NASA and Blue Origin's Flight Assessment of Precision Landing Algorithms Computing Performance

David Rutishauser[1] and Gavin Mendeck[2]
NASA Johnson Space Center, Houston, Texas, 77058, United States

Ray Ramadorai[3]
Blue Origin, Kent, WA 98032, United States

John Prothro[4]
Avenue Technologies and Commodities Inc., Ponca City, Oklahoma, 74604, United States

Thadeus Fleming[5]
Odyssey Space Research, Houston, Texas, 77058, United States

Peter Fidelman[6]
Blue Origin, Kent, WA 98032, United States

NASA's Safe and Precise Landing - Integrated Capabilities Evolution (SPLICE) project continues NASA's work in the development and testing of technologies for Precision Landing and Hazard Avoidance (PL&HA). This paper presents results characterizing how SPLICE flight software utilizes the shared computing resources of the Descent Landing Computer (DLC), one of the PL&HA technologies under development. The SPLICE technologies are being tested as an integrated payload on Blue Origin's New Shephard suborbital vehicle. The results presented in this paper are measured by applications running in and with the flight software both in flight, and in a high-fidelity Hardware-in-the-Loop (HWIL) simulation environment. Linux utilities to measure performance are also executed from the command line in the HWIL configuration. Performance measurements of the SPLICE workloads executing on the DLC provide insight on how efficiently the software is utilizing the DLC resources. Examples of how these measurements have guided improvements in the flight code are presented. In addition, the DLC uses a commercial processor as a surrogate for NASA's High Performance Spaceflight Computing (HPSC) processor. This work provides insight on how an HPSC system may perform delivering PL&HA capabilities on a future mission. The measurements also can be used to infer architectural requirements for PL&HA capabilities, informing the HPSC project and other flight computer development efforts. Examples of the measurements collected include processor utilization, I/O bandwidth, cache and branch misses, and application profiles.

[1] SPLICE Avionics Lead, Avionic Systems Division.
[2] SPLICE Flight Software Lead, Aerosciences and Flight Mechanics Division.
[3] Principal Technologist Avionics.
[4] Software Engineer, Avionic Systems Division, NASA JSC.
[5] Software Engineer, Aeroscience and Flight Mechanics Division, NASA JSC.
[6] Software Engineer, Advanced Development Programs.

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

I.Nomenclature

DDL = De-orbit Descent and Landing
DLC = Descent and Landing Computer
DQG = Dual Quaternion Guidance
HPSC = High Performance Space Computer
LIDAR = Laser Imaging, Detection and Ranging
IMU = Inertial Measurement Unit
PL&HA = Precision Landing and Hazard Avoidance
RAM = Random Access Memory
SBC = Single-Board Computer
SPLICE = Safe and Precise Landing Integrated Capabilities Evolution
TRN = Terrain Relative Navigation

II. Introduction

The NASA/Blue Origin 2018 De-orbit Descent and Landing (DDL) Tipping Point includes two flights on the New Shepard launch vehicle with NASA and commercial sensor payloads. Incorporating results from these sensors into guidance and navigation solutions requires a high-performance computer. For the test flights on the New Shepard vehicle, NASA has developed a Descent and Landing Computer (DLC) as the compute platform to integrate the Safe and Precise Landing - Integrated Capabilities Evolution **(**SPLICE) project technologies under test. Today's DLC is a surrogate unit based on the Xilinx Multi-Processor System on a Chip (MPSoC) Ultrascale+ and does not meet environmental requirements for long duration spaceflight. Future SPLICE development will replace today's surrogate DLC with space-grade hardware such as the High Performance Spaceflight Computing (HPSC) processor. Multicore or other high-performance processing such as Single-Instruction, Multiple Data (SIMD) is becoming increasingly relevant to capabilities planned for future space missions, including Precision Landing & Hazard Avoidance (PL&HA). An understanding of how the SPLICE algorithms utilize the shared resources of the DLC, during integrated operation in flight, provides valuable insight on the SPLICE payload performance. These measurements not only reveal where performance optimizations are most necessary, but also inform the selection of space-grade hardware.

