# Fault Tolerance Support for Supercomputers with Multicore Nodes

Esteban Meneses
Xiang Ni


**Exascale Supercomputer:**
100 M of cores

*"an Exascale system could be expected to have a failure ... every 35–39 minutes"*
**Exascale Computing Study**

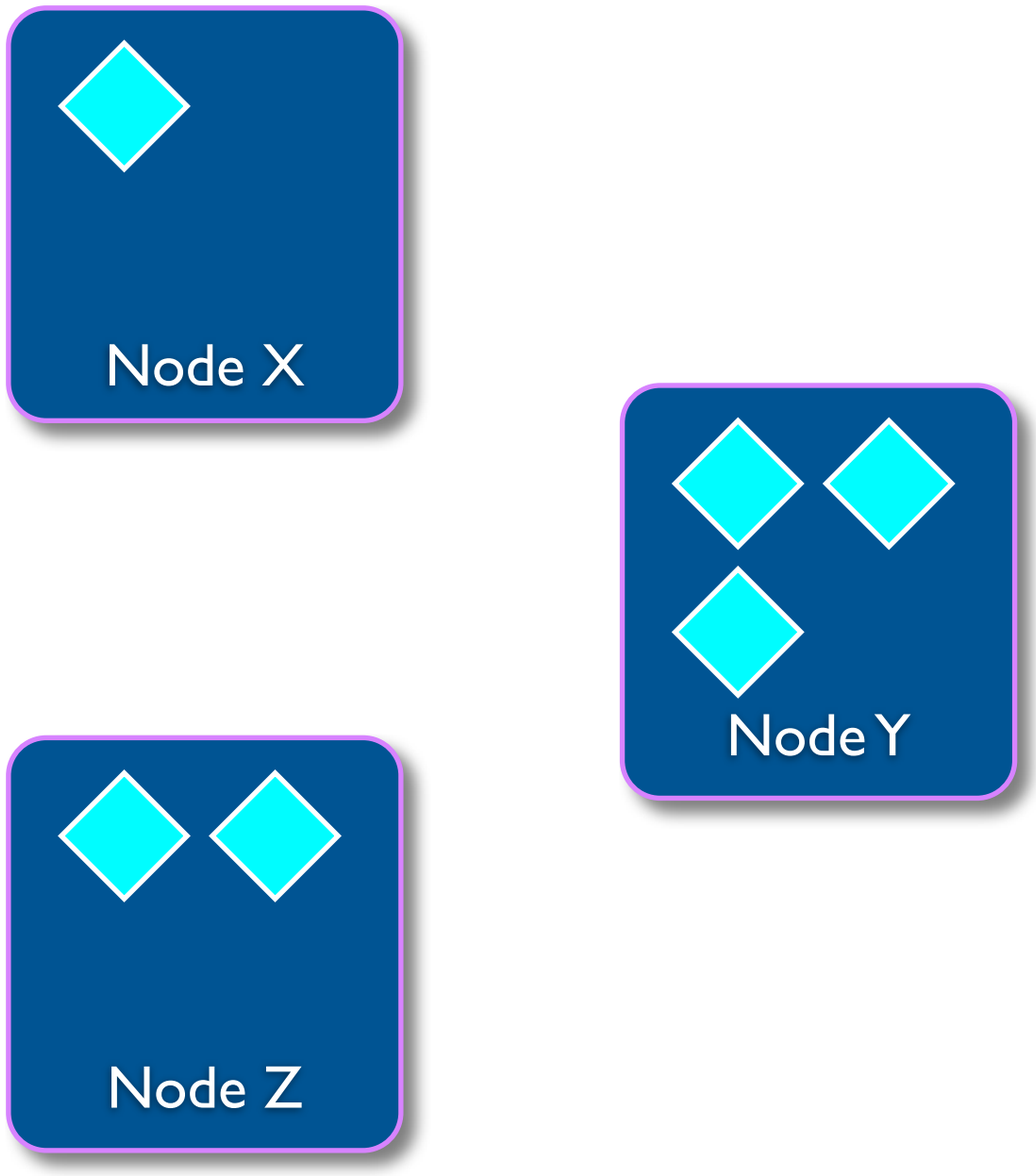
*"insufficient resilience of the software infrastructure would likely render extreme scale systems effectively unusable"*
**The International Exascale Software**

# Contents

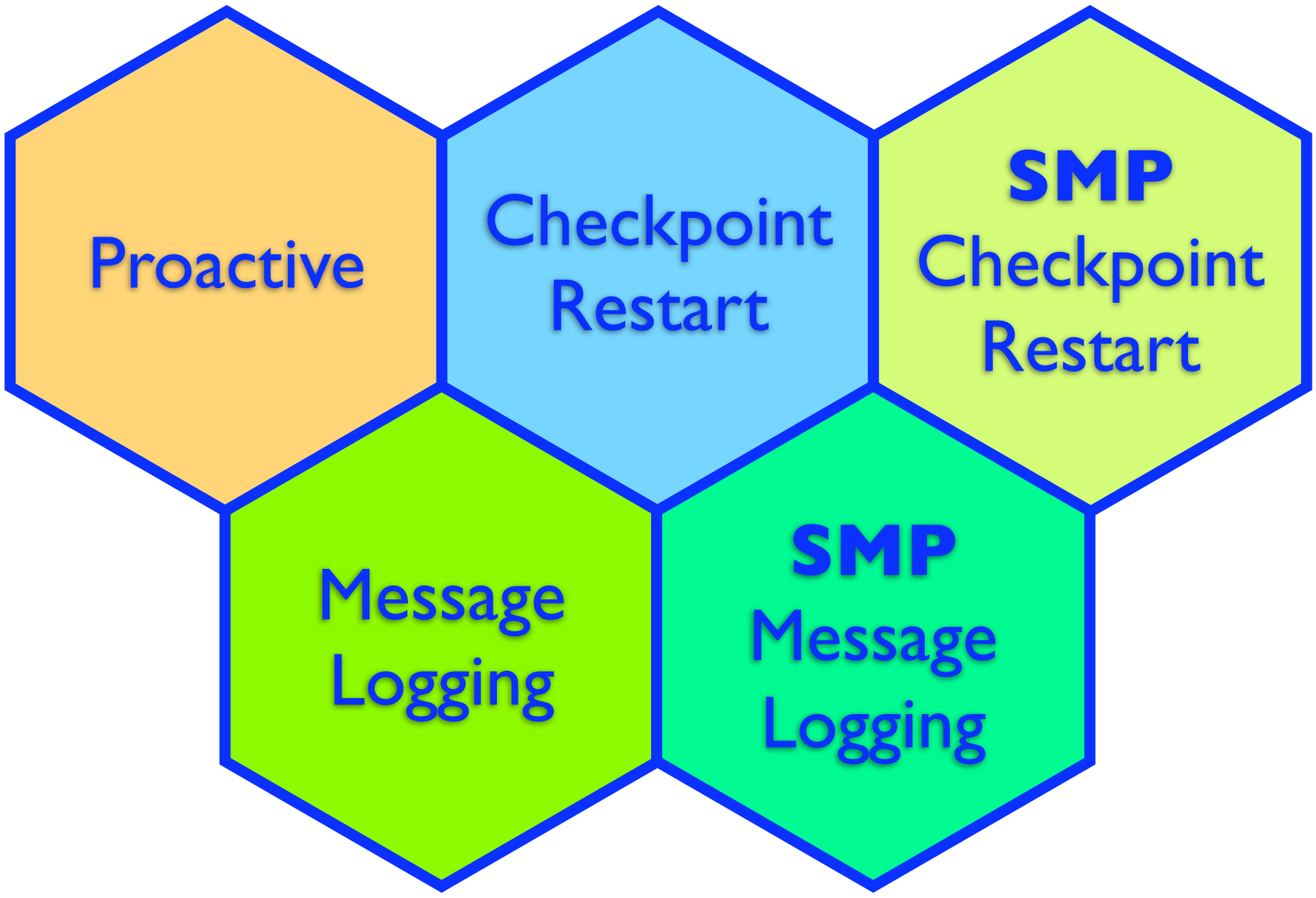
- Charm++ Fault Tolerance Infrastructure.
- Fault Tolerance in SMP.
- Preliminary Results.
- Multiple Concurrent Failure Model.
- Future Work.

