UT-BATTELLE

A Proactive Fault Tolerance Framework for High-Performance Computing

Antonina Litvinova[1,2], Christian Engelmann[2], and Stephen L. Scott[2]

[1] Department of Computer Science
[2] The University of Reading, Reading, UK

[2] Computer Science and Mathematics Division
[2] Oak Ridge National Laboratory, Oak Ridge, USA

9th IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN), Innsbruck, Austria, Feb. 16-18, 2010

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

Extreme-Scale High-Performance Computing Systems for Computational Science

top500.org processor count:

- About three years ago the entire 500 list broke the million processor mark
- Now the top 7 add up to over a million

#1

#3

OAK RIDGE National Laboratory

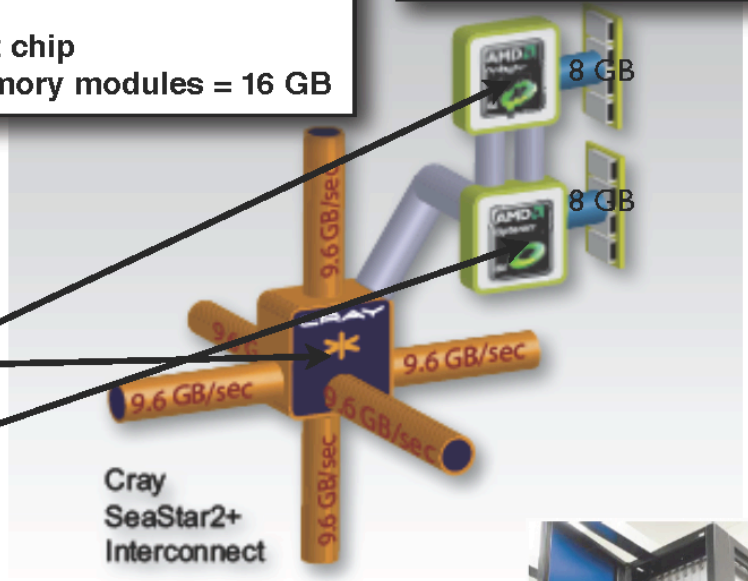
#1: Jaguar at Oak Ridge National Laboratory

Blade = 4 Nodes
8 processors
48 cores
4 interconnect chips
16 (4 GB) memory modules = 64 GB
6 voltage converters

Node = 2 Processors
6 cores per processor

1 interconnect chip
4 x (4 GB) memory modules = 16 GB

Processor = 6 Cores
2 memory modules



Cabinet = 24 Blades
1152 cores
96 interconnect chips
384 memory modules (1.5 TB)
144 voltage converters
+ power supply, liquid cooling, etc.
Power 480V, ~40,000 Watt per cabinet

