Fault-Tolerant Spaceborne Computing Employing New Technologies

Workshop Report and Vision

This is a report on the May 28-30, 2008 workshop on Fault-Tolerant Spaceborne Computing Employing New Technologies held in Albuquerque, NM. This workshop was attended by about 70 people representing Government, industry, and universities. This document was constructed by reorganizing and editing the out brief recommendations (which are included as an appendix).

Applications Drivers [recommendation 8]

This workshop put forth the vision of vastly more computing power in space by adapting emerging "multi-core" and related technologies to the extended requirements for environmental and fault tolerance in space. The increased computing power could extend existing mission classes and could become an enabler for new missions. Many space missions in the abstract deliver varying mission value as a function of the space and ground computing capability. With 1000× or more compute capability potentially available in space through the use of new technologies, some spacecraft will be able to perform their "computational kernel" algorithms on larger data than before, in real time rather than intermittently, or more effectively in space than transmitting the data to ground. In other cases, a mission believed to be infeasible today could be enabled (or enabled sooner) by the availability of the larger amount of compute capability.

Mission Classes [recommendation 7]

The table below gives an initial example of this analysis. The plan over time is to analyze each mission class in terms of metrics, algorithms, and kernels, matching these against architectures and technology roadmaps to create a mission-specific roadmap for each mission class. The roadmap would be validated by developing benchmarks of the underlying algorithms and running them on terrestrial computers representative of what could be implemented in space.

Mission Class	**Compute Requirement Addressed**
Man-rated systems	More highly fault tolerant at constant SWP
Large Earth orbiters	Much higher performance at slightly higher SWP
Interplanetary probes	Longer life, lower power
Robotics systems	Fault tolerant in increasingly extreme environments, lower power
Sensor webs/large space structures	Ultra low power
Instruments	Specialized high efficiency, high bandwidth (e. g. data flow or hybrid μP + FPGA)

Architecture [recommendation 4]

We envision new computing architectures being developed for space missions based on emerging commercial “multi-core” and related technologies. Semiconductor performance boosts due to Moore’s Law are the fundamental advance that allows us to consider modest-scale parallel computers in spacecraft. However, Moore’s Law is no longer simply ratcheting up the computing power of microprocessors and FPGAs exponentially with time. Due to shifts in the technology, engineers of terrestrial systems must now use new architectures such as multi-core processors, GPUs, accelerators, and so forth to realize the performance boost offered by the underlying physical devices.

While the new commercial architectures may be useful to space, they need additional work to make them sufficiently fault and radiation hard. Over the years, parallel processing tools have been developed to give higher performance on terrestrial systems. The need for extreme reliability in space could be addressed in principle by using some of the parallelism for the purpose of redundancy and fault tolerance. Repurposing emerging commercial technology is an obvious future task and a part of this space computing vision.

With the preceding paragraphs as background, the vision for space computing will be to use emerging “multi-core” and related technologies and investigate their effectiveness for various classes of missions. These emerging architectures are expected to offer teraflops level performance at similar size, weight and power as microprocessor- or FPGA-based solutions today. Some of the issues to be considered are: fault models, interchangeability of components during mission, power efficiency, and application fit to different architectures (image processing, high speed positioning, data routing).

With this long term vision in mind, the participants at the Albuquerque workshop agreed on some initial steps in the areas of interconnect and memory.

Interconnect [recommendation 1]

The community agreed to an initial framework for developing space architectures by agreeing to support certain board-level interconnects. Modern architectures are generally interconnections of functional modules, each specific architecture differing on the topology and performance of the interconnect and the functions of the modules. The workshop agreed that there should be emphasis placed on PCI-Express, Rapid I/O, and Gigabit Ethernet (each of which addresses a different level of interconnect and are thus not competing), as spacecraft interconnect.

The workshop participants agreed to support a process for investigating space architectures based on standardized interconnects. The hardware sector would support rad hardening and space qualifying interface chips, switches, and so forth for these initial interconnects as well as encourage use of functional module boards compatible with these interconnects.

The architecture and software participants agreed that the initial set of architectures could be investigated by table-top bread boarding commercial versions of these parts. The

software participants would be able to use the commercial setups to develop the appropriate middleware, benchmarks, and so forth.