# Fault Tolerance in Charm++

- Object Migration
- Load Balancing
- Runtime Support
- SMP version



# Strategies


Proactive
Checkpoint
Restart
SMP
Checkpoint
Restart
Message
Logging
SMP
Message
Logging

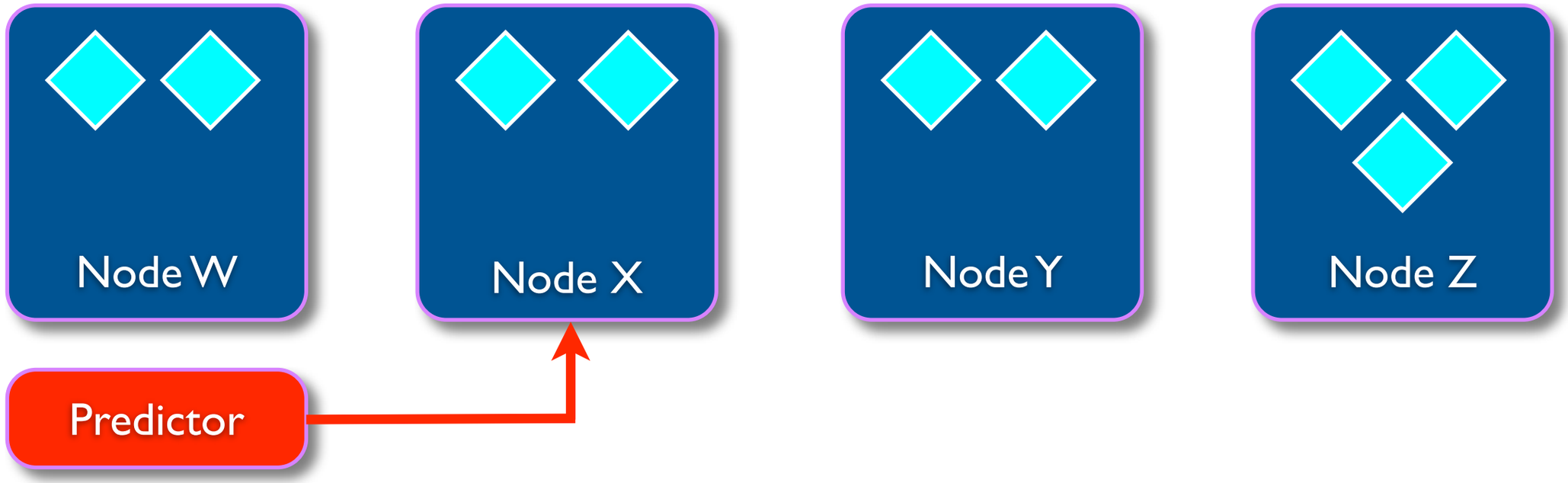
Proactive

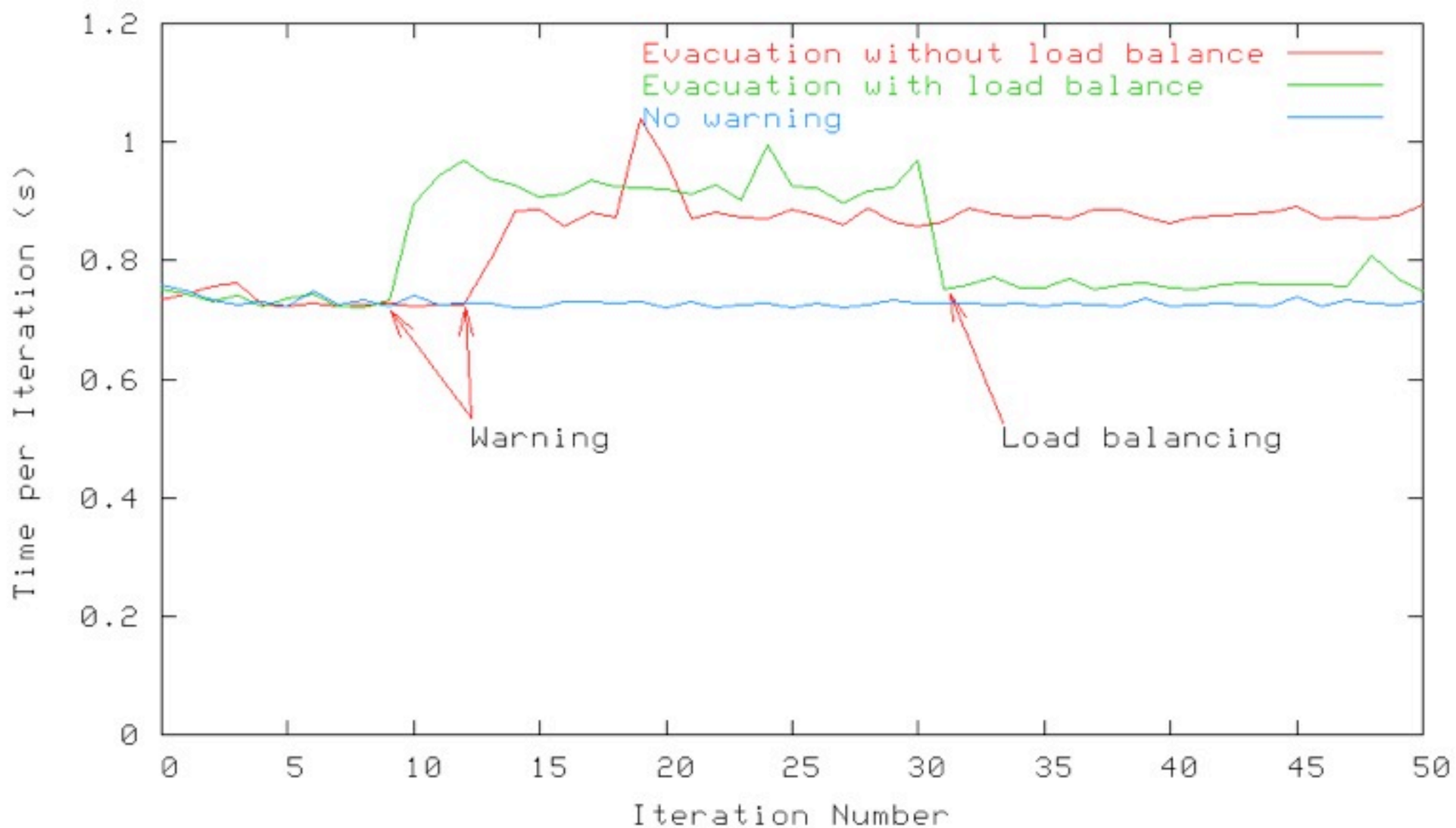
Checkpoint Restart

Message Logging

**SMP** Checkpoint Restart

**SMP** Message Logging



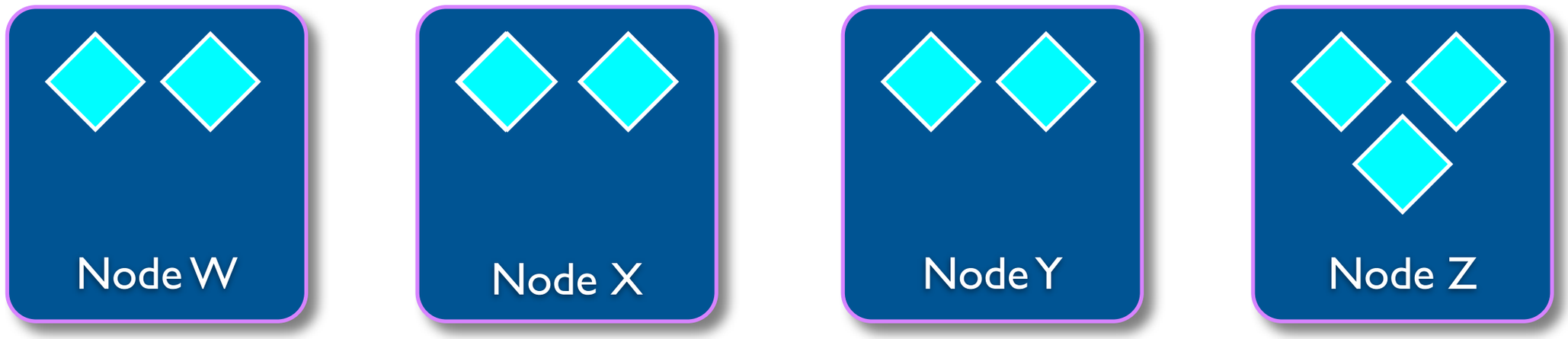
