UT-BATTELLE

Reliability, Availability, and Serviceability (RAS) for Petascale High-End Computing and Beyond

Christian Engelmann and Stephen L. Scott

Computer Science and Mathematics Division
Oak Ridge National Laboratory

www.fastos.org/ras

Office of Science
U.S. DEPARTMENT OF ENERGY

OAK RIDGE National Laboratory

NC STATE UNIVERSITY

LOUISIANA TECH UNIVERSITY®

Motivation

- Large-scale PFlop/s systems have arrived
  - #1 ORNL Jaguar XT5: 1.759 PFlop/s LINPACK, 224,162 cores
  - #2 NSCS Nebulae: 1.271 PFlop/s LINPACK, 120,640 cores
  - #3 LANL Roadrunner: 1.042 PFlop/s LINPACK, 122,400 cores
- Other large-scale systems exist
  - #4 NICS Kraken XT5: 0.831 PFlop/s LINPACK, 98,928 cores
  - #5 Juelich JUGENE: 0.825 PFlop/s LINPACK, 294,912 cores
  - #6 NASA Pleiades: 0.773 PFlop/s LINPACK, 81,920 cores
- The trend is toward even larger-scale systems

Toward Exascale Computing (My Roadmap)

Based on proposed DOE roadmap with MTTI adjusted to scale linearly

Systems	2009	2011	2015	2018
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System memory	0.3 PB	1.6 PB	5 PB	10 PB
Node performance	125 GF	200GF	200-400 GF	1-10TF
Node memory BW	25 GB/s	40 GB/s	100 GB/s	200-400 GB/s
Node concurrency	12	32	O(100)	O(1000)
Interconnect BW	1.5 GB/s	22 GB/s	25 GB/s	50 GB/s
System size (nodes)	18,700	100,000	500,000	O(million)
Total concurrency	225,000	3,200,000	O(50,000,000)	O(billion)
Storage	15 PB	30 PB	150 PB	300 PB
IO	0.2 TB/s	2 TB/s	10 TB/s	20 TB/s
MTTI	4 days	19 h 4 min	3 h 52 min	1 h 56 min
Power	6 MW	~10MW	~10 MW	~20 MW

Factors Driving up the Error Rate

- Significant growth in component count (up to 50x nodes) results in respectively higher system error rate
- Smaller circuit sizes and lower voltages increase soft error vulnerability (bit flips caused by thermal and voltage variations as well as radiation)
- Power management cycling decreases component lifetimes due to thermal and mechanical stresses
- Hardware fault detection and recovery is limited by power consumption requirements and costs
- Heterogeneous architectures (CPU & GPU cores) add more complexity to fault detection and recovery

Risks of the Business as Usual Approach

- Increased error rate requires more frequent checkpoint/ restart, thus lowering efficiency (application progress)
- Current application-level checkpoint/restart to a parallel file system is becoming less efficient and soon obsolete
- Memory to I/O ratio (dump time) improves from 25 min to 8.3 min, but concurrency for coordination and I/O scheduling increases significantly (50x nodes, 444x cores)
- Missing strategy for silent data/code corruption will cause applications to produce erroneous results or just hang

Objectives

- Develop scalable system software technologies to achieve high-level RAS for next-generation petascale scientific high-end computing resources
- Provide for non-stop scientific computing on a 24x7 basis without interruption with virtualized adaptation, reconfiguration, and preemptive measures
- Address the RAS research challenges outlined by DOE's Forum to Address Scalable Technology for Runtime and Operating Systems (FAST-OS)

Approach

- Leverage virtualization for transparent fault tolerance on extreme scale computing systems
- Perform reliability analysis to enable failure prediction
- Investigate proactive fault tolerance using migration away from components that are “about to fail”
- Develop reactive fault tolerance enhancements, such as checkpoint interval and placement adaption
- Offer holistic fault tolerance through combination of adaptive proactive and reactive fault tolerance

