

# Resiliency for high-performance computing

**Presented by**

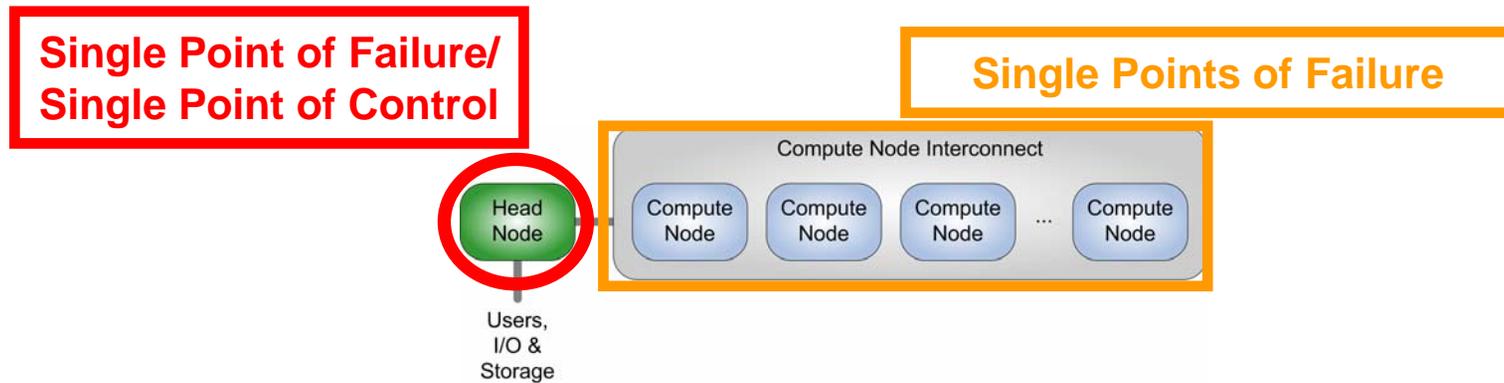
Christian Engelmann

Computer Science and Mathematics Division  
Oak Ridge National Laboratory, Oak Ridge, TN, USA

# Outline

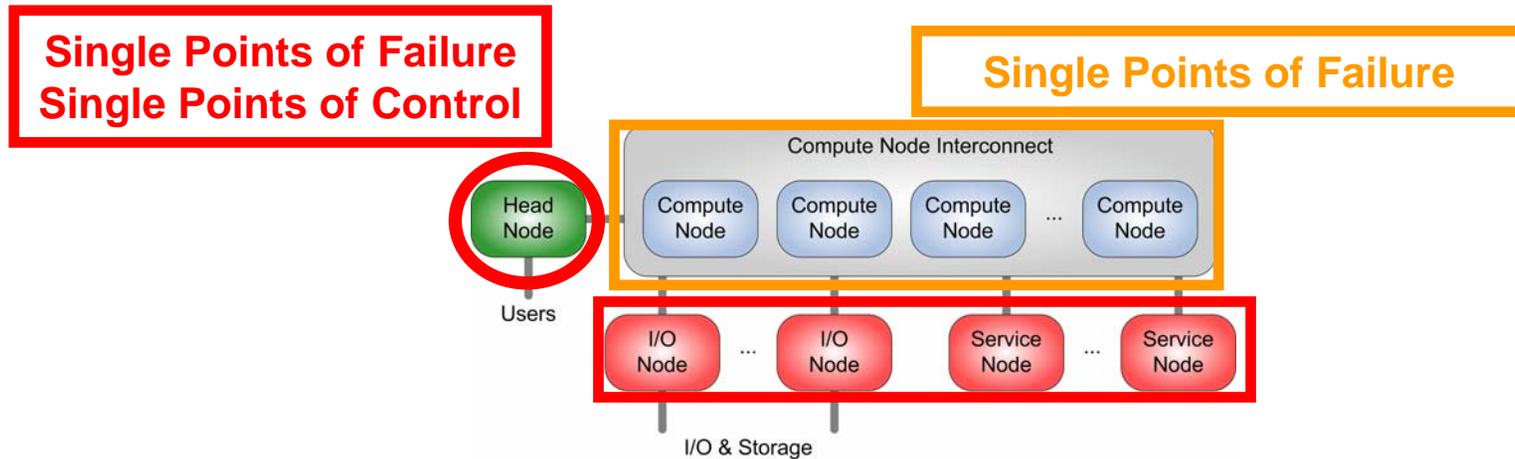
- **Resilience deficiencies of high-performance computing architectures**
- **Research and development goals and projects overview**
- **Symmetric active/active redundancy for head and service nodes**
- **Job pause approach for reactive compute node fault tolerance**
- **Pre-emptive migration for proactive compute node fault tolerance**
- **Reliability analysis and modeling for fault prediction and anticipation**
- **Evaluation of compute node fault tolerance policies using simulation**
- **Vision of a holistic resiliency framework concept**
- **Conclusion: Achievements, ongoing work and future plans**

# Traditional Beowulf cluster computing architecture



- **Single head node manages entire HPC system**
- **System-wide services are provided by head node:**
  - Job & resource management, networked file system, ...
- **Local services are provided by compute nodes**
  - Message passing (MPI, PVM), ...

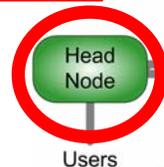
# Recent trend toward massively parallel processing (MPP) architectures



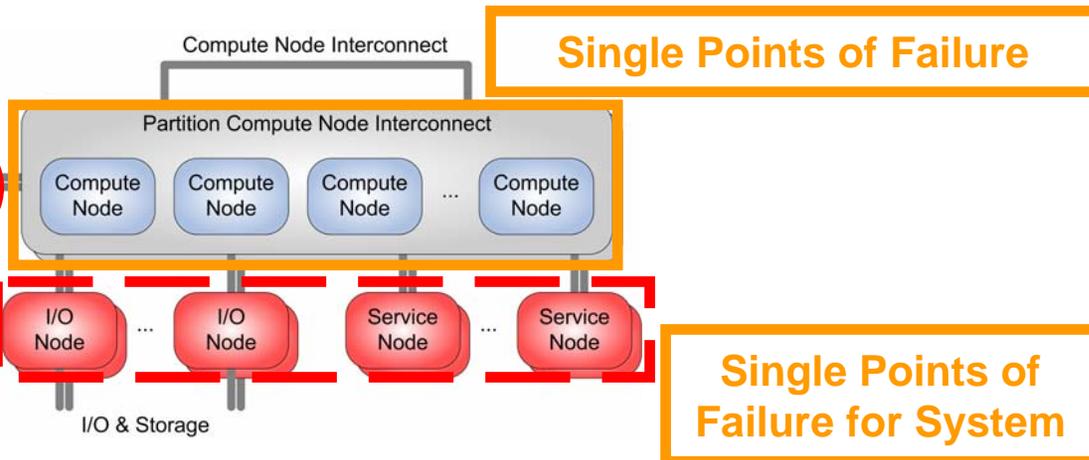
- **Single head node and additional service nodes manage the entire HPC system**
- **System-wide services are provided by head node and are offloaded to service nodes, e.g., networked file system**
- **Local services are provided by service nodes and compute nodes, e.g., message passing**

# Recent trend toward partitioned MPP architectures

**Single Point of Failure/  
Single Point of Control**



**Single Points of Failure/Single  
Point of Control for Partition**



**Single Points of Failure**

**Single Points of  
Failure for System**

- **Single head node manages entire HPC system**
- **Service nodes manage and support compute nodes belonging to their partitions**

# Availability Measured by the Nines

<http://info.nccs.gov/resources> - HPC system status at Oak Ridge National Laboratory

9's	Availability	Downtime/Year	Examples
1	90.0%	36 days, 12 hours	Personal Computers
2	99.0%	87 hours, 36 min	Entry Level Business
3	99.9%	8 hours, 45.6 min	ISPs, Mainstream Business
4	99.99%	52 min, 33.6 sec	Data Centers
5	99.999%	5 min, 15.4 sec	Banking, Medical
6	99.9999%	31.5 seconds	Military Defense

- Enterprise-class hardware + Stable Linux kernel = 5+
- Substandard hardware + Good high availability package = 2-3
- Today's supercomputers = 1-2
- My desktop = 1-2

# Typical Failure Causes in HPC Systems

- **Hardware failures due to wear/age of:**
  - Hard drives, memory modules, network cards, and processors
- **Software failures due to bugs in:**
  - Operating system, middleware, applications
- **Stress-related failures that exceed design specifications in:**
  - Hardware, e.g. due to excessive heat radiation
  - Software, e.g. due to (unintentional) denial of service
- **Radiation-induced soft errors (bit flips due to electromagnetic interference, heat radiation, natural neutron radiation) in:**
  - Memory modules, network cards, and processors
- ➔ **Different scale requires different solutions:**
  - Compute nodes (up to 150,000)
  - Front-end, service, and I/O nodes (1 to 150)

# Outline

- Resilience deficiencies of high-performance computing architectures
- **Research and development goals and projects overview**
- Symmetric active/active redundancy for head and service nodes
- Job pause approach for reactive compute node fault tolerance
- Pre-emptive migration for proactive compute node fault tolerance
- Reliability analysis and modeling for fault prediction and anticipation
- Evaluation of compute node fault tolerance policies using simulation
- Vision of a holistic resiliency framework concept
- **Conclusion: Achievements, ongoing work and future plans**

# Research and development goals

- **Efficient redundancy strategies** for head and service nodes in HPC systems to provide high availability as well as high performance of critical infrastructure services
- **Reactive fault tolerance** for HPC compute nodes utilizing the job pause approach as well as checkpoint interval and placement adaptation to actual and predicted system health threats
- **Proactive fault tolerance** using system-level virtualization in HPC environments for pre-emptive migration of computation away from compute nodes that are about to fail
- **Reliability analysis** for identifying pre-fault indicators, predicting failures, and modeling and monitoring of individual component and overall HPC system reliability
- **Holistic fault tolerance** technology through combination of adaptive proactive and reactive fault tolerance mechanisms in conjunction with system health monitoring and reliability analysis

# MOLAR: Adaptive runtime support for high-end computing operating and runtime systems

- **Previous effort: 2004-2007**
- **Addresses the challenges for operating and runtime systems to run large applications efficiently on large-scale supercomputers**
- **Part of the Forum to Address Scalable Technology for Runtime and Operating Systems (FAST-OS)**
- **MOLAR is a collaborative research effort ([www.fastos.org/molar](http://www.fastos.org/molar)):**



