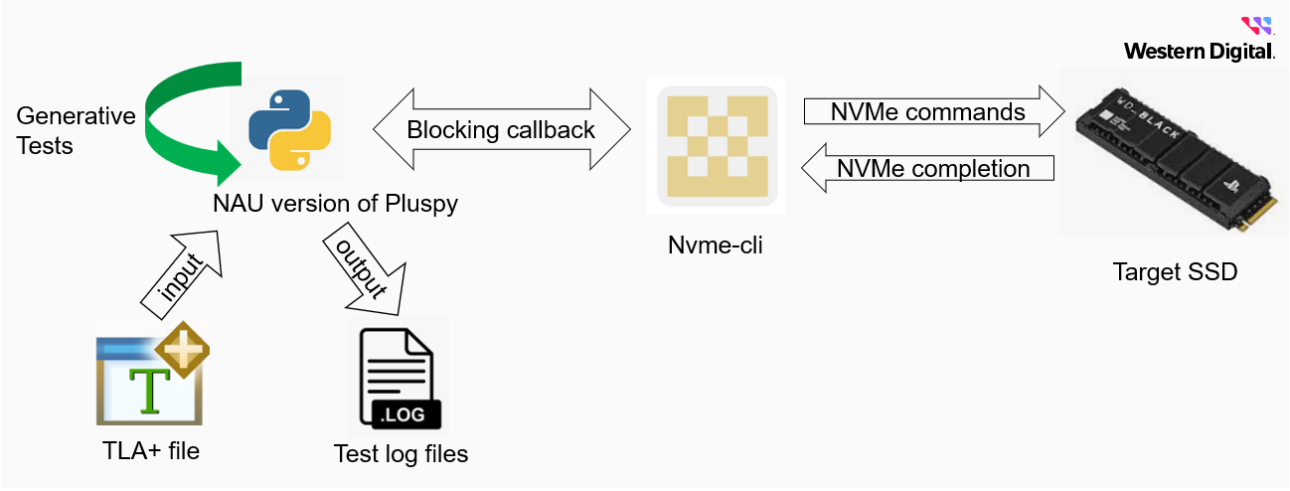
The AWS video can be found here: https://www.infoq.com/presentations/aws-testing-tla/.

TLA+ is high-level formal specification language developed by Leslie Lamport. Lamport is a Turing Award winner, inventor of Time Clock, Bakery mutual exclusion, and Paxos algorithms which are widely taught and used as references in Computer Science academia and industries. He is also the inventor of LaTEX. TLA+ is widely used in Microsoft for 20 years, AWS for 15 years, and Oracle Cloud Infrastructure for 9 years. These 3 companies joined together to create TLA+ Foundation: https://foundation.tlapl.us/. There are also other companies like Crowdstrike, LinkedIn, MongoDB, Google, and academia participating in TLA+ Conferences: https://conf.tlapl.us/home/.

During validation we found out that large quantities of SSD devices we test can sit idle once all the test plan developed by test engineers has been exhausted. Running the same test plan again and again does not give us any value in terms of test coverage. This affects our resource allocation and impact us with large cost. We like to improve our SSD utilization by testing them with uncovered test sequences that generated automatically by TLA+. This way the SSD devices are not sitting idle but rather being exercised or tested with new possible test sequences

that is not developed or anticipated by test engineer. In effect, this will improve our test coverage at the same time our SSD and test system utilization as well.

The solution we are thinking is that given a TLA+ specification, the NAU modified Pluspy which is a TLA+ python interpreter will generate a random seed and number of steps to take in generating a possible test sequence. This Pluspy will have blocking callbacks to nvme-cli which will issue the NVMe commands to and receive back result of that command from the SSD device. Each test run will generate different possible test sequence. In case a particular test sequence detected an NVMe failure, it can retry the same random seed and the same number of steps to reproduce the failure because the modified Pluspy will sample the same path taken given the same set of inputs. These automatic retries will increase our confidence that the issue found is reproduceable and valid which we can file JIRA bug to our product teams. It is important for us to have log of the test sequences and test results which will be generated after each test run. These logs will serve as records of evidence when filing JIRA bug and also for us to review on how to improve our TLA+ specification if we need to steer it in other ways.



Knowledge, skills, and expertise required for this project:

- A basic understanding of TLA+ which is a high-level formal specification language.
  - Video course here: https://www.youtube.com/@tlavideocourse8540/playlists
  - TLA+ Home Page here: https://lamport.azurewebsites.net/tla/tla.html
- Knowledge in issuing NVMe command to SSD:
  - Github here: https://github.com/linux-nvme/nvme-cli;
  - Ubuntu manpages here: https://manpages.ubuntu.com/manpages/mantic/man1/nvme.1.html
- Knowledge in Python.

Equipment Requirements:

- There should be no equipment or software required other than a development platform and software/tools freely available online.
- Desktop PC with secondary target SSD that is NVMe and PCIe Compliant.
- Latest Ubuntu with Python 3
- Pluspy python TLA+ interpreter module: https://github.com/tlaplus/PlusPy

Software and other Deliverables:

- Detailed, clear, and professionally composed Design document and literature which will be used for future enhancement of the tool in Western Digital's SSD Validation.
- Complete and well commented codebase will be shared to Western Digital in ZIP file format.