Reactive vs. Proactive Fault Tolerance

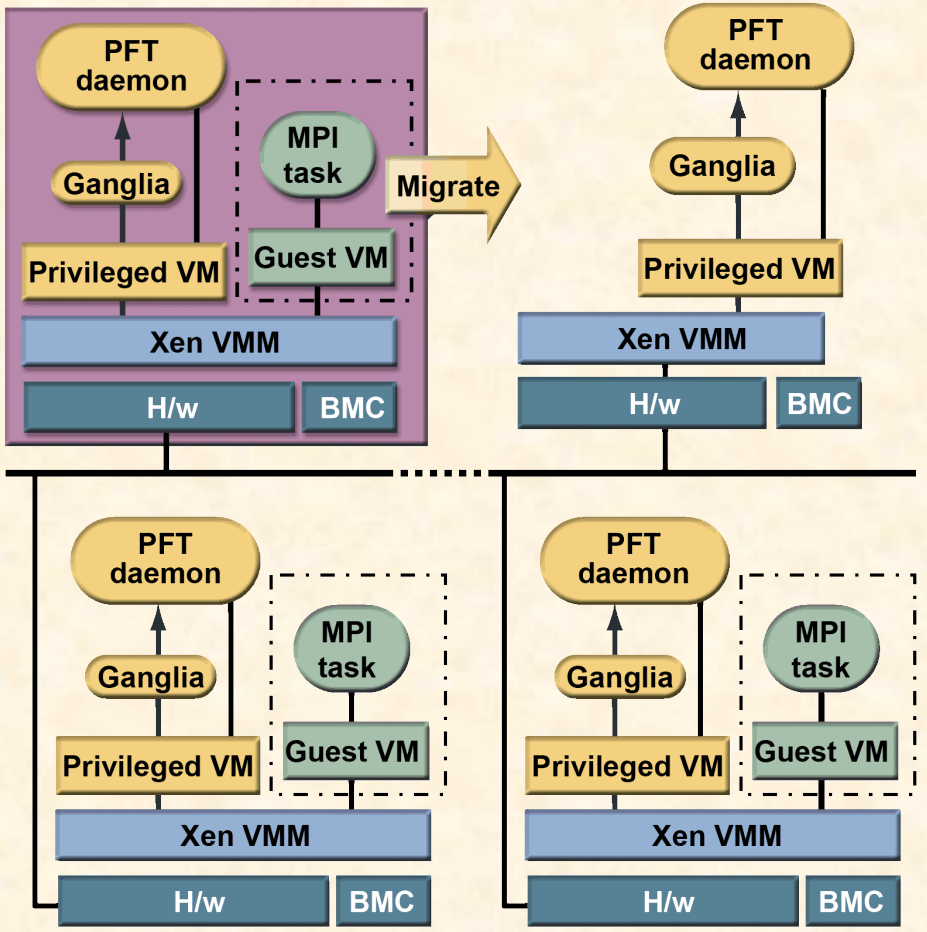
- **Reactive fault tolerance**
  - Keeps parallel applications alive through recovery from experienced failures
  - Employed mechanisms react to failures
  - Examples: Checkpoint/restart and message logging/replay
- **Proactive fault tolerance**
  - Keeps parallel applications alive by avoiding failures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

Proactive Fault Tolerance using Migration

- Relies on a feedback-loop control mechanism
  - Application health is constantly monitored and analyzed
  - Application is reallocated to improve to 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