Jaguar = 284 cabinets (XT5 and XT4), ~ 6.5 Megawatts

Cray XT5

Motivation

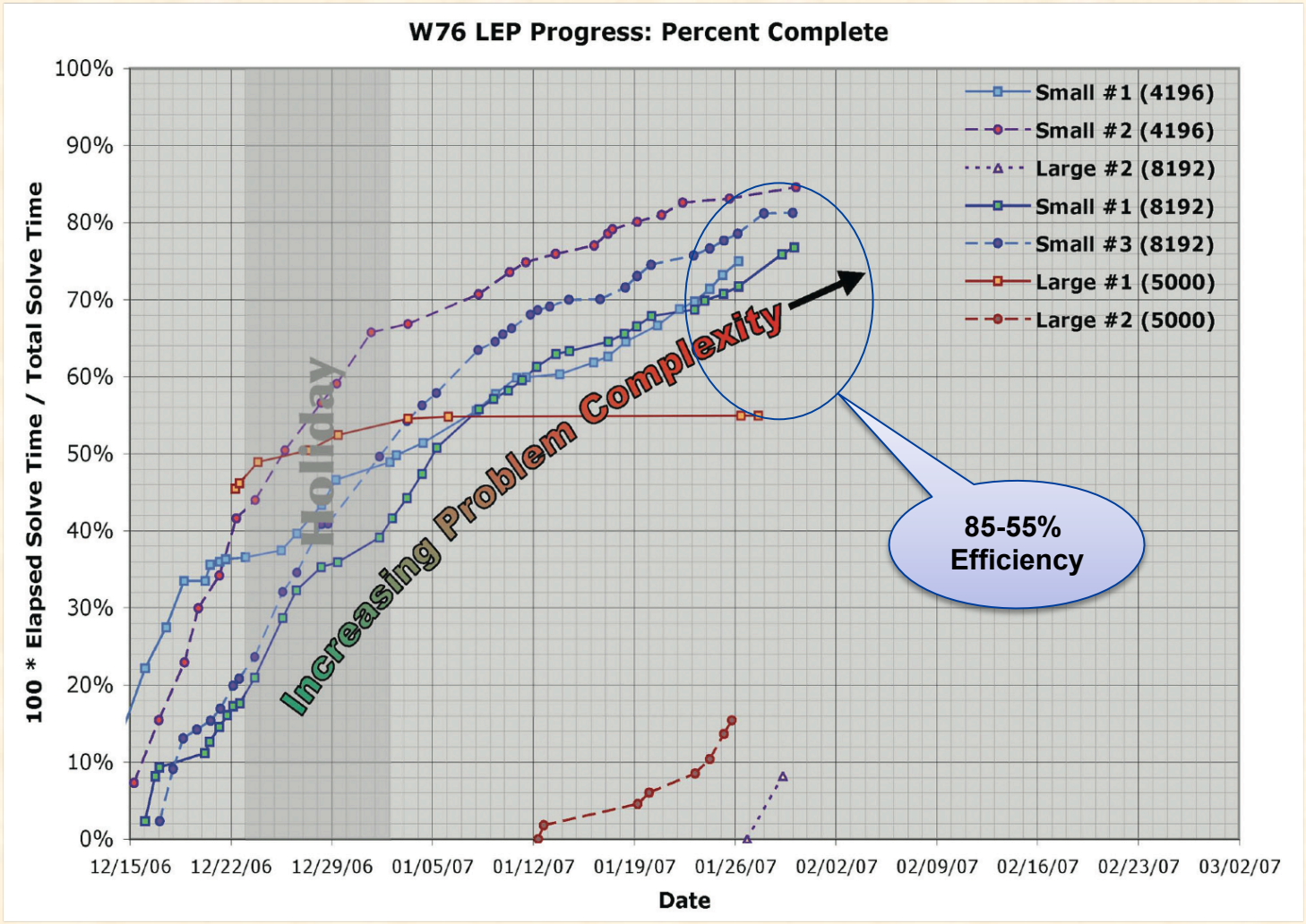
- Large-scale PFlop/s systems have arrived
  - #1 ORNL Jaguar XT5: 1.759 PFlop/s LINPACK, 224,162 cores
  - #2 LANL Roadrunner: 1.042 PFlop/s LINPACK, 122,400 cores
- Other large-scale systems exist
  - #3 NICS Kraken XT5: 0.831 PFlop/s LINPACK, 98,928 cores
  - #4 Juelich JUGENE: 0.825 PFlop/s LINPACK, 294,912 cores
- The trend is toward larger-scale systems
  - Exascale (1,000 PFlop/s) system with 100M-1B cores by 2018
- Significant increase in component count and complexity
- Expected matching increase in failure frequency
- Checkpoint/restart is becoming less and less efficient

The Road to Exa-Scale: Challenges Ahead

Systems	2009	2018	Difference Today & 2018
System peak	2 Pflop/s	1 Eflop/s	O(1000)
Power	6 MW	~20 MW	
System memory	0.3 PB	32 - 64 PB **[.03 Bytes/Flop]**	O(100)
Node performance	125 GF	1,2 or 15TF	O(10) - O(100)
Node memory BW	25 GB/s	2 - 4TB/s **[.002 Bytes/Flop]**	O(100)
Node concurrency	12	O(1k) or 10k	O(100) - O(1000)
Total Node Interconnect BW	3.5 GB/s	200-400GB/s (1:4 or 1:8 from memory BW)	O(100)
System size (nodes)	18,700	O(100,000) or O(1M)	O(10) - O(100)
Total concurrency	225,000	O(billion) [O(10) to O(100) for latency hiding]	O(10,000)
Storage	15 PB	500-1000 PB (>10x system memory is min)	O(10) - O(100)
IO	0.2 TB	60 TB/s (how long to drain the machine)	O(100)
MTTI	days	O(1 day)	- O(10)

