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
{mjames,zima}@jpl.nasa.gov

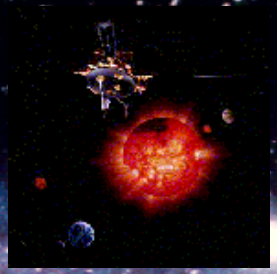
High Performance Embedded Computing (HPEC) Workshop

MIT Lincoln Laboratory, 23-25 September 2008

Contents

More than 50 NASA Missions Explore Our Solar System

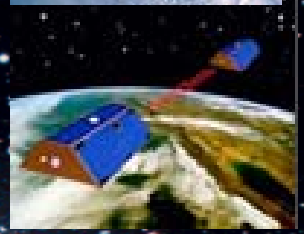
Spitzer studying stars and galaxies in the infrared

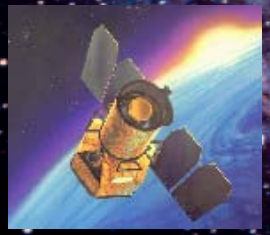


Ulysses studying the sun

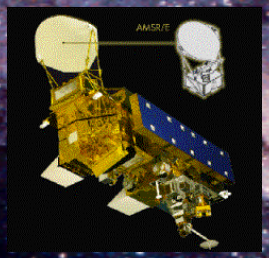
Mars Odyssey, rovers "Spirit" and "Opportunity" studying Mars

Cassini studying Saturn



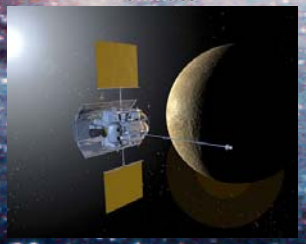


GALEX surveying galaxies in the ultraviolet

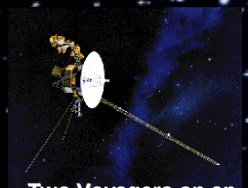


Aqua studying Earth's oceans

CALIPSO studying Earth's climate



MESSENGER on its way to Mercury



Two Voyagers on an interstellar mission

Aura studying Earth's atmosphere

Hubble studying the universe

QuikScat, Jason 1, CloudSat, and GRACE (plus ASTER, MISR, AIRS, MLS and TES instruments) monitoring Earth.

Chandra studying the x-ray universe

New Horizons on its way to Pluto

Space Challenges: Environment

Constraints on Spacecraft Hardware

- **Radiation**
  - *Total Ionizing Dose (TID)—amount of ionizing radiation over time: can lead to long-term cumulative degradation, permanent damage*
  - *Single Event Effects—caused by a single high-energy particle traveling through a semiconductor and leaving a ionized trail*
    - *Single Event Latchup (SEL)—catastrophic failure of the device (prevented by Silicon-On-Insulator (SOI) technology)*
    - *Single Event Upset (SEU) and Multiple Bit Upset (MBU)—change of bits in memory: a transient effect, causing no lasting damage*
- **Temperature**
  - *wide range (from -170 C on Europa to >400 C on Venus)*
  - *short cycles (about 50 C on MER)*
- **Vibration**
  - *launch*
  - *Planetary Entry, Descent, Landing (EDL)*