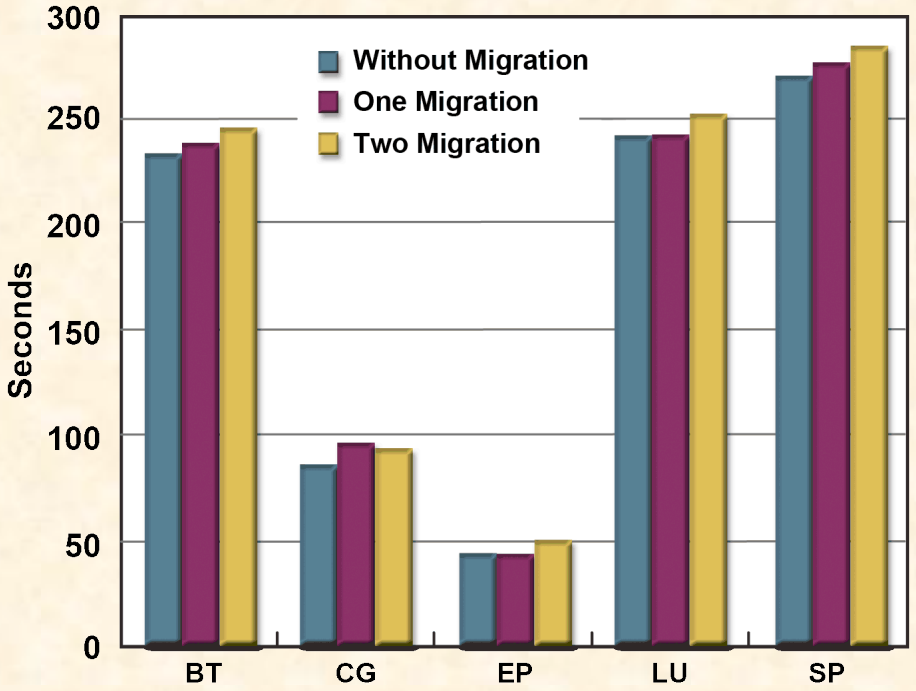
VM-level Migration with Xen

- Type 1 system setup
  - Xen VMM on entire system
  - Host OS for management
  - Guest OS for computation
  - Spares without Guest OS
  - Monitoring in Host OS
  - Decentralized scheduler/ load balancer w/ Ganglia
- Deteriorating node health
  - Ganglia threshold trigger
  - Migrate guest OS to spare
  - Utilize Xen migration



VM-level Migration Performance Impact

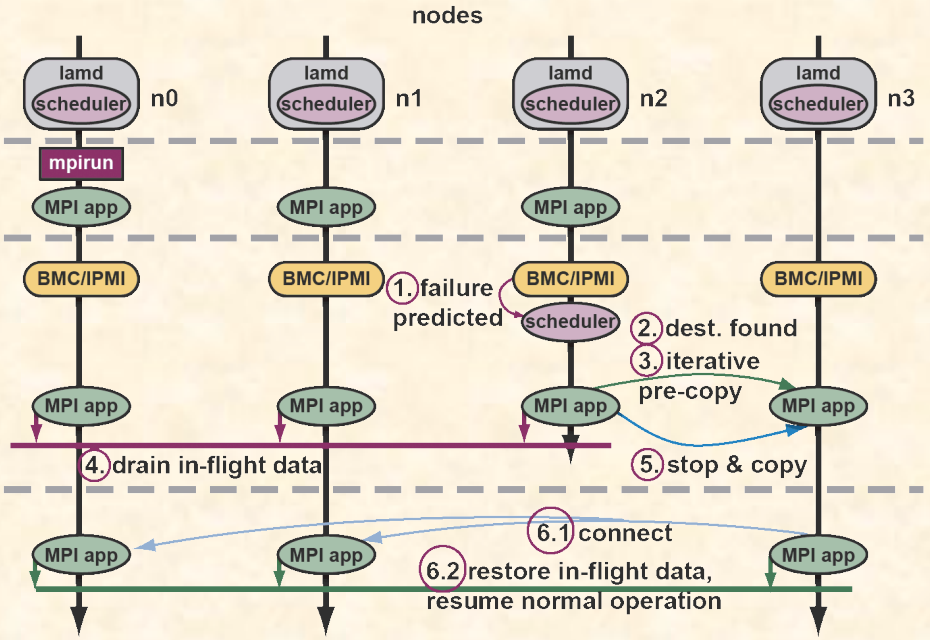
- Single migration overhead
  - Live : 0.5-5.0%
- Double migration overhead
  - Live : 2.0-8.0%
- Migration duration
  - Stop & copy : 13-14s
  - Live : 14-24s
- Application downtime
  - Stop & copy > Live



