

Dependable Multiprocessor (DM) Support for Diverse and Heterogeneous Processing

Mathew Clark, John R. Samson, Jr., Susan Van Portfliet, Eric Grobelny
Honeywell Aerospace, Defense and Space Systems
Clearwater, FL
matthew.clark@honeywell.com

Abstract

Flying high-performance COTS-based supercomputing in space is a long-standing desire of many in the NASA and DoD communities. The objective of the Dependable Multiprocessor (DM) technology development effort is to demonstrate COTS-based, high-performance cluster processing that can operate in a natural space environment. The goals of the project are to provide high-throughput, low power, scalable, and fully programmable processing, while achieving high throughput density, technology independent system software that manages the cluster of COTS processing elements, technology independent system software that enhances radiation upset tolerance, high system availability, and high system computational correctness in terms of the probability of delivering correct and timely data.

To satisfy this need, over the past five years the DM concept has been developed and demonstrated as one of the experiments for NASA's NMP ST8 project. DM combines high performance, fault tolerant, COTS-based cluster processing with replication services, Algorithm-Based Fault Tolerance (ABFT), and fault tolerant middleware in an architecture and software framework capable of supporting a wide variety of mission applications. DM technology is currently targeted towards application domains with substantial backend processing (e.g., HSI, SAR and similar DSP applications) that only require radiation tolerance. This makes DM suitable for a variety of platforms including NASA science and autonomy missions, landers and rovers, DoD UAVs, USVs, Stratellites and Operationally Responsive Space (ORS).

The DM TRL6 technology validation demonstration was held in September of 2008. The demonstration included comprehensive software-based fault injection testing which profiled the fault/error syndrome mapping, and system-level radiation beam testing in which one (1) COTS data-processor board was exposed to a proton beam while executing an application suite and operating in the context of a DM flight system including all experiment interfaces and data collection software. The system-level testing validated DM design and operation in a radiation environment. Additionally, the system-level experiment validated the predictive DM Availability, Computational Consistency, and Performance models so that they can be

used to predict DM performance for different missions and in different radiation environments

DM technology is an experimental system embodying 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.

A generic DM hardware architecture is depicted in Figure 1. The basic architecture consists of a radiation-hardened system controller which acts as the highly reliable manager of a parallel processing cluster of COTS-based, high-performance data processing (DP) nodes, a high-speed network interconnect, and a spacecraft interface. As Figure 1 shows, one or more of these hardware elements may be redundant, depending on mission requirements and platform capacity. The system can also be augmented with mission-specific elements, including mass storage, custom interfaces, and radiation sensors, as required.

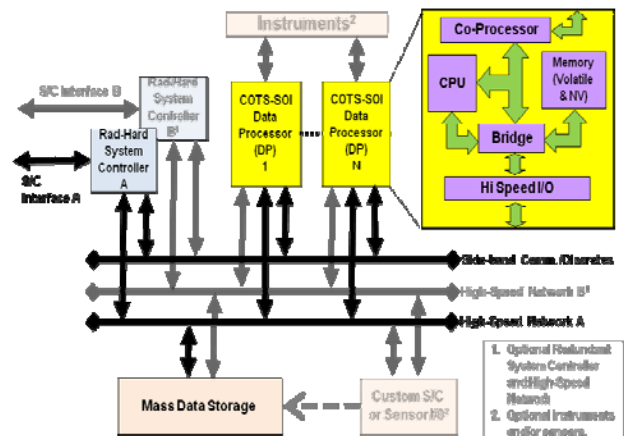


Figure 1: DM Top Level HW Architecture

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. The first is the reliable system controller, which is responsible for managing the cluster, including system initialization, DP node discovery/membership, self-test, the establishment of communication, and the establishment of system resource 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); 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. The second type is the high performance data-processing (DP) node, which executes user jobs/assignments issued to it by the system controller.

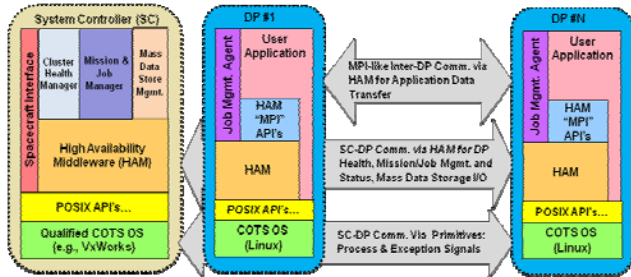


Figure 2: DM Top Level SW Architecture

Communications take place at different levels, from primitives for DP process and exception signals, HAM (High Availability Middleware) messaging between the DP's and the system controller for health, job and mission management, and an MPI-like API set provided to user applications to conduct reliable high-speed data transfer.

A key aspect of the DM's 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, as well as the underlying platform (HW and OS). DM uses well-defined interfaces, such as the MPI-like API for application data transfer and adherence to system software standard, such as POSIX. As a result, DM maintains a level of platform agnosticism that greatly simplifies the porting of DM to other platforms and, more

importantly, allows users to move their application from the lab to space with reduced cost/risk/schedule.

DM's platform agnosticism allows heterogeneous operation with different processing engines and different POSIX-compliant operating systems. To date, DM technology has been demonstrated with PowerPC 970, 750FX, 7447A, 7455 & 603e and Intel/AMD x86 single core processing engines, and with PA Semiconductor PA6T-1682M and Freescale 8641D dual core processing engines. The DM project is researching the extension of DM software to the Broad Reach Engineering BRE440 processor, the Tiler Tile64/MAESTRO, and other multi/many-core architectures such as the IBM Cell. The poster will discuss the results of these investigations, efforts toward diversity and heterogeneity and additional demonstrations of DM heterogeneous operation.

References

- [1] Samson, John, Jr., and E. Grobelny., "NMP ST8 Dependable Multiprocessor: TRL6 Validation - Preliminary Results," *Proceedings of the 2009 IEEE Aerospace Conference*, Big Sky, MT, March 8-13, 2009.
- [2] Samson, Jr. John R., et. al., "NMP ST8 High Performance Dependable Multiprocessor," 12th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 23 - 26, 2008.
- [3] Samson, Jr. John R., et. al., "NMP ST8 High Performance Dependable Multiprocessor," 11th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 18, 2007.
- [4] 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.
- [5] Samson, John, Jr., et. al., "High Performance Dependable Multiprocessor II," *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2007.
- [6] 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.
- [7] 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.