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

- **Current effort: 2008-2011**
- **Addresses the fault resilience challenges for operating and runtime systems of next-generation large-scale supercomputers**
- **Part of the Forum to Address Scalable Technology for Runtime and Operating Systems (FAST-OS)**
- **RAS HPC is a collaborative research effort ([www.fastos.org/ras](http://www.fastos.org/ras)):**



**NC STATE UNIVERSITY**



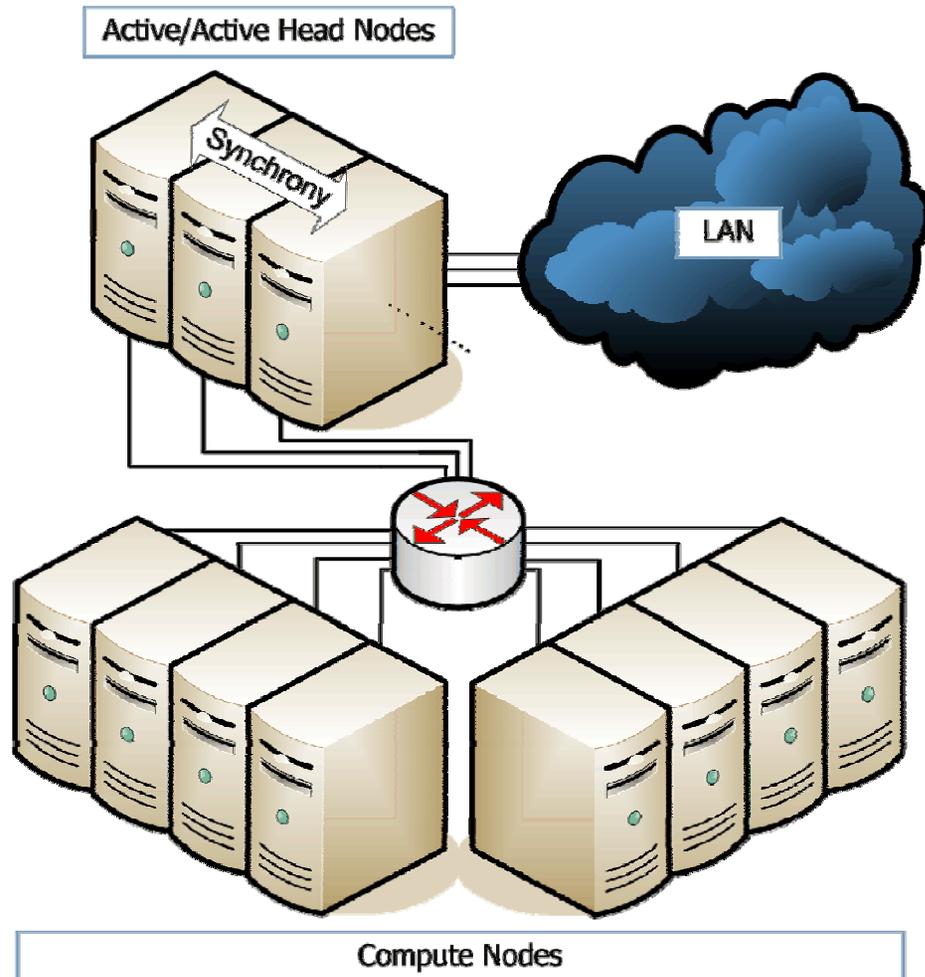
The University of Reading

**LOUISIANA TECH**  
UNIVERSITY®

# Outline

- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- **Symmetric active/active redundancy for head and service nodes**
- Job pause approach for reactive compute node fault tolerance
- Pre-emptive migration for proactive compute node fault tolerance
- Reliability analysis and modeling for fault prediction and anticipation
- Evaluation of compute node fault tolerance policies using simulation
- Vision of a holistic resiliency framework concept
- Conclusion: Achievements, ongoing work and future plans

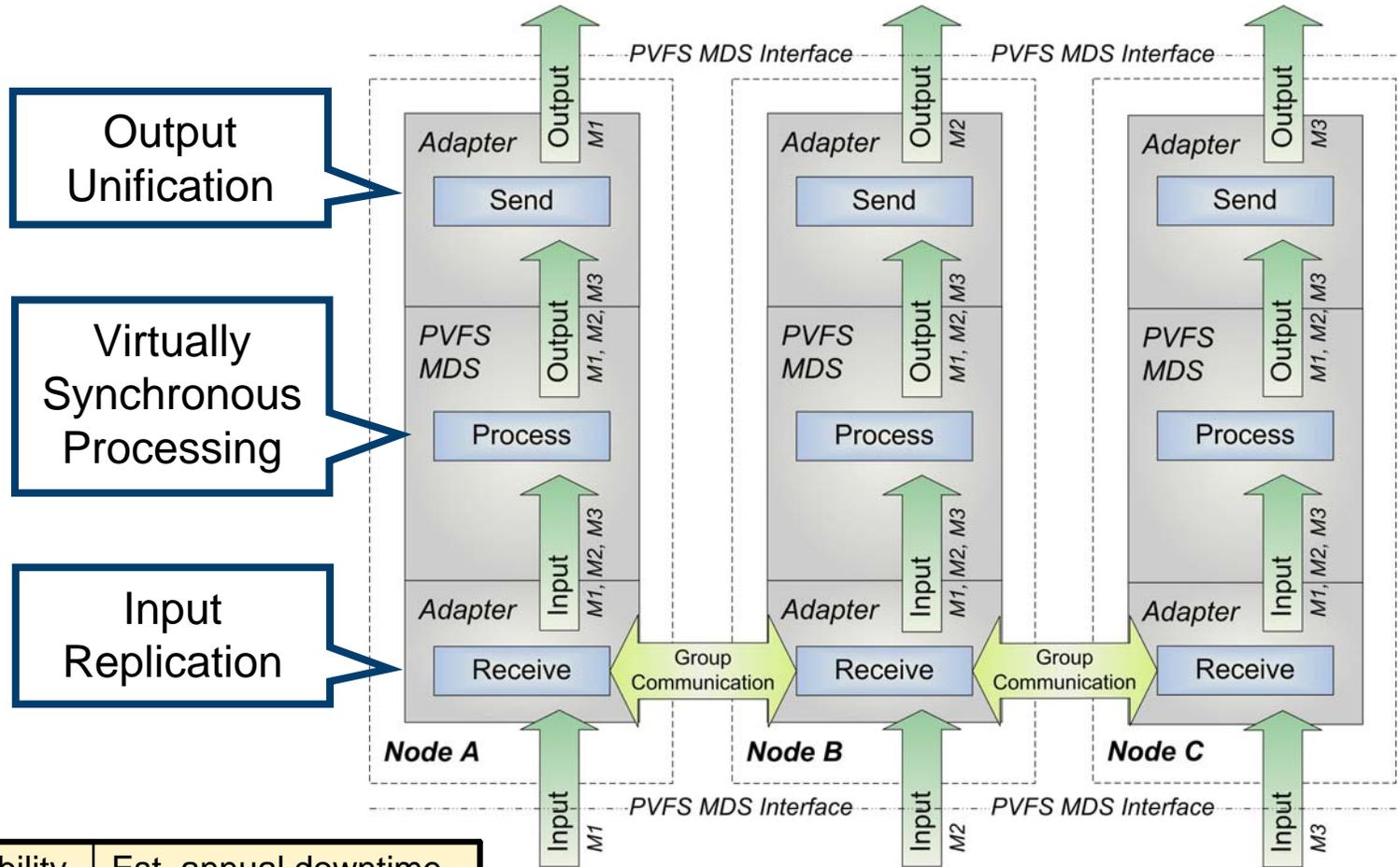
# Symmetric active/active redundancy for head and service nodes



- Many active head nodes
- Work load distribution
- Symmetric replication between head nodes
- Continuous service
- Always up to date
- No fail-over necessary
- No restore-over necessary
- Virtual synchrony model
- **↓** Complex algorithms
- **→** Prototypes for PBS Torque and PVFS metadata server