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

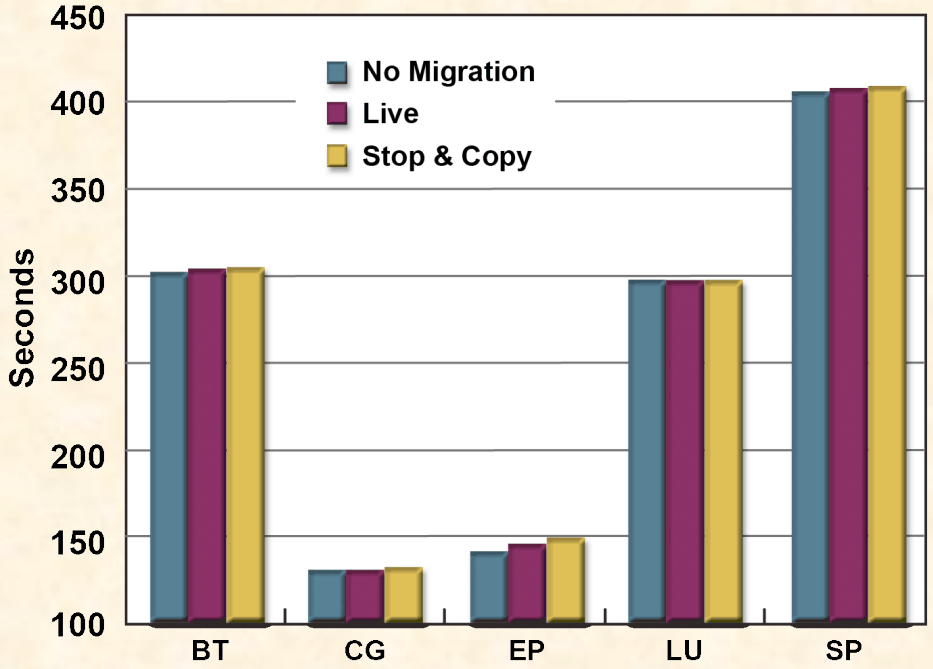
Process-Level Migration with BLCR

- LAM/MPI with Berkeley Lab Checkpoint/Restart (BLCR)
- Per-node health monitoring
- 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
- Available through BLCR distribution



Process-Level Migration Performance Impact

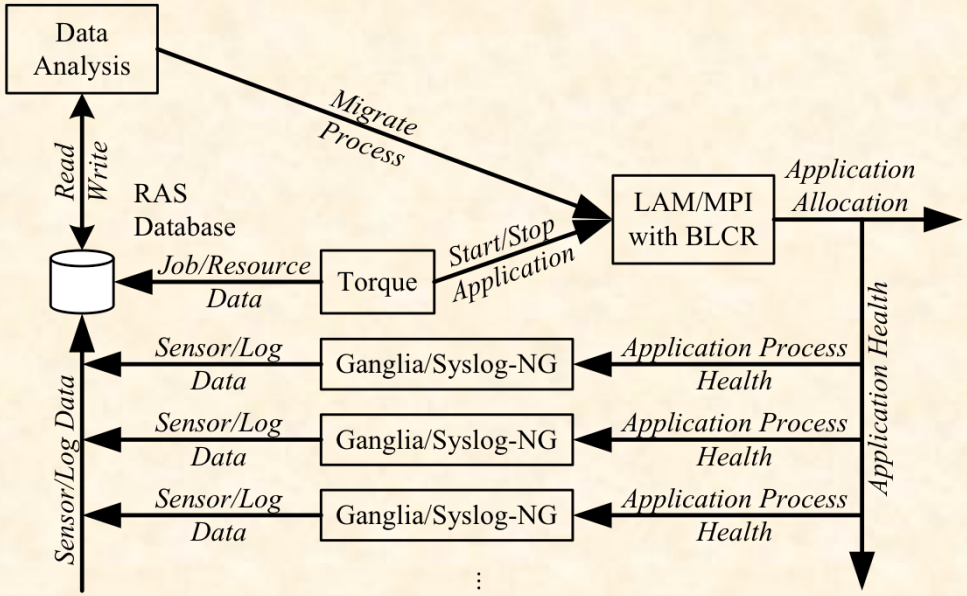
- Single migration overhead
  - Stop & copy : 0.09-6.00%
  - Live : 0.08-2.98%
- Single migration duration
  - Stop & copy : 1.0-1.9s
  - Live : 2.6-6.5s
- Application downtime
  - Stop & copy > Live
- Node eviction time
  - Stop & copy < Live



