Introspection-Based Fault Tolerance for Future On-Board Computing Systems

Mark L. James and Hans P. Zima
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109
{mjames,zima}@jpl.nasa.gov

Introduction

NASA's and JPL's future missions of deep space exploration face the challenge of designing, building, and operating progressively more capable autonomous spacecraft and planetary rovers. Given the latencies of spacecraft-Earth communication for such missions, the need for enhanced autonomy becomes obvious. Earth-based mission controllers will be unable to directly control distant spacecraft and robots to ensure timely precision and safety, and to support "opportunistic science" by capturing rapidly changing events, such as dust devils on Mars or volcanic eruptions on a remote moon in the solar system [OASIS.07]. Furthermore, the high data volume yielded by smart instruments on-board the spacecraft would overwhelm the limited bandwidth of spacecraft-Earth communication, enforcing on-board data analysis, filtering, and compression. Finally, on-board science processing, such as Synthetic Aperture Radar (SAR) applications, may need more than 100 Gigaflops computational power.

Scalable COTS-Based Multi-Core On-Board Architectures

Traditional missions have been relying on radiation-hardened hardware architectures for providing fault tolerance, using fixed redundancy schemes with voting. Currently, the performance of such architectures lags that of commercial products by orders of magnitude; this gap is expected to even widen in the future. Therefore, this approach will not scale with the requirements for future on-board computing. A radical departure is necessary.

An extrapolation of the current trends for Commercial-Off-the-Shelf (COTS) multi-core architectures suggests the possibility of a multi-Teraflops on-board architecture by 2015, with about 100 processing cores operating within the power budget of the spacecraft. This leads to a new paradigm for spacecraft architectures: the ultra-reliable core component of the spacecraft responsible for control, navigation, data handling, and communication needs to be complemented by a COTS-based high-performance parallel processor (Figure 1). However, despite recent advances in technology, COTS architectures in space are subject to transient faults caused by radiation. Thus, there is the problem of providing fault tolerance for the COTS subsystem such that safe operation of the spacecraft is ensured. Addressing this issue is the main focus of this paper, which describes work in progress at the Jet Propulsion Laboratory.

Introspection-Based Fault Tolerance

We present an approach that has the goal of providing adaptive fault tolerance for mission software executing on

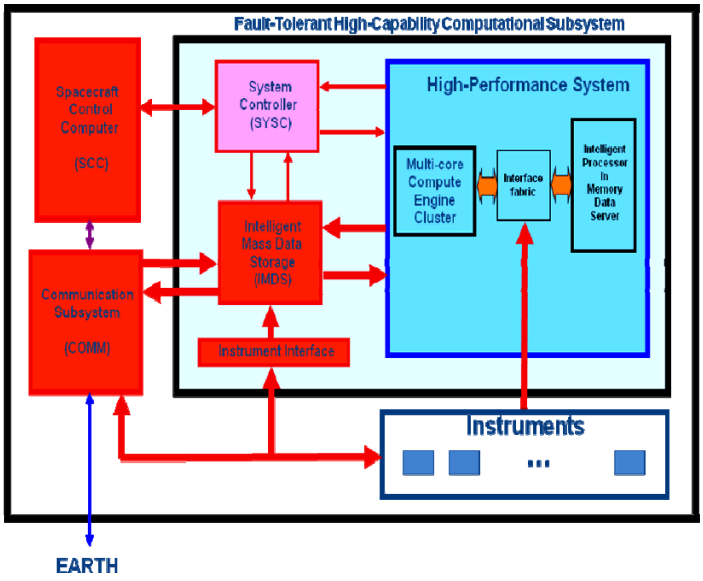

Figure 1: Scalable COTS-based on-board architecture

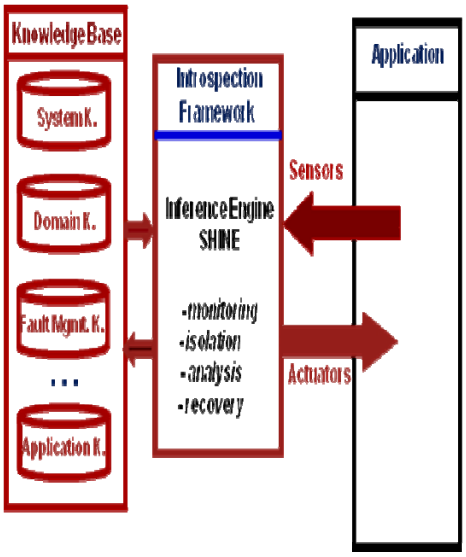

Figure 2: Introspection Framework

COTS-based high-performance on-board architectures (Figure 2). This approach relies on an introspection mechanism that supports automatic recovery in deep space or time-critical situations, minimizing the loss of function or data, and reducing the overhead of fixed redundancy schemes such as Triple Modular Redundancy (TMR) or N-Modular Redundancy (NMR).

