NMP ST8 Dependable Multiprocessor

John Samson,
Honeywell International, Aerospace Systems
john.r.samson@honeywell.com

Alan George
University of Florida
george@hcs.ufl.edu

Rafi Some
Jet Propulsion Laboratory, California Institute of Technology
Raphael.Some@jpl.nasa.gov

Abstract

Space exploration, science, and defense missions are requiring ever-increasing bandwidth and processing capacity to the extent that the ability to apply high performance COTS processors for onboard computing in space is becoming a critical need. NASA NMP ST8's Dependable Multiprocessor (DM) is an experimental system embodying an architecture and a methodology which enable COTS-based, high performance, scalable multi-computer systems to operate in space environments. DM technology encompasses fault tolerant middleware which manages a cluster of high performance COTS processors while providing enhanced SEU-tolerance, supports parallel and distributed processing for science applications, provides an application software development and runtime environment that is familiar to science application developers and facilitates porting of applications from the laboratory to payload data processors, accommodates future COTS parts/standards through HW/SW upgrades, and includes the capability for autonomous and adaptive control of fault tolerant configurations that are responsive to the environment, to application criticality, and to system modes which maintain the required dependability and availability, while optimizing resource utilization and system efficiency.

Flying high-performance COTS-based supercomputing in space is a long-standing desire of many in the NASA and DoD communities. Doing so is not without its problems. Three key problems that needed to be overcome in order to fly COTS in space are: 1) development of an effective approach to handle SEUs in high performance cluster processors, 2) handling the thermal issues associated with state-of-the-art COTS components, and 3) achieving high power efficiency (throughput per watt). The DM project has addressed and solved all three problems. DM solved the SEU problem by combining cluster management software with SEU tolerance-enhancing software in a flexible, efficient, integrated DM middleware suite. DM solved the thermal issue by mining the ruggedized, conductive-cooled, COTS airborne embedded processing domain. DM solved the power efficiency problem by mining the high performance, low power mobile computing processing domain.

The top-level, generic, Dependable Multiprocessor hardware architecture is depicted in Figure 1. The basic architecture consists of a redundant radiation-hardened system controller which acts as the highly reliable controller for a parallel processing cluster of COTS-based, high-performance, data processing nodes, a redundant network interconnect, and a redundant spacecraft interface. The system can be augmented with mission-specific elements, including mass storage, custom interfaces, and radiation sensors, as required.

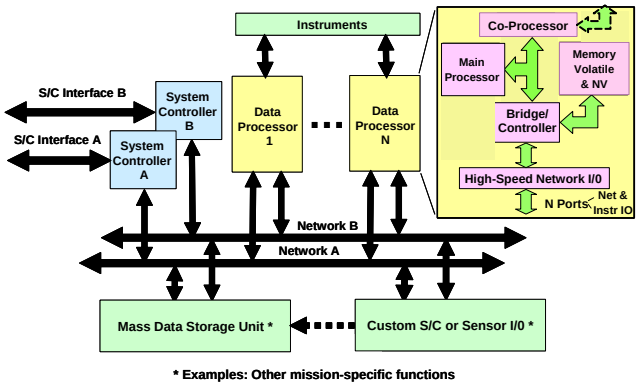

Figure 1 - Dependable Multiprocessor – Top Level Hardware Architecture

The top-level DM software architecture framework is depicted in Figure 2. The function of the DM system software is two-fold: 1) to support cluster operation for scalable high performance systems, and 2) to provide a system environment that enhances SEU-tolerances through software fault tolerance techniques. Figure 2 shows the two types of processing nodes: one type, the reliable system controller for control functions, and the second type, a high performance, COTS-based, cluster processing node. A key feature of the DM software architecture is the incorporation of a set of generic fault tolerant middleware techniques implemented in a software framework that is independent of and transparent to the specific-mission application, and independent of and transparent to the underlying platform (HW and Operating System). This independence and transparency is achieved through well-defined, high-level, application interfaces, an API (Application Programming Interface) to support mission-specific application needs, and an SAL (System Abstraction Layer) which isolates the remainder of the software system from the underlying platform, simplifying the porting of this software system to other platforms and allowing the generic fault tolerance middleware services to be available to future mission applications on future onboard processing platforms.

