



SSDynamics

Generative Testing of NVMe using TLA+ Model-Simulation

Team Members: Carter Kaess, Charles(Chas) Diaz, Connor Aiton, Charles Descamps

Mentors: Brian Donnelly, Savannah Chappus

Client: Chris Ortiz, John Lee – Sandisk/Western Digital



Scan for website

Introduction

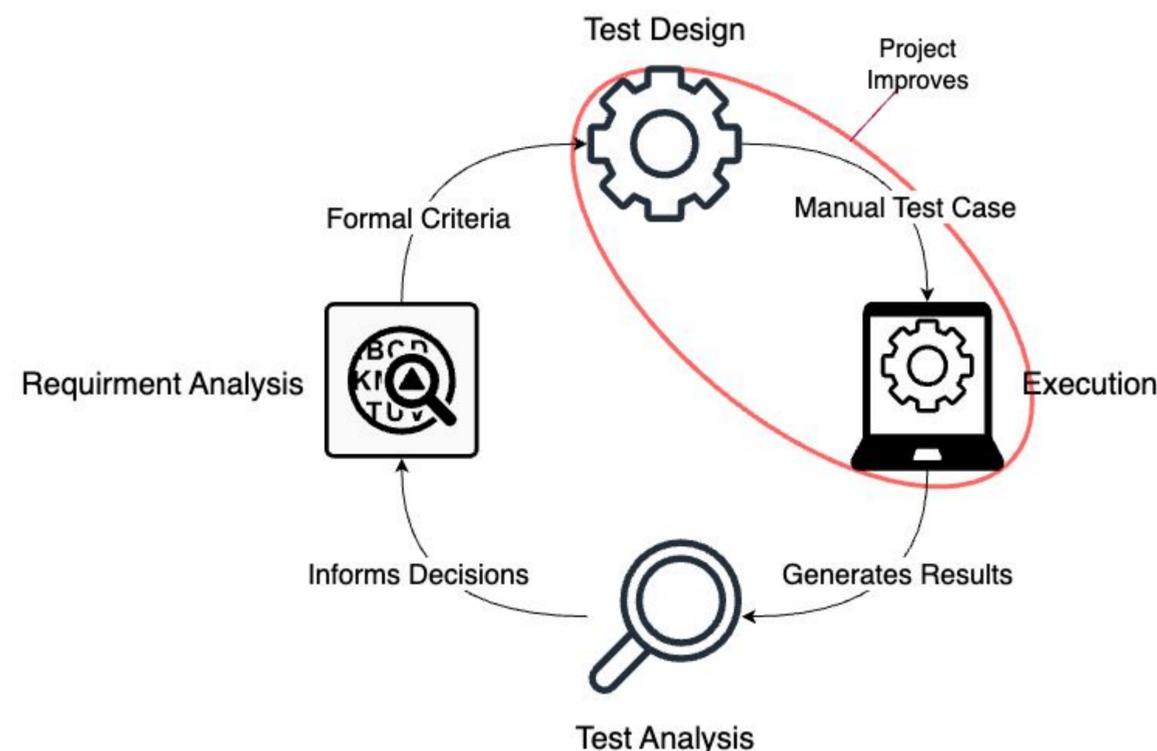
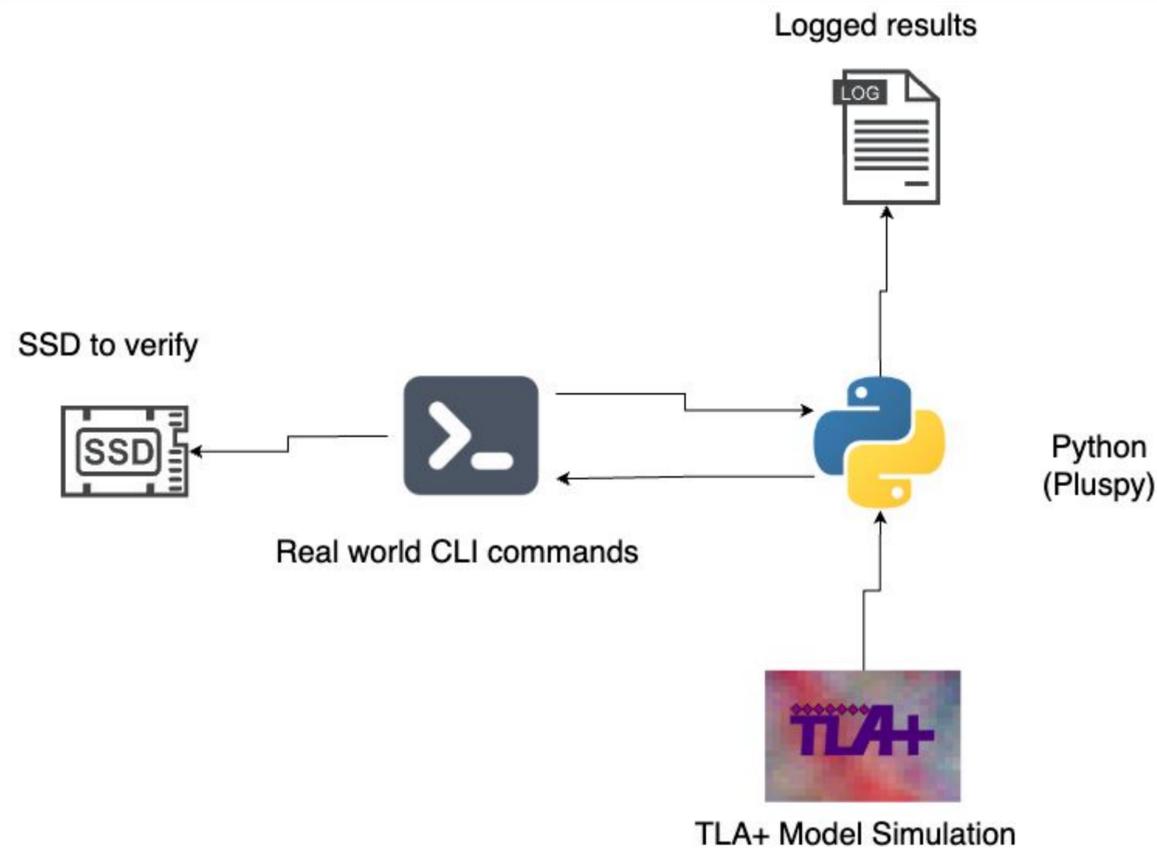
The need for **fast solid-state** storage is growing as computers become faster. Applications demand quick storage solutions to keep up with performance requirements. Storage drives are improving in speed **but testing processes have not kept pace**. Exploring additional testing options during downtime can help identify obscure bugs or specific violations.

Motivation

- Long-term storage, like **NVMe drives**, is crucial for saving files, media, and running applications smoothly.
- NVMe drives power modern computing but must be **rigorously tested** to prevent data loss.
- Traditional tests exhaust quickly, causing idle time and wasted computing resources.

Solution

- We're developing a **brute force approach** to maximize testing, finding more bugs and edge cases.
- **NVMe-CLI** is a tool we use to send commands to an SSD.
- A Python script connects to NVMe-CLI and runs tests using a TLA+ file and a **random seed**.
- The TLA+ file contains predefined tests, and the seed selects tests randomly.
- This approach allows extra tests during idle time, helping find SSD errors.



Testing

- Ability to dynamically interact with the simulation
- NVMe CLI functionality
- Logging testing
- Integration testing with various NVMe components

Challenges

- Interpreting the TLA+ output and using it to drive commands to NVMe controller
- Logging management and organization
- How to test a tool that is used to test a component
- Team NVMe command validation and knowledge

Future work

- Asynchronous event handling (adding watchdog timing and concurrency)
- Dynamic TLA+ variable functionality
- Better I/O pattern recognition
- Multi-threaded execution to multiple NVMe devices

Technologies