This arrangement would allow cost effective prototyping of architectures and software with a path to space qualification.

Memory [recommendation 3]

The workshop participants reiterated and extended the view that there is a need for a new non-volatile memory technology. This would be a memory technology that can reach the gigabit size and which can be integrated with semiconductor logic and computing devices/circuits. In addition to general storage, such a technology could become the configuration memory in FPGAs. The solution is expected to be a non-transistorized memory technology and there are efforts underway for the development of various options at the physical sciences level. (For example: 6KB CNT memories have been flown and 4Mb are in development; there is also CRAM memories.)

The extended plan for memory technology is for the architecture and software groups to get enough knowledge about the physics of future solutions to be able to design suitable architectures and software. For example, the way a future memory solution integrates with CMOS or speed and reliability characteristics may influence architecture and software.

System and Standards [recommendation 5]

The long term computing vision involves the development of standards following the same general structure as the High Performance Computing (HPC) industry. The HPC industry leverages commercial microprocessors through a set of hardware and software standards, benchmarks, and a limited number of custom chip designs. The space computing community at the workshop endorsed a similar structure but technically targeted at low power, high reliability, and radiation hardness. Over time, the government-sector consumers would develop a set of standards and metrics that over time could become requirements in procurements.

The initial proposed set of standards would include

1. CPU ISA specification (to enhance software robustness)
2. Fault protection (and traceability) design
3. Modular software design techniques to maximize reliability and reuse

There would be a series of design principles that could become the basis of benchmarks or procurement requirements

1. Features to enhance V&V
2. Features to enhance fault robustness
3. Features for real time design
4. Features for scalability
5. Features to maximize portability across different processor families

Reusable Software [recommendation 6]

The vision includes controlling software development costs through software reuse. The reference model is again from the HPC industry. The government consumers of HPC standardized on Fortran and C/C++ as supported languages and MPI and OpenMP as parallel communications APIs. These standards enabled consumers to develop code portable across all HPC computers. The space computing vision is to apply this model to space computing, which will involve different languages and new standards that offer control of faults and redundancy and are real-time.

The proposed initial set of standards includes:

1. interface drivers
2. numerical libraries
3. communications coders/decoders
4. fault protection methods

The space computing vision includes a cross-industry organizational structure, which is impacted by the prevalence of export-controlled information and other sensitivities. The following functions are seen to be part of this structure:

- [recommendation 3] Workshops in the 2008 workshop format, which included sessions for sensitive topics, export controlled topics, and public sessions that can be attended by foreign nationals.
- [recommendation 2] A password-protected electronic forum where approved US Persons could coordinate across the community. This facility would be administered by the Government or FFRDC.
- [recommendation 10] Technical working groups to apprise the Government of gaps and needs so Government agencies and coordinate investment avoid duplication. The groups would maintain a list of “top 10” challenges. These challenges will be made available to agencies such as NASA, DARPA, DOD, DOE, AF, and NSF. Challenges currently include interconnect and memory (as described above) and could later include issues such as multicore V&V, multicore R/T programming models, multicore fault protection methods, and state synchronization methods across multicore systems.
- [recommendation 11] Operating a multi-tiered cross-agency clearinghouse to organize needs and opportunities for flight validation of space computing hardware. This activity would collect and maintain lists of technologies ready to schedule flight infusion trials as well as a list of upcoming flight opportunities and their launch windows and other critical requirements. Software may have additional opportunities for flight testing through uploads to a spacecraft after the primary mission is complete. The task would be to create public and non-public lists of needs and opportunities and points of contact in various agencies capable of acting on matches as appropriate.

Appendix

Workshop out brief plus comments via e-mail from two reviewers.

Draft Recommendations from Fault-Tolerant Spaceborne Computing Employing New Technologies

Recommendations from Plenary Session

1. The workshop participants believe by consensus that the community should create rad-hard boards compatible with a rad-hard physical interface standard. While the group would have liked to have a single standard, PCI-Express and RapidIO were both deemed acceptable but neither obviously wins over the other. This would define a de-facto space architecture of functional modules connected through multiple instances (busses) of this interconnect. Aside from giving flexibility to the hardware designer, the software community is eager to use the hardware as a target for middleware. (The group also considered PCI-X, 1553b, gigabit Ethernet, Spacewire, but these options did not get consensus support of the group.)