Introspection enables a software system to become self-aware of its health, performance, and power consumption by monitoring its execution behavior, reasoning about its internal state, making decisions or recommendations about appropriate changes of the system or system state when necessary, and supporting recovery from faults. A *Core Capability for Introspection (CCI)* is reusable across multiple application domains and hardware platforms, providing system and application independence. As a proof of concept for the CCI, a *No-Loss Computation (NLC)* capability is being implemented that will plug in to the CCI infrastructure. The NLC will detect, analyze, and isolate the loss of function or data resulting from a fault by using a distributed class of no-loss computation agents in combination with automatically generated directives for the continuation of the computation to provide seamless autonomous recovery. Emphasis is placed on application-oriented fault tolerance that takes into account knowledge about the application and the algorithms and programs with which it is implemented.

Related Work

Fault tolerance for computing systems has been studied since the 1950s. Our work is specifically related to efforts for providing fault-tolerant on-board computing in space. Early work in this area includes the Advanced On-Board Signal Processor (AOSP), the Advanced Architecture On-Board Processor (AAOP) developed by Raytheon Corporation in the 1970s and 1980s, and the Space Touchstone computer developed at Honeywell Corporation that used COTS components in a space-borne computing system.

The Remote Exploration and Experimentation (REE) project conducted at NASA was among the first to consider putting a COTS-based parallel machine into space and address the resulting problems related to application-adaptive fault tolerance [Some.99]. The project developed Software-Implemented Fault Tolerance (SIFT) methods and libraries, and made important contributions to the conceptual foundations of this field. More recently, NASA's Millenium ST-8 project develops a “Dependable Multiprocessor” around a COTS-based cluster using the IBM PowerPC 750FX as a data processor, with a Xilinx VirtexII 6000 FPGA co-processor for the support of application-specific modules for digital signal processing,data compression, and vector processing [Ramos.06, Samson.07].

Our approach, as outlined in Figure 2, generalizes this work by: (1) developing application-oriented fault tolerance based on a newly developed introspection framework, (2) exploiting knowledge about applications and leveraging existing analysis methods for more efficient fault detection and analysis, (3) taking advantage of the opportunities offered by emerging multi-core systems for the efficient support of introspection, and (4) implementing the system on a cluster of Cell Broadband Engines, which are among the fastest systems available for embedded computing.

As mentioned above, our introspection-based approach relies exclusively on runtime detection, analysis, and recovery, although it can be effectively supported by static analysis. This is an immediate consequence of the necessity to deal with faults caused by external events, such as radiation, in logically correct programs. In contrast, verification and validation (V&V) methods, which are mainly concerned with verifying the logical correctness of programs, rely exclusively on static methods---i.e., methods that need to deal with the set of all inputs to a given program---or tests, where a program is run with a fixed input set. Runtime verification, as used by Haveland and co-workers for testing automatically generated programs for autonomy, also belongs in this category.

Finally, the concept of introspection, as used in our work, has been proposed by Peter Kogge, and outlined in [PARCO.04] and more recently described in [IEEE-Aero.08]. A similar idea has been used by Iyer and co-workers for application-specific security [Iyer.07].

References

[IEEE-Aero.08] M.L.James and H.P.Zima: An Introspection Framework for Fault Tolerance in Support of Autonomous Space Systems.
Proc.2008 IEEE Aerospace Conference, Big Sky, MT (March 2008)

[Iyer.07] R.K.Iyer et al.: Toward Application-Aware Security and reliability.
IEEE Security and Privacy, 5(1):57-62, 2007.

[PARCO.04] H.P.Zima: Introspection in a Massively Parallel PIM-Based Architecture.
Advances in Parallel Computing Vol.13,pp.441-448, Elsevier B.V., 2004

[OASIS.07] R.Castano et al.: OASIS: Onboard Autonomous Science Investigation System for Opportunistic Rover Science.
Journal of Field Robotics, 24(5),pp.379-397, 2007

[Ramos.06] J.Ramos et al.: High Performance Dependable Multiprocessor.
Proc.2006 IEEE Aerospace Conference, Big Sky, MT (March 2006)

[Samson.07] J.Samson et al.: High Performance Dependable Multiprocessor II.
Proc.2007 IEEE Aerospace Conference, Big Sky, MT (March 2007)

[Some.99] R.Some and D.Ngo: REE: A COTS-Based Fault-Tolerant Parallel Processing Supercomputer for Spacecraft Onboard Scientific Data Analysis.
Proc.Digital Avionics Systems Conference, pp.7.B.3-1-7.B.3-12, 1999