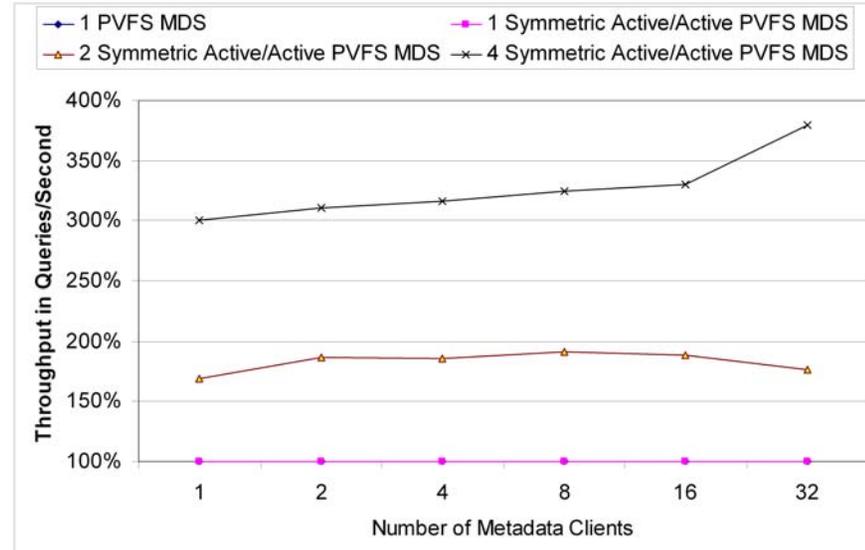
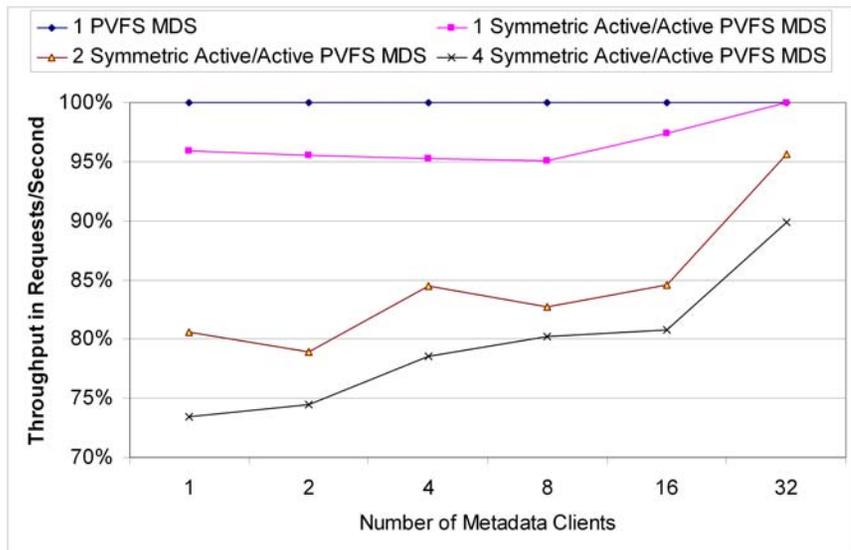
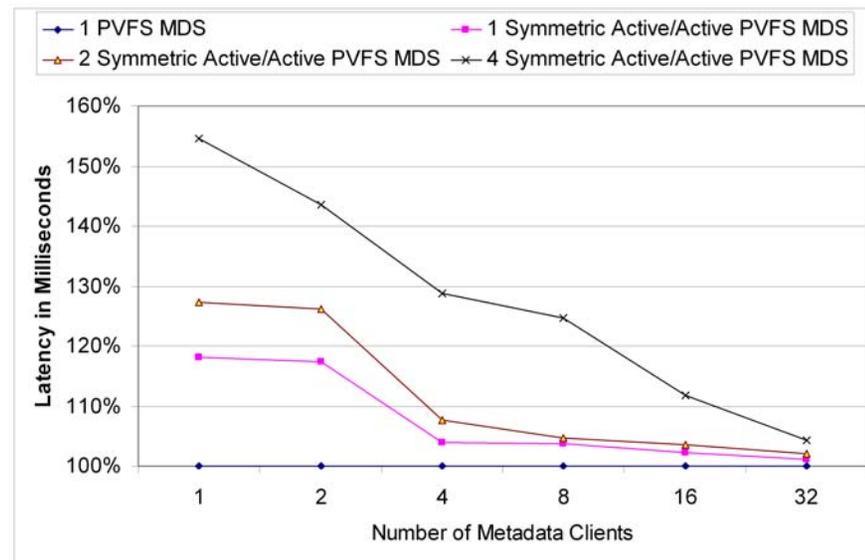
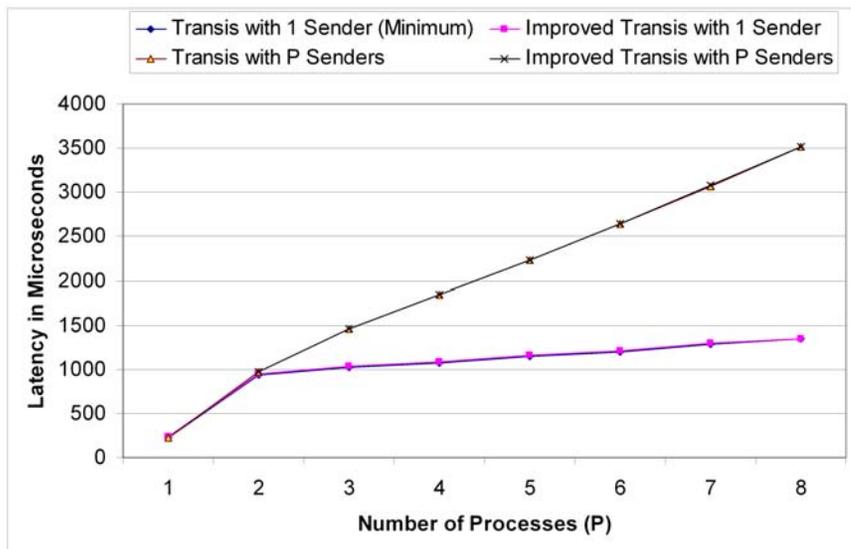


# Symmetric active/active replication



Nodes	Availability	Est. annual downtime
1	98.58%	5d 4h 21m
2	99.97%	1h 45m
3	99.9997%	1m 30s

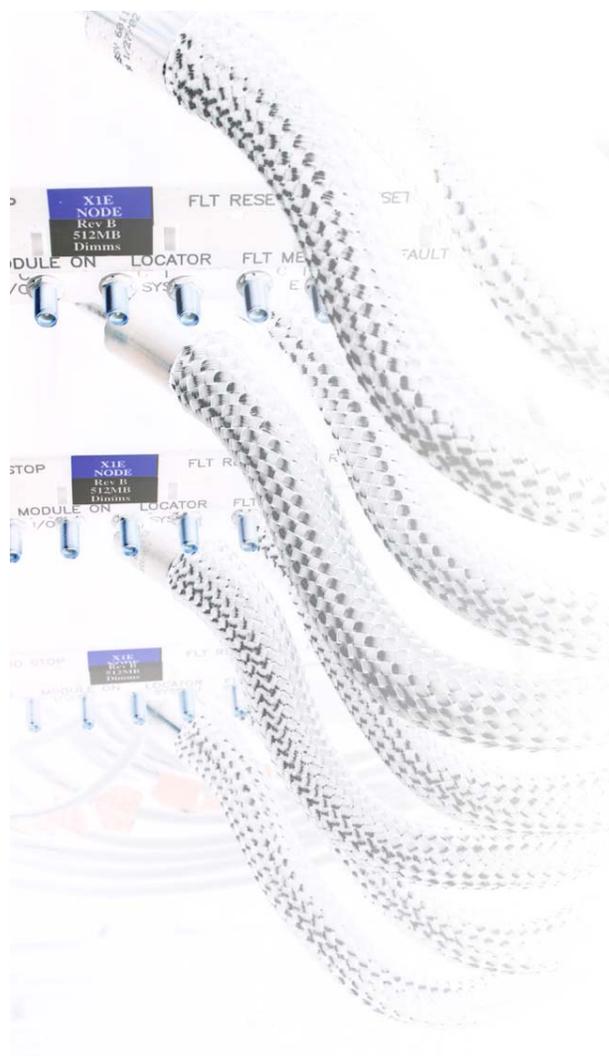
# Symmetric active/active Parallel Virtual File System metadata service



# Outline

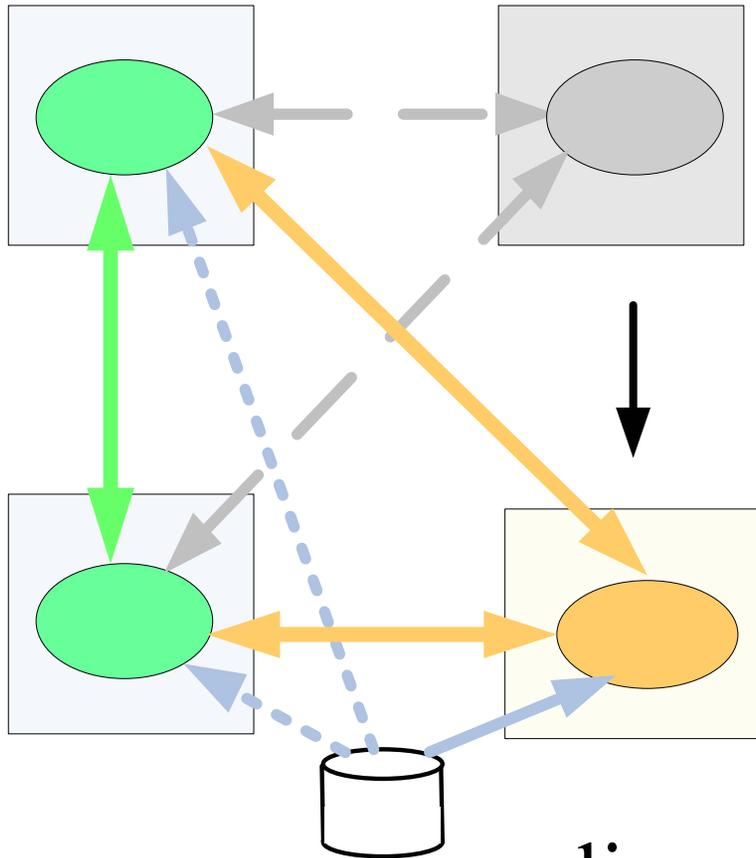
- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- Symmetric active/active redundancy for head and service nodes
- **Job pause approach for reactive compute node fault tolerance**
- Pre-emptive migration for proactive compute node fault tolerance
- Reliability analysis and modeling for fault prediction and anticipation
- Evaluation of compute node fault tolerance policies using simulation
- Vision of a holistic resiliency framework concept
- Conclusion: Achievements, ongoing work and future plans

# Reactive vs. proactive fault tolerance for compute nodes



- **Reactive fault tolerance:**
    - **State saving during failure-free operation**
    - **State recovery after failure**
    - **Assured quality of service, but limited scalability**
  - **Proactive fault tolerance:**
    - **System health monitoring and online reliability modeling**
    - **Failure anticipation and prevention through prediction and reconfiguration before failure**
    - **Highly scalable, but not all failures can be anticipated**
- ➔ **Ideal solution: Matching combination of both**