2. Develop a password-protected electronic forum. This forum is expected to be used by users with wide-ranging interests, but a group of participants with an interest in fault-tolerance was most immediately interested. (This facility has been set up at Sandia; contact erikdebenedictis@sandia.gov for a password.)

3. Meet again in a year.

Hardware Group Recommendations

4. There is a compelling need for a rad-hard memory technology that can reach the gigabit size and which can be integrated. In addition to general storage, such a technology would become a “holy grail” for FPGAs. FPGAs need a way to store the active configuration in a rad-hard medium. This contrasts with the current method of storing the configuration data in an external rad-hard memory and then loading it into SRAM and scrubbing. The existing solution is insufficient because radiation hits to the SRAM will cause too much damage to the computation during the latency interval before scrubbing. The need is for the real active configuration storage to be rad-hard.

5. The hardware group is intrigued by multi-core, GPU, accelerators, and other emerging architectures and advises them to be investigated for a potential future role. These have the same power as a PC processor, but 1-2 orders of magnitude more throughput. Some of the issues to be considered are: fault models, interchangeability of components during mission, power efficiency, and application fit to different architectures (image processing, high speed positioning, data routing).

Software Group Recommendations

6. Form a **Hardware/Software Interoperability WG** to devise standards and architectural principles that ultimately will simplify software design and overall

system robustness. Will provide feedback to hardware designers (and vice versa). A few example areas of scope:

a. CPU ISA specification (to enhance software robustness)
b. Fault Protection (and traceability) design
c. Modular software design techniques to maximize reliability and reuse.
d. Develop a list of hardware features (do's and don'ts) that will enhance V&V, fault robustness, real time design, scalability, and portability across different processor families.

7. Form an "Open Source Multicore Software" committee to identify, standardize, and archive common spacecraft software that can be (re)used across various systems and agencies to avoid NIH. Examples:
   e. Interface drivers
   f. Numerical Libraries
   g. Communications Coder/Decoders
   h. Fault Protection Methods
   i. ….

8. Form a group to study if it makes sense to devise different classes of spacecraft architectures depending on the target end mission (rather than one size fits all). Examples might include:

	Spacecraft Class	**Avionics Architecture**
a.	Man-rated systems	- highly fault tolerant
b.	Large Earth orbiters	- high performance
c.	Interplanetary probes	- long life, low power
d.	Robotic systems	- extreme environments, - low power
e.	Sensor webs / large space structures	- ultra low power
f.	Instruments	- data flow arch, hybrid - MC + FPGA
g.	….	

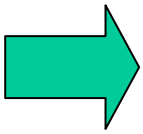
9. Form a team to oversee the development of roadmaps for "applications trends" as well as a "processor evolution" (similar to the SIA roadmap)
   a. Create a space applications taxonomy, metrics, and analyze how key applications algorithm kernels drive spacecraft computer architecture
   b. Spans wide range of apps, e.g., science, instrument, health management, control, navigation, …
   c. Possible focus group task: develop an applications survey and matrix from which space computer requirements can be extracted. (Note: Should precede prior recommendation #8)

10. Need to maintain a list of top challenges and then create a clearing house for various agencies to draw on for future BAAs

Technology R&D Challenges

- Multicore V&V
- Multicore R/T programming models
- Multicore fault protection methods
- State synchronization methods across multicore systems

….



Potential Funding Agencies

- NASA SMD
- NASA ESMD
- DARPA
- DOE
- AF
- NSF
- Intel

11. Form a working group to act as a broker to help identify opportunities and facilitate MC technology infusion into missions for flight validation
    a. Would be nice to have a running list of mid TRL technologies ready for flight infusion trials as well as a list of upcoming flight opportunities and their launch windows and other critical requirements.
    b. Software may have slightly more opportunities than hardware to be “flight tested” through uploads to a spacecraft after the primary mission is completed.