Space Challenges: Communication and Navigation

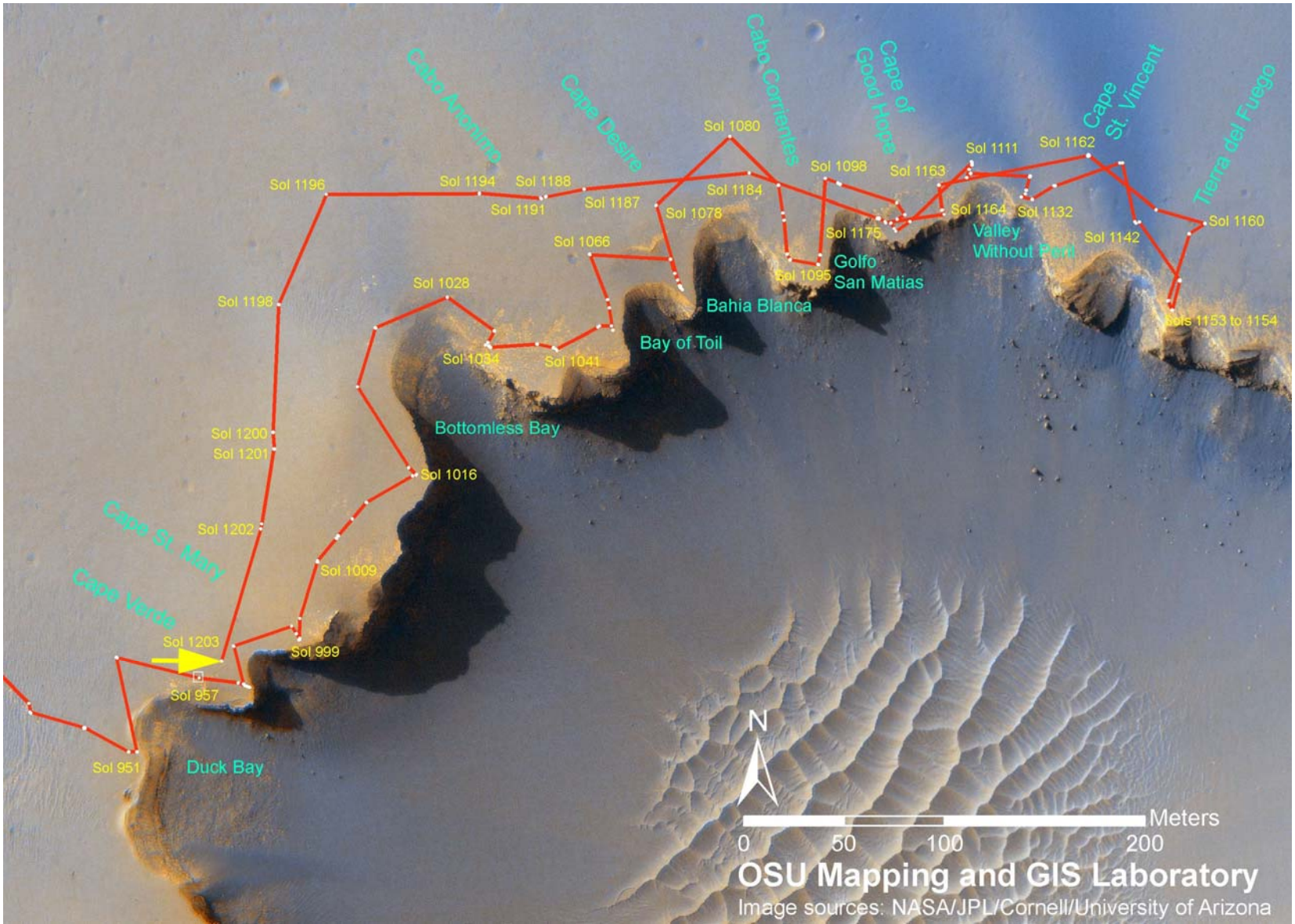
Constraints on mission operations

- **Bandwidth**
  - *6 Mbit/s maximum, but typically much less (100 b/s)*
  - *spacecraft transmitter power less than light bulb in a refrigerator*
- **Latency (one way)**
  - *20 minutes to Mars*
  - *13 hours to Voyager 1*
- **Navigation**
  - *Position*
  - *Velocity*

Space Challenges: Engineering

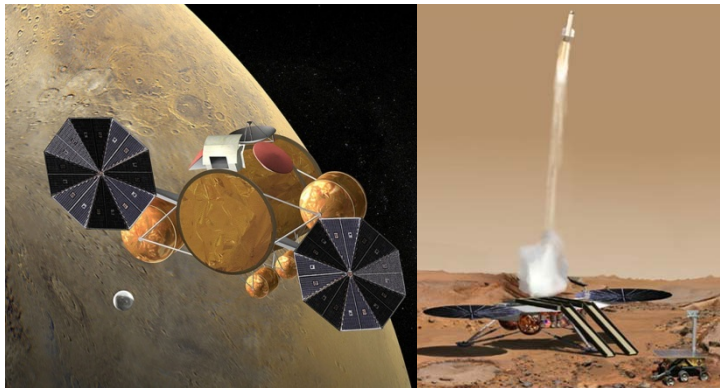
- **Only flight qualified parts are typically used**
  - *systems are at least 5 years out of date when launched—two generations behind commercial state-of-the-art*
- **Power and Mass Restrictions**
  - *20-30 W for a flight computer*
- **Often test of final system possible only when it is flown**
  - *importance of modeling and simulation*
- **Long mission duration challenges maintainability of ground assets in operations phase**
  - *Voyager is based on custom flight computer designed with MSI parts and ferrite core memory of the late 1960's (programmed in assembler)*

Duck Bay: Site of Opportunity's descent into Victoria Crater

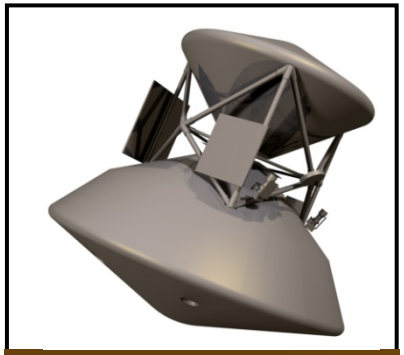


NASA/JPL: Potential Future Missions

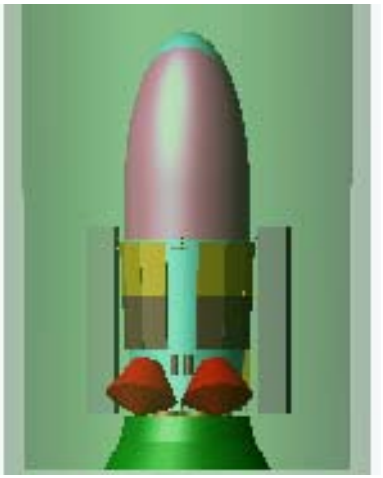
Artist Concept



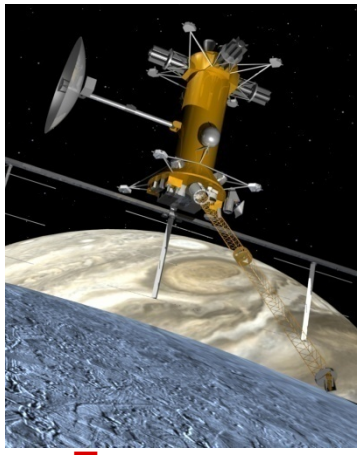
Mars Sample Return



Titan Explorer



Neptune Triton Explorer



Europa Explorer

Europa Astrobiology Laboratory

Future Mission Applications

- **New Types of Science**
  - *Opportunistic science (event detection: e.g., dust devils or volcanic eruptions)*
  - *Model-based autonomous mission planning*
  - *Smart high resolution sensors (e.g., Gigapixel, SAR,…)*
  - *Hyperspectral imaging*
- **Entry Descent & Landing**
  - *Flight control through disparate flight regimes*
  - *Landing zone identification*
  - *Lateral winds*
  - *Soft touchdown*
- **Surface Mobility**
  - *Terrain traversal, obstacle avoidance*
  - *Science Target identification*
  - *Image/video Compression*
- **Communication with Earth is a limiting factor**
  - *Small bandwidth requires reduction of data transfer volume; on-board data analysis, filtering, and compression*