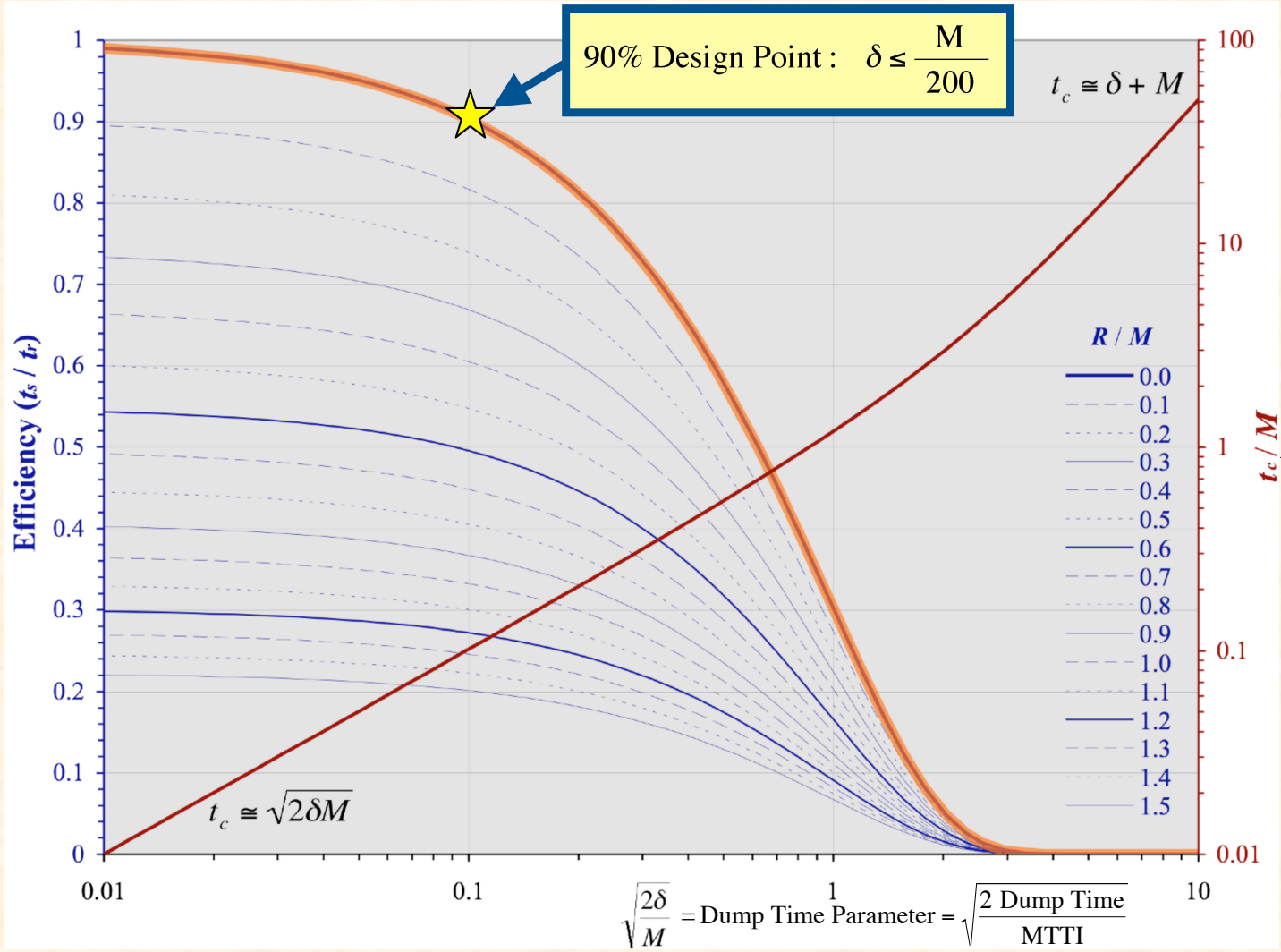
Parallel File System Checkpoint/Restart Efficiency Study (2006 @ LANL)

J. T. Daly. ADTSC Nuclear Weapons Highlights: Facilitating High-Throughput ASC Calculations. Technical Report LALP-07-041, Los Alamos National Laboratory, June 2007.



Parallel File System Checkpoint/Restart Efficiency Model

J. T. Daly. Methodology and metrics for quantifying application throughput. In Proceedings of the Nuclear Explosives Code Developers Conference (NECDC) 2006, Los Alamos, NM, USA, Oct. 23-27, 2006.