# Enhanced reactive fault tolerance with LAM/MPI+BLCR job pause mechanism



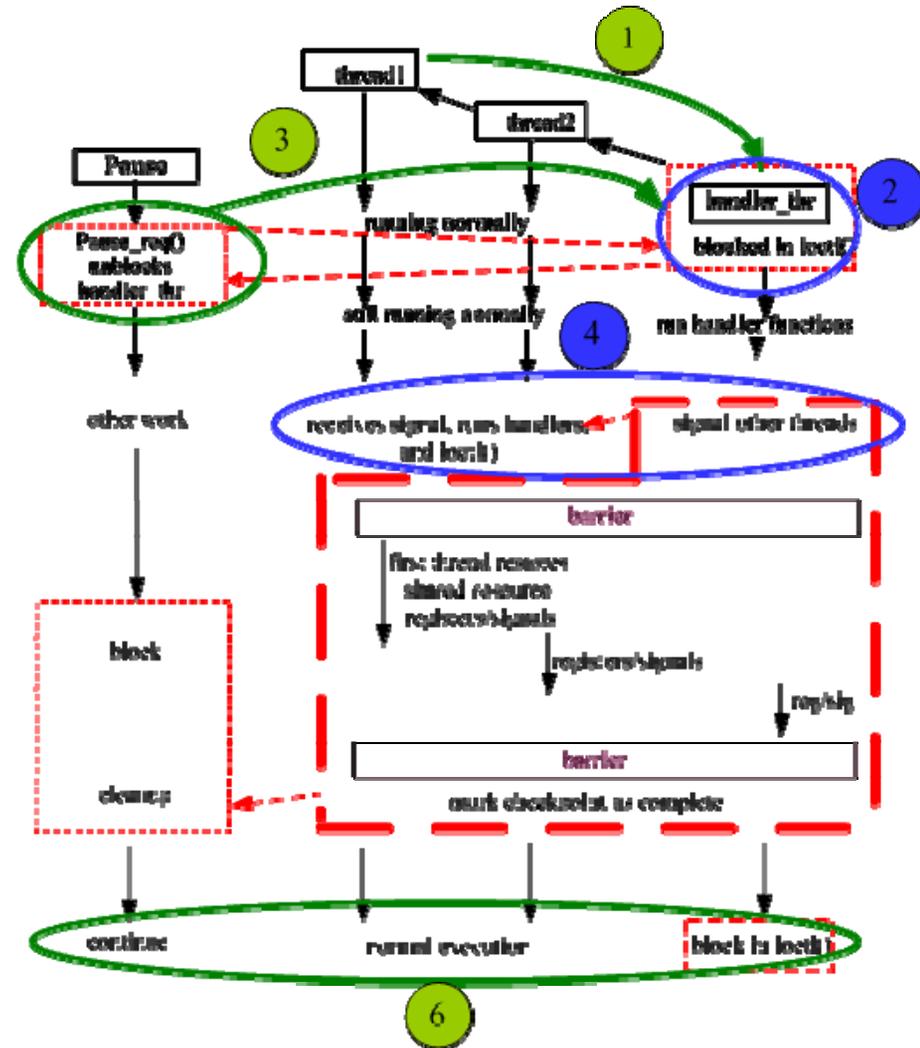
- **Operational nodes: Pause**
  - BLCR reuses existing processes
  - LAM/MPI reuses existing connections
  - Restore partial process state from checkpoint
- **Failed nodes: Migrate**
  - Restart process on new node from checkpoint
  - Reconnect with paused processes

➔ **Scalable MPI membership management for low overhead**

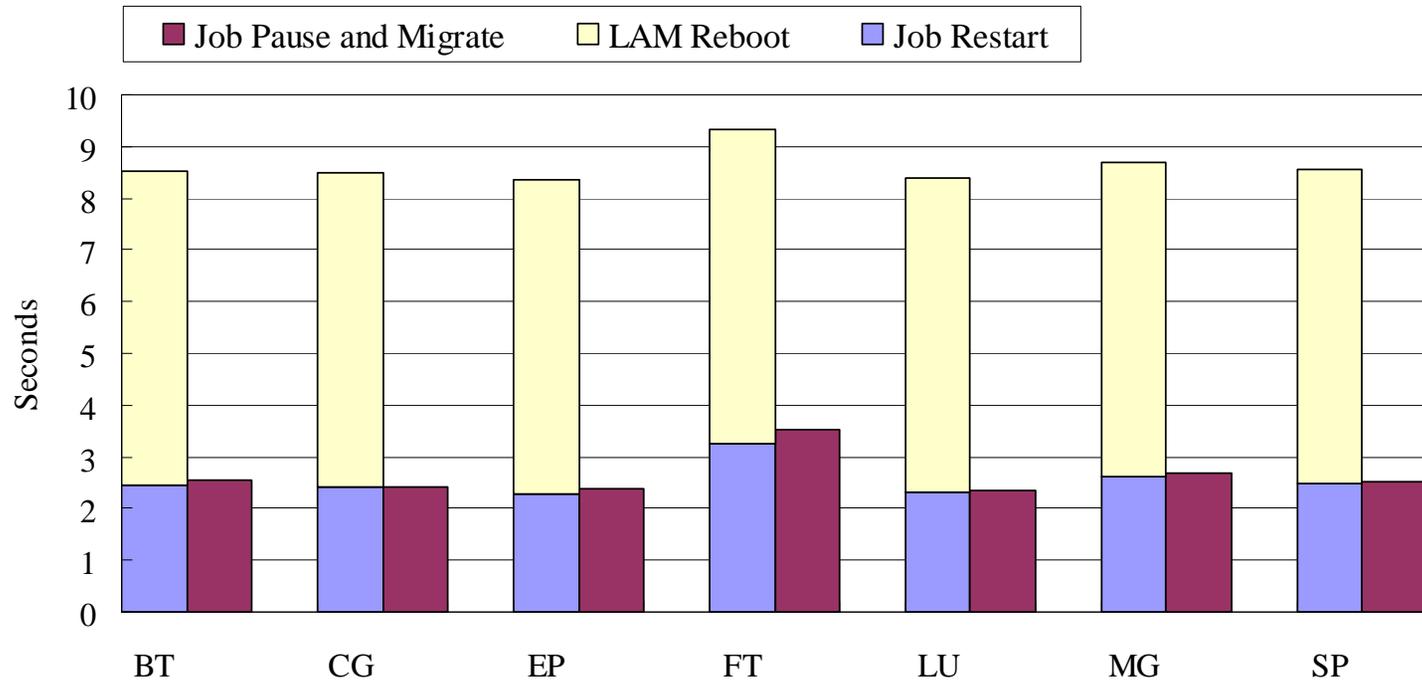
live node ➔ **Efficient, transparent, and automatic failure recovery**

# New job pause mechanism in BLCR

- Application registers threaded callback → spawns callback thread
- Thread blocks in kernel
- Pause utility calls `ioctl()`, unblocks callback thread
- All threads complete callbacks and enter kernel
- New: All threads restore part of their states
- Run regular application code from restored state



# LAM/MPI+BLCR job pause performance

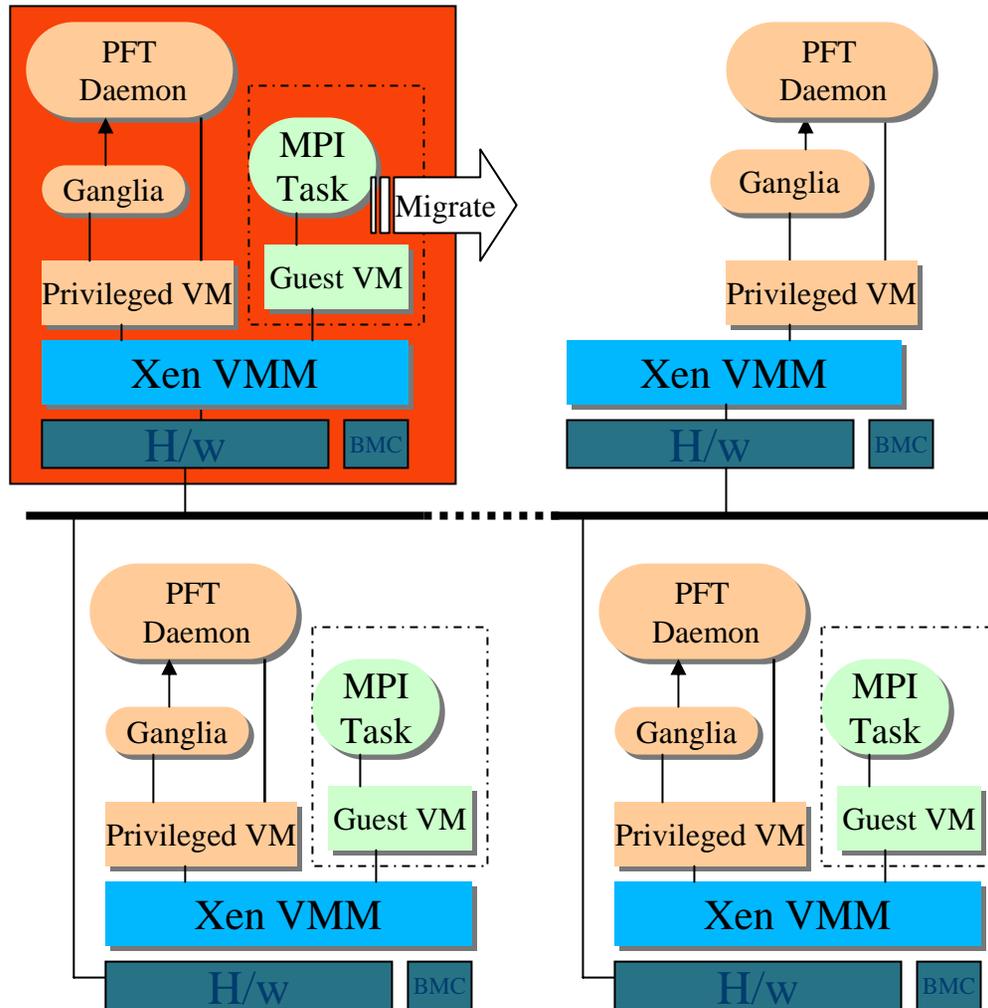


- **3.4% overhead over job restart, but**
  - No LAM reboot overhead
  - Transparent continuation of execution
- **No re-queue penalty**
- **Less staging overhead**

# Outline

- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- Symmetric active/active redundancy for head and service nodes
- Job pause approach for reactive compute node fault tolerance
- **Pre-emptive migration for proactive compute node fault tolerance**
- Reliability analysis and modeling for fault prediction and anticipation
- Evaluation of compute node fault tolerance policies using simulation
- Vision of a holistic resiliency framework concept
- Conclusion: Achievements, ongoing work and future plans