New Requirements

New applications and the limited downlink to Earth lead to two major new requirements:

1. Autonomy

2. High-Capability On-Board Computing

Such missions require on-board computational power ranging from tens of Gigaflops to hundreds of Teraflops

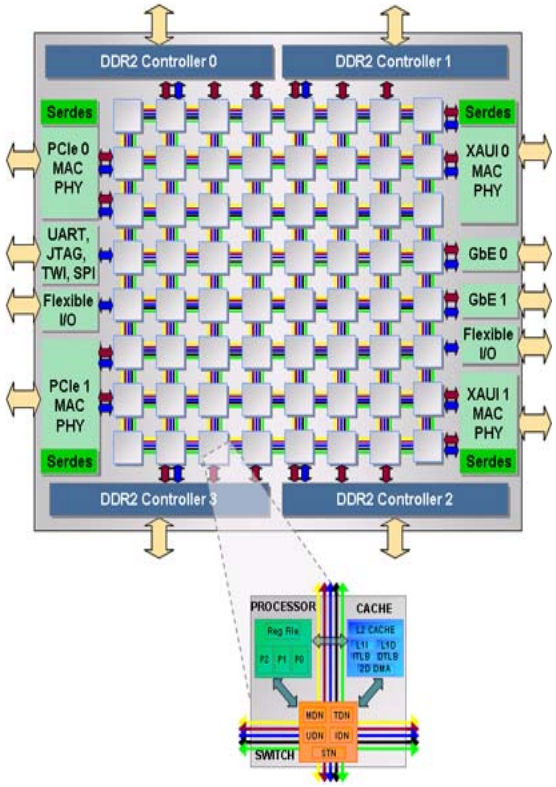
The Traditional Approach will not Scale

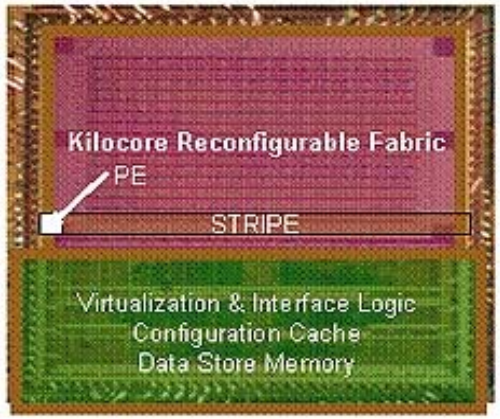
- **Traditional approach based on radiation-hardened processors and fixed redundancy (e.g.,Triple Modular Redundancy—TMR)**
  - ***Current Generation (Phoenix and Mars Science Lab –'09 Launch)***
    - ***Single BAE Rad 750 Processor***
    - ***256 MB of DRAM and 2 GB Flash Memory (MSL)***
    - ***200 MIPS peak, 14 Watts available power (14 MIPS/W)***
- **Radiation-hardened processors today lag commercial architectures by a factor of about 100 (and growing)**
- **By 2015: a single rad-hard processor may deliver about 1 GFLOPS—orders of magnitude below requirements**

Contents

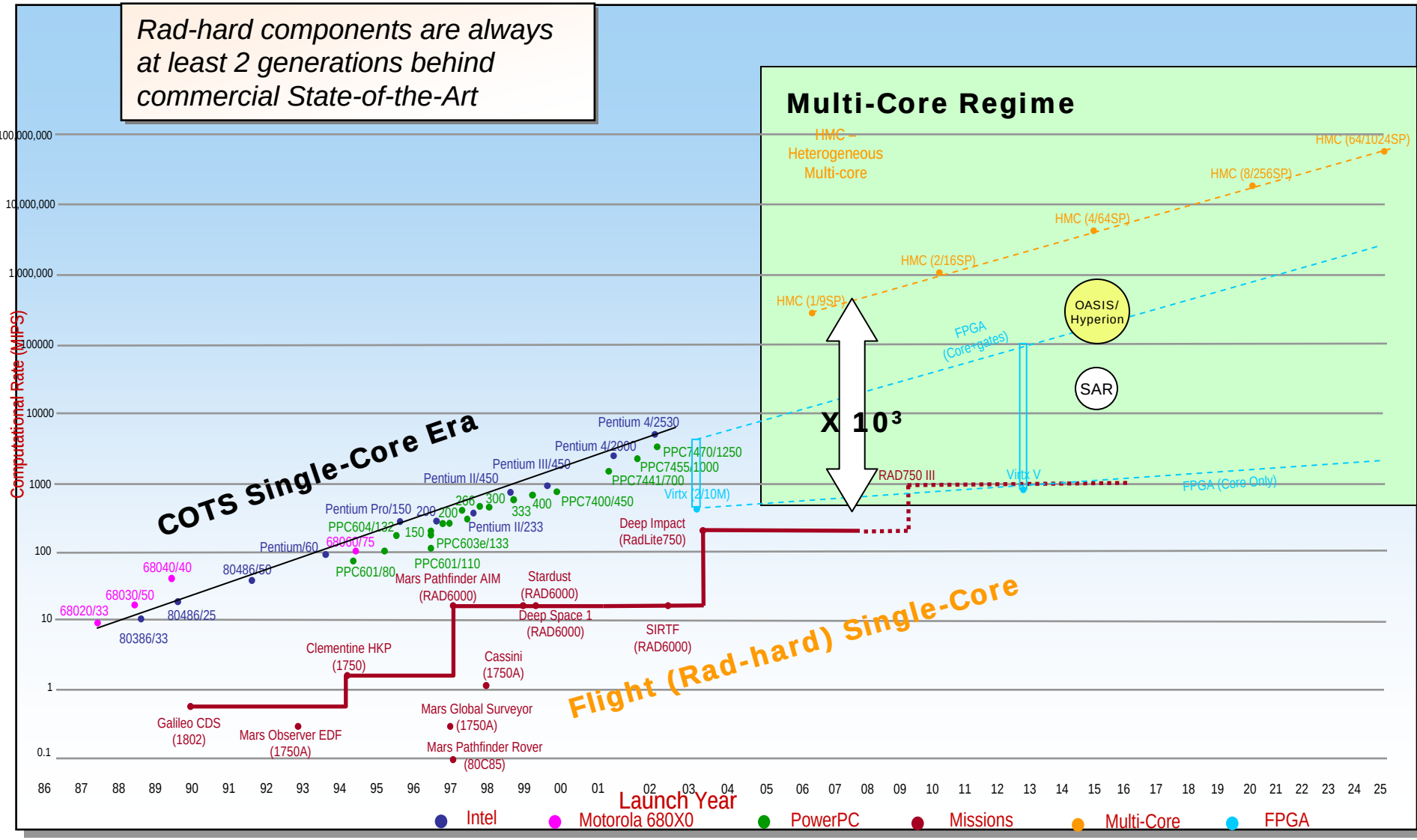
Future Multicore Architectures: From 10s to 100s of Processors on a Chip

