Tiered Error Detection and Recovery

Z. Kalbarczyk
Center for Reliable and High-Performance Computing
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

http://www.crhc.uiuc.edu/DEPEND

Fault Models

- Transients are a major factor in upsetting the correct operation of digital circuits.
  - Single and multiple upsets – device scaling and decrease in power make electronic devices highly sensitive to transients induced by ionizing particles and current and voltage spikes. *(Single Event Upsets in Future Computing Systems, NASA-JPL Workshop, May 2003)*
  - Wide range of software and hardware errors
- Significant fraction of processor logic is not visible and unprotected against transients
  - Combinatorial/control logic becomes major contributor to system failures??

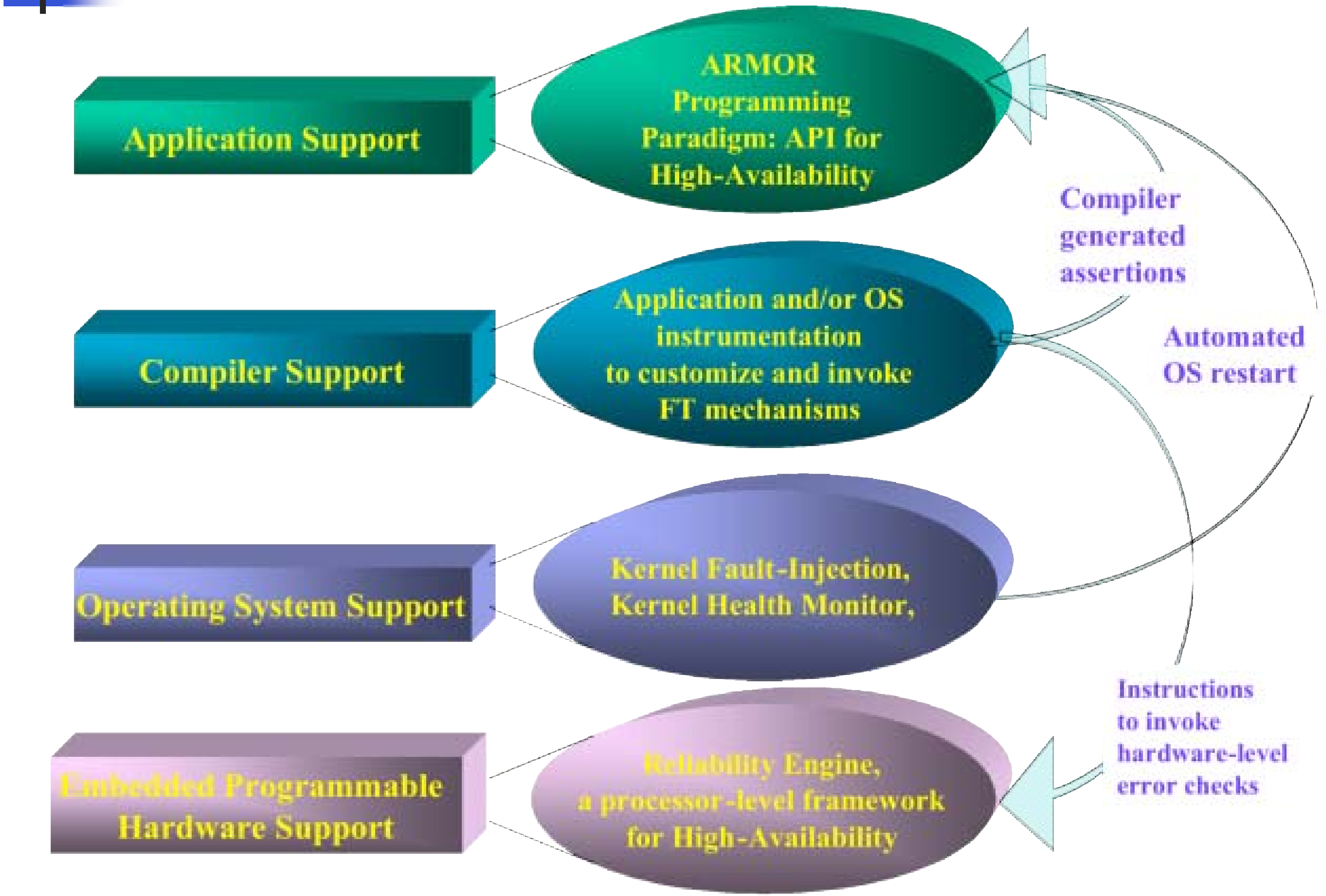
Error Propagation

- Errors do propagate!!!!
- While relatively small percentage of errors propagate from the lower levels to the system-level, the system-wide impact can be catastrophic
  - hangs or crashes, prohibitively long recovery times.
- Studies on networks and operating systems
  - 11% of faults at the electric level propagated to the system level, e.g., LAN of computers connected via Myrinet switch.
  - less than 10% of errors propagate between the OS subsystems, or OS and applications, e.g. Unix, Linux, LynxOS.
  - 18% of software design errors cause error propagation, e.g., Tandem Guardian operating system

What Do We Need to Meet Technology Challenges?

- Tiered system of detection and recovery schemes and mechanisms
  - embedded into the hardware (e.g., processor or dedicated FPGA-based modules)
  - integrated with the operating system or application (e.g., a robust middleware).
- Traditional coding schemes (e.g., ECC) and spatial and temporal redundancy need to be re-evaluated in the *power-performance-reliability* space
- Standard or well-defined procedures (e.g., operational conditions) for assessing soft error rates
- Apply research to realistic benchmarks