Proactive

Checkpoint Restart

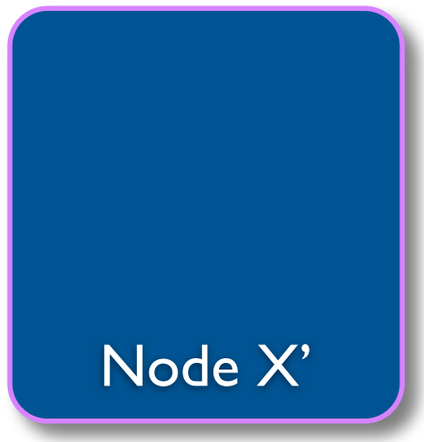
Message Logging

SMP Checkpoint Restart

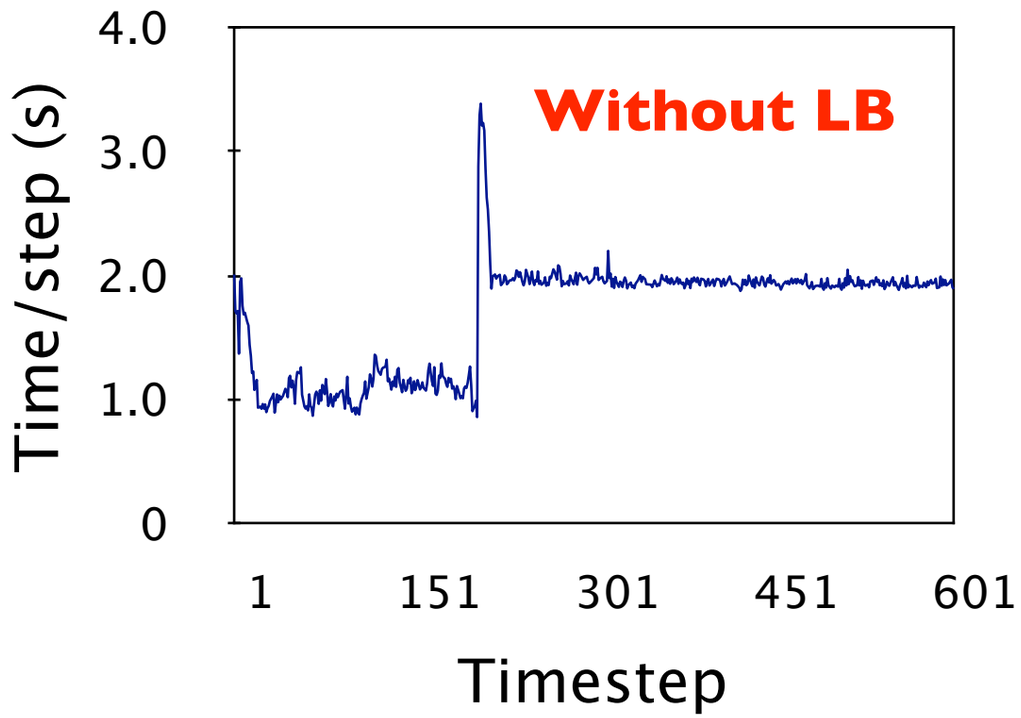
SMP Message Logging

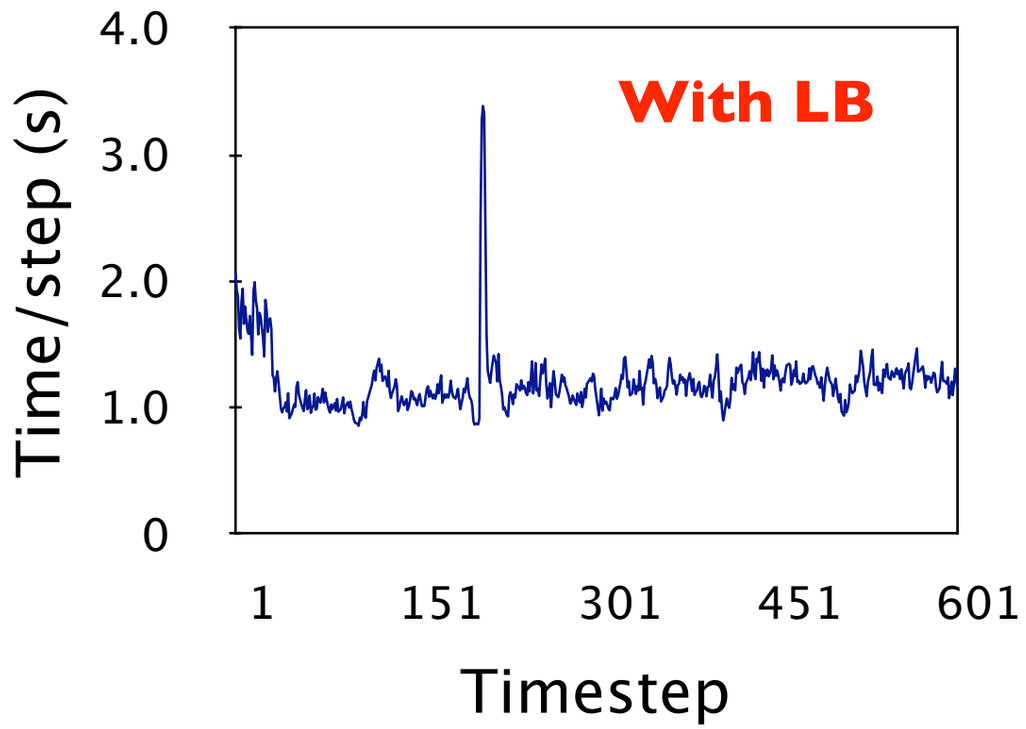



Checkpoint in buddy's memory




Restart with or without spare nodes



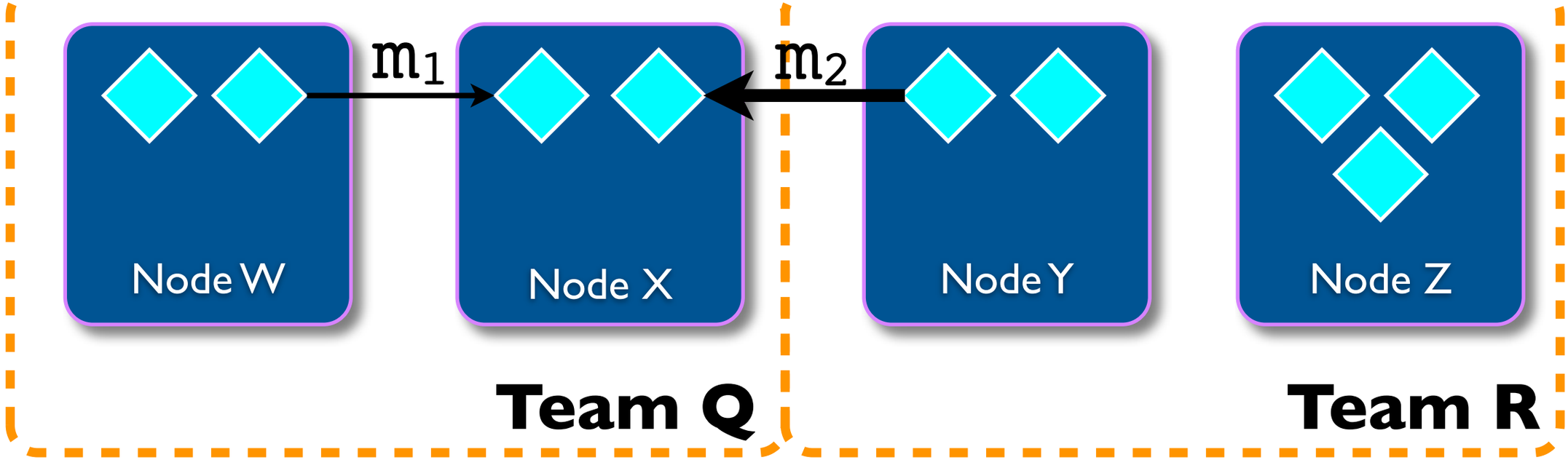
Proactive

Checkpoint Restart

Message Logging

