Human and Robotic Space Mission Use Cases for High-Performance Spaceflight Computing

Richard Doyle, Larry Bergman,
Raphael Some, William Whitaker
Jet Propulsion Laboratory
California Institute of Technology

Wesley Powell, Michael Johnson
NASA Goddard Space Flight Center

Montgomery Goforth
NASA Johnson Space Center

Michael Lowry
NASA Ames Research Center

Extended Abstract

Spaceflight computing is a key resource in NASA space missions and a core determining factor of spacecraft capability, with ripple effects throughout the spacecraft, end-to-end system, and the mission; it can be aptly viewed as a "technology multiplier" in that advances in onboard computing provide dramatic improvements in flight functions and capabilities across the NASA mission classes, and will enable new flight capabilities and mission scenarios, increasing science and exploration return per mission-dollar.

Space-qualified computing technology, however, has not advanced significantly in over ten years and the current state of the practice fails to meet the near- to mid-term needs of NASA missions. To sharpen understanding of this gap, a multi-center NASA team conducted a study to address the following questions:

- What are the paradigm shifting NASA space-based applications that drive flight computing?
- What are the requirements imposed on flight computing by these applications?

To answer the first question, a series of workshops was held with scientists and engineers from NASA Johnson Space Center (JSC), NASA Goddard Space Flight Center (GSFC) and the Jet Propulsion Laboratory (JPL). Both robotic and human

spaceflight mission applications were examined. The approach taken was to identify the high priority capabilities and mission scenarios, or "use cases" from the NASA future mission set, and to focus on those use cases that required high performance computing. Table 1 lists the identified use cases, organized for the two primary NASA mission directorates: the Human Exploration Mission Operations Diretorate (HEOMD) and the Science Mission Directorate (SMD). A straightforward conclusion is that a high performance spaceflight computer will indeed be game changing because the capability is needed for many planned space missions across the agency, and will enable new and dramatic mission applications that are strongly desired by advanced mission planners.

Table 1 — Future NASA Mission Application Use Cases Requiring High Performance Spaceflight Computing

HEOMD Use Cases	SMD Use Cases
Cloud Services	Extreme Terrain Landing
Advanced Vehicle Health Management	Close Proximity Operations/Formation Flying
Crew Knowledge Augmentation	Fast Traverse
Improved Displays and Controls	New Surface Mobility Systems
Augmented Reality for Recognition and Cataloging	Imaging Spectrometers
Tele-Presence	Radar
Automated Guidance, Navigation and Control	Low Latency Products for Disaster Response
Human Movement Assist	Space Weather
Autonomous & Tele-Robotic Construction	Autonomous Mission Planning
	Immersive Environments - Science Ops/Outreach

To answer the second question, a series of discussions was held with engineers and mission personnel from the NASA Centers, including the engineers and scientists who developed the use cases as shown in Table 1. By characterizing the required computing, the environment, the criticality of the application, and the system constraints, computing system and processor chip requirements were derived for each use case. In doing so, over 60 variations of the applications shown in the table above were examined. To reduce this to a manageable number, the most stressing and cross cutting of these applications were condensed into ten representative applications, which we termed '"Eigen-Applications." Each Eigen-Application represents a broad class of mission applications with similar requirements.

Expanding on our methodology, we surveyed upcoming NASA mission scenarios and for each scenario, we identified the flight software applications that drive flight computing needs. Many of the scenarios we examined were already under development within mission programs, such as pinpoint landing on planetary surfaces, or autonomous rendezvous and docking. But aware that mission design teams often size their ambitions to their default expectations about flight computing resources, we encouraged the teams to look at desirable mission capabilities

separately from computational constraints and then, after the mission scenarios were defined, to revisit the computational requirements.

The sessions were supported by systems, engineering, and science subject matter experts from GSFC, JPL and JSC. Each session had the following study objectives:

- Identify desired high value NASA mission scenarios and identify the applications, science and/or exploration return, and the required computing capabilities
- Describe the added value quantitatively in terms of extended mission benefits, e.g., science return, using appropriate metrics
- Describe the added value qualitatively in terms of new mission capabilities and/or new mission architectures that become realizable
- Identify required changes to other elements of the flight/ground system architectural elements in order to realize the benefit

With the set of use cases capturing future NASA mission scenarios in hand, our team determined the need for a framework within which to derive flight computing requirements from the set of identified flight software applications. The requirements template we developed for this purpose is shown here:

- Space Environment(s)
  - Radiation environment at the time of application; e.g., geosynchronous (GEO), low-Earth orbit (LEO)
- Spacecraft Power Environment(s) / Constraint(s)
  - Small spacecraft or rover with limited power availability, large spacecraft with large power budget
- Criticality / Fault Tolerance
  - Is this application life or mission critical, must it operate through faults, can it tolerate errors if detected?
- Real-time
  - Does the application have a hard real-time deadline; if so, what is the required timing?
- Type(s) of Processing
  - Algorithm kernel(s). Is it primarily e.g., digital signal processing (DSP), data base query, parallelizable, amenable to a data flow approach?
- Memory Access Pattern
  - Fetch data from memory once and then flow through processing chain, or continuous random memory access required
- Duty Cycle

— Continuous operation over a long period of time, short duration one-time, spiky and/or unpredictable

- Data Rate
  — I/O and memory data rates

In the full paper, we describe in detail a representative sample of the 19 NASA mission use cases. We also characterize the generalized Eigen-Applications that were derived from the full set of use cases, and serve to summarize future NASA flight computing requirements. As noted above, some of the mission applications we examined had already been identified within NASA programs. In these cases, our effort served to articulate the relevant flight computing requirements with greater fidelity. Some of the identified mission applications, however, were new or significantly extended, and serve to broaden the base of NASA mission-pull for high performance flight computing.

It has been about 15 years since the previous investment at NASA in a flight computer. Our study provides a clear and broad basis that a next-generation flight computing capability will be a necessary and timely ingredient for future mission success.

References

1. Tara Estlin, Benjamin Bornstein and Larry Matthies, "Software on Mars," *Communications of the ACM*, November 2012.

2. David R. Thompson, Daniel Q. Tran, David McLaren, Steve A. Chien, Larry Bergman, Rebecca Castano, Richard Doyle, Tara Estlin and Matthew Lenda, "Autonomous Onboard Science Data Analysis for Comet Missions," *11th International Symposium on Artificial Intelligence and Robotics Applications for Space,* Torino, Italy, September 2012.

3. James Alexander, Bradley J. Clement, Kim P. Gostelow, John Y. Lai, "Fault Mitigation Schemes for Future Spaceflight Multicore Processors, *AIAA Infotech*, Garden Grove, CA, June 2012.

4. James Alexander, Yang Cheng, William Zheng, Nikolas Trawny and Andrew Johnson, "A Terrain Relative Navigation Sensor Enabled by Multi-Core Processing," *IEEE Aerospace*, Big Sky, MT, March 2012.

5. Peter M. Kogge, Benjamin J. Bornstein and Tara A. Estlin, "Energy Usage in an Embedded Space Vision Application on a Tiled Architecture," *AIAA Infotech*, St. Louis, March 2011.

6. Kim P. Gostelow, "The Design of a Fault-Tolerant, Real-Time, Multi-Core Computer System, *AIAA Infotech*, St. Louis, March 2011.

7. Wesley A. Powell, Michael A. Johnson, Jonathan Wilmot, Raphael Some, Kim P. Gostelow, Glenn Reeves and Richard J. Doyle, "Enabling Future Robotic Missions with Multicore Processors," *AIAA Infotech*, St. Louis, March 2011.