- **Tile64 (Tilera Corporation, 2007)**
  - *64 identical cores, arranged in an 8X8 grid*
  - *iMesh on-chip network, 27 Tb/sec bandwidth*
  - *170-300mW per core; 600 MHz – 1 GHz*
  - *192 GOPS (32 bit)—about 10 GOPS/Watt*
- **Kilocore 1025 *(Rapport Inc. and IBM, 2008)***
  - *Power PC and 1024 8-bit processing elements*
  - *125 MHz per processing element*
  - *32X32 "stripes" dedicated to different tasks*
- **512-core SING chip (Alchip Technologies, 2008)**
  - *for GRAPE-DR, a Japanese supercomputer project*
- **80-core research chip from Intel (2011)**
  - *2D on-chip mesh network for message passing*
  - *1.01 TF (3.16 GHz); 62W power—16 GOPS/Watt*
  - ***Note: ASCI Red (1996): first machine to reach 1 TF***
    - *4,510 Intel Pentium Pro nodes (200 MHz)*
    - *500 KW for the machine + 500 KW for cooling of the room*





Space Flight Avionics and Microprocessors History and Outlook



Source: Contributions from Dan Katz (LSU), Larry Bergman (JPL), and others

Multi-Core Challenges for Space

- **General**
  - *parallel programming and execution models*
  - *complex hardware architectures*
  - *porting of legacy codes*
  - *programming environments*
  - *new methods for exploiting hardware: introspection, automatic tuning, power management*
- **Space Critical**
  - *real-time*
  - *fault tolerance*
  - *verification and validation*