This paper describes the study approach, results and conclusions associated with the SPLICE DLC performance evaluation. Section III gives a brief description of the system's hardware and software architecture and the algorithms that it executes. Section IV describes the performance metrics collected during the DLC's ground and flight testing and the methodology by which they were collected. Section V provides a summary of the results. Section VI contextualizes the results and describes how they have informed improvements to system performance. Finally, we offer conclusions in section VII.

III.System Description

The DLC system includes a high-performance computer, camera, navigation Doppler LIDAR, and an IMU for measuring attitude rates and lateral accelerations. A Hazard Detection LIDAR will be added for future test flights. The purpose of the DLC's computer is to execute SPLICE algorithm technologies under development, such as extended Kalman Filter guidance, Dual Quaternion Guidance (DQG), and Terrain Relative Navigation (TRN). Hazard Detection and Avoidance (HDA) will be added as a future capability. The Hardware-in-the-Loop (HWIL) configuration used for the ground data collection is shown in Figure 1.

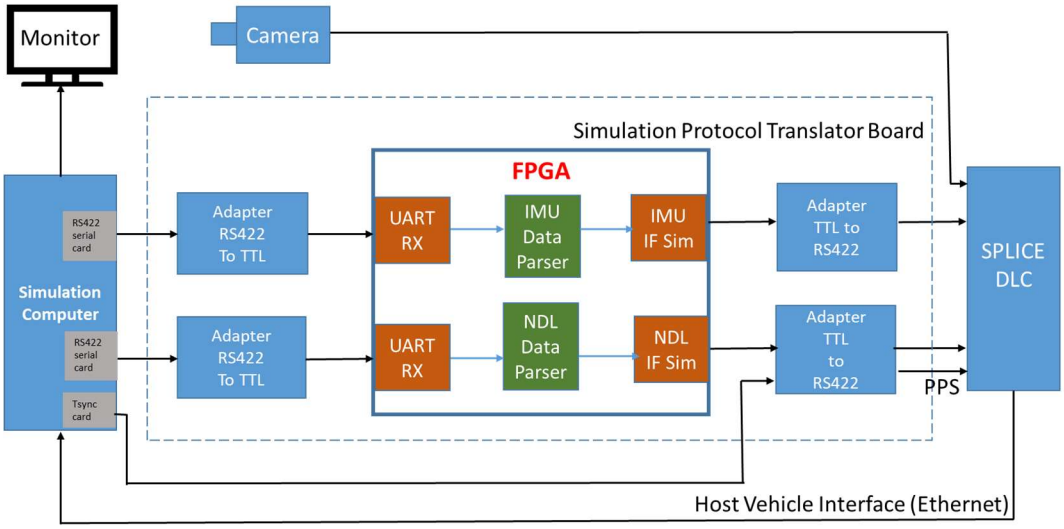


Fig. 1 SPLICE HWIL Test System Configuration.

IV.Performance Metrics

Measurements were collected both by manually executed Linux command line tools, applications running independent of the flight software, and by the flight software's built-in monitoring features. Performance measurements collected in the HWIL test configuration using the Tipping Point flight DLC build, and during the flight, are compared in this paper. Table 1 shows a run matrix detailing the measurements collected in the HWIL configuration. Nine runs were performed. Each run included the flight software's built-in utilization measurement CFS HS, a before and after *top* and *df* snapshot, and the standalone application multi-perf. The command-line utilities *mpstat* and *iotop,* and the in-house threadmon standalone application were alternative used in different runs. This was done to reduce overhead and interference from the measurement utilities.

