Applications Development for a Parallel COTS Spaceborne Computer

Daniel S. Katz, Paul L. Springer, Robert Granat, and Michael Turmon

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

Contact: Daniel S. Katz, JPL/Caltech, 4800 Oak Grove Drive, MS 168-522, Pasadena, California, 91109-8099, USA, Daniel.S.Katz@jpl.nasa.gov, phone: 818.354.7359, fax: 818.393.3134
REE Vision

Move Earth-based Scalable Supercomputing Technology into Space

Background

- Funded by Office of Space Science (Code S) as part of NASA's High Performance Computing and Communications Program
- Started in FY1996

REE Impact on NASA and DOD Missions by FY03

Faster - Fly State-of-the-Art Commercial Computing Technologies within 18 months of availability on the ground

Better - Onboard computer operating at > 300MOPS/watt scalable to mission requirements (> 100x Mars Pathfinder power performance)

Cheaper - No high cost radiation hardened processors or special purpose architectures
Objectives

- **High Power Performance:**
  - Obtain power efficiencies of 300-1000 MOPS per watt
  - Develop an architecture that scales to 100 watts
    (depending on mission needs)

- **Fault-tolerance through system software:**
  - Enable reliable operation for 10 years and more
    (tolerate transient as well as permanent errors)
  - Using commercially available or derived components
  - Includes application services
    (such as Algorithm-Based Fault Tolerance)

- **New spaceborne applications:**
  - Run in embedded high-performance computers
  - Return analysis results to the earth; not just raw data
Remote Exploration and Experimentation Project

Overview

Feasibility?

Study Phase

>30 MOPS/watt

Scalable Testbed

>300 MOPS/watt

Flight Prototype

Fault-Tolerance

Real-Time

Spaceborne Applications

Demo spaceborne applications on embedded high-performance computing testbed

Scalable Applications I

Scalable Applications II

Activity Type:

- Computing Testbed
- System Software
- Science Applications
REE Implementation

- Use COTS hardware and software to the maximum extent possible
  - Assume that memory supports EDAC
  - Assume hardware detection of “standard” exceptions, but assume that some faults will go undetected
  - Fault tolerance achieved through software
- Keep overhead low
  - Emphasize techniques which do not require replication
- Maintain architecture independence
  - Design should not be tied to any particular hardware architecture
- “95%” rule
  - System does not have to be continuously available
  - Reset is acceptable recovery technique
- Target large applications, both parallel and distributed
  - Gigabytes of memory, gigaflops of processing
  - Scalable with high efficiency
  - Static load balancing sufficient
Current Partnerships

USAF Phillips Lab
Improved Space Architecture Concepts (ISAC)

- Inter-program coordination on a regular basis
- Joint participation on technical reviews and procurement actions
- Technical interactions to avoid duplicate investments and identify possibilities for joint investment