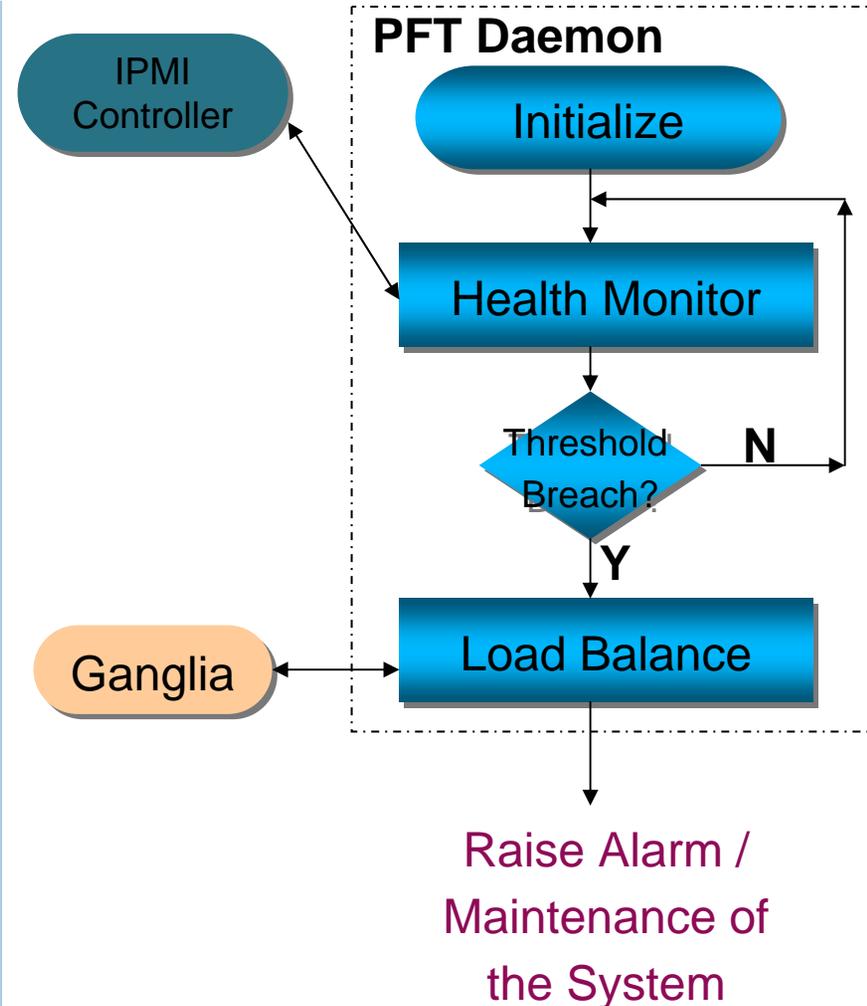
# Proactive fault tolerance using Xen virtualization



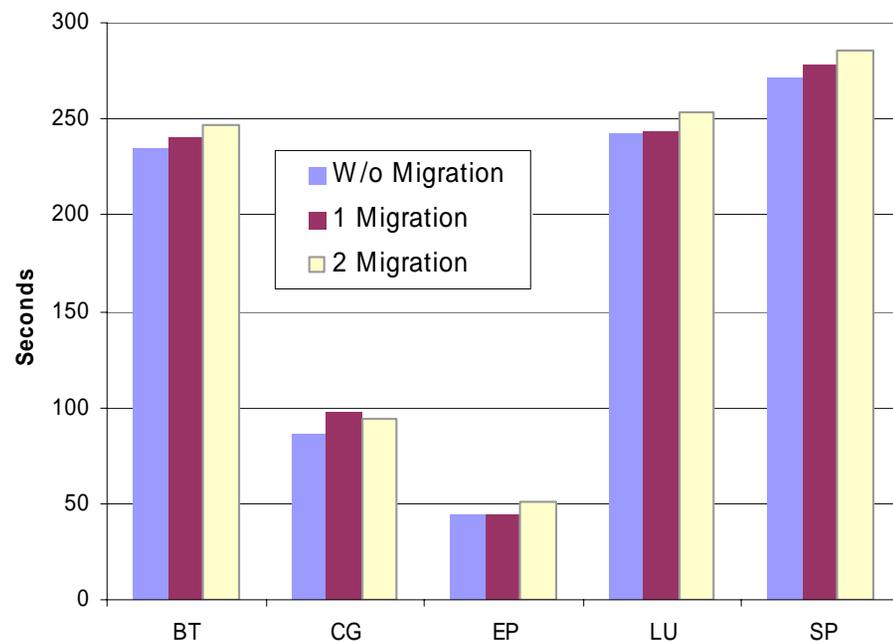
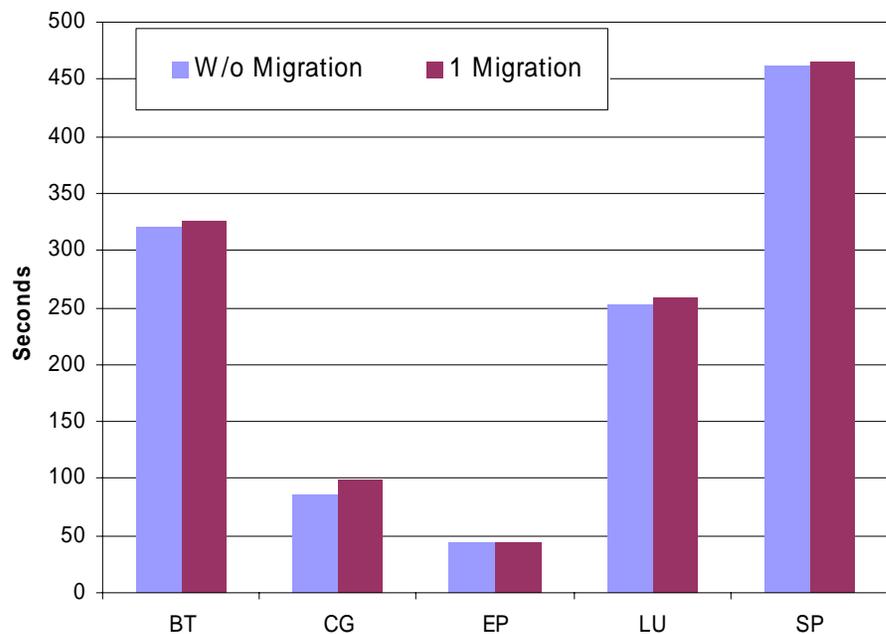
- **Stand-by Xen host (spare node without guest VM)**
  - **Deteriorating health:**
    - Migrate guest VM to spare node
  - **New host generates unsolicited ARP reply**
    - Indicates that guest VM has moved
    - ARP tells peers to resend to new host
- ➔ **Novel fault tolerance scheme that acts before a failure impacts a system**

# Proactive fault tolerance (PFT) daemon

- **Runs in privileged domain (host)**
- **Initialization**
  - Read safe threshold from config file
  - Init connection with IPMI controller
  - Obtain/filter set of available sensors
- **Health monitoring**
  - Read sensors from IPMI controller
  - Periodically sample data
  - Trigger load balancing if exceeding sensor threshold
- **VM migration**
  - Select target based on load
  - Invoke Xen live migration for VM



# VM migration performance impact



- **Single node failure: 0.5-5% additional cost over total wall clock time**
- **Double node failure: 2-8% additional cost over total wall clock time**

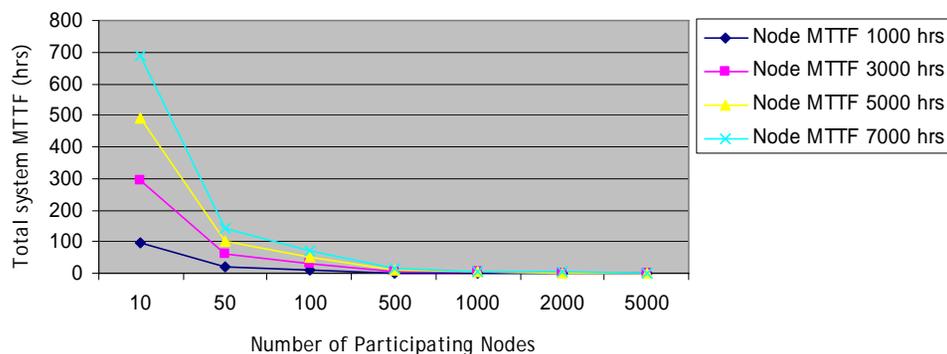
# Outline

- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- Symmetric active/active redundancy for head and service nodes
- Job pause approach for reactive compute node fault tolerance
- Pre-emptive migration for proactive compute node fault tolerance
- **Reliability analysis and modeling for fault prediction and anticipation**
- Evaluation of compute node fault tolerance policies using simulation
- Vision of a holistic resiliency framework concept
- Conclusion: Achievements, ongoing work and future plans

# HPC reliability analysis and modeling for prediction and anticipation

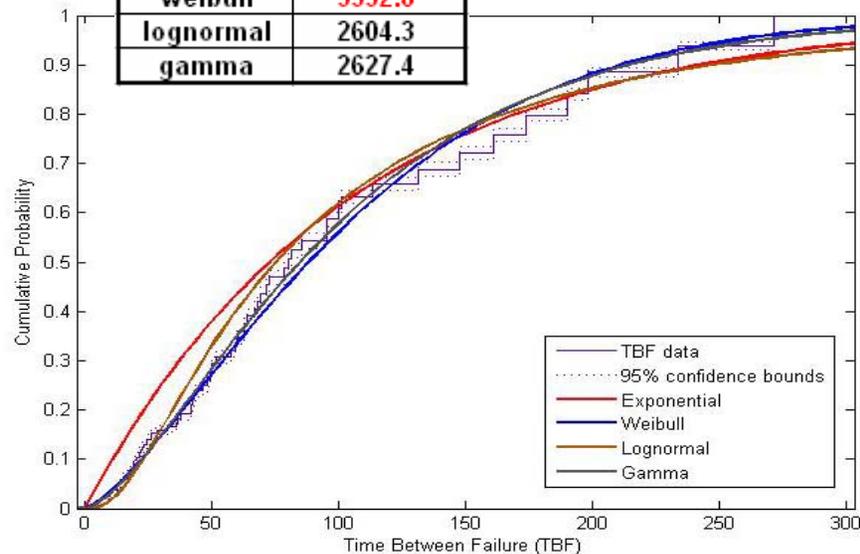
- Programming paradigm and system scale impact reliability
- Reliability analysis:
  - Estimate mean time to failure (MTTF)
  - Obtain failure distribution: Exponential, Weibull, Gamma, ...
- Feedback into fault tolerance schemes for adaptation

