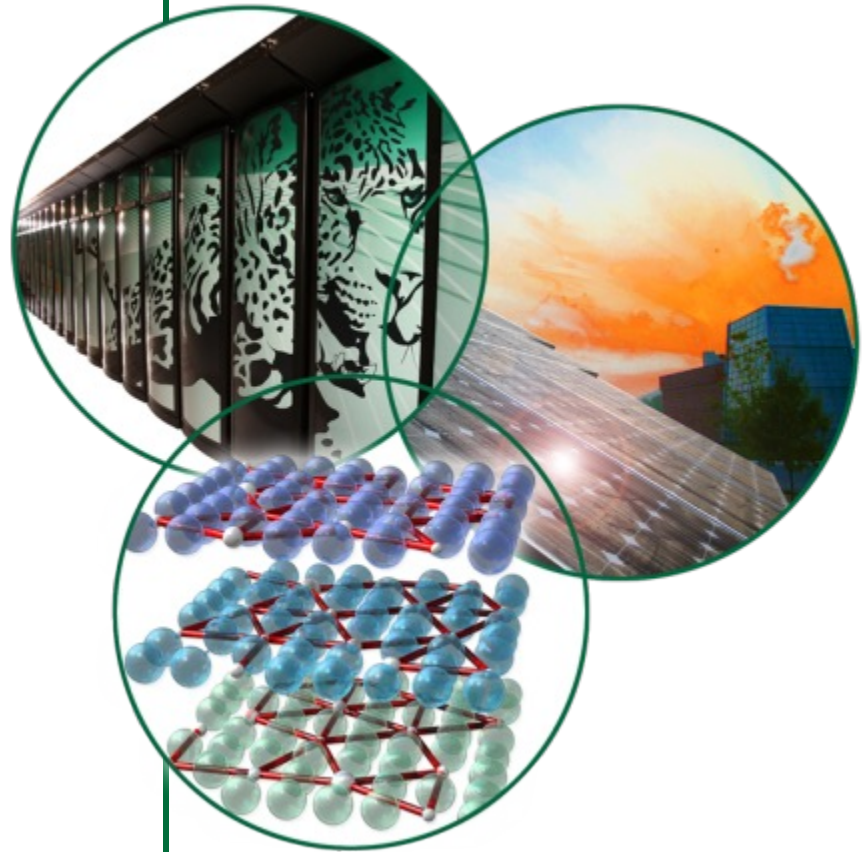


Resilience Challenges and Solutions for Extreme-Scale Supercomputing

Christian Engelmann

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

SOS Workshop, Park City, UT, 3/2-5, 2015.



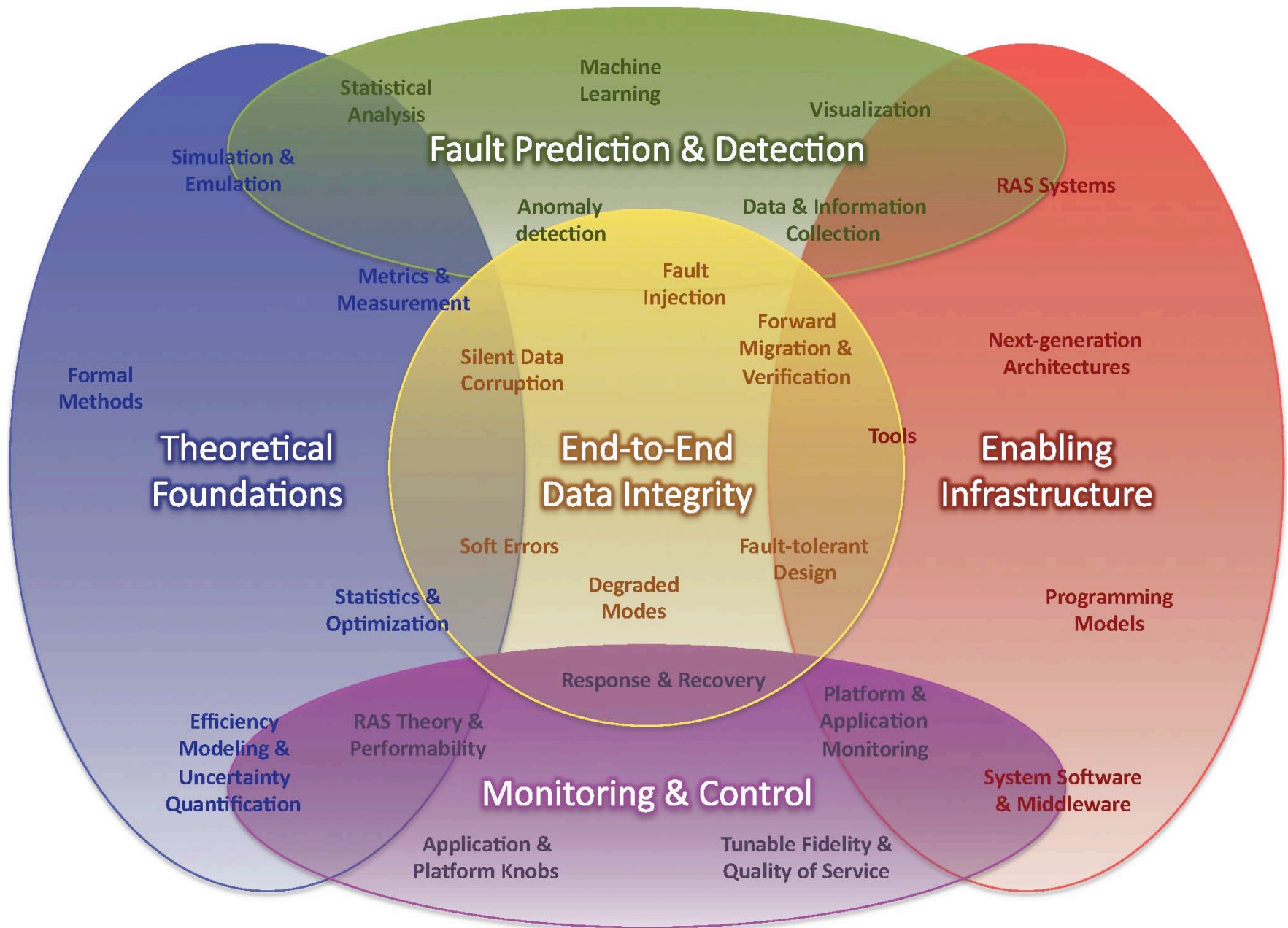
Why is Resilience a Challenge?