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

Proactive Fault Tolerance Framework

- Central MySQL database
- Environmental monitoring
  - OpenIPMI and Ganglia
- Event logging and analysis
  - Syslog forwarding
- Job & resource monitoring
  - Torque (epilogue/ prologue)
- Migration mechanism
  - Process-level with BLCR



System Monitoring with Ganglia and Syslog

Experiment #1:

- 32-node Linux cluster
- 30 second interval
- 40 Ganglia metrics
- ≈20 GB of data in 27 days
- ≈33 MB/hour
- ≈275 kb/interval

Experiment #2:

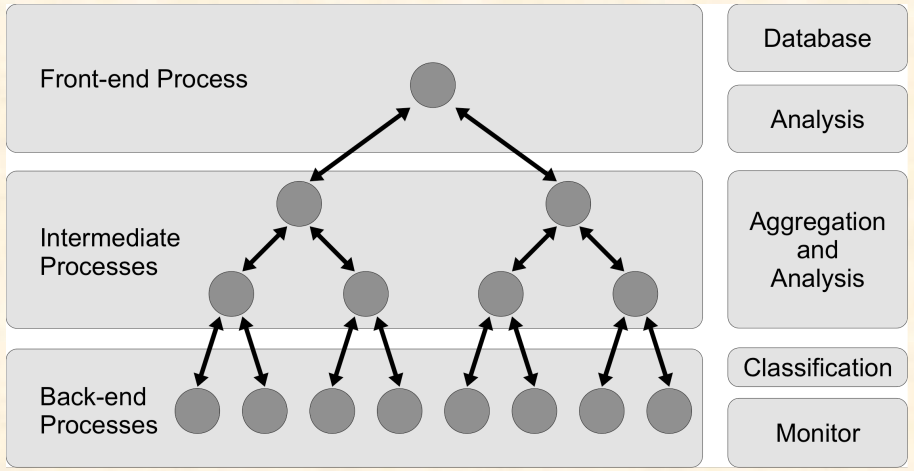
- 32-node Linux cluster
- 30 second interval
- 40 Ganglia metrics
- No measurable impact on NAS benchmarks

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)

MRNet-based System Monitoring

- Aggregation of metrics
- Tree-based overlay network
- Fan-in for metric data
- Fan-out for management
- Classification of data on back-end nodes
- In-flight processing on intermediate nodes
- Collection and storing on front-end node



- 1 MB of data in 4 hours
- ≈250 kB/hour
- ≈2 kb/interval
- ≈56x less than Ganglia

Incremental Checkpointing with BLCR

- Recent enhancement for Berkeley Lab Checkpoint/ Restart (BLCR)
- Track differences with dirty bit at PTE
- Hybrid: 1 full and *k* incremental checkpoints
- Available through BLCR distribution