System reliability (MTTF) for k-of-n AND Survivability (k=n) Parallel Execution model



negative likelihood value

exponential	2653.3
weibull	3532.8
lognormal	2604.3
gamma	2627.4

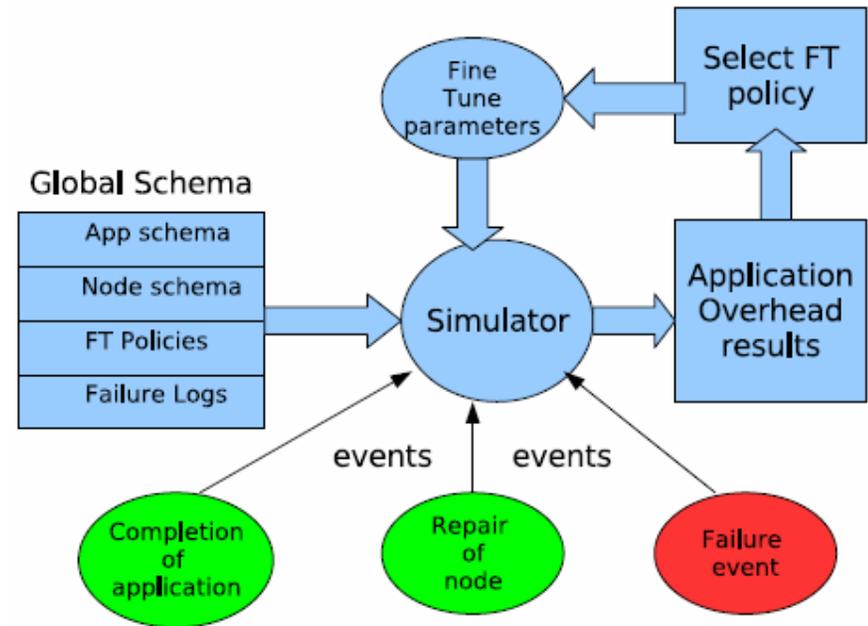


# Outline

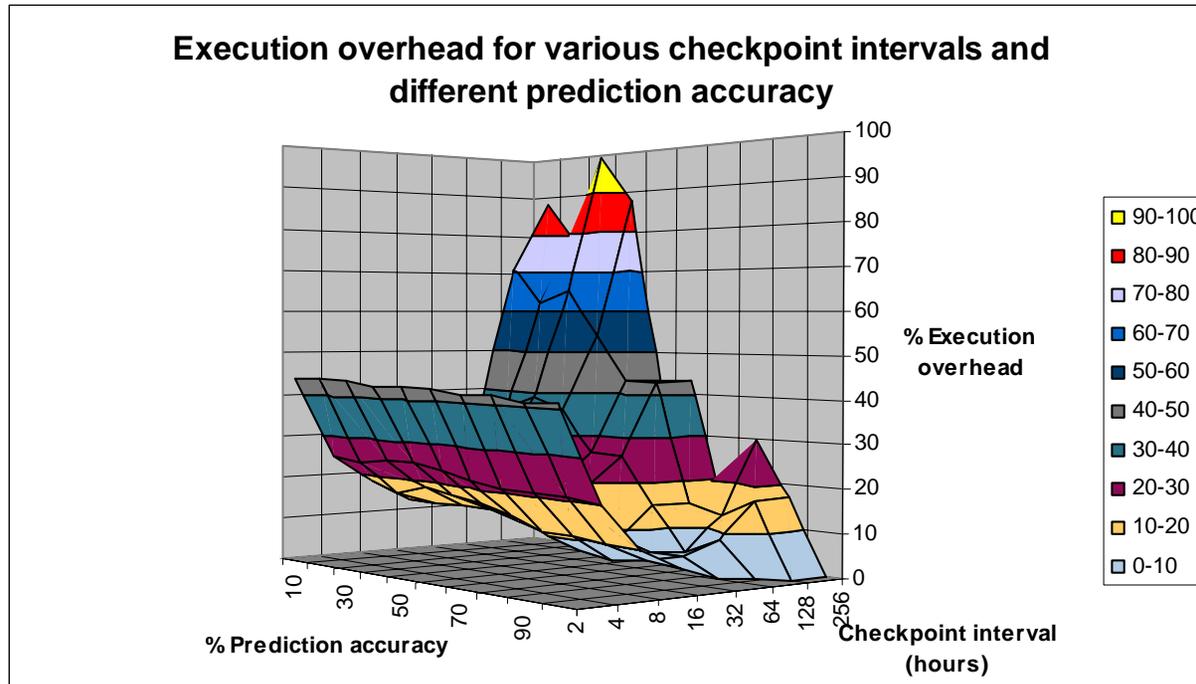
- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- Symmetric active/active redundancy for head and service nodes
- Job pause approach for reactive compute node fault tolerance
- Pre-emptive migration for proactive compute node fault tolerance
- Reliability analysis and modeling for fault prediction and anticipation
- **Evaluation of compute node fault tolerance policies using simulation**
- Vision of a holistic resiliency framework concept
- Conclusion: Achievements, ongoing work and future plans

# Simulation framework for HPC fault tolerance policies

- **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



# Combination of proactive and reactive fault tolerance: Simulation example 1

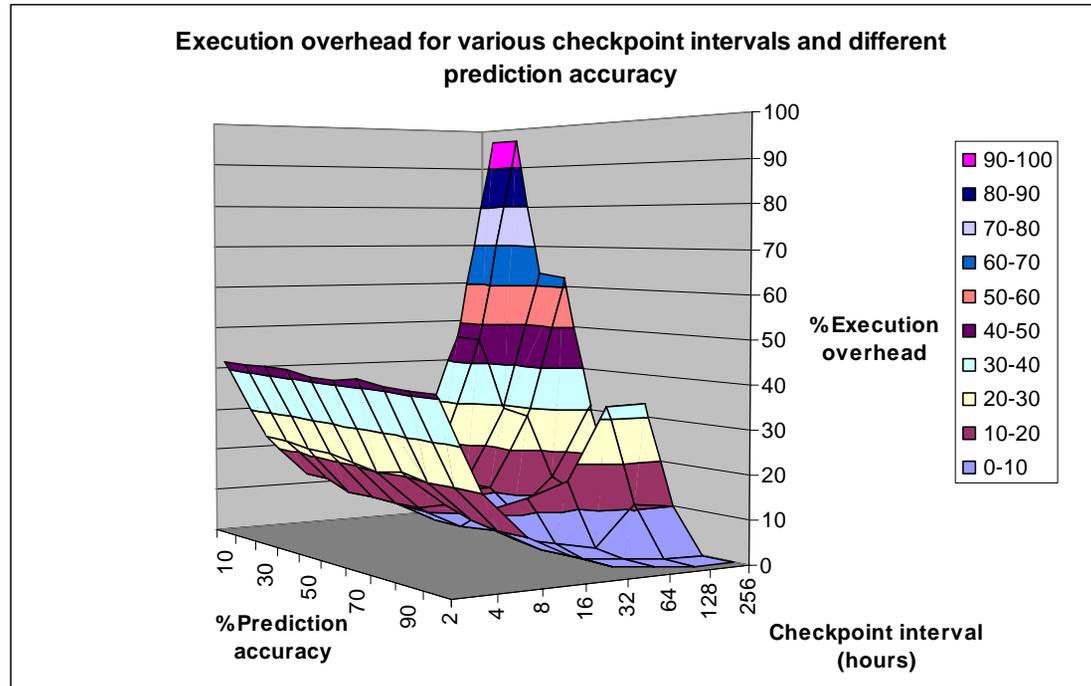


- **Best: Prediction accuracy >60% and checkpoint interval 16-32 hours**
- **Better than only proactive or only reactive**
- **Results for higher prediction accuracies and very low checkpoint intervals are worse than only proactive or only reactive**

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

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

# Combination of proactive and reactive fault tolerance: Simulation example 2



- **Best: Accuracy >60%, interval 16-64h**
- **70% and 32 hours:**
  - 8% gain over reactive only
  - 24% gain in over proactive only
- **80% and 32 hours:**
  - 10% gain over reactive only
  - 3% loss over proactive only

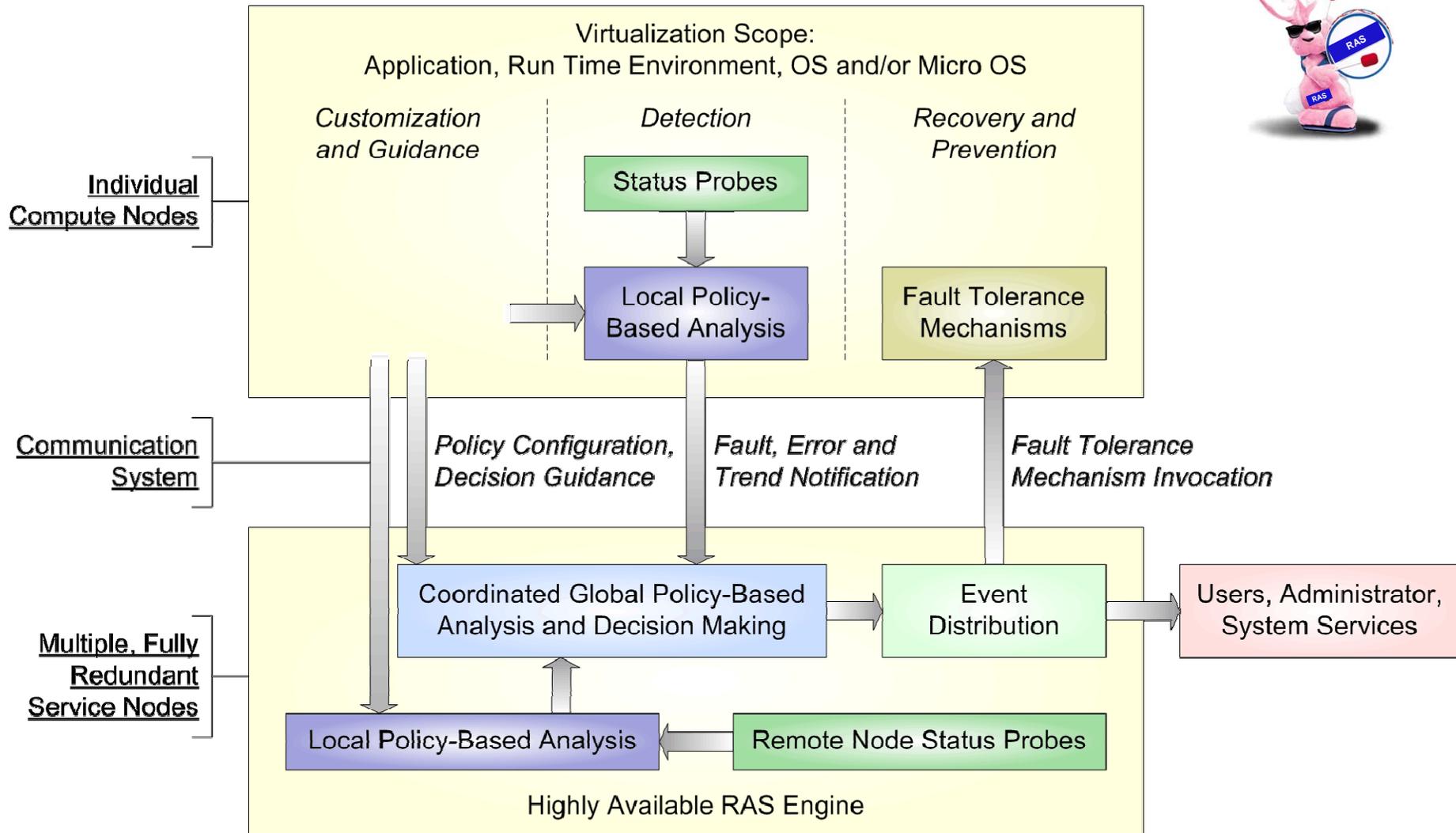
Number of processes	125
Active nodes / Spare nodes	125 / 12
Checkpoint overhead	50 min/Checkpoint
Migration overhead	1 min/Migration