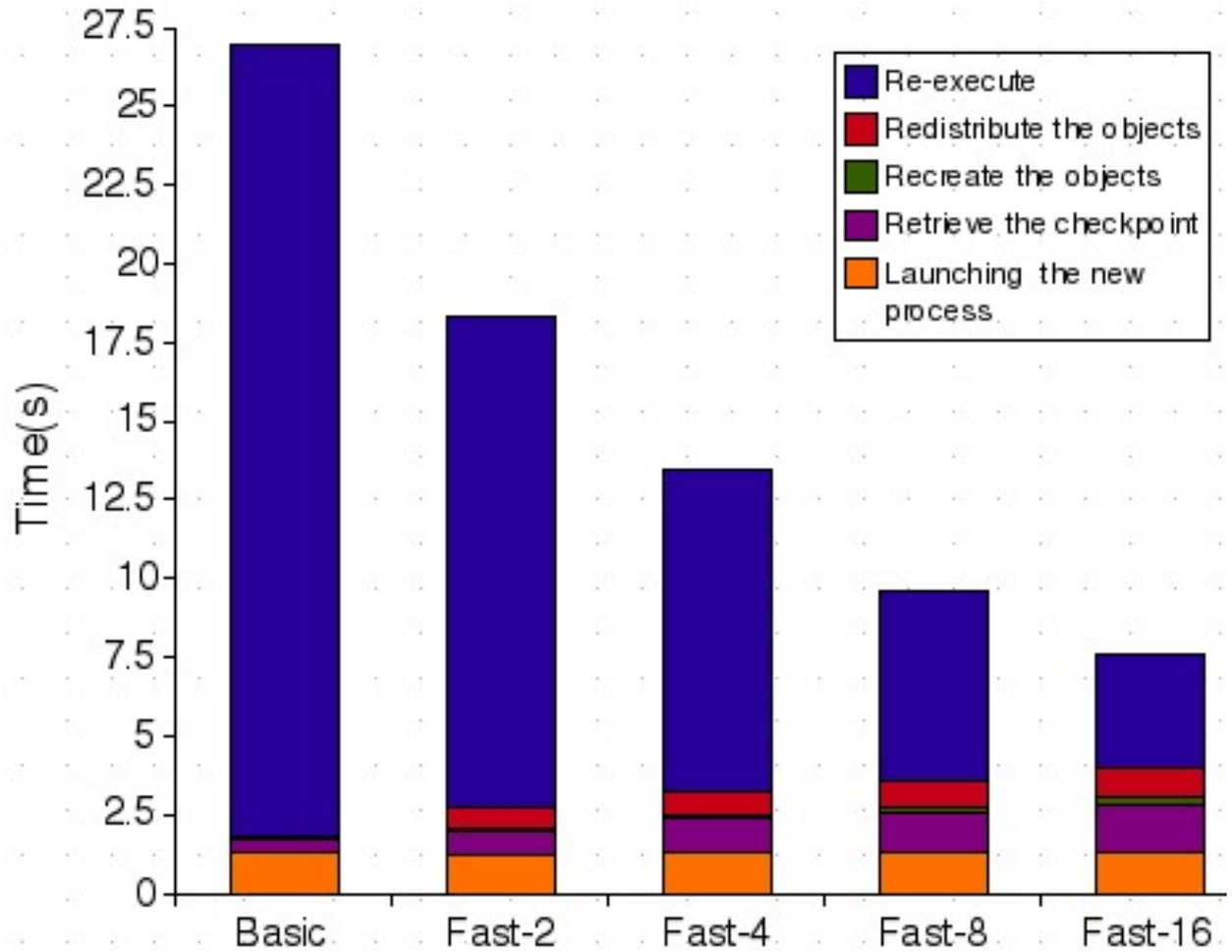
SMP Checkpoint Restart

SMP Message Logging




**Team-based Message Logging**

**Parallel Restart**



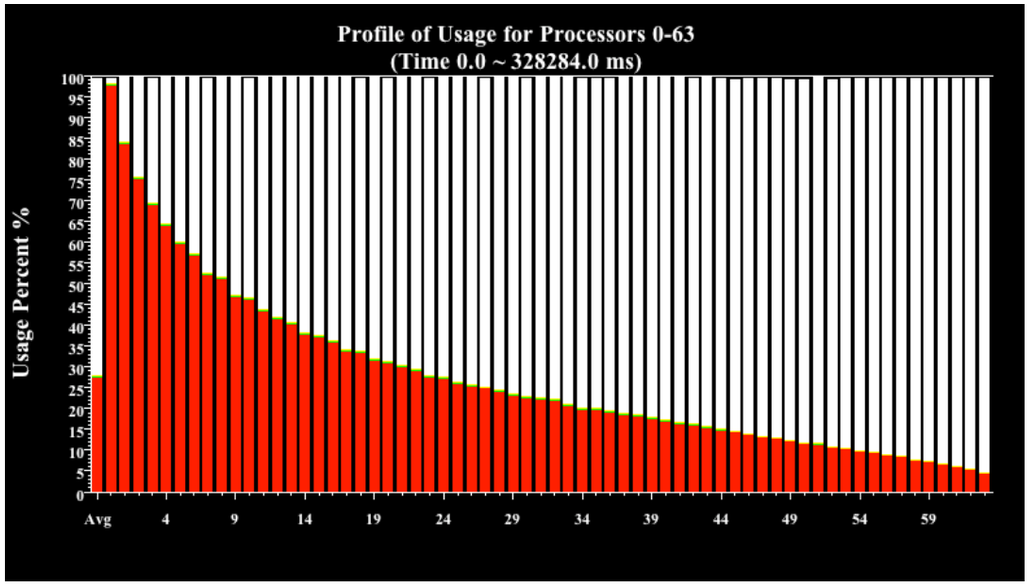
Proactive

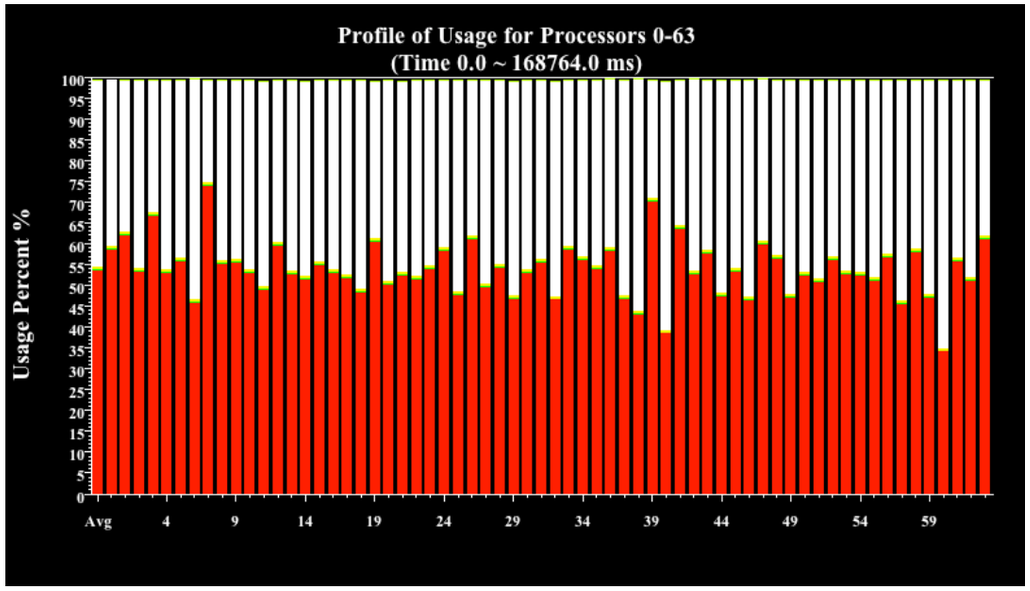
Checkpoint Restart

Message Logging

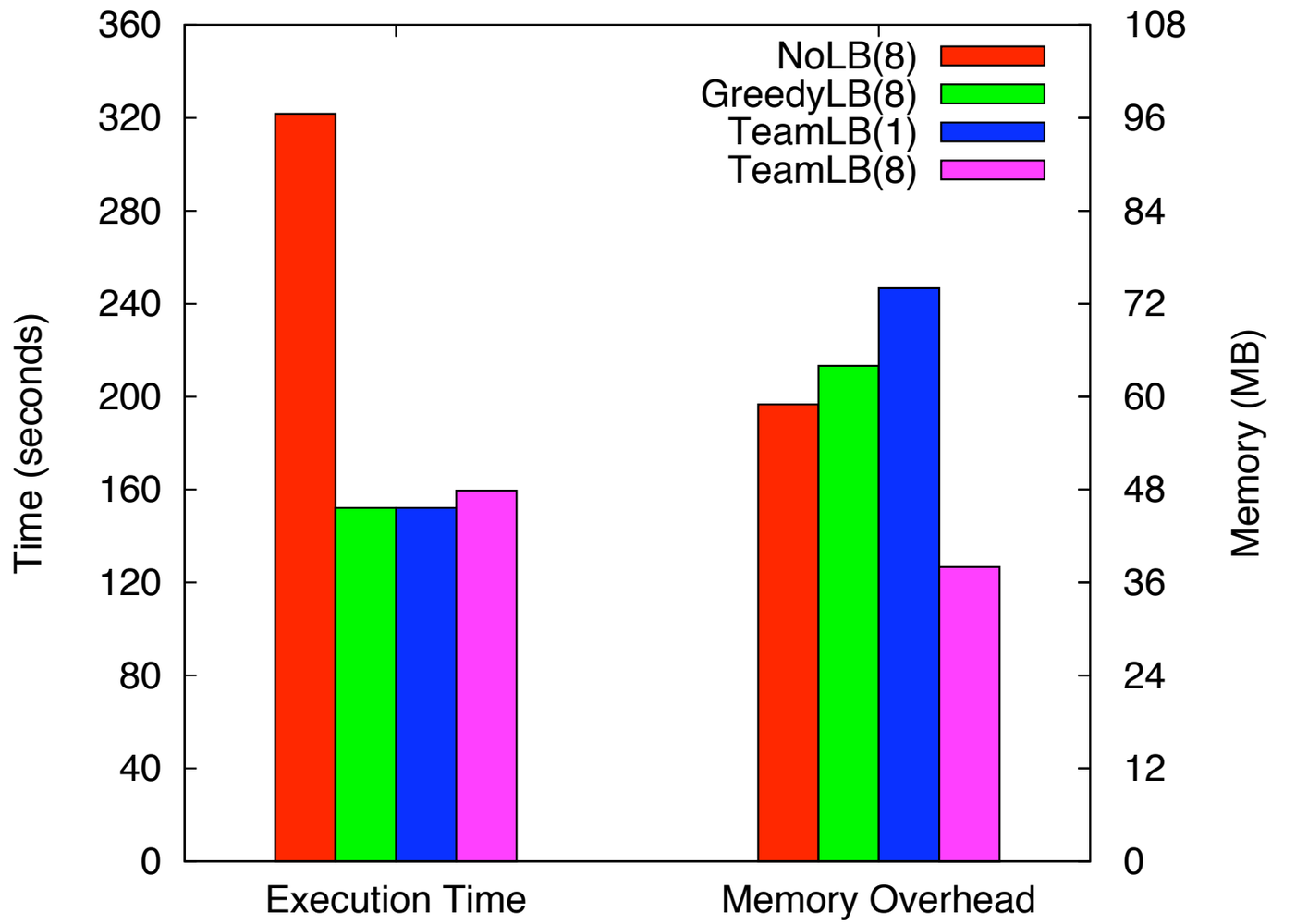
SMP Checkpoint Restart

SMP Message Logging




**Team-based Load Balancer**



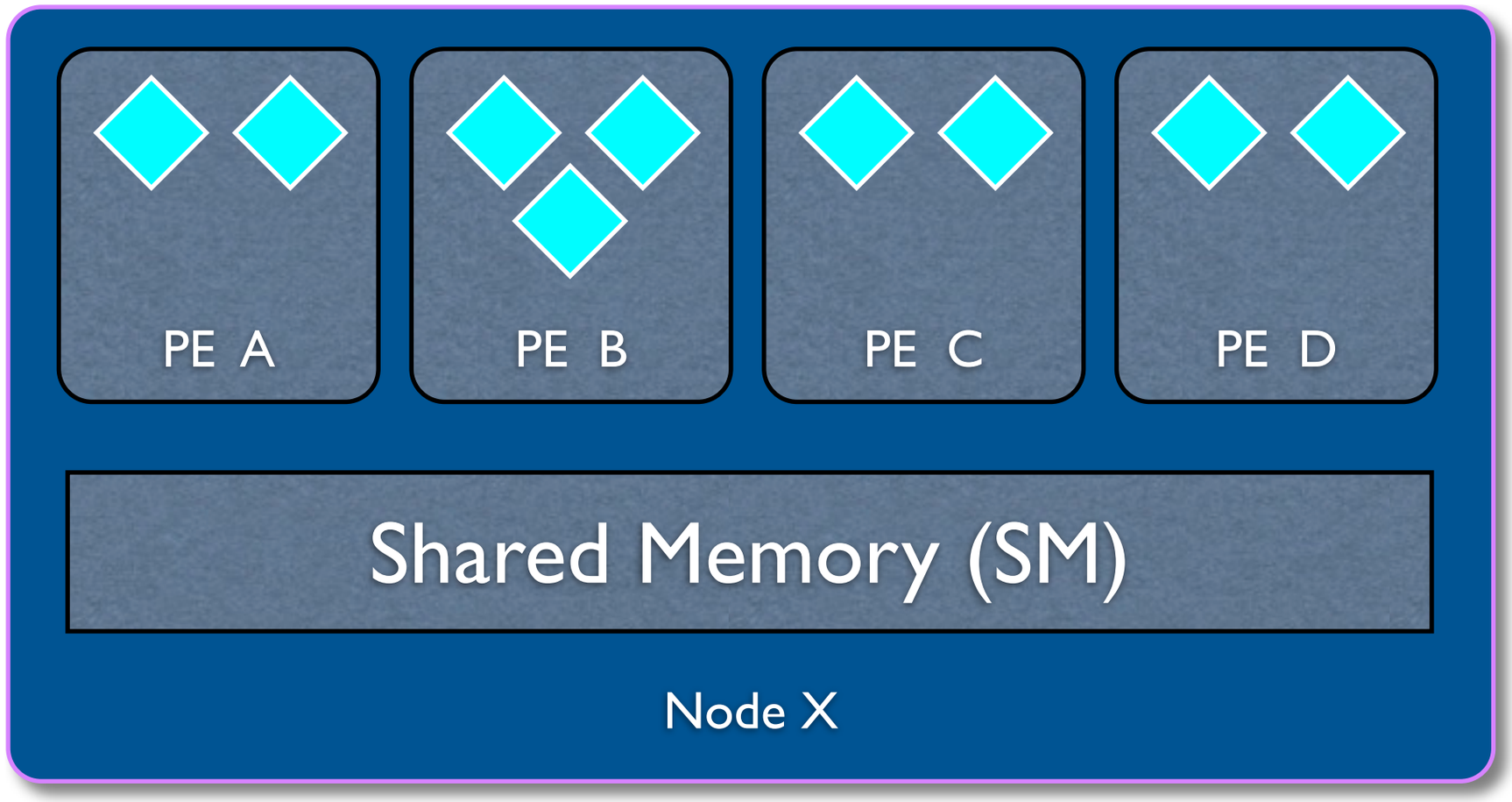