Contents

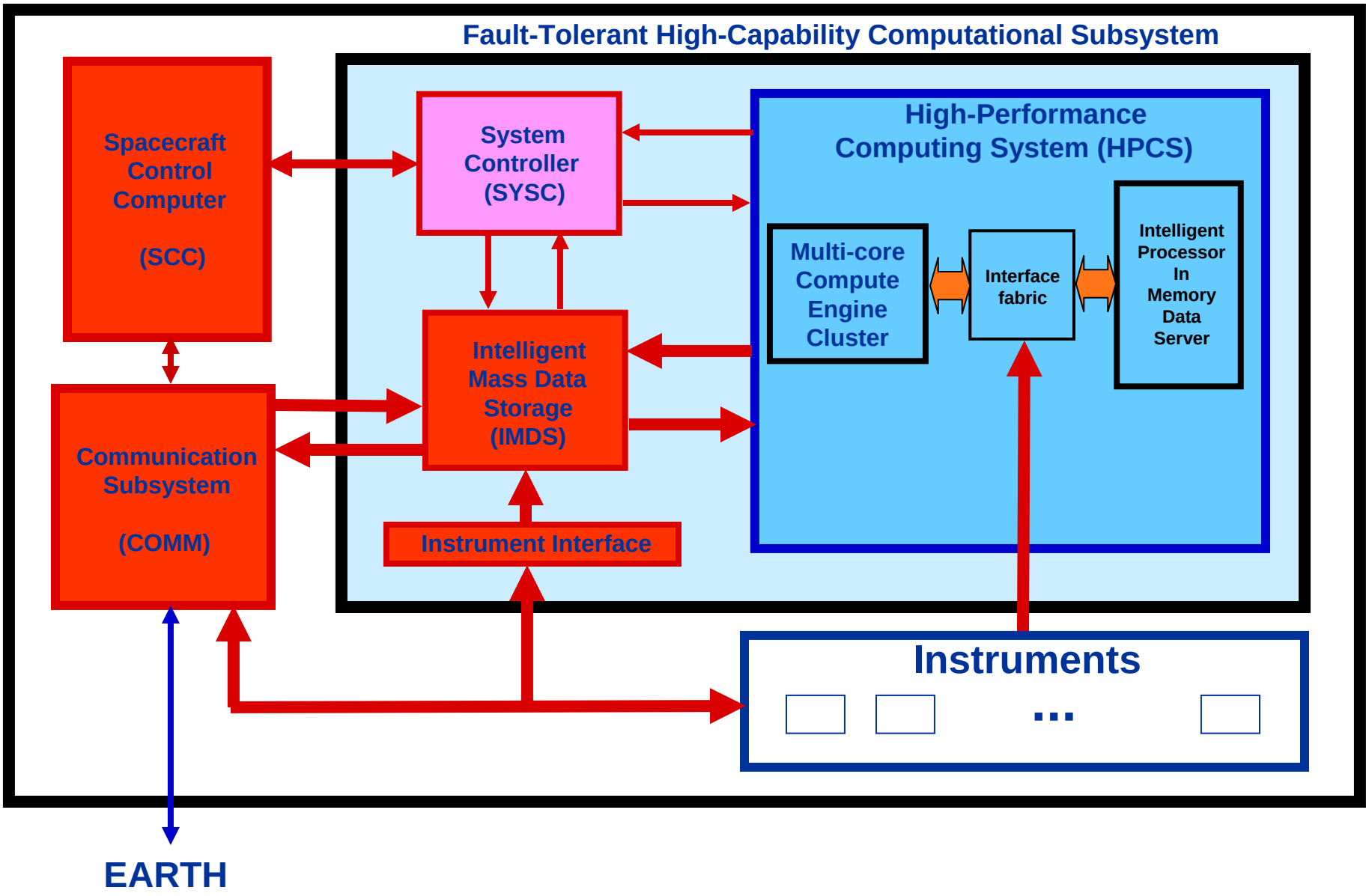
COTS-Based On-Board Systems

- **Basic Idea: augment the radiation-hardened core on-board system with a commodity high-performance computing system (HPCS) based on multi-core technology**

- **Earlier approaches—based on traditional multiprocessors**
  - ***Remote Exploration and Experimentation (REE) project at NASA***
  - ***ST8 Dependable Multiprocessor (DM) project (Honeywell, U. Florida, JPL)***

- **Key issue: provide fault tolerance for HPCS without relying on rad-hard processors or special-purpose architectures**

High-Capability On-Board System: An Example



Transient Faults

- **SEUs and MBUs are radiation-induced transient hardware errors, which may corrupt software in multiple ways:**
  - ***instruction codes and addresses***
  - ***user data structures***
  - ***synchronization objects***
  - ***protected OS data structures***
  - ***synchronization and communication***
- **Potential effects include:**
  - ***wrong or illegal instruction codes and addresses***
  - ***wrong user data in registers, cache, or DRAM***
  - ***control flow errors***
  - ***unwarranted exceptions***
  - ***hangs and crashes***
  - ***synchronization and communication faults***

Focus of this Work

- **Support for application-oriented, adaptive, and dynamic fault tolerance in the HPCS component**
- **Assumptions**
  - *HPCS: homogeneous cluster using COTS-based multi-core components*
  - *applications are non-critical, parallelization based on MPI*
  - *focus on hard and transient faults*
- **Approach**
  - *replacing fixed redundancy schemes with an application-adaptive approach, exploiting application and system knowledge, user input*
  - *based on an introspection framework providing a real-time inference engine*
  - *prototype implementation on a cluster of Cell Broadband Engines*