In support of scalable cluster processing, DM system software encompasses: system initialization including discovery/membership, self-test, the establishment of communication, and the establishment of system resource

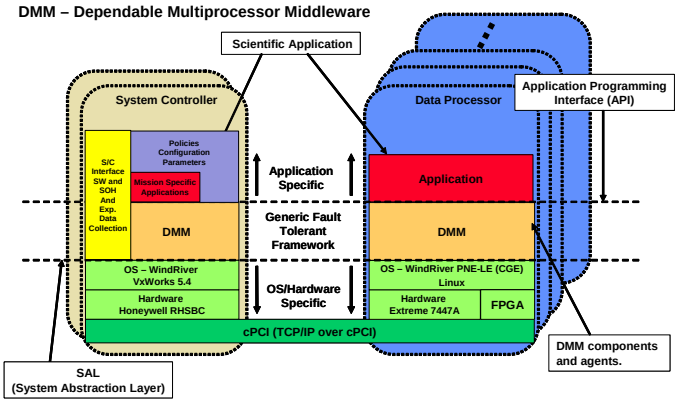

Figure 2 - Dependable Multiprocessor – Top Level Software Architecture

tables; basic job management services including loading/unloading, starting/stopping, pausing/resuming, transition handling, and dynamic maintenance of job and scheduling (periodic scheduling, frame-based gang resource tables; basic job execution services including job scheduling, a-periodic scheduling, triggered scheduling, continuous scheduling, and single executions) and job synchronization/coordination (application-based, process-based, task-based, event-based, and data-based); basic communication services including reliable messaging and user level APIs (Application Interfaces); and basic resource management services including effecting established mission policies and application execution modes, keeping track of resource status (busy/active/halted nodes, busy/active halted jobs and processes) and dynamic maintenance of resource tables.

As mentioned previously, one of the main problems flying COTS in space is the occurrence of SEUs. Physically, SEUs affect the hardware, but manifest themselves in software as errors. The types of soft errors encountered in applications and middleware include: data errors, control flow errors, hangs and crashes, OS exceptions induced by the applications, and communication errors and time-outs. The types of soft errors encountered in the OS include: kernel PANICs causing an OS hang or crash, HW error interrupts, and communication protocol errors. In the application and middleware domains, soft errors are detected by OS exception captures, by replication, by heartbeat and thread monitoring, by hang and crash timers, by exit handlers, by message traffic monitors, and by message error checking. Application data errors can be detected with spatial and temporal replication (SCP and TMR) and with ABFT (Algorithm-Based Fault Tolerance) techniques. Soft errors in the OS are detected with exception handlers, heartbeat monitors, processor and bridge chip exception capture, and communication error checking.

The DM project has been proceeding toward a flight validation experiment in 2009. DM cluster management and enhanced software-based SEU-tolerance were shown at the TRL5 technology validation demonstration in May 2006. Radiation testing of key COTS components selected for the flight experiment showed that these components exhibited no catastrophic latch-up and a sufficient number of SEUs to support the flight validation experiment. The DM project held a successful CDR (Critical Design Review) in June of 2007. The TRL7 flight experiment, including command and telemetry, has been defined and is currently being prototyped. In August 2007, NASA funding cuts eliminated the ST8 flight experiment. The ST8 project will end with the ground-based TRL6 technology validation demonstrations currently scheduled to be completed no later than October 2008.

One goal of the DM project is to develop platform-independent technology. DM technology has been successfully demonstrated on multiple x86 and PPC-based platforms, with heterogeneous processing clusters including FPGAs, and with heterogeneous operating systems, e.g., VxWorks, and Linux. This paper fits well with the theme of HPEC 2007 because DM technology is being extended to include multi-core processors. The details of the integration of IBM Cell technology in the DM framework are being reported in a second paper, "Dependable Multiprocessing with the Cell Broadband Engine," presented at HPEC 2007. DM technology has also demonstrated ease of use. Previously sight-unseen applications have been ported to DM testbeds in a few hours to less than a man-week.

While DM technology is currently being develop by NASA, primarily to support NASA science and autonomy missions including future lander and rover applications, the technology is also applicable to a wide range of DoD missions including UAVs, USVs, Stratolites, and ORS (Operationally Responsive Space).

The poster presentation provides an overview of the DM architecture, an update of the flight system hardware and software, an update of the ST8 flight experiment, and a summary of the work with multi-core processors. More detailed information about DM technology can be found in references [1] – [4].

References

[1] Samson, John, Jr., et. al., "Technology Validation: NMP ST8 Dependable Multiprocessor Project II," *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2007.

[2] Samson, John, Jr., et. al., "High Performance Dependable Multiprocessor II," *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2007.

[3] Samson, Jr. John R., et. al.,"NMP ST8 High Performance Dependable Multiprocessor," 10th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 20, 2006.

[4] Samson, Jr. John R., J. Ramos, A. George, M. Patel, and R. Some, "Environmentally-Adaptive Fault Tolerant Computing (EAFTC)," 9th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 22, 2005.