Simulation based on ASCI White system logs  
(nodes 126 – 250 and 500-512)

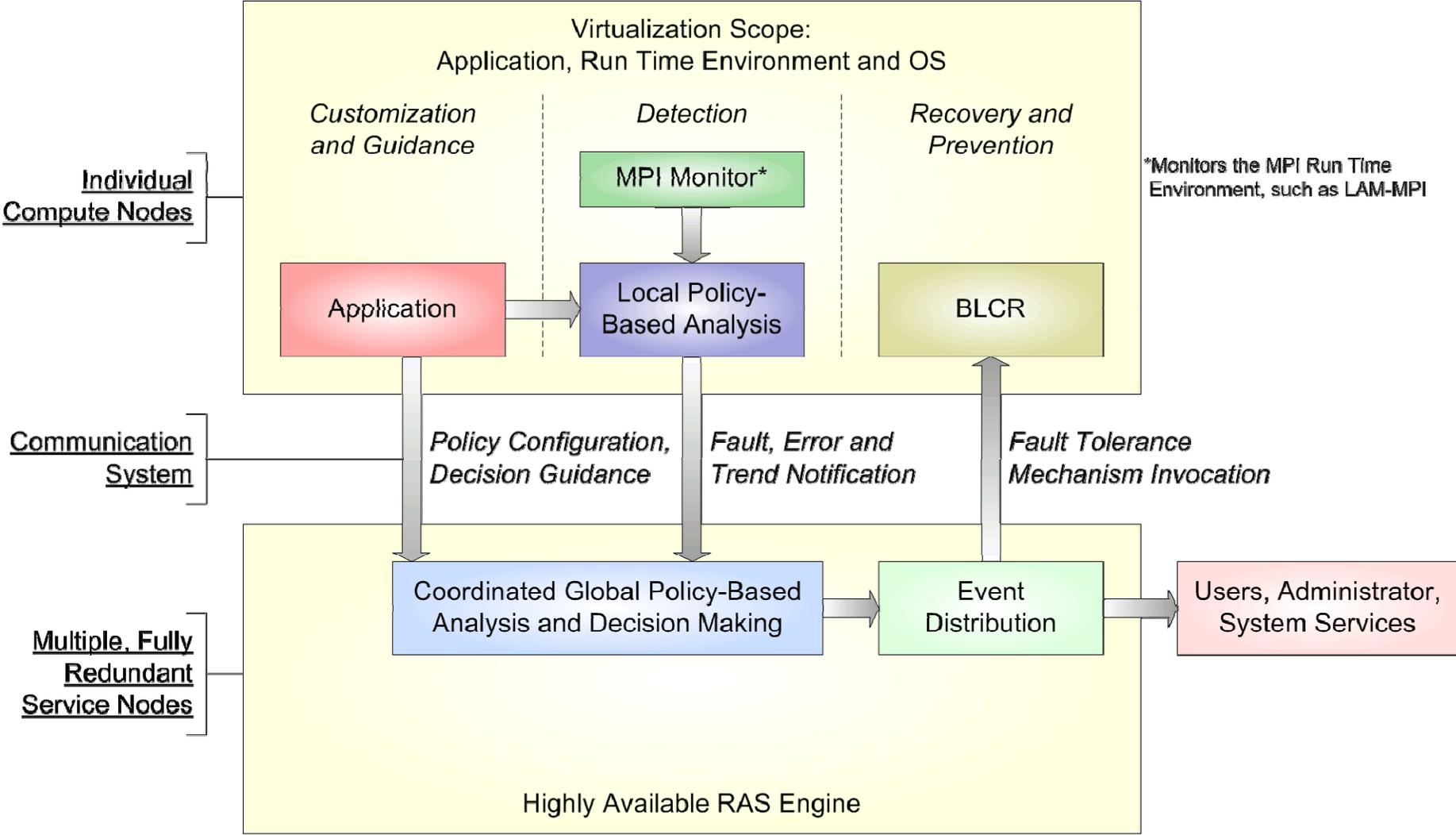
# Outline

- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- Symmetric active/active redundancy for head and service nodes
- Job pause approach for reactive compute node fault tolerance
- Pre-emptive migration for proactive compute node fault tolerance
- Reliability analysis and modeling for fault prediction and anticipation
- Evaluation of compute node fault tolerance policies using simulation
- **Vision of a holistic resiliency framework concept**
- Conclusion: Achievements, ongoing work and future plans

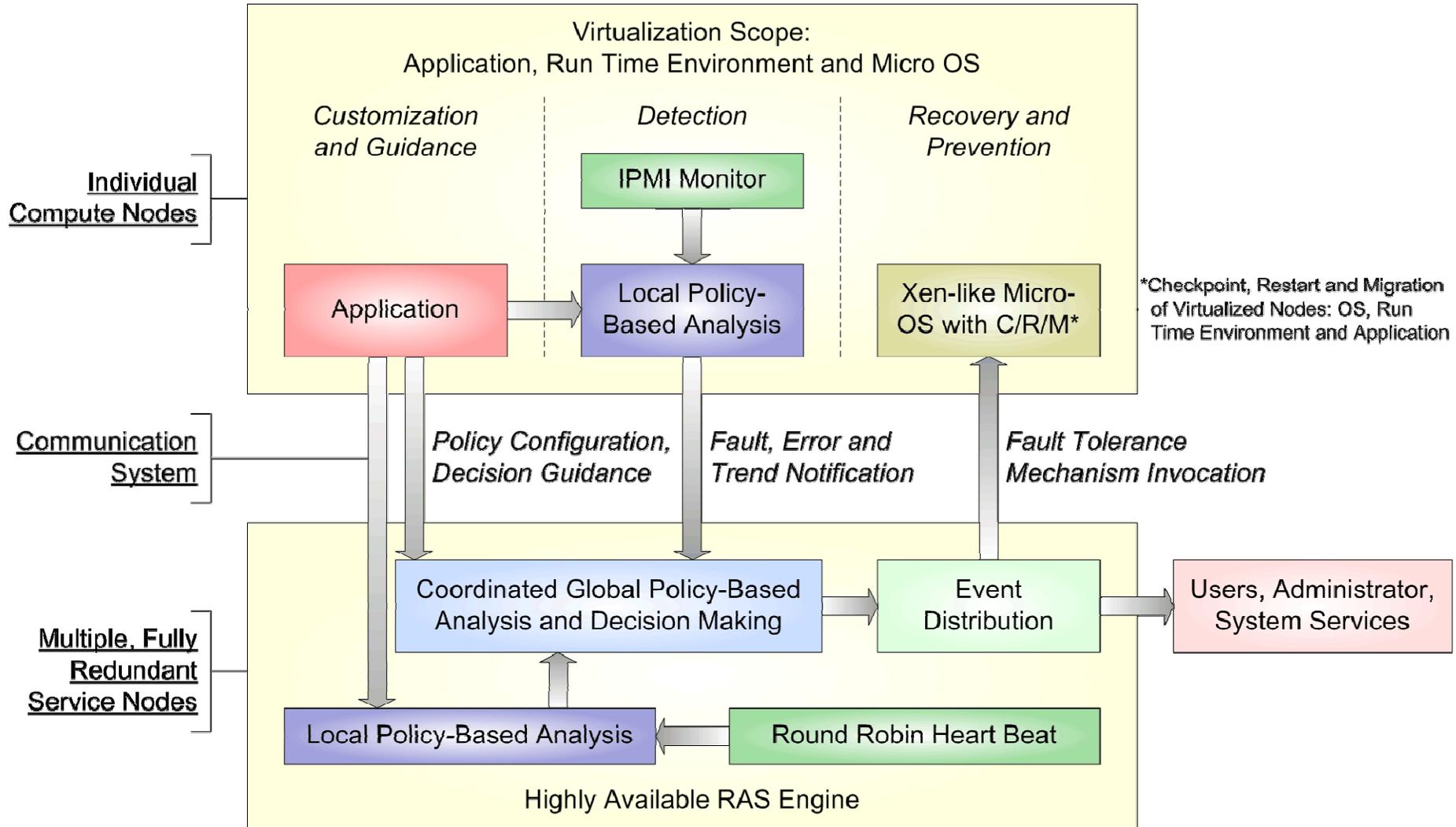
# A holistic resiliency framework concept for high-performance computing



