Corruption Tolerant Operating System for GPUs in Low Earth Orbit

Multi-view Onboard Computational Imager (MOCI)

Eric Miller[*1], David L. Cotten[^1,22], Deepak Mishra[1,2]

[*]EricMiller@uga.edu, [^]dcotte1@uga.edu, [1]Small Satellite Research Laboratory, University of Georgia,

[2]Center for Geospatial Research, University of Georgia

Overview

The Multiview Onboard Computational Imager (MOCI) is the second satellite to be launched by the Small Satellite Research Lab (SSRL) at the University of Georgia. The satellite mission involves the monitoring of environmental phenomena, which entails three-dimensional terrain modeling. Such models require high-power data processing, which, in most cases, would be completed from the ground. However, as this would involve downlinking large amounts of data, the SSRL plans to add automated, more efficient onboard data processing to MOCI through the inclusion of a GPU—the Nvidia Jetson TX2i.

Unlike on Earth, radiation causes non-negligible problems for computers, particularly bit flips [1]. While the onboard computer (OBC) contains built-in software-level redundancy for fault tolerance, the same is not provided on the TX2i as a backup to physical shielding.

Methods

To mitigate potential corruption to the operating system, the boot image was triplicated on the TX2i's filesystem. Then, modifications were implemented on its stage 3 bootloader, U-Boot [2], in order to implement triple-modular redundancy.

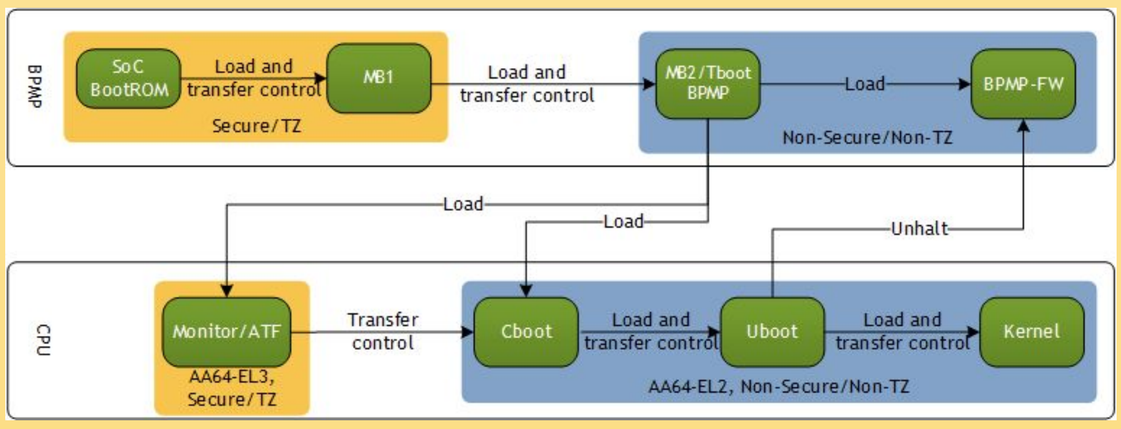

Figure 1: The control flow of the TX2i's boot process.

In addition to the triplicated boot image, each image was stored with a pre-calculated hash (MD5). Upon boot, the custom U-Boot script began by:

- Rehashing each of the images
  - If any of these hashes matched the stored hash on the filesystem, such image would be assumed non-corrupt and therefore chosen for boot.
- In the scenario where all three images are corrupt, and none match their hashes:
  - The three images would be fed into a majority voting logical gate one bit at a time. This would construct a new (probabilistically non-corrupt) image from which the TX2i could boot.

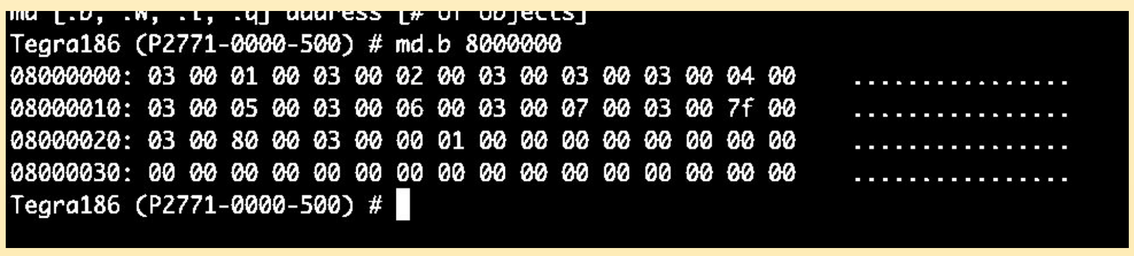

Figure 2: Sample U-Boot command environment. `md.b` accesses the bytes that have been loaded at the given U-Boot memory address.

These custom commands were developed on a Linux-for-Tegra [3] host device and flashed onto the TX2i. Specifically, they were developed within U-Boot's command-line environment (see Figure 2) for ease-of-testing, and eventually ported over to the bootloader's main loop.

Results

In practice, the use of hashes to check for corruption, rather than direct or automatic bit-voting, allowed the TX2i to maintain a relatively-quick average boot time.

Multiple tests were conducted on the implementation once ported to U-Boot's main loop in order to ensure reliability:

- single-image corruption
- random-bit corruption (to, in a way, simulate the radiation environment)
- all-byte corruption, where all bytes were changed on all images (albeit on different bits)

The TX2i passed all tests and was therefore able to self-correct corruption and reboot.

Future Work

The SSRL is currently partnering with the Johns Hopkins Applied Physics Lab [4] in order to perfect a fault-tolerant TX2i operating system. This primarily involves operating system minimization, which entails the removal of unnecessary programs or features of the software in order to reduce potential vulnerabilities.

In addition, we are working to add filesystem checks, to keep filesystem tables in place for safe indexing.

References

[1] "IEEE standard for environmental specifications for spaceborne computer modules," *IEEE Std 1156.4-1997*. [Online]. Available: www.ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=603627&tag=1

[2] "The DENX U-Boot and Linux guide (DULG) for canyonlands," 2018. [Online]. Available: www.denx.de/wiki/DULG/Manual

[3] NVIDIA Tegra Linux driver package development guide. [Online]. Available: https://docs.nvidia.com/jetson/ archives/l4t-archived/l4t-322/

[4] Johns Hopkins Applied Physics Laboratory. [Online]. Available: https://www.jhuapl.edu/