

Software-Implemented Fault Tolerance for Supercomputing in Space

John A. Rohr
Jet Propulsion Laboratory
Pasadena, California, USA
John.A.Rohr@Jpl.Nasa.Gov

The NASA Jet Propulsion Laboratory Remote Exploration and Experimentation (REE) Project is a large multi-year technology demonstration project which will develop low-power, scalable, fault-tolerant, high-performance computing for use in space and will demonstrate that significant onboard processing capability enables a new class of science missions. This will permit increased data collection rates, mitigate downlink limitations, and reduce ground station operations, enabling greater science return at lower cost.

The REE Project plans to attain these goals by adapting for routine use in outer space, earth-based supercomputing technology developed by industry, while dramatically reducing the mass, size, and power consumption of these systems. REE will adapt commercially-developed, ultra-low-power components for use in fault-tolerant architectures which can be scaled with available power to provide high performance and which can handle the high rate of radiation-induced transient errors that are expected in space. A testbed is currently under development which will be used to experiment with fault-tolerance implementations and demonstrate space science applications. After the experiments have been completed and the results analyzed, a flight prototype will be built.

A primary goal of the REE Project is to validate computing efficiencies on the order of hundreds of MIPS per watt in a multiprocessor architecture that can scale from 1 to 100 watts, depending on the specific application and mission requirements. At the low end of this spectrum, REE will develop spacecraft data systems, including mass storage, that are capable of operating on less than one watt of electrical power for extremely power-constrained missions. At the high end, REE plans to develop spacecraft data systems which will scale up to multiple processors to provide thousands of MIPS of performance for missions that are not severely power-constrained. These designs must be capable of reliable operation in space for 10

years or more using commercially-available components.

The REE Project intends to leverage commercial computing technology to the greatest extent possible to maximize performance and minimize power and cost while taking advantage of available software, support tools, and standards that are available. The use of components based on commercial designs creates a significant problem with transient errors (called single-event upsets or SEU's in space terminology) which can occur frequently in the space environment. Thus fault-tolerant designs will be required to provide satisfactory operation, and the fault-tolerance design approaches will face two fundamental constraints:

1. Since the REE Project has chosen to use components that are functionally identical to commercial components, no changes can be made to their designs. While this will enable use of all software developed for the commercially-available components, it also results in limiting the fault-tolerance mechanisms which can be incorporated to interface interconnection circuits surrounding the commercial components and software written in the environments provided.
2. The requirement for extremely low power precludes the use of massive redundancy techniques for all but the most critical functions and data.

Thus fault-tolerance mechanisms that are used within the constraints of the REE Project will be required to handle errors which result from single-event upsets and permanent failures of system components, and

they must facilitate system recovery and resumption of normal operation after a fault occurs.

It is expected that the usual fault-tolerance features which are incorporated in commercial technology will be available. For example, the communications systems can include multiple paths, error-detecting/correcting codes, retries, automated routing, and other capabilities. Memories can utilize multiple copies, error-correcting codes, and address protection registers. Overall control can include timers, heartbeats, and redundancy. Finally, any fault-tolerance features which are available in the processors which are used may be utilized to further increase the fault tolerance of the entire system. Additional hardware features which can be incorporated into the systems will be limited to only interconnection circuits that support fault tolerance since the processors themselves cannot be altered.

Although architectural mechanisms can be expected to contribute to the fault tolerance of computer systems developed by the REE Project, the limited flexibility in hardware design and implementation will require that much of the fault-tolerance capability be provided by the use of software-implemented fault-tolerance techniques. Because of the desire to maximize the use of commercial software already developed for the processors used in REE computer systems and because of the complexity of modifying such software, many of the software fault-tolerance capabilities provided in computer systems developed by the REE Project will most likely be provided by middleware, which is a layer of software which is inserted between the operating system and the applications programs. Middleware will be used to provide checkpointing and restart, multiple levels of fault tolerance for different applications running concurrently, monitoring of transient faults, and other capabilities such as consensus determination of redundant processes and results.

The use of middleware for software fault tolerance will not provide all the software capabilities needed. Faults that occur during execution of the operating system will largely be beyond the reach of middleware. Thus fault-tolerance mechanisms will be needed in the operating system itself. However, this area may be the most difficult to handle, since no major work on operating systems is planned as a part of the project. Applicable independent work which is done in this area, however, will be incorporated into the REE project.

Much of the fault-tolerance capability for the REE Project is expected to be provided by the applications programs themselves. Often only the programmer can best specify optimal placement for the checkpoints and the algorithms for acceptance checks and calibrated computations that are needed for fault detection and recovery. Although automated insertion of such capabilities would be desirable, it is beyond the intended scope of the REE Project to develop such techniques. However, many existing software fault-tolerance mechanisms and techniques are available which can be used by the REE applications programs. An initial check which can be done by all programs is to perform range checks and reasonableness tests. Data which is clearly out of range will indicate the possible presence of errors. Also, programs and constant data can be checkpointed and periodically checked to ensure that they have not changed. Beyond these simple checks, the computational results of many algorithms can be checked by applying inverse algorithms or other data manipulations. [For example, the inverse of a matrix can be multiplied ~~the~~ original matrix.] If the result is the identity matrix, both the original calculations and the check can be presumed to be correct. Another example is the use of assertions which can be dynamically checked for validity within the program.

It is well-known that software-implemented error detection coupled with the constraint of not using massive redundancy due to severe power limitations will limit the overall fault-tolerance coverage of the REE system. Thus the challenge lies in providing satisfactory performance within these constraints.

Since the primary goal for use of the high performance REE system is science data processing rather than spacecraft control, the primary objective is high availability rather than guaranteeing that no errors will be made in computations. Thus the system will contain a small, heavily-protected hard core that can periodically reload, diagnose, and restart the system to handle those errors and faults that may have been missed by the software-implemented fault-tolerance mechanisms.

The development of software-implemented fault tolerance is an integral part of the REE Project that will involve the JPL REE team, its contractors, the science application developers, and university researchers. Each has a part to contribute to ensuring the success of this important technology demonstration project.