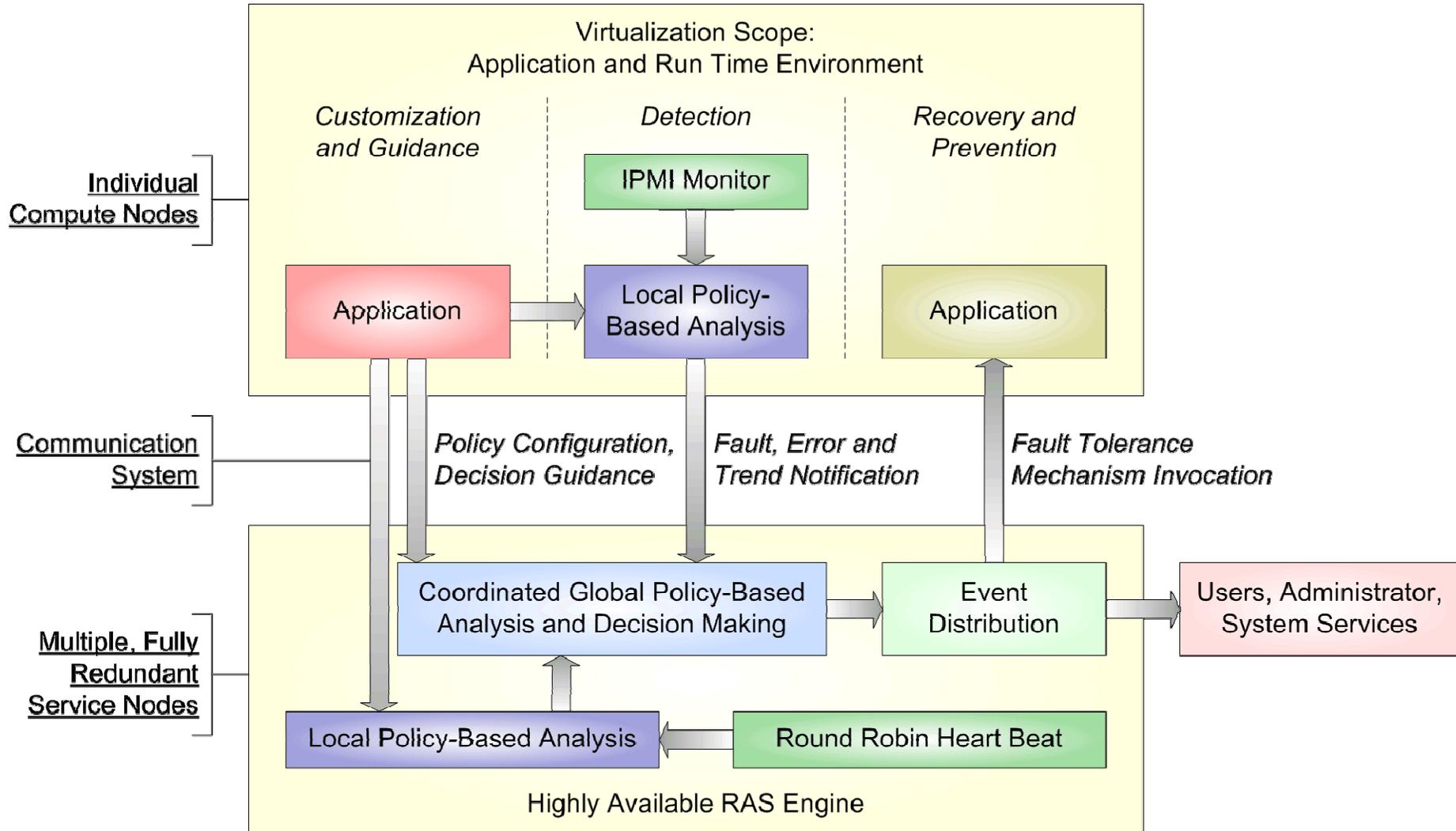
# Example 1: Automatic Runtime Environment Checkpoint/Restart



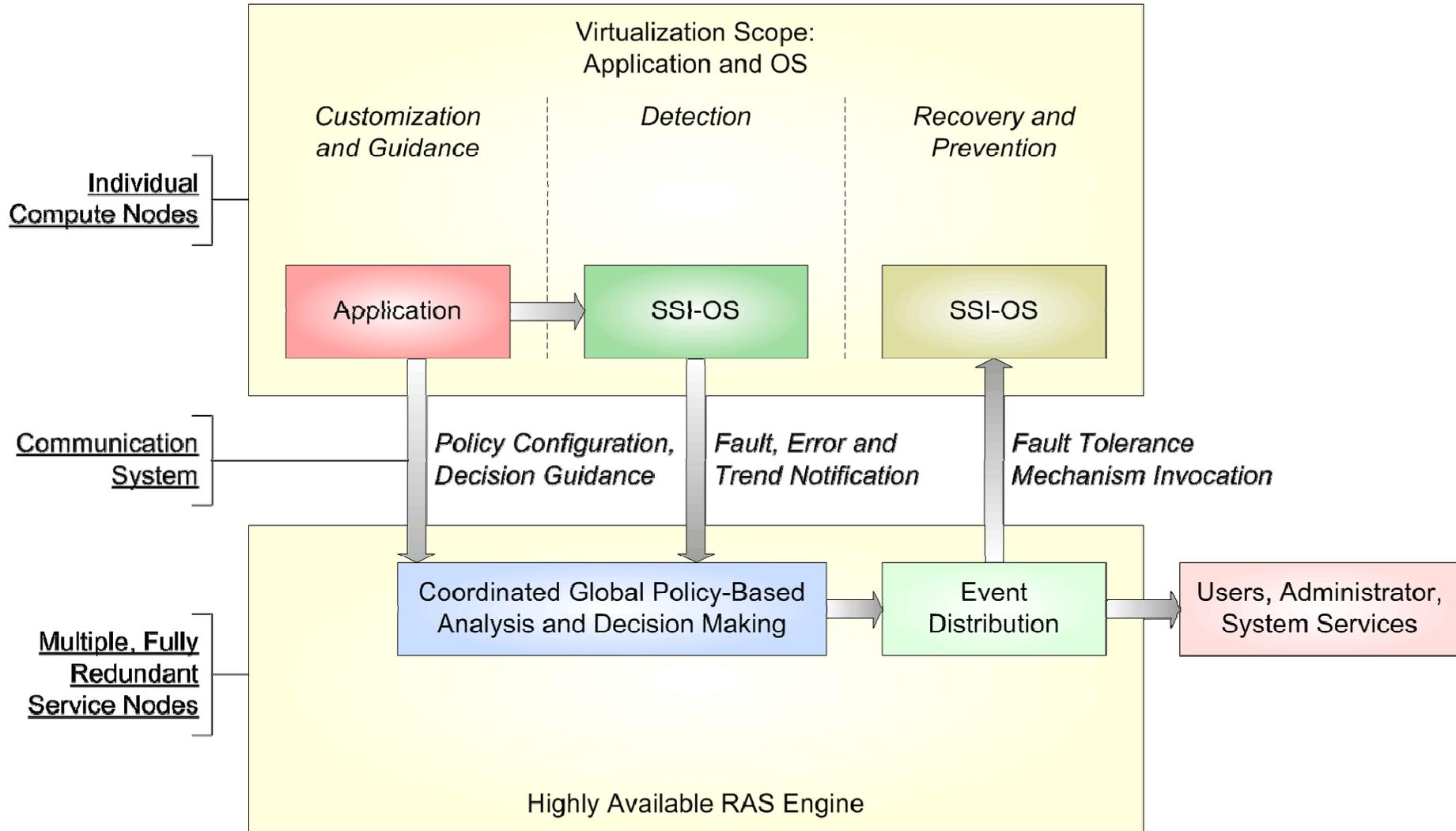
# Example 2: Automatic Virtual Machine Checkpoint/Restart/Migration



# Example 3: Automatic Application-Level Checkpoint/Restart



# Example 4: Automatic OS-Level Checkpoint/Restart/Migration



# Outline

- Resilience deficiencies of high-performance computing architectures
- Research and development goals and projects overview
- Symmetric active/active redundancy for head and service nodes
- Job pause approach for reactive compute node fault tolerance
- Pre-emptive migration for proactive compute node fault tolerance
- Reliability analysis and modeling for fault prediction and anticipation
- Evaluation of compute node fault tolerance policies using simulation
- Vision of a holistic resiliency framework concept
- **Conclusion: Achievements, ongoing work and future plans**

# Achievements

- Developed efficient redundancy strategies for critical infrastructure services in HPC systems
- Added the job pause approach for HPC compute nodes to enhance reactive fault tolerance
- Implemented threshold-based preemptive migration using system-level virtualization for proactive fault tolerance
- Investigated mean-time to failure characteristics of HPC systems using reliability analysis
- Simulated and evaluated various fault tolerance strategies using actual failure data from HPC system logs
  - Reactive fault tolerance only
  - Proactive fault tolerance only
  - Holistic fault tolerance through reactive/proactive combination

# Ongoing work and future plans

- Finishing development on threshold-based process-level preemptive migration for proactive fault tolerance
- Engaging in prediction-based preemptive migration using for proactive fault tolerance
- Investigating failure mode, failure detection, and failure distribution characteristics of HPC systems using reliability analysis
- Plan to further enhance reactive fault tolerance with checkpoint placement adaptation to actual and predicted system health threats
- Plan to develop a holistic fault tolerance framework that offers the mix-and-match approach for various fault resilience strategies

# Acknowledgements

- **Investigators at Oak Ridge National Laboratory:**
  - *Stephen L. Scott*, Christian Engelmann, Hong H. Ong, Geoffroy R. Vallée, Thomas Naughton, Anand Tikotekar, George Ostrouchov
- **Investigators at Louisiana Tech University:**
  - *Chokchai (Box) Leangsuksun*, Nichamon Naksinehaboon, Raja Nassar, Mihaela Paun
- **Investigators at North Carolina State University:**
  - *Frank Mueller*, Chao Wang, Arun Nagarajan, Jyothish Varma
- **Investigators at Tennessee Technological University:**
  - *Xubin (Ben) He*, Li Ou, Xin Chen
- **Funding sources:**
  - U.S. Department of Energy, U.S. National Science Foundation, Oak Ridge National Laboratory, Tennessee Technological University



The University of Reading



# Contacts

- **Christian Engelmann**

System Research Team  
Computer Science Research Group  
Computer Science and Mathematics Division  
Oak Ridge National Laboratory

engelmannc@ornl.gov

- **Stephen L. Scott**

System Research Team  
Computer Science Research Group  
Computer Science and Mathematics Division  
Oak Ridge National Laboratory

scottsl@ornl.gov

- **[www.fastos.org/ras](http://www.fastos.org/ras) | [www.fastos.org/molar](http://www.fastos.org/molar)**

- **[www.csm.ornl.gov/srt](http://www.csm.ornl.gov/srt)**