Proactive

Checkpoint Restart

Message Logging

**SMP** Checkpoint Restart

**SMP** Message Logging




The minimum unit of failure is a **node**

**Single** node failure support

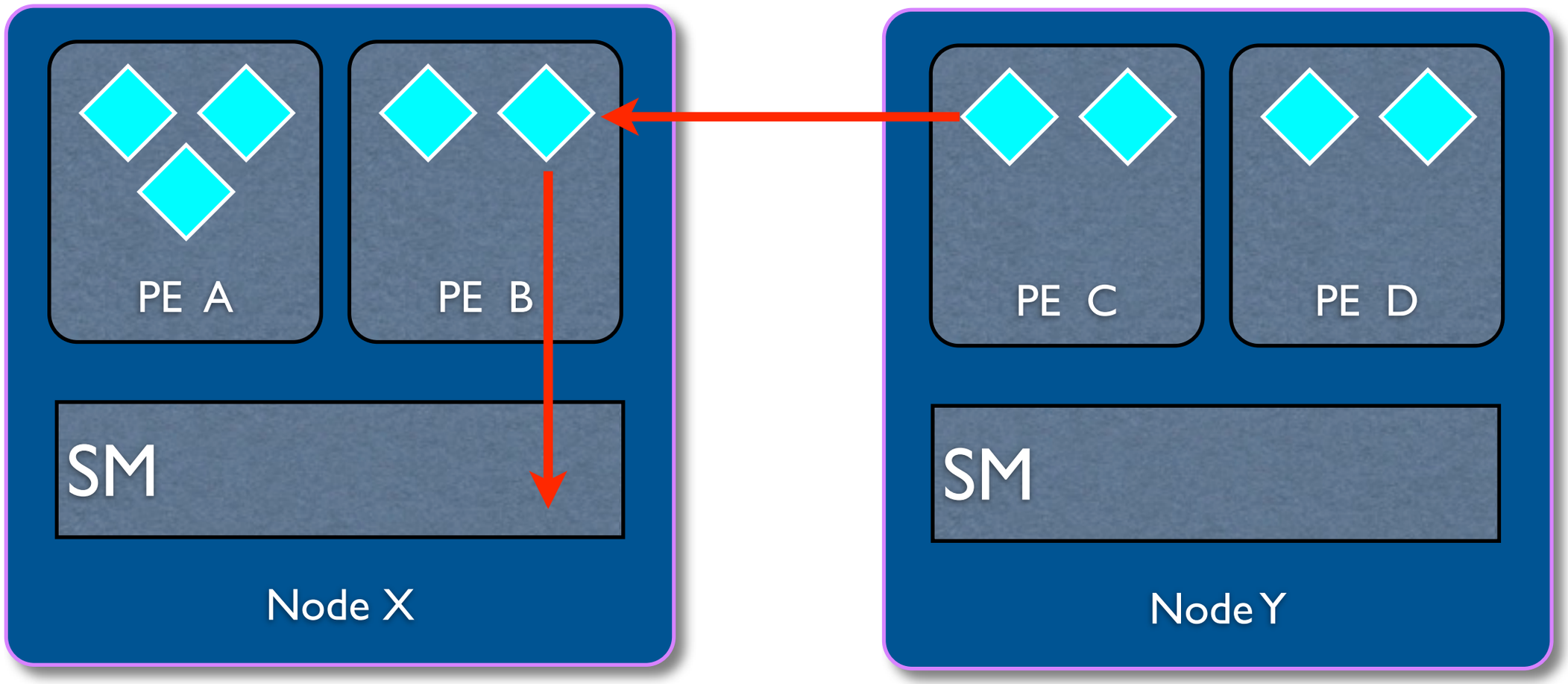
Proactive

Checkpoint Restart

Message Logging

**SMP** Checkpoint Restart

**SMP** Message Logging




Causal Message Logging → **determinants** in shared memory