Reactive vs. Proactive Fault Tolerance

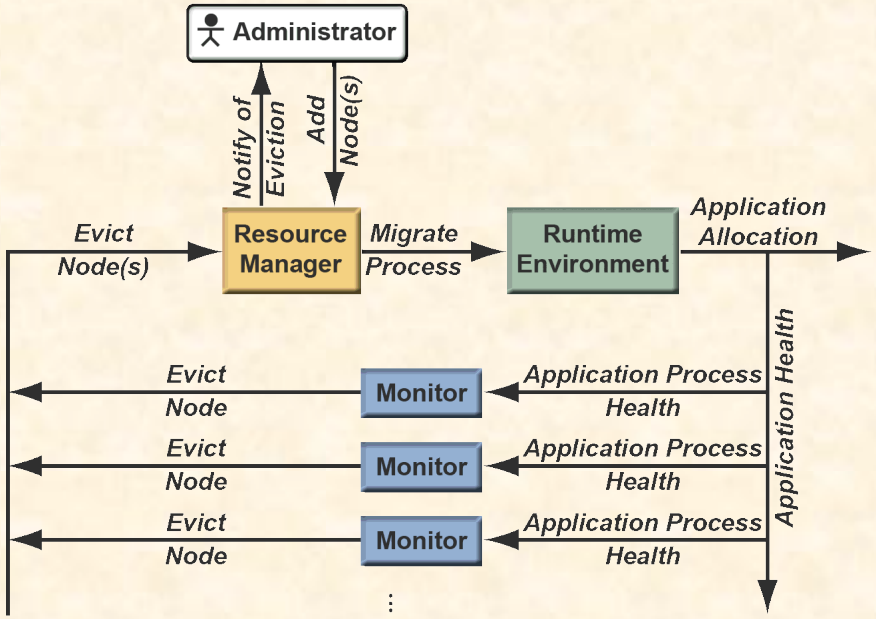
- Reactive fault tolerance
  - Keeps parallel applications alive through recovery from experienced failures
  - Employed mechanisms react to failures
  - Examples: Checkpoint/restart, message logging/replay
- Proactive fault tolerance
  - Keeps parallel applications alive by avoiding failures through preventative measures
  - Employed mechanisms anticipate failures
  - Example: Preemptive migration

Proactive Fault Tolerance using Preemptive Migration

- Relies on a feedback-loop control mechanism
  - Application health is constantly monitored and analyzed
  - Application is reallocated to improve its health and avoid failures
  - Closed-loop control similar to dynamic load balancing
- Real-time control problem
  - Need to act in time to avoid imminent failures
- No 100% coverage
  - Not all failures can be anticipated, such as random bit flips

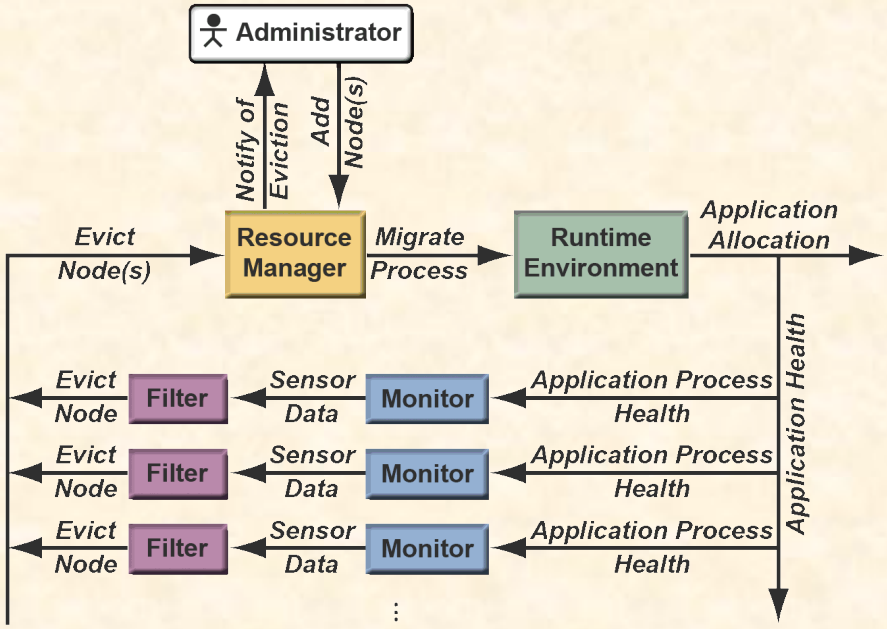
Type 1 Feedback-Loop Control Architecture