- Smaller and smaller process technology
 - Unknown aging effects
 - Increased variability
 - Increased soft error vulnerability
- Near-threshold voltage to achieve energy efficiency
 - Increased soft error vulnerability
 - Decreased noise immunity
- System MTTI decreases as component count increases
 - Can not offset component growth with reliability improvements anymore
- System software complexity
 - Most failures are due to system software (e.g. parallel file system)

Why is Resilience a Challenge?

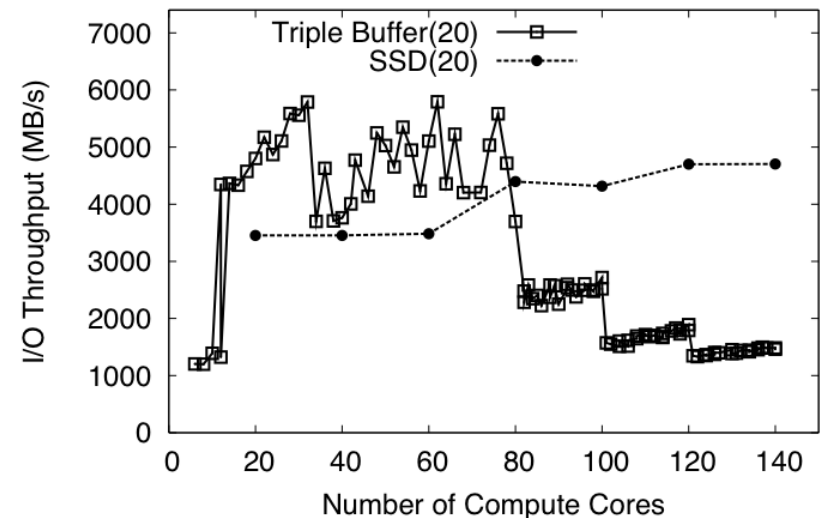
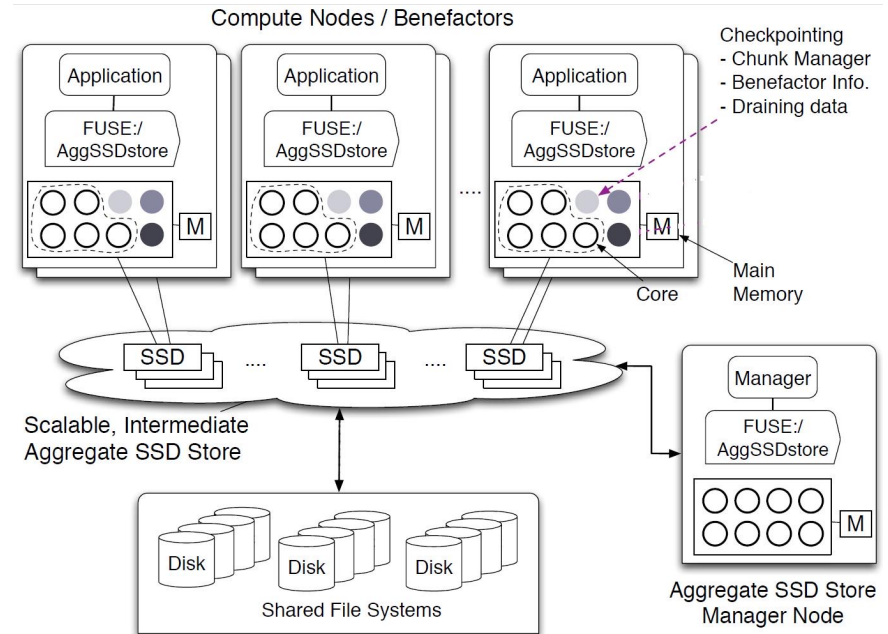
- **NO: It's a cost problem**
 - **This isn't NASA's Moon Program (\$25.4 billion in 1973 Dollars - \$170 billion in 2005 Dollars - over an 11-year period)**
- **The challenge is to build a reliable system within a given cost budget that achieves the expected performance**