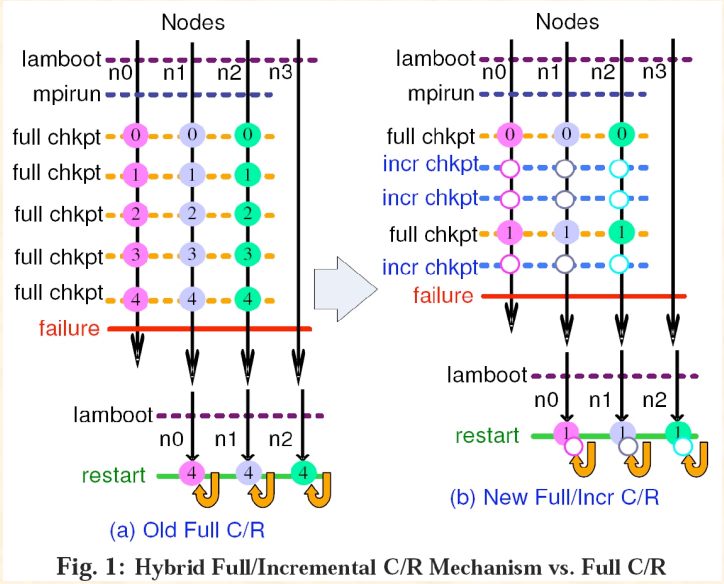
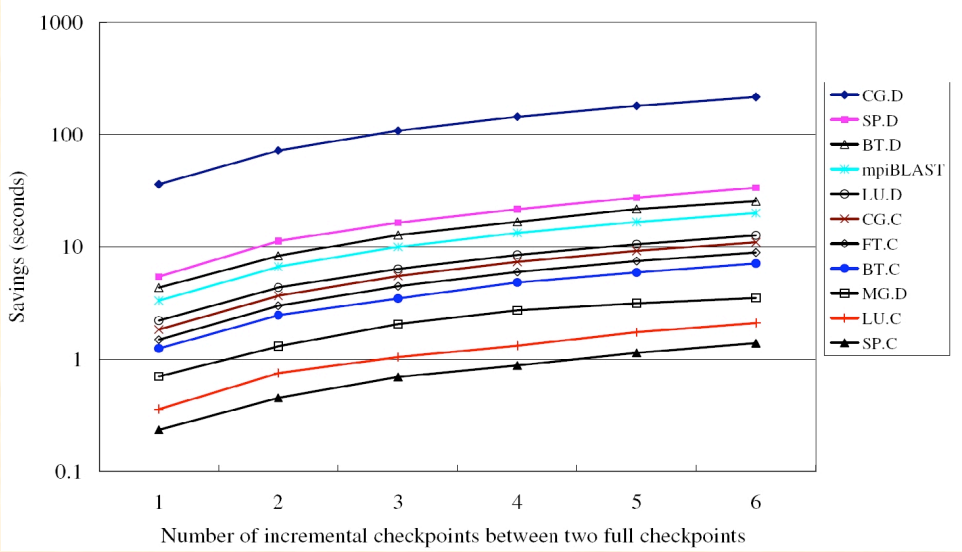
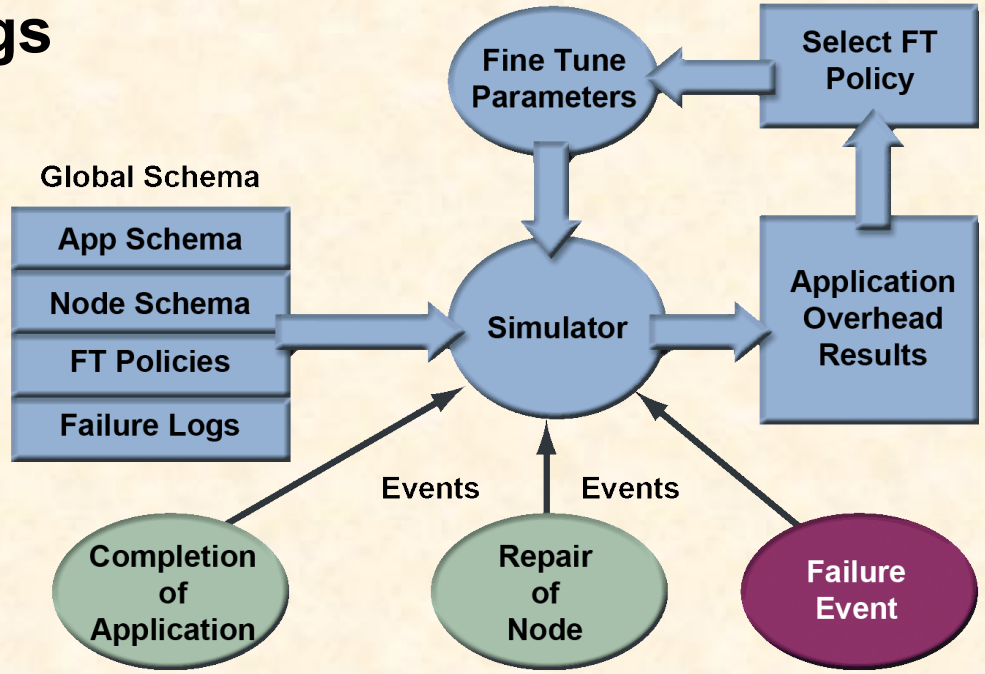