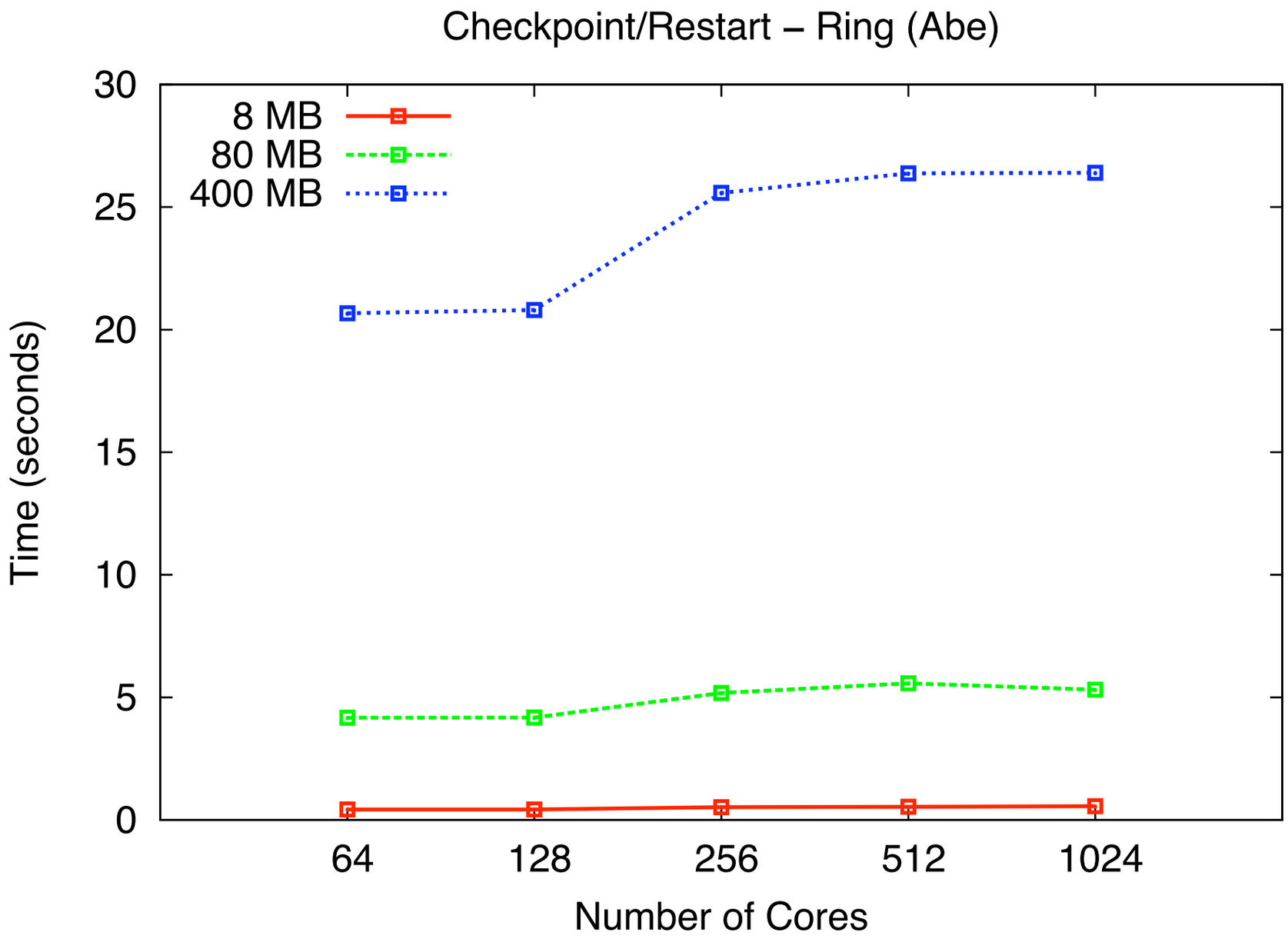
**Lock contention** → hybrid scheme

**Load balancing** → increase communication inside a node

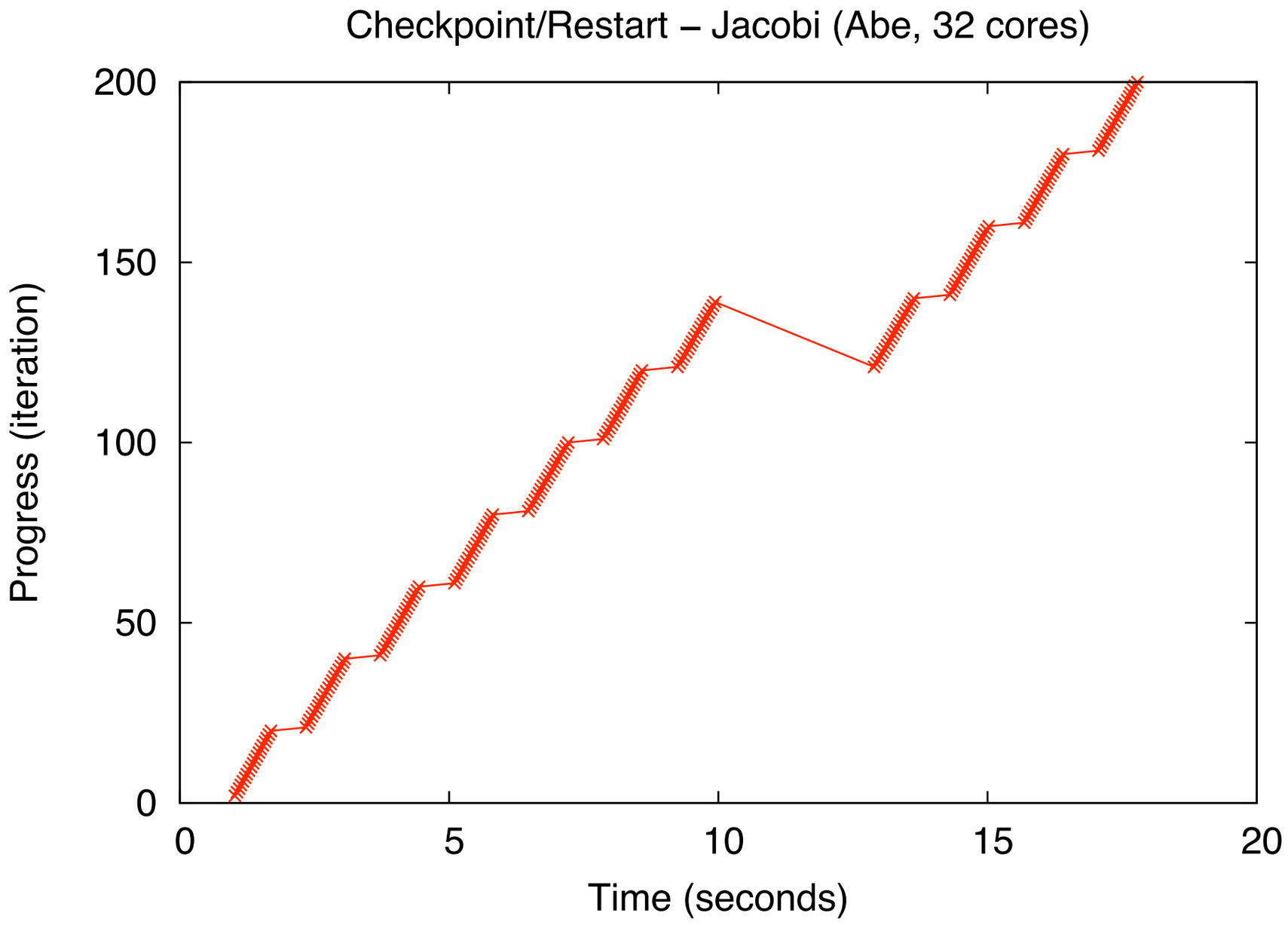
# Experiments

- *Hardware*:
  - **Abe**@NCSA: 1200 8-way SMP nodes.
  - **Ranger**@TACC: 3936 16-way SMP nodes.
- *Benchmarks*:
  - **Ring**: Charm++ nearest neighbor exchange.
  - **Jacobi**: 7-point stencil.