Key Areas of HPC Resilience Research



Advanced Checkpoint Storage Architecture

- Nodes offer available node-local SSD or memory space
- Manager process maintains metadata for each client
- FUSE client offers mount point w/o POSIX locking
- Data striped across nodes
- Tested on system with 160 cores using 140 as clients
- Solution helps maintaining I/O throughput



Full/Incremental Checkpointing with BLCR

- Hybrid checkpointing:
1 full and k incremental
(part of BLCR distribution)
- Tracks dirty memory pages
- Full: Saves all pages
- Incremental: Appends dirty pages to checkpoint file
- Recovery: Scans file in reverse sequence
- Optimal at 1 full / 9 incr.

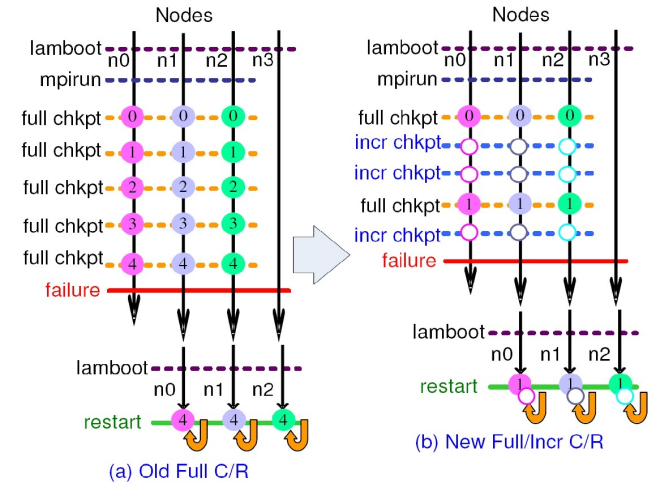
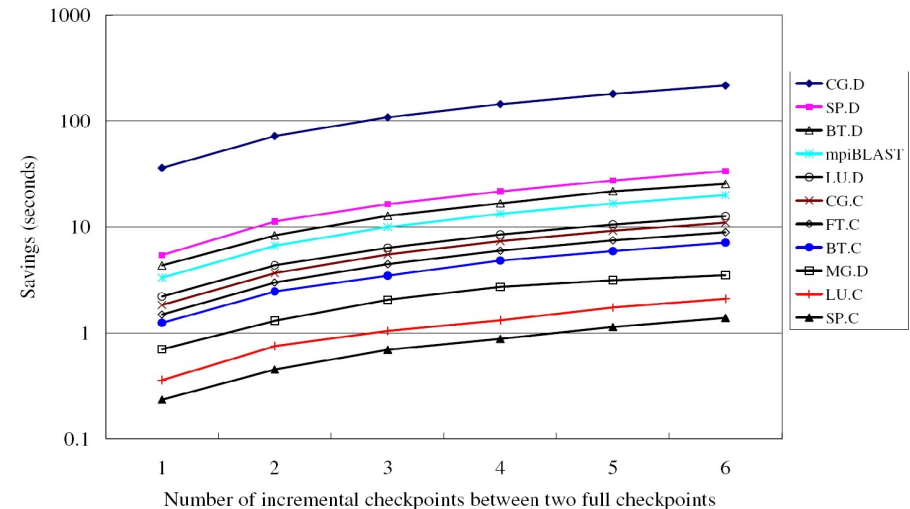