Fig. 1: Hybrid Full/Incremental C/R Mechanism vs. Full C/R



Simulation of Fault Tolerance Policies

- DES with actual system logs
- Evaluation of policies
  - Reactive only
  - Proactive only
  - Reactive/proactive combination
- Evaluation of parameters
  - Checkpoint interval
  - Prediction accuracy
- Customizable simulation
  - # of active/spare nodes
  - Checkpoint and migration overheads



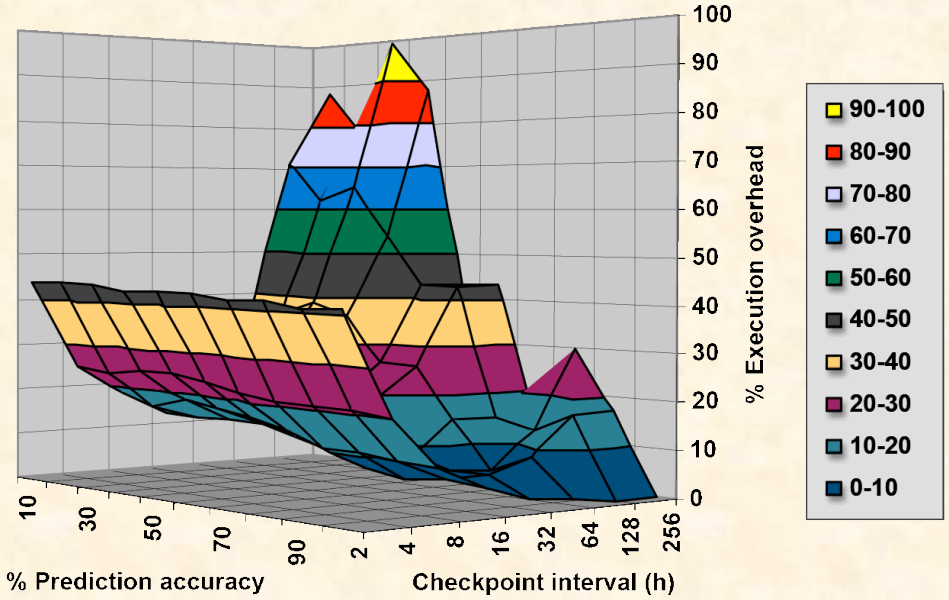
Combining Proactive & Reactive Approaches

- Optimum for the given logs:
  - Prediction accuracy > 60%
  - Checkpoint interval 16-32h
- 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

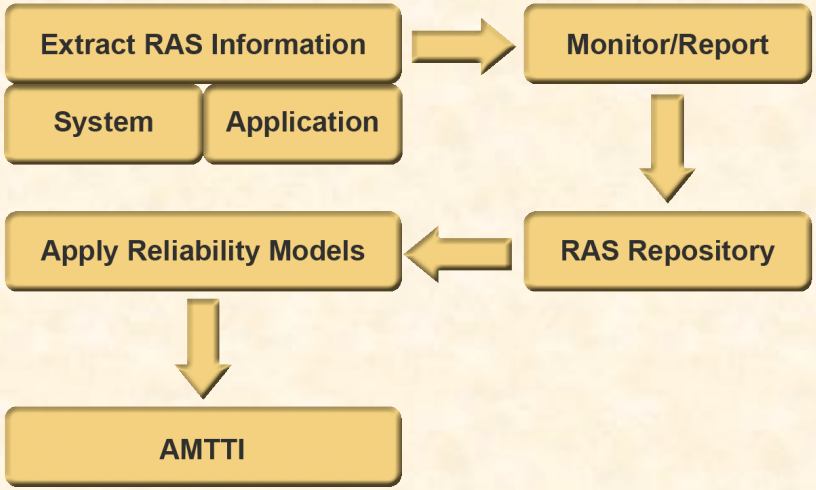
Simulation based on ASCI White logs (nodes 1-125 and 500-512)