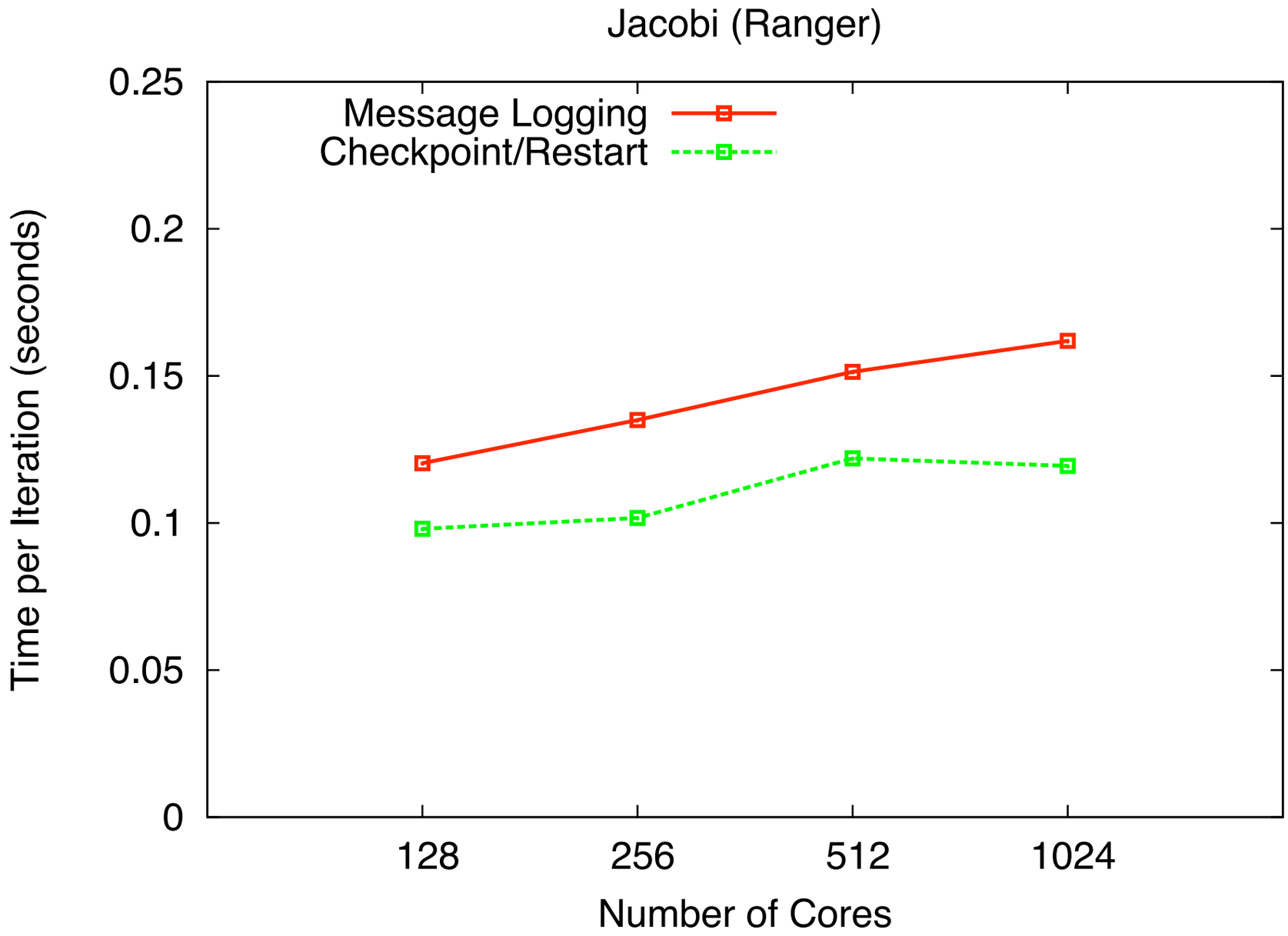
# Checkpoint Time



# Restart Time



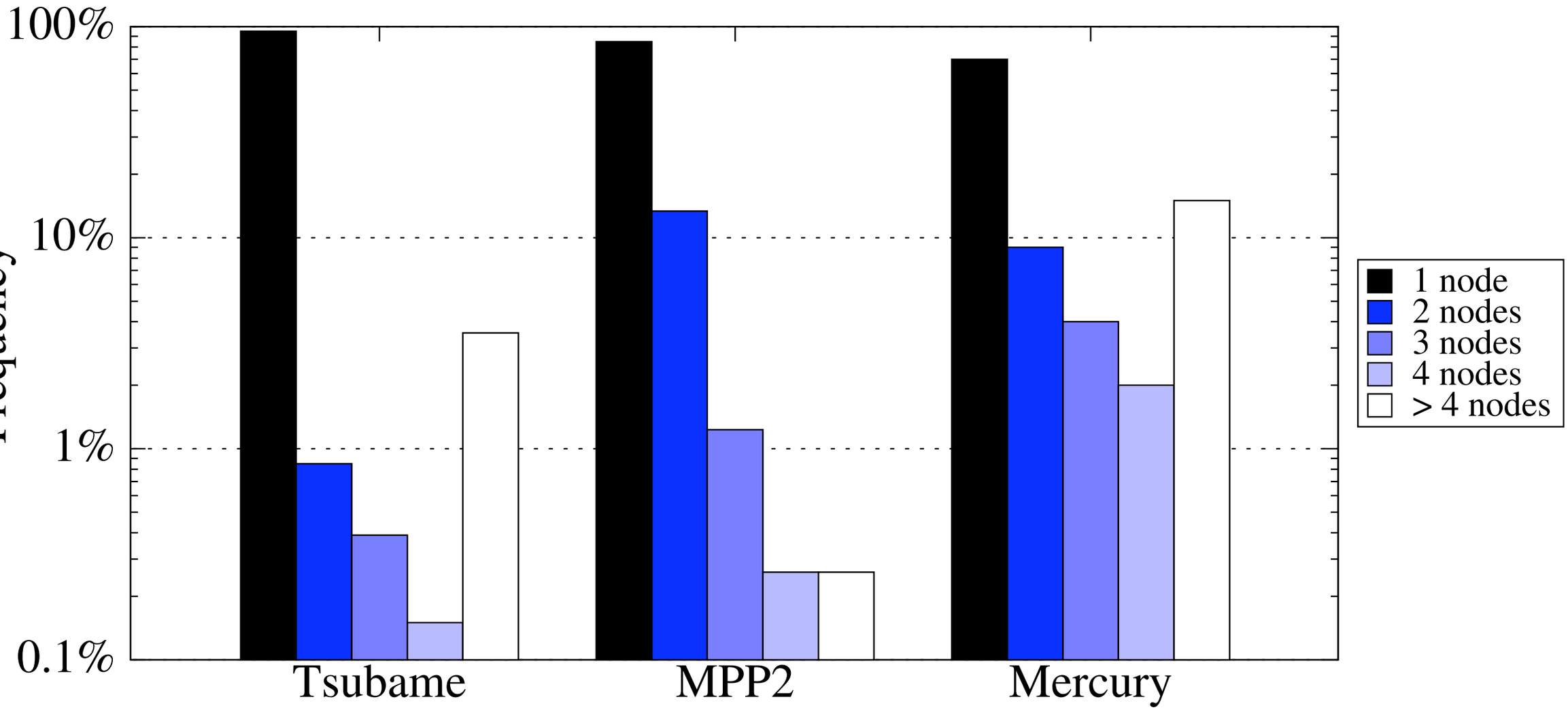
# Message Logging Overhead



# Single Node Failure

- All protocols presented tolerate a single node failure.
- They *may* recover from a multiple failure.
- Multiple concurrent failures are rare.
- Cost to tolerate them is high:
  - Checkpoint/restart: more checkpoint buddies.
  - Causal Message logging: determinants must be stored in more locations.

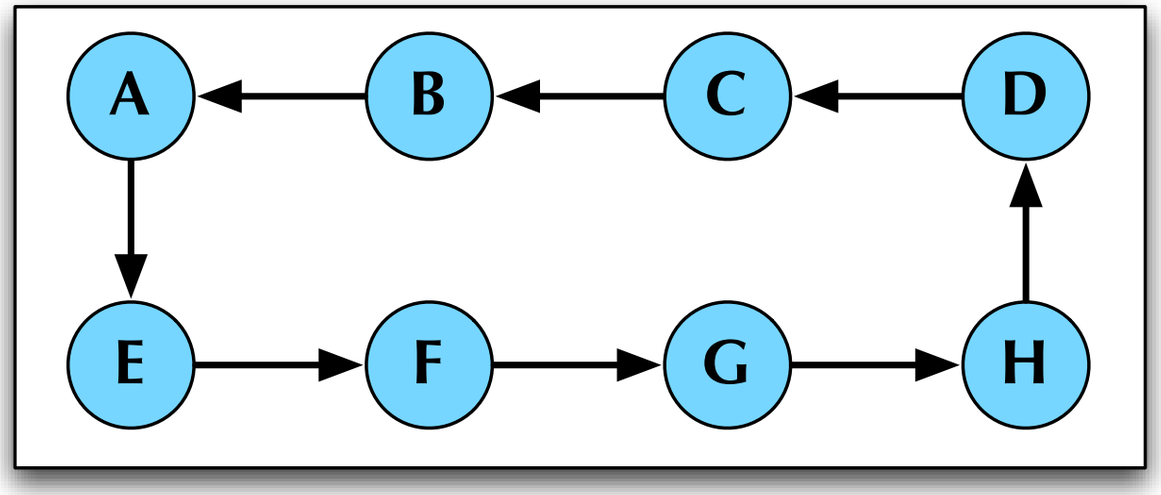
# Distribution of Multiple Failures



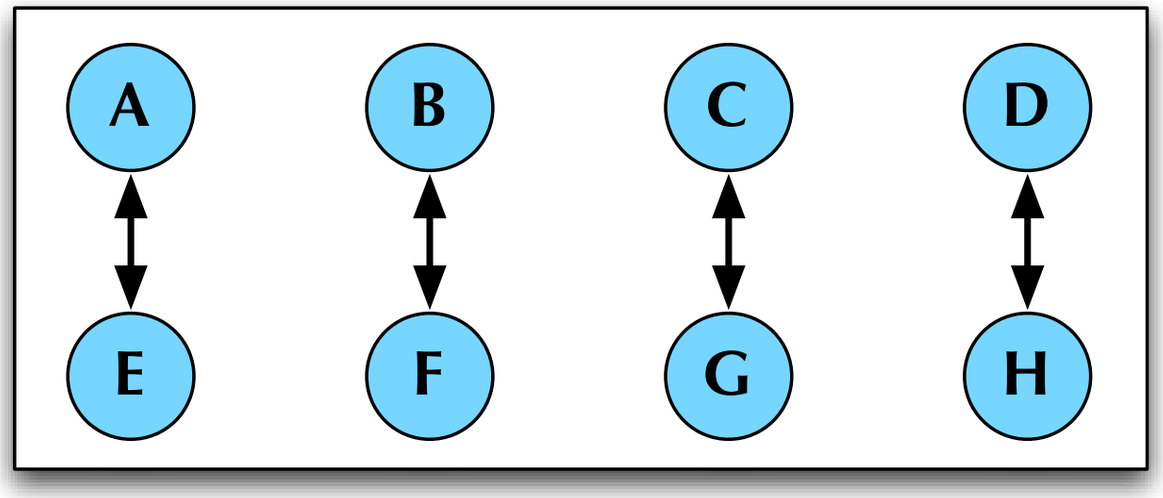
# Multiple Concurrent Failures

- Analytical Model:
  - **Multiple Failure Distribution**: (heavy-tailed).
  - **Checkpoint/Restart**: probability of losing both a node and its buddy.
  - **Message Logging**: probability of losing a node and another node it contacts.