Four-Tiered Approach to High Dependability



Application Support

ARMOR Programming Paradigm: API for High-Availability

Compiler Support

Application and/or OS instrumentation to customize and invoke FT mechanisms

Operating System Support

Kernel Fault-Injection, Kernel Health Monitor,

Embedded Programmable Hardware Support

Reliability Engine, a processor-level framework for High-Availability

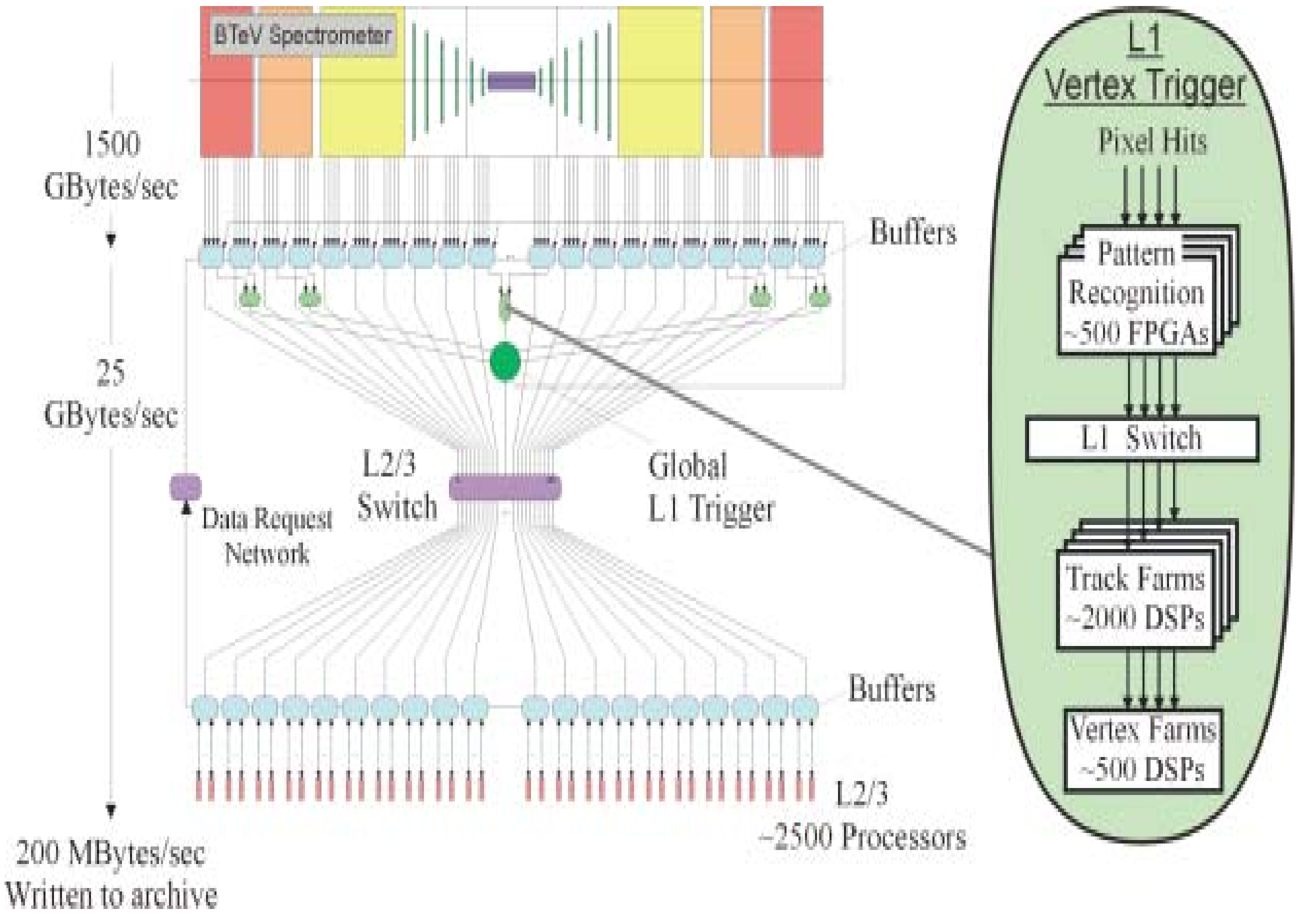
Compiler generated assertions

Automated OS restart

Instructions to invoke hardware-level error checks

Large Scale Real Time System: Trigger and Data Acquisition System

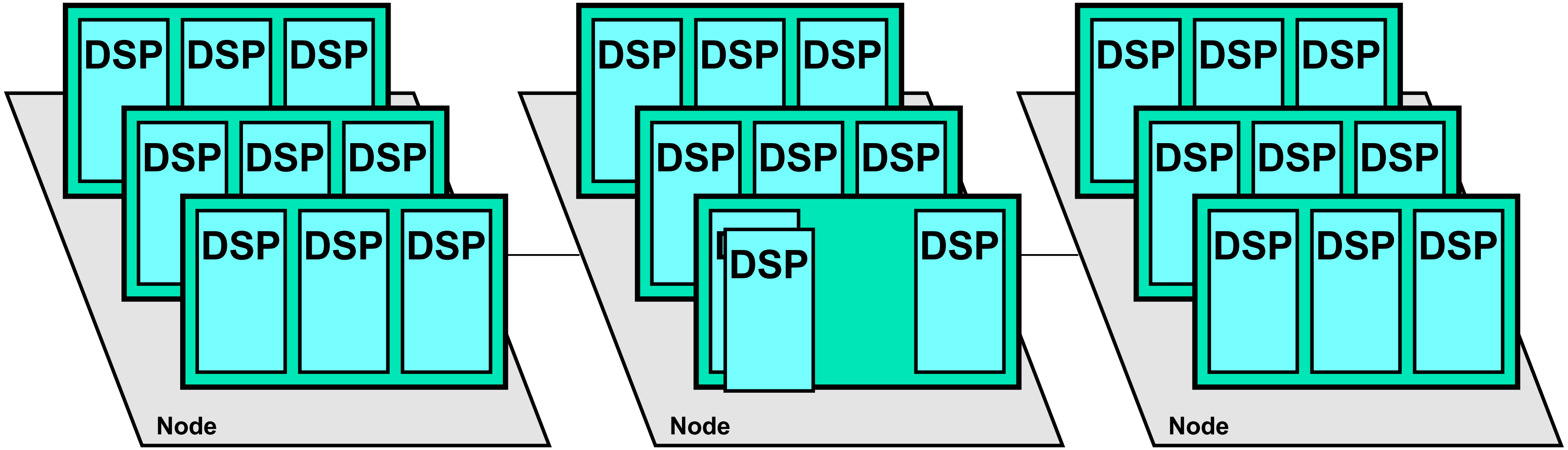
- High availability:
  - 24x7 uptime to perform physics experiments.
- Data integrity:
  - Avoid loss of physics data at all cost.
- Self-diagnosis:
  - Rapid error detection.
  - Automatic response to an error (recovery may be slower)
- Adaptivity:
  - Key operating parameters can change during experiment
  - Error detection and recovery policies evolve throughout experiment



Joint NSF-ITR project with FermiLab

RTES Computational Platforms

- Level 1 computation on DSP farm: embedded hardware support



- Level 2/3 computation on Linux cluster: robust OS and software detection and recovery (e.g., hierarchy of ARMOR processes)