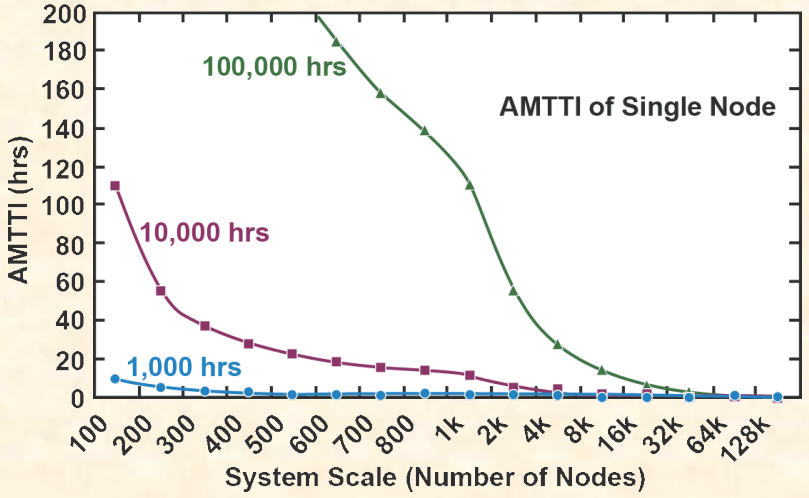
Execution overhead for various checkpoint intervals and different prediction accuracy



Research in Reliability Modeling

- Application MTTI estimation
  - Monitoring & recording of application & system health
  - Reliability analysis on recorded data
  - Adaptation of checkpoint interval to system health
- Finding failure patterns
  - Additional recording of application interrupts
  - Reliability analysis on recent and historical data





Accomplishments

- Select developed software
  - BLCR-based process migration
  - BLCR-based incremental/adaptive checkpointing
  - MRNet-based scalable system monitoring
  - Fully integrated RAS framework for Linux clusters
  - Simulation framework for fault tolerance policies
  - Xen-based virtual machine migration framework
- Select developed theoretical foundations:
  - Analyzed system and component reliability
  - Identified needs for more accurate failure reporting
- Numerous high-level publications
  - ICS'06, ICS'07, IPDPS'07, SC'08, …

Ongoing FAST-OS Work

- Extending the MRNet-based system monitoring
  - In-flight statistical processing of monitoring data
  - In-flight pattern matching of syslog data (for reduction)
- Advanced statistical analysis for *anomaly* detection
- Compute-node rejuvenation, e.g., reboot between jobs
- Checkpointing GPUs using CUDA streams
- Next HA-OSCAR release:
  - Extends the OSCAR Linux cluster installation and management suite with the developed RAS mechanisms

Acknowledgements

- Investigators at Oak Ridge National Laboratory:
  - *Stephen L. Scott [Lead PI]*, Christian Engelmann, Geoffroy Vallée, Thomas Naughton, Anand Tikotekar, George Ostrouchov
- Investigators at Louisiana Tech University:
  - *Chokchai (Box) Leangsuksun [Lead PI]*, Nichamon Naksinehaboon, Raja Nassar, Mihaela Paun
- Investigators at North Carolina State University:
  - *Frank Mueller [Lead PI]*, Chao Wang, Arun Nagarajan, Jyothish Varma
- Funding sources:
  - U.S. Department of Energy, Office of Science, FASTOS 2