- Alert-driven coverage
  - Basic failures
- No evaluation of application health history or context
  - Prone to false positives
  - Prone to false negatives
  - Prone to miss real-time window
  - Prone to decrease application heath through migration
  - No correlation of health context or history



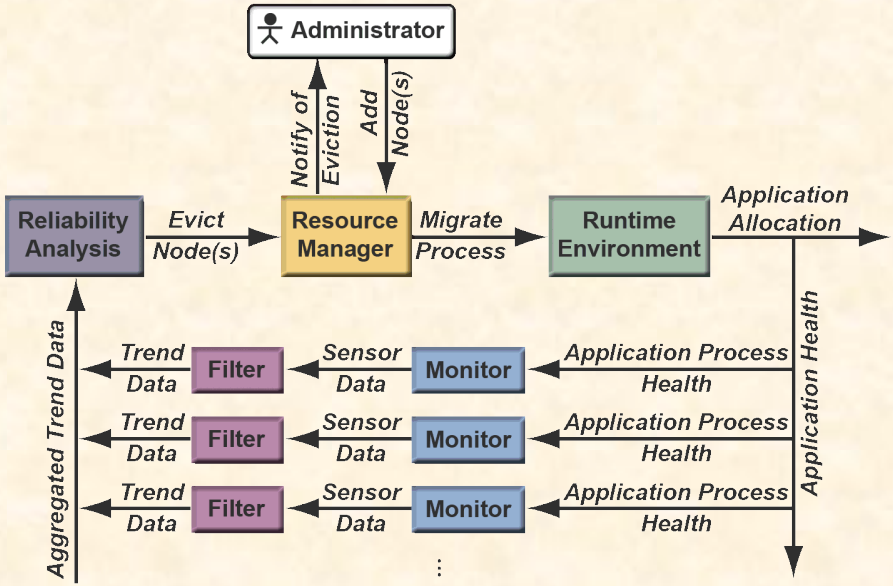
Type 2 Feedback-Loop Control Architecture

- Trend-driven coverage
  - Basic failures
  - Less false positives/negatives
- No evaluation of application reliability
  - Prone to miss real-time window
  - Prone to decrease application heath through migration
  - No correlation of health context or history



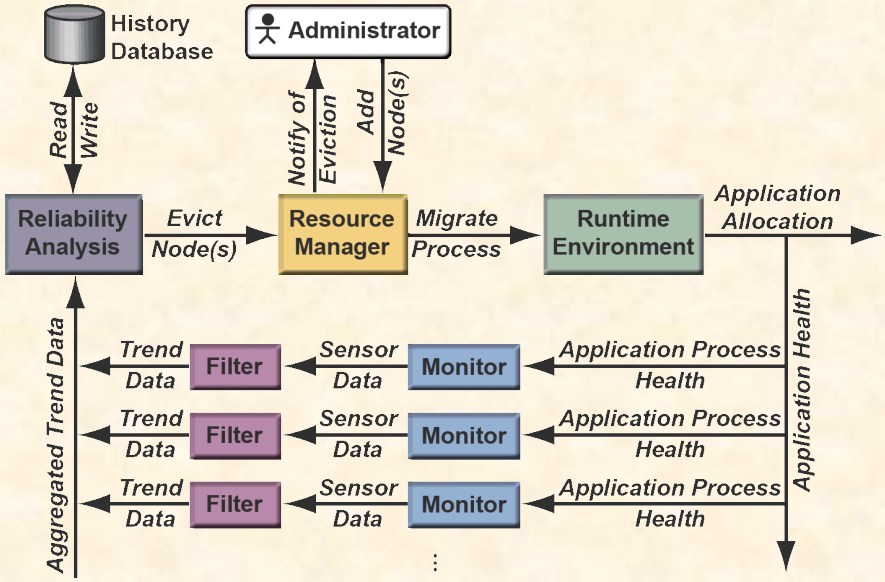
Type 3 Feedback-Loop Control Architecture

- Reliability-driven coverage
  - Basic and correlated failures
  - Less false positives/negatives
  - Able to maintain real-time window
  - Does not decrease application heath through migration
  - Correlation of short-term health context and history
- No correlation of long-term health context or history
  - Unable to match system and application reliability patterns