Table 1. Performance Measurement Summary

		Quality					Flight1_Playback								
Measurement	**Tool**	**FSW**	DQG	TRN	NLU	NLP	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
processor utilization	**CFS HS, (1/sec)**	●	◔	◔	○	○	x	x	x	x	x	x	x	x	x
	mpstat	●	○	○	○	○			x						
	threadmon (1/sec)	●	●	●	●	●		x							
	top (before/after)	◑	◑	◑	◑	◑	x	x	x	x	x	x	x	x	x
SSD usage	df (before/after)	◕	○	○	○	○	x	x	x	x	x	x	x	x	x
SSD bandwidth	iotop	●	◔	◔	◔	◔				x					
DDR usage	**cFS HS?**	◔	○	○	○	○									
	OS ?	◑	○	○	○	○									
DDR bandwidth	?	○	○	○	○	○									
Cache misses	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Cache references	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Branch misses	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Branch references	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Instructions	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Cycles	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Context switches	**multi-perf, (1/sec)**	○	●	●	●	●	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*	*all*
Application profile	perf record	●	●	●	●	●					*all*	DQG	TRN	NLU	NLP

V. Results

Periods of high processor utilization were observed for the DQG and TRN algorithms, and a profile of usage is shown for TRN in Figure 2. Between February and April 2021, optimizations were made to the TRN code with the intent of reducing its processor utilization. Results show that these optimizations have been effective: while the algorithm still saturates a CPU core at 100% utilization for some portion of the mission, the duration of the saturation has decreased. TRN's increased efficiency is also visible in other statistics, with some other improvements being fewer cycles and fewer missed branches. The navigation algorithms (NLP and NLU) used so little processor time that they are not candidates for further optimization.

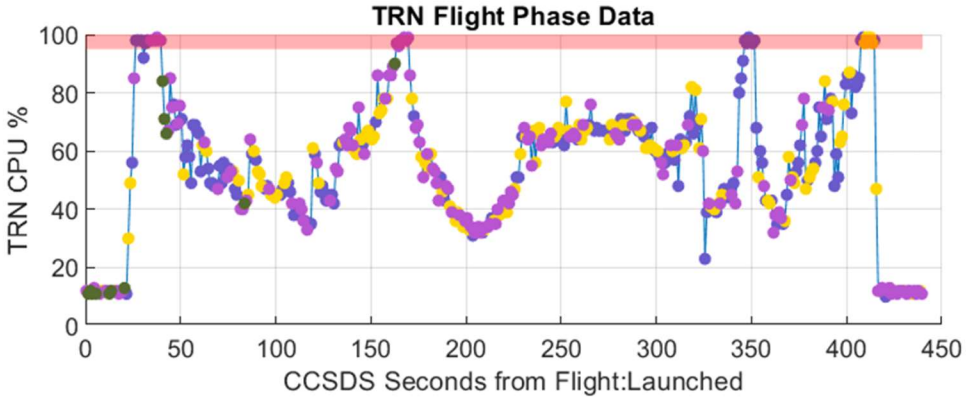


Fig. 2 HWIL Processor Utilization of TRN Algorithm.

VI. Discussion

The DLC system's processing bottleneck arises due to a mismatch between its processing capabilities and the algorithms' demands. The Ultrascale+ processor used on the surrogate DLC is a multi-core processor, (as are all other processors in its performance class), but the DQG and TRN navigation algorithms running on the DLC are mostly single-threaded and therefore unable to fully utilize the DLC's four cores. Parallelizing these algorithms, resulting in full utilization of four cores instead of two, would improve performance. The performance improvement of parallelization would be even more apparent on the HPSC because that computer is planned to incorporate eight cores instead of the surrogate's four. Another possible solution to the processing bottleneck would be to select a processor with greater single-threaded performance.

VII. Conclusion

The results presented in this paper represent a single step in an ongoing cycle of performance optimization. The team will continue working with algorithm and software developers to appraise the performance improvements made so far and to determine what additional software changes are most likely to yield further speedups. These results will also influence the selection of space-grade DLC hardware compatible with the SPLICE algorithms' processing requirements.

Acknowledgements

This work was performed under Space Act Agreement # ACO-2019-051 between NASA and Blue Origin.

References

To be provided in the final paper