NASA KSC/Intern Abstract – 07/05/18

Yale

Command and Control Software Development

Student Name: Zhengdong Wang
Academic Level: Undergraduate
Academic Major: Computer Science
Academic Institution: Yale University

Mentor Name: Jamie Szafran
Job Title: Computer Engineer, AST
Org Code/Branch or Division: NE-XS
Directorate: Engineering

Abstract

The National Aeronautics and Space Administration (NASA) has entered a new chapter of its history. As technology in spaceflight advances, new participants have joined in the journey to explore the stars. NASA's Kennedy Space Center (KSC), the central site of many of the agency's historic launches, has in turn diversified its role. Evolving into a multi-user spaceport, commercial partners such as SpaceX, Blue Origin, and United Launch Alliance can largely independently run their own spaceflight operations.

As NASA's role in spaceflight operations change, so too does its design priorities. NASA's flagship venture, the Space Launch System (SLS), is being created primarily to explore deep space. While commercial partners conduct low Earth orbit resupplies, tests, and even transportation of astronauts, NASA can divert more resources to more exciting projects — going back to the moon, or onwards to Mars. In the short term, this goal manifests in the upcoming Exploration Mission 1 (EM1) test of the SLS, slated for as early as 2019. SLS, with the Orion capsule, will travel farther than any human-rated spacecraft has gone before. As the most powerful launch vehicle ever created, SLS requires many new innovations to ensure mission success. One such technology is the launch control software, which is the focus of this internship.

The SLS launch control software is composed of many functions, all of which require rigorous testing to meet the standard of life-critical code. To facilitate easier testing, the first project I undertook was to customize the open-source tool Wireshark to the software team's needs. Wireshark is a network protocol analyzer that can take in information about custom information packets. The launch control software will have several protocols that are custom to NASA. I ensured that all necessary component files were the correct file type and in the correct structure for Wireshark to compile. Having allowed Wireshark access to understand custom NASA data packets, testers of the launch control software will be able to find anomalies easier.

It is also imperative that the launch control software runs as efficiently as possible — it is close to a real time system. In instances where multiple operations can be done concurrently, the

launch control software employs multithreading, or executing multiple threads to increase the use of individual central processing unit cores. As SLS is a complex piece of machinery, the launch control software needs to constantly read in changing data from multiple sources. Reading and updating the vehicle's data needs to happen quickly and accurately to ensure the safety of the astronauts and the onboard systems. I improved on the multithreading of one component of the launch control software. Previously, a particular operation was single-threaded because it only collected data from a few sources. As the launch control software grew, however, using a single thread to read an ever-growing list of data sources became a bottleneck. I contributed to an effort to redesign, implement, and test a multithreaded approach to data reading in this component of the launch control software. Users will not need to change how they use the software. But under the hood, the launch control software will run faster, using better practices.

EM1 and later exploration missions will take the achievements of human spaceflight farther than ever before — and the mission begins with launch at KSC. With any life-critical mission, every part of the process, no matter how insignificant it may seem, could mean the difference between mission success and failure. NASA takes as many precautions as possible, from strict testing to performance standards. I am grateful to have the opportunity to contribute to NASA's next human spaceflight endeavor through this internship.

Acknowledgements

I would like to thank Jamie Szafran for selecting me for this internship and giving me everything I need to succeed. I learned a lot about not only computer science but also how to tackle my first triathlon. I am extremely grateful also to everyone in the OSBII intern room, especially my project teammates Brandon Echols and Elijah Spiro, without whom it would have been impossible to do my job. From Nova-feeding to coffee (and Haroon) roasting to the Robot troubleshooting that I never had to experience, you all made this experience unforgettable.