Type 4 Feedback-Loop Control Architecture

- Reliability-driven coverage of failures and anomalies
  - Basic and correlated failures, anomaly detection
  - Less prone to false positives
  - Less prone to false negatives
  - Able to maintain real-time window
  - Does not decrease application heath through migration
  - Correlation of short and long-term health context & history



Existing Work

- Environmental monitoring
  - OpenIPMI, Ganglia, OVIS 2
  - HPC vendor RAS systems
- Event logging and analysis
  - USENIX Computer Failure Data Repository
  - System log analysis efforts
- Job and resource monitoring
  - Torque, Moab, SGE, …
- Migration mechanisms
  - Process-level using BLCR
  - VM-level using Xen
- Proactive FT Frameworks
  - Type 1 based on Xen & Ganglia
  - Type 1 based on BLCR & Ganglia
  - Type 1 to investigate interfaces, coordination and protocols
- Fault tolerance policies
  - Simulation framework to evaluate trade-off for combining migration with checkpoint/restart

Holistic Fault Tolerance Framework

Virtualization Scope:
Application, Run Time Environment, OS and/or Micro OS

Customization and Guidance

Detection

Recovery and Prevention

Status Probes

Local Policy-Based Analysis

Fault Tolerance Mechanisms

Individual Compute Nodes

Communication System

Policy Configuration, Decision Guidance

Fault, Error and Trend Notification

Fault Tolerance Mechanism Invocation

Coordinated Global Policy-Based Analysis and Decision Making

Event Distribution

Users, Administrator, System Services

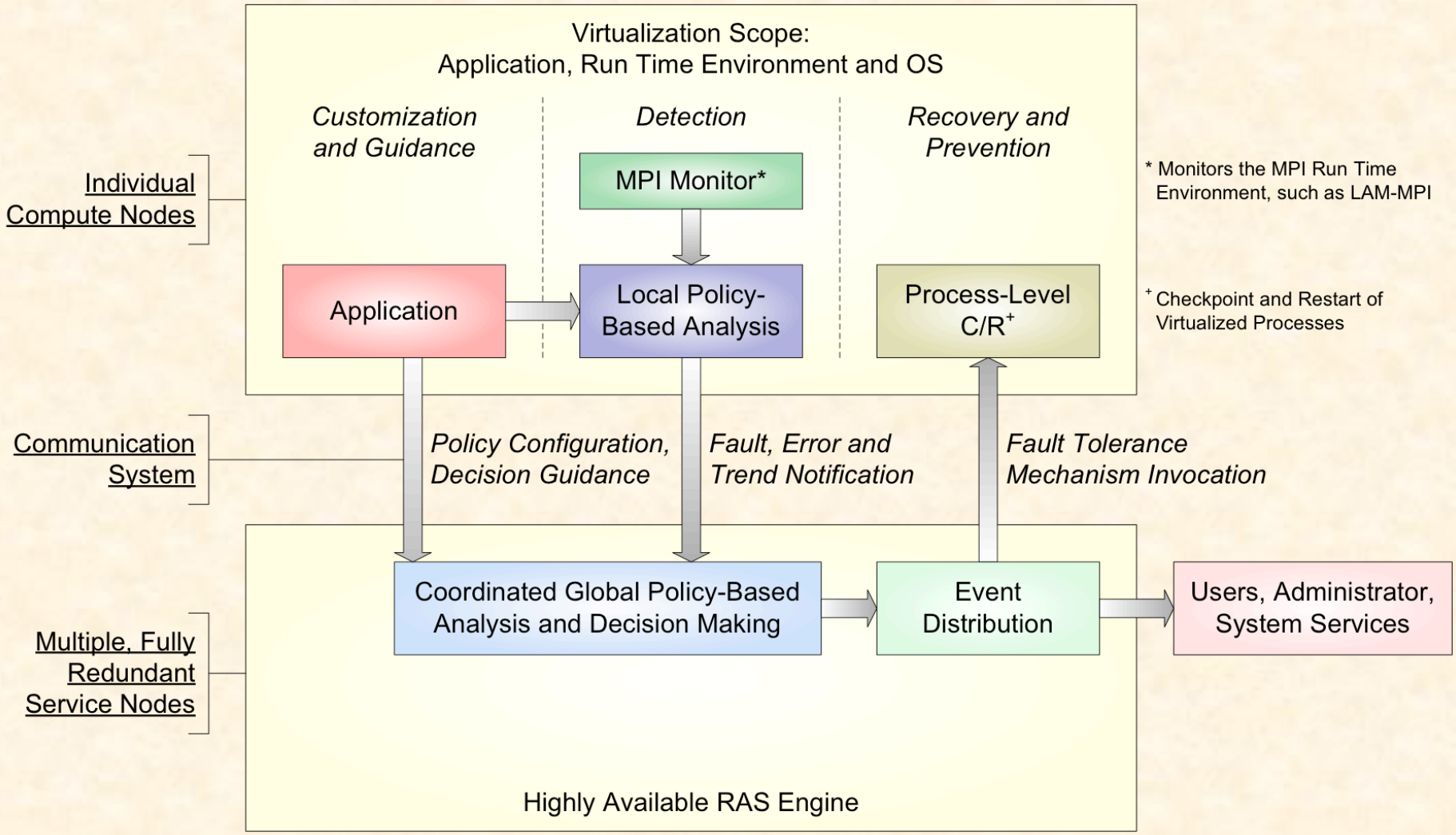
Multiple, Fully Redundant Service Nodes

Local Policy-Based Analysis

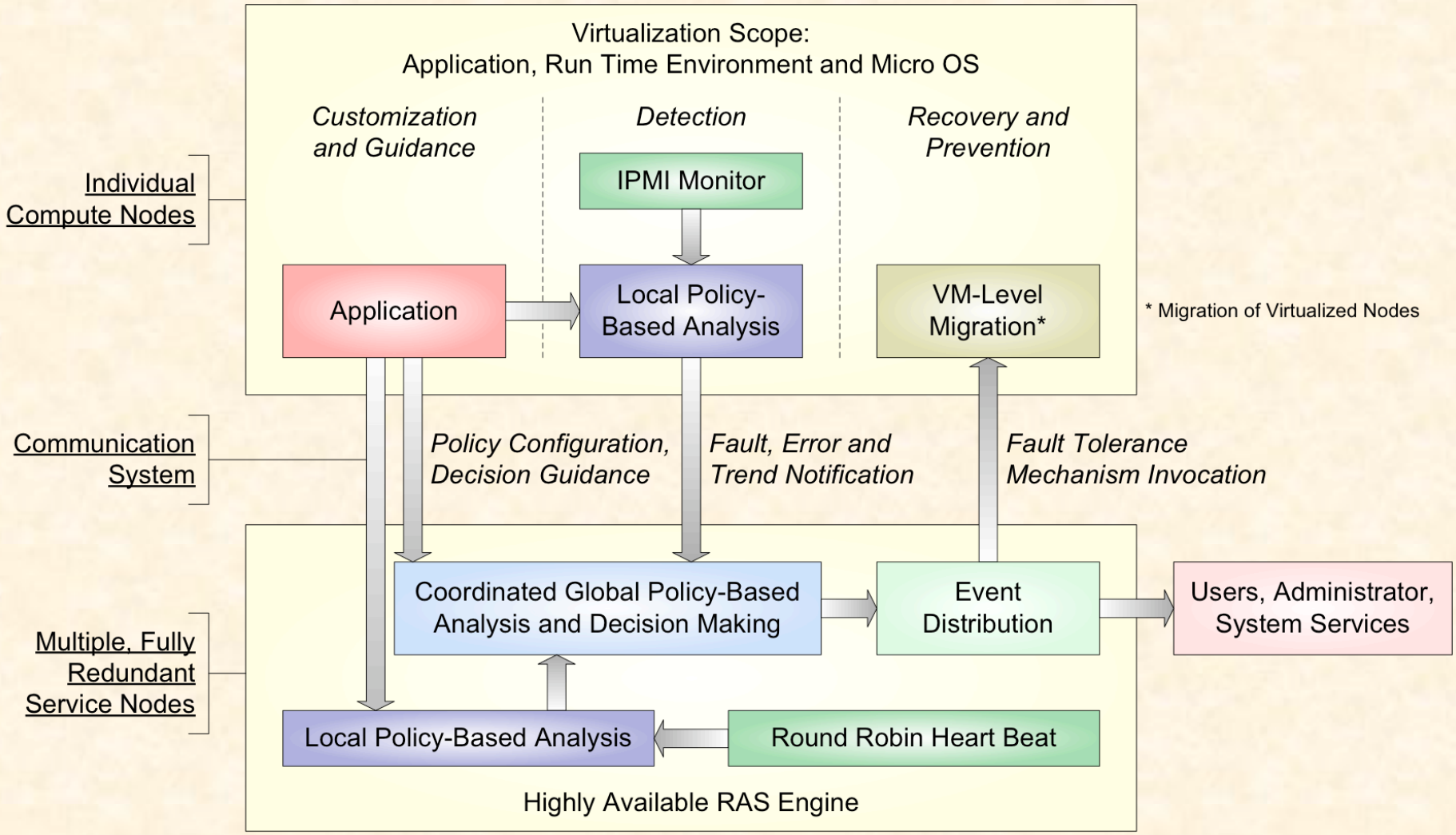
Remote Node Status Probes

Highly Available RAS Engine

Holistic Fault Tolerance Framework: Reactive Fault Tolerance

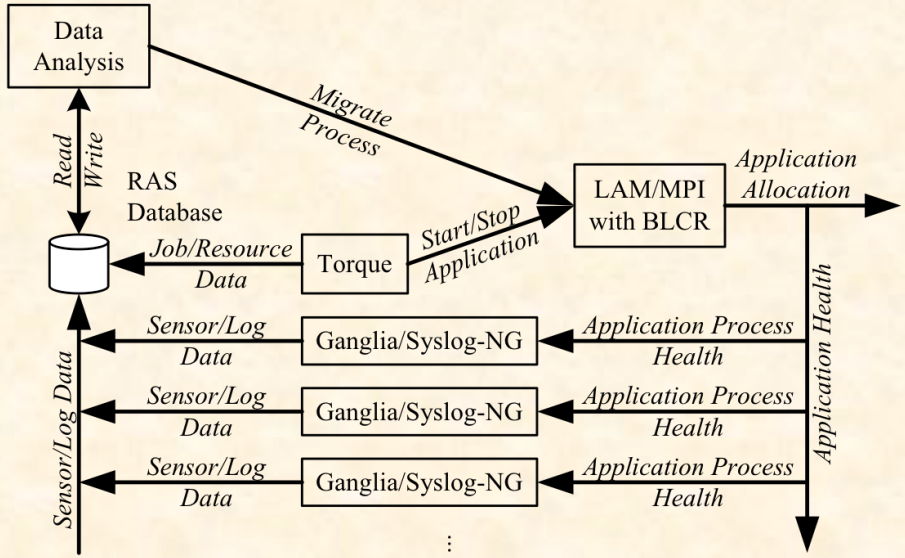


Holistic Fault Tolerance Framework: Proactive Fault Tolerance



Framework Implementation

- Focus on proactive FT approach
- Central MySQL database for data logging and analysis
- Environmental monitoring
  - OpenIPMI and Ganglia
- Event logging and analysis
  - Syslog (node-local logging and forwarding to central server)
- Job and resource monitoring
  - Torque (epilogue/prologue)
- Migration mechanisms
  - Process-level using BLCR



Results

- Deployed on XTORC @ ORNL
  - 64-node Intel-based Linux cluster
- MySQL, Gangila, Torque, Syslog, LAM-MPI+BLCR with migration
- Experiment #1:
  - Fully deployed on 64 nodes
  - 30 second data collection interval
  - Collection of 20 metrics resulted in over 20GB of data in 27 days (~33MB/hour or ~275kb/interval)
  - Basic temperature threshold triggers for migration resulted in migration when covering up air intake holes
- Experiment #2:
  - Fully deployed on 32 nodes
  - Collection of 40 metrics
  - 30 second data collection interval
  - No measurable impact on NAS benchmarks (see Figure below)

Class C NPB on 32 nodes	CG	FT	LU
Average time in seconds	264	235	261
Average time under load in seconds	264	236	260

Table 2. NPB test results (averages over 10 runs)

Conclusions

- Developed a proactive FT framework that performs
  - Environmental monitoring
  - Event logging
  - Parallel job monitoring
  - Resource monitoring
  - Online and offline HPC system reliability analysis
- It permits fault avoidance through process migration
- Deployed on a 64-node system to gain hands-on experience and to investigate the challenges ahead
  - The biggest challenge is the amount of stored data
  - Optimal pre-processing, scalable data aggregation and combined (all sources, in-flight) data analysis is needed

UT-BATTELLE

Questions?