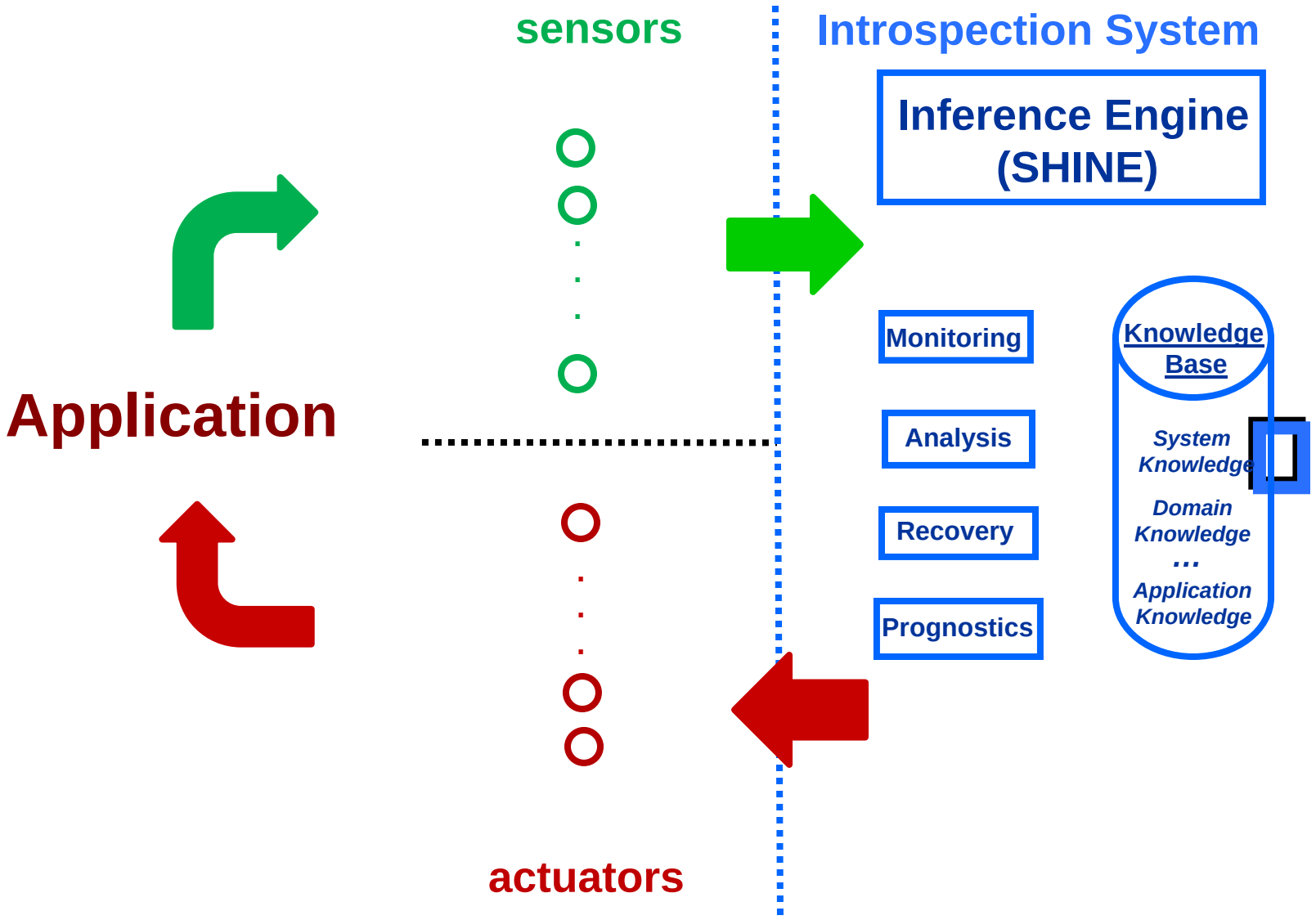
Contents

A Framework for Introspection

Introspection…

- provides *dynamic* monitoring, analysis, and feedback, enabling system to become self-aware and context-aware:
  - *monitoring execution behavior*
  - *reasoning about its internal state*
  - *changing the system or system state when necessary*
- exploits adaptively the available threads
- can be applied to different scenarios, including:
  - *fault tolerance*
  - *performance tuning*
  - *power management*
  - *behavior analysis*
  - *intrusion detection*

An Introspection Module (IM)



Sensors and Actuators

- ***Sensors*** **and** ***actuators*** **link the introspection framework to the application and the environment**
- **Sensors: provide** ***input*** **to the introspection system**

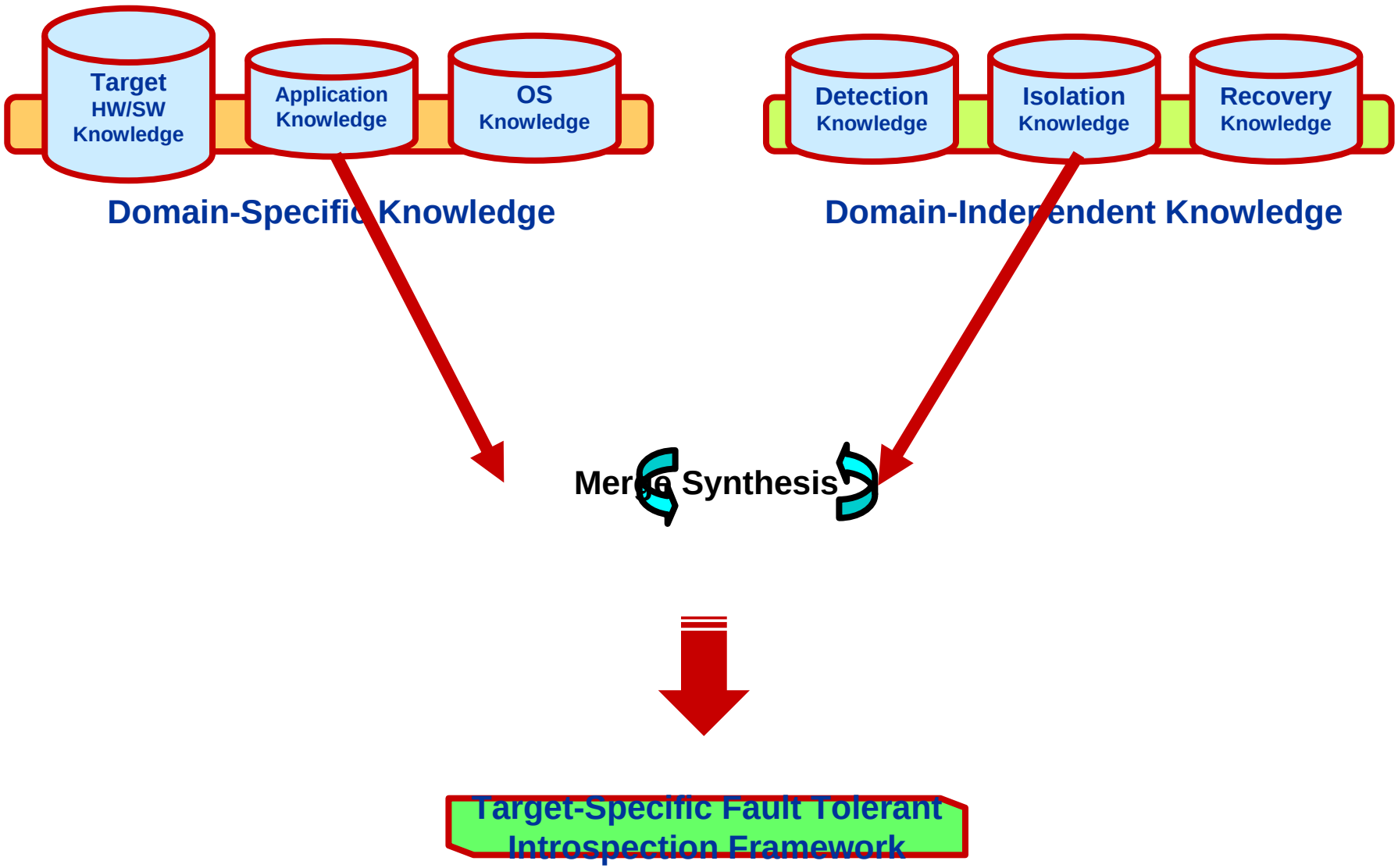
  Examples for sensor-provided inputs:
  - ***state of a variable, data structure, synchronization object***
  - ***value of an assertion***
  - ***state of a temperature sensor or hardware counter***
- **Actuators: provide** ***feedback*** **from the introspection system**