Fig. 1: Hybrid Full/Incremental C/R Mechanism vs. Full C/R



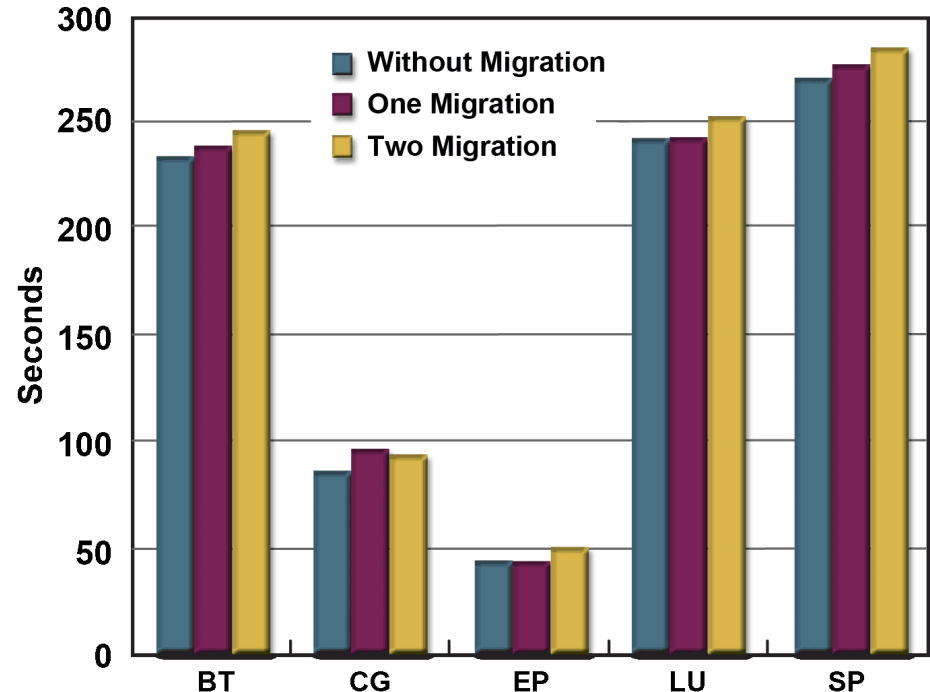
Proactive Fault Tolerance using Migration

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



VM-level Migration with Xen

- 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
- 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



NC STATE UNIVERSITY

Process-Level Migration with BLCR

Single migration overhead

- Stop & copy : 0.09-6.00%
- Live : 0.08-2.98%

Single migration duration

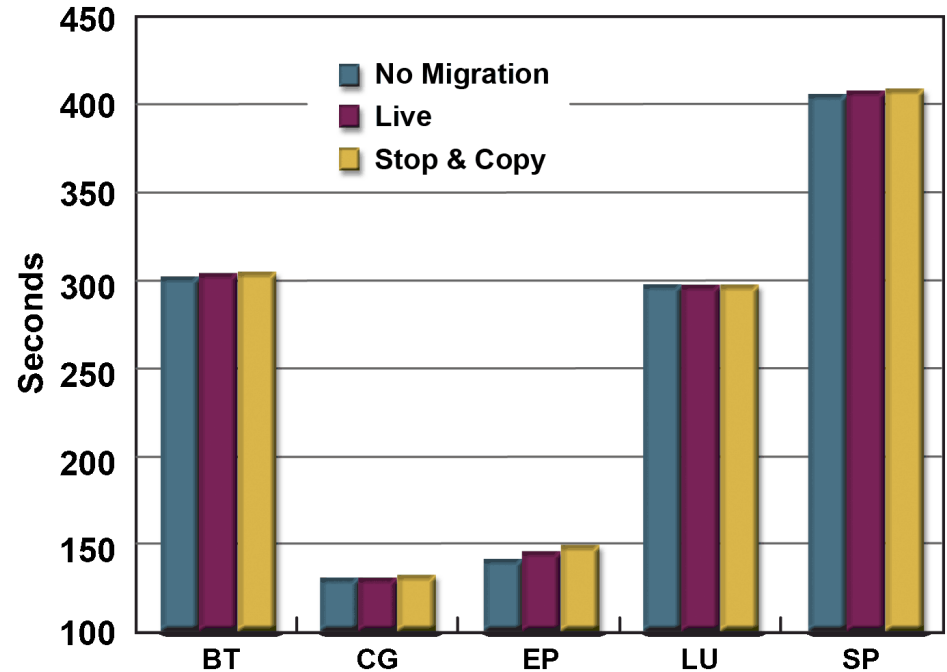
- Stop & copy : 1.0-1.9s
- Live : 2.6-6.5s

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



NC STATE UNIVERSITY

Process-level Redundancy atop MPI for Soft Error Resilience

- Transparent redundant execution of MPI apps. (redMPI)
- Interposition library between MPI and the app.
- App. runs with $r * m$ ranks:
 - r ranks visible to the app.
 - m is the replication degree
- Fault model is fail-stop
- All messages are replicated
- File I/O is unified or replicated



Using Redundancy for Soft Error Injection

- Study propagation of silent data corruption at runtime
- Taint one replica and use the other one as live control
- Disable error correction by the redundant MPI (redMPI)
- Compare MPI messages and record mismatch
- Obtain some sense of silent data corruption vulnerability
 - Corruption patterns
 - Propagation speed

