A Tunable Holistic Resiliency Approach for High-Performance Computing Systems

Stephen L. Scott, Christian Engelmann, Geoffroy Vallée, Thomas Naughton, Anand Tikotekar, George Ostrouchov (Oak Ridge National Laboratory)
Chokchai Leangsuksun, Nichamon Naksinehaboon, Raja Nassar, Mihaela Paun (Louisiana Tech University)
Frank Mueller, Chao Wang, Arun Nagarajan, Jyothish Varma (North Carolina State University)

Motivation

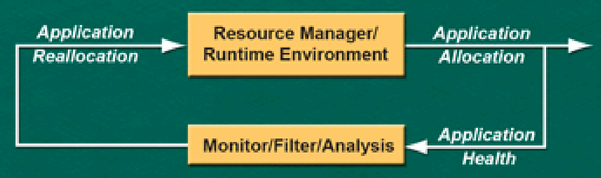
- The 1PFlop/s (10^{15} Floating Point Operations Per Second) barrier has been broken
  - #1: LANL Roadrunner with 129,600 processor cores
  - #2: ORNL Jaguar with 150,152 processor cores
- Other large-scale systems exist
  - LLNL @ 212,992, ANL @ 163,840, TACC @ 62,976
- The trend is toward even larger-scale systems
- The significant increase in component count and complexity leads to an increase in failure frequency
- Checkpoint/restart is becoming less and less efficient

Reactive vs. Proactive Fault Tolerance

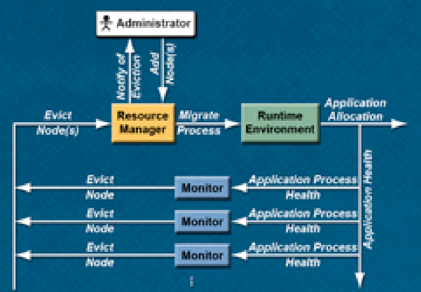
- Reactive fault tolerance
  - Keeps parallel applications alive through recovery from experienced failures
  - Employed mechanisms react to failures
  - Examples: Checkpoint/restart, message logging/replay
- Proactive fault tolerance
  - Keeps parallel applications alive by avoiding experiencing failures 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 double-bit ECC errors

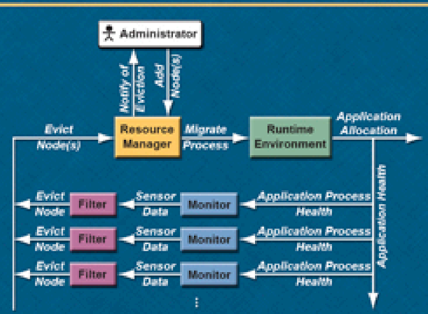


Feedback-Loop Control Architecture



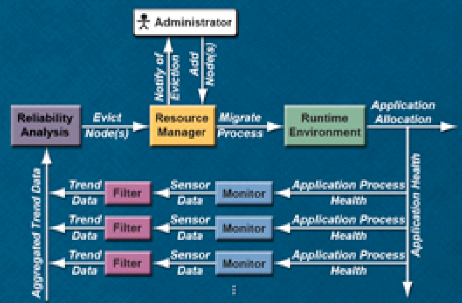
Type 1

- Alert-driven coverage for basic failures
  - Fan fault, overheating and other precursors to hard errors
- 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 health through migration
  - No correlation of health context (space) or history (time)



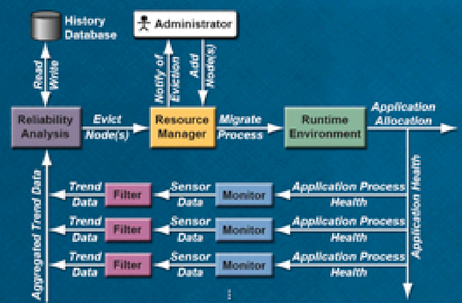
Type 2

- Trend-driven coverage for basic failures
  - Fan fault, overheating and other precursors to hard errors
  - Less prone to false positives
  - Less prone to false negatives
- No evaluation of application reliability
  - Prone to miss real-time window
  - Prone to decrease application health through migration
  - No correlation of health context (space) or history (time)



Type 3

- Reliability-driven coverage of failures
  - Basic and correlated failures
  - Even less prone to false positives
  - Even less prone to false negatives
  - Able to maintain real-time window
  - Not prone to 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