  Examples for actuator-triggered actions:
  - ***modification of program components (methods and data)***
  - ***modification of sensor/actuator sets (including activation and deactivation)***
  - ***local recovery***
  - ***signaling fault to next higher level in an introspection hierarchy***
  - ***requesting actions from lower levels in a hierarchical system***

The Spacecraft Health Inference Engine (SHINE)

- **A tool for building and deploying real-time rule-based reasoning systems for detection, diagnostics, prognostics, and recovery**
- **Outperforms commercial products by orders of magnitude**
  - ***Inference speed is achieved using graph transformations based on data flow analysis***
  - ***Rules are statically analyzed for all interactions***
- **The underlying structure is mapped into temporally invariant dataflow elements for execution on sequential or parallel hardware**
- **The final representation is either executed in a development environment or can be translated to a target language (C/C++)**
- **Deliveries**
  - ***NASA (Deep Space Network, applied to five NASA missions)***
  - ***Military (Lockheed JSF program, F-18 with 25+ flights)***
  - ***Aerospace (Northup, Lockheed, Boeing)***
  - ***Commercial (ViaChange, Vialogy, VIASPACE, Aerosciences, etc.)***

Knowledge Synthesis



Application-Oriented Introspection-Based Fault Tolerance in the HPCS: Research Issues

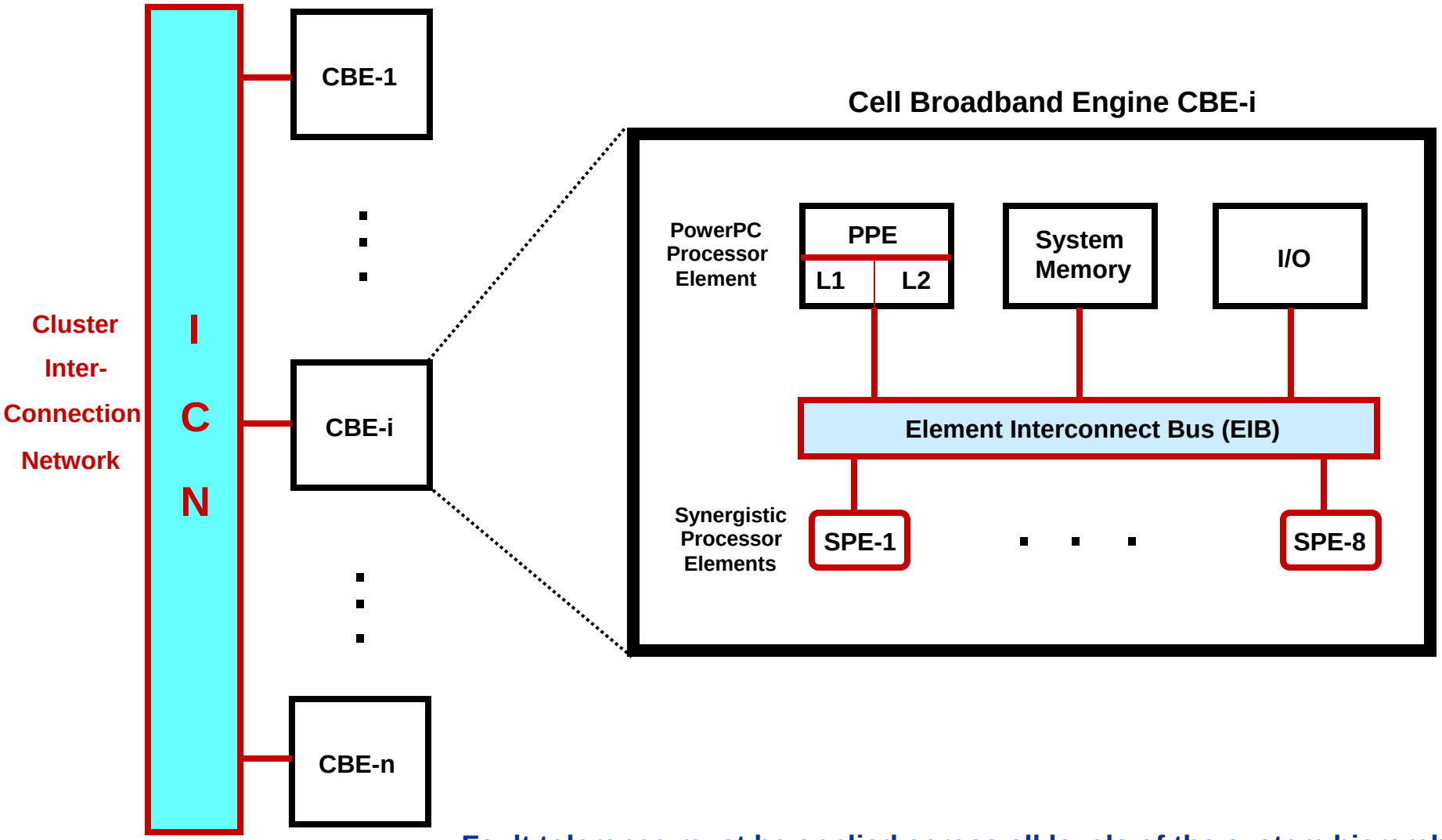
- **Current focus**
  - *transient and hard faults; fault detection*
  - *goal: reducing overhead of fixed-redundancy schemes*
- **Based on a (mission-dependent) fault model**
  - *classifies faults (fault types, severity)*
  - *specifies fault probabilities, depending on environment*
  - *prescribes recovery actions*
- **Exploiting knowledge from different sources**
  - *results of static analysis, dynamic analysis, profiling*
  - *target system hardware and software*
  - *application domain (libraries, data structures, data distributions)*
  - *user-provided assertions and invariants*
- **Leveraging existing technology**
  - *Algorithm-Based Fault Tolerance (ABFT)*
  - *naturally fault-tolerant algorithms*
  - *integration of high-level generator systems such as CMU's "SPIRAL"*
  - *fixed redundancy for small critical areas in a program*