# Buddy Assignment

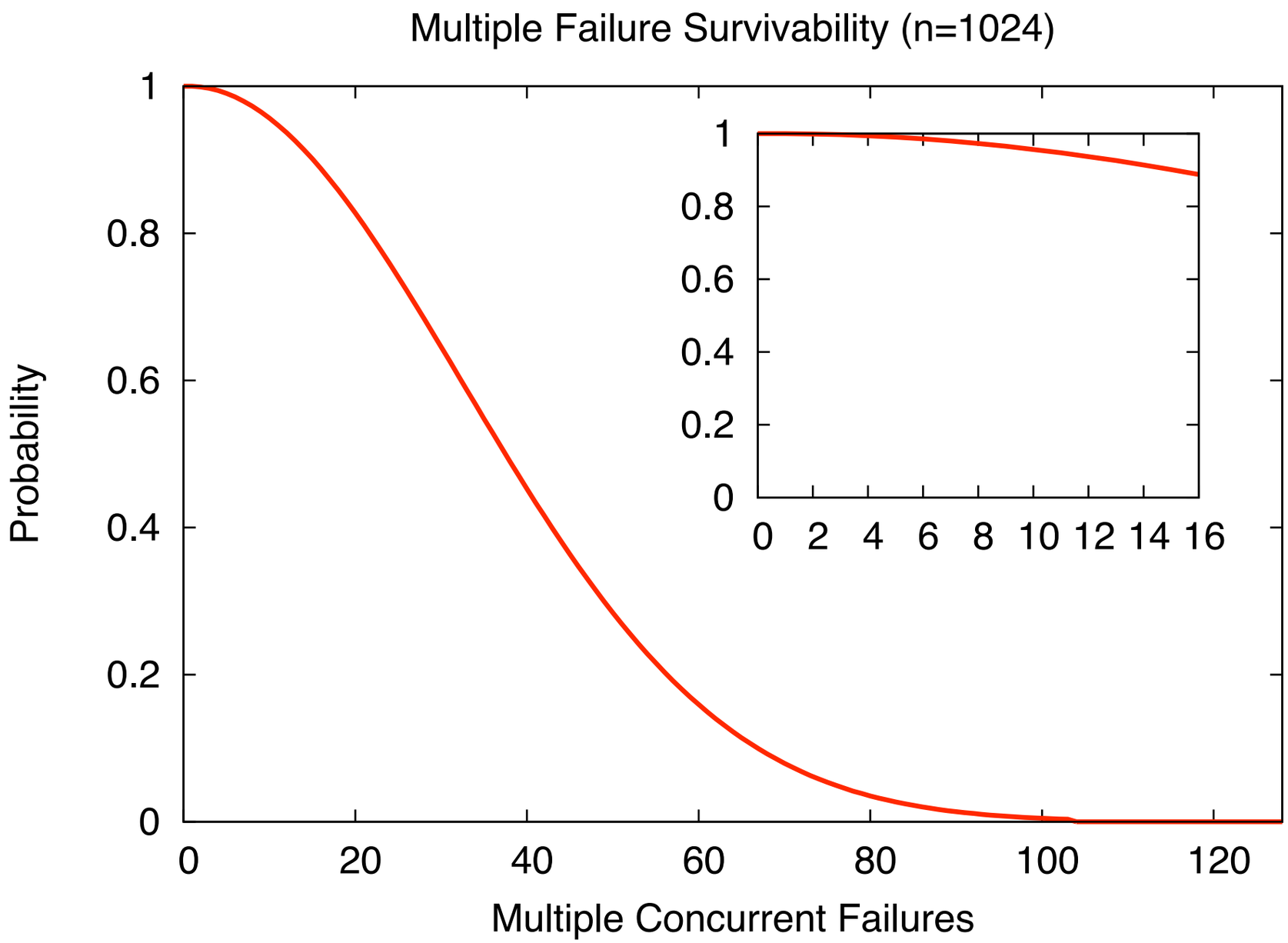



Ring Mapping

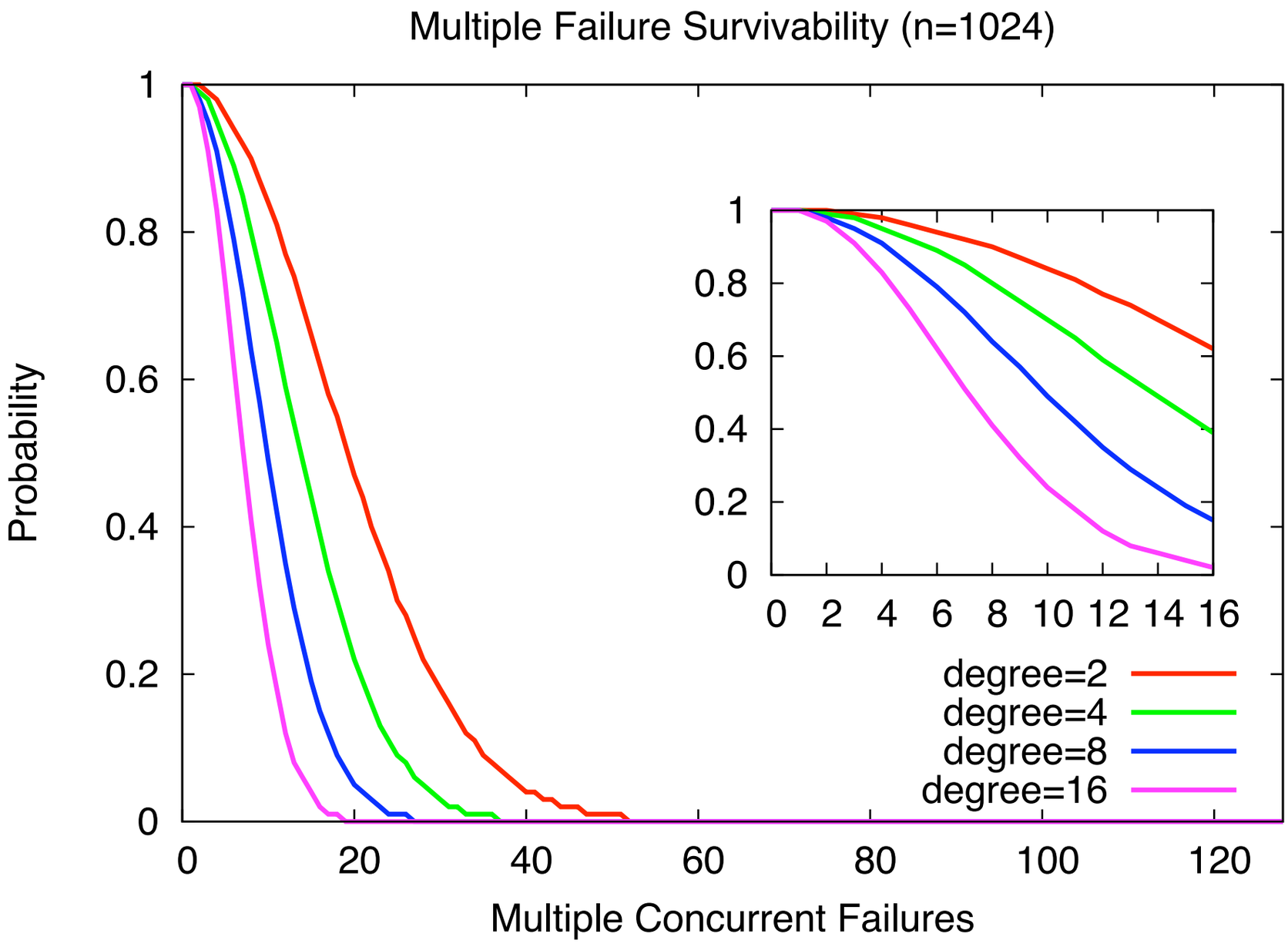



Pair Mapping

# Checkpoint/Restart



# Message Logging



# Conclusions

- Fault Tolerance for SMP better matches the failure reality of supercomputers.
- Single node failure support is robust enough for failure pattern in supercomputers.
- Load balancer is key to enhance fault tolerance in SMP.

# Future Work

- Optimize message logging in SMP.
- Add load balancer to reduce communication overhead.
- Early stages of supercomputer: *correlated* failures.

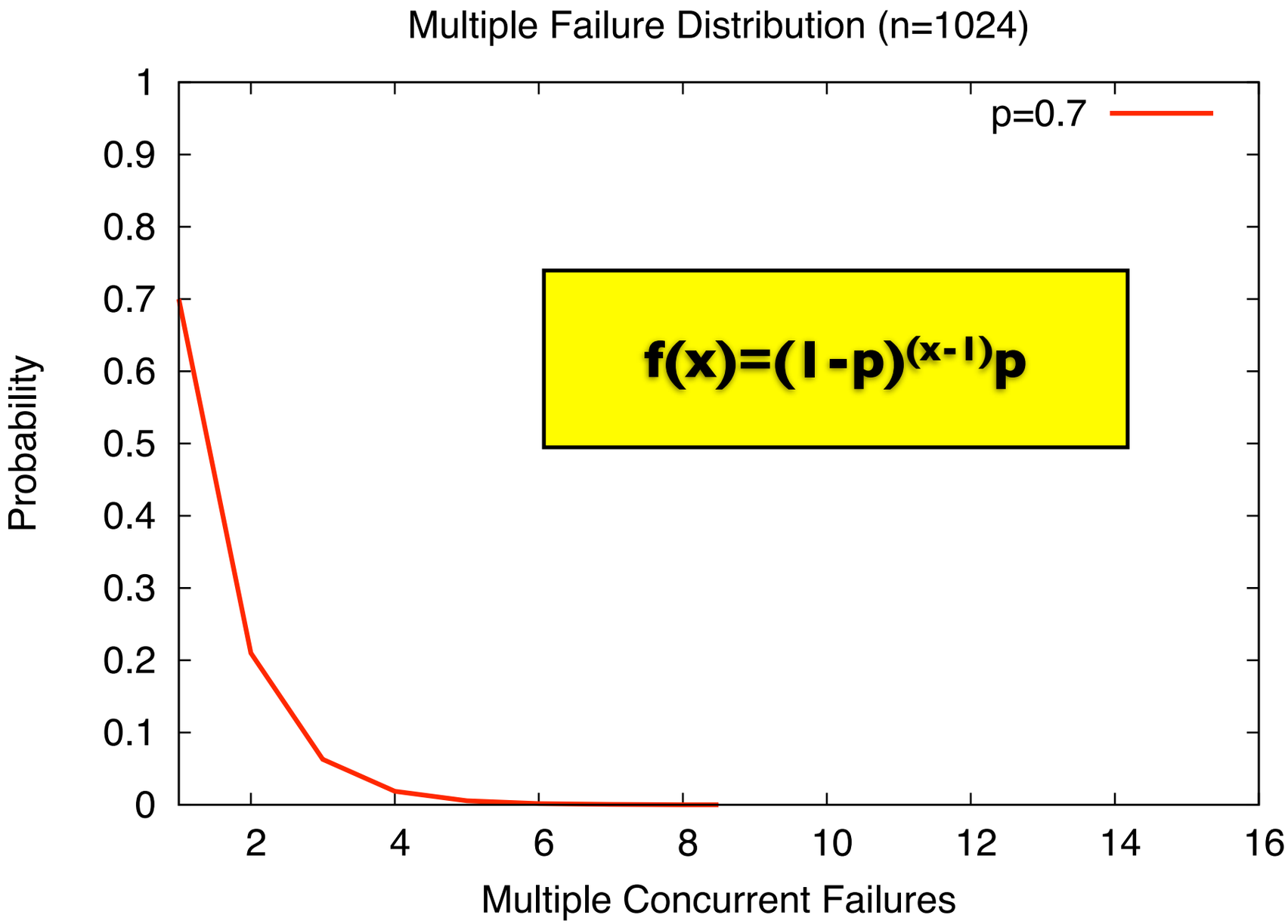
# Aknowledgments


- Ana Gainaru (NCSA).
- Leonardo Bautista Gómez (Tokyo Tech).
- This research was supported in part by the US Department of Energy under grant DOE DE-SC0001845 and by a machine allocation on the Teragrid under award ASC050039N.

# Thanks!

# Q&A

# Multiple Failures Model



# Survivability

| | S |
|---|---|
| Checkpoint/Restart | 0.999402 |
| Message Logging (2) | 0.997624 |
| Message Logging (4) | 0.995285 |
| Message Logging (8) | 0.990716 |
| Message Logging (16) | 0.981973 |