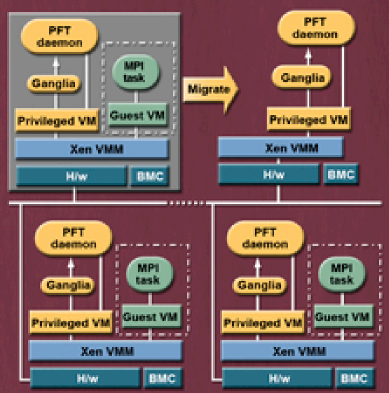
Type 4

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

Prototype 1

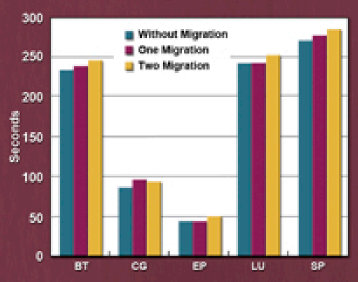
VM-level Preemptive Migration using Xen

- Type 1 system setup
  - Xen VMM on entire system
  - Host OS for management
  - Guest OS for computation
  - Spare nodes without Guest OS
  - System monitoring in Host OS
  - Decentralized scheduler/load balancer using Ganglia
- Deteriorating node health
  - Ganglia threshold trigger
  - Migrate guest OS to spare
  - Utilize Xen's migration facility



VM-level Migration Performance Impact

- Single node migration
  - 0.5-5% longer run time
- Double node migration
  - 2-8% longer run time
- Migration duration
  - Stop & copy : 13-14s
  - Live : 14-24s
- Application downtime
  - Stop & copy > Live

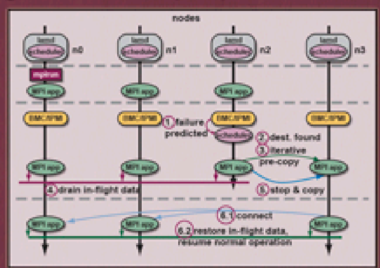


16-node Linux cluster at NCSU with dual core, dual-processor AMD Opteron and Gigabit Ethernet

Prototype 2

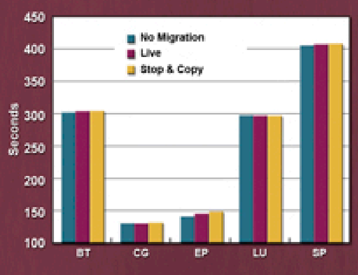
Process-Level Preemptive Migration using BLCR

- Type 1 system setup
  - LAM/MPI with Berkeley Lab Checkpoint/Restart (BLCR)
  - Per-node health monitoring
    - Baseboard management controller (BMC)
    - Intelligent platform management interface (IPMI)
  - New decentralized scheduler/ load balancer in LAM
  - New process migration facility in BLCR (stop&copy and live)
- Deteriorating node health
  - Simple threshold trigger
  - Migrate process to spare



Process-Level Migration Performance Impact

- Single node migration overhead
  - Stop & copy : 0.09-6 %
  - Live : 0.08-2.98%
- Single node migration duration
  - Stop & copy : 1.0-1.9s
  - Live : 2.6-6.5s
- Application downtime
  - Stop & copy > Live
- Node eviction time
  - Stop & copy > Live

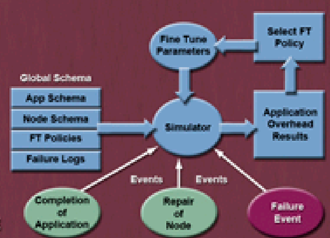


16-node Linux cluster at NCSU with dual core, dual-processor AMD Opteron and Gigabit Ethernet

Prototype 3

Simulation Framework for Fault Tolerance Policies Tolerance

- Evaluation of fault tolerance policies
  - Reactive only
  - Proactive only
  - Reactive/proactive combination
- Evaluation of fault tolerance parameters
  - Checkpoint interval
  - Prediction accuracy
- Event-based simulation framework using actual HPC system logs
- Customizable simulated environment
  - Number of active and spare nodes
  - Checkpoint and migration overheads

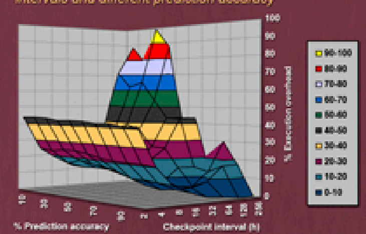


Combining Proactive and Reactive Fault Tolerance

- Best: Prediction accuracy >60% and checkpoint interval 16-32h
- Better than only proactive or only reactive
- Results for higher accuracies and very low intervals are worse than only proactive or only reactive

Number of processes	125
Active/Spare nodes	125/12
Checkpoint overhead	50min
Migration overhead	1 min

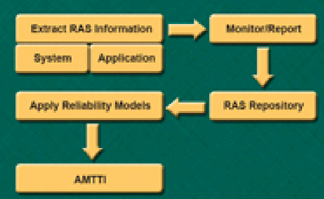
Execution overhead for various checkpoint intervals and different prediction accuracy



Simulation based on ASCI White system logs (nodes 1-125 and 500-512)

Research in Reliability Modeling

- Type 3 system setup
  - Monitoring of application and system health
  - Recording of application and system health monitoring data
  - Reliability analysis on recorded data
  - Application mean-time to interrupt (AMTTI) estimation



- Type 4 system setup
  - Additional recording of application interrupts
  - Reliability analysis on recent and historical data