Introspection Versus Traditional V&V

- **Verification and Validation (V&V)**
  - *focuses on design errors*
  - *applied before actual program execution*
  - *theoretical limits of verification: undecidability and NP-completeness*
  - *model checking: scalability challenge (exponential growth of state space)*
  - *tests can only identify faults, not prove their absence for all inputs*
  - *V&V cannot deal with transient errors or execution anomalies*

- **Introspection can complement traditional V&V technology**
  - *performs execution time monitoring, analysis, recovery*
  - *fault tolerance approach can be extended to address design errors*
  - *can deal with transient errors, execution anomalies, intrusion detection*
  - *can be integrated into a comprehensive V&V scheme*

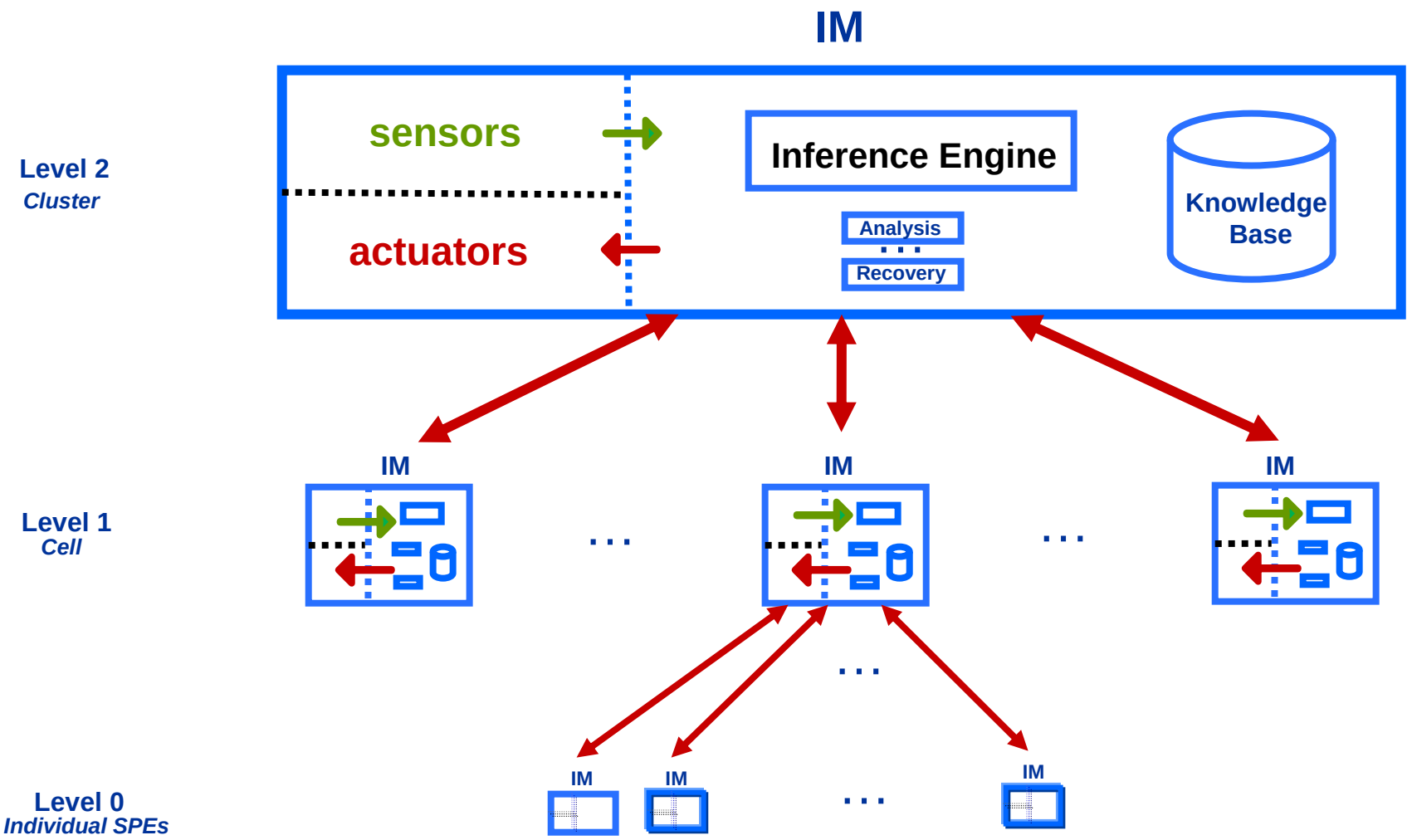
Implementation Target Architecture: Cluster of Cell Broadband Engines



Fault tolerance must be applied across all levels of the system hierarchy:

SPE → PPE → CBE → Cluster

Introspection Hierarchy for a Cluster of Cells



Concluding Remarks

- **Deep-space missions require space-borne high-capability computing for support of autonomy and on-board science**
- **Traditional approaches will not scale sufficiently**
- **Our approach:**
  - ***augment the radiation-hardened core of the on-board system with a commodity cluster of multi-core components***
  - ***develop an introspection framework for execution time monitoring, analysis, and recovery***
  - ***provide application-oriented adaptive fault tolerance for the HPCS***
- **Future Work**
  - ***completion of a prototype implementation for the Cell (and possibly ST8)***
  - ***application of the framework to mission codes (Synthetic Aperture Radar)***
  - ***integration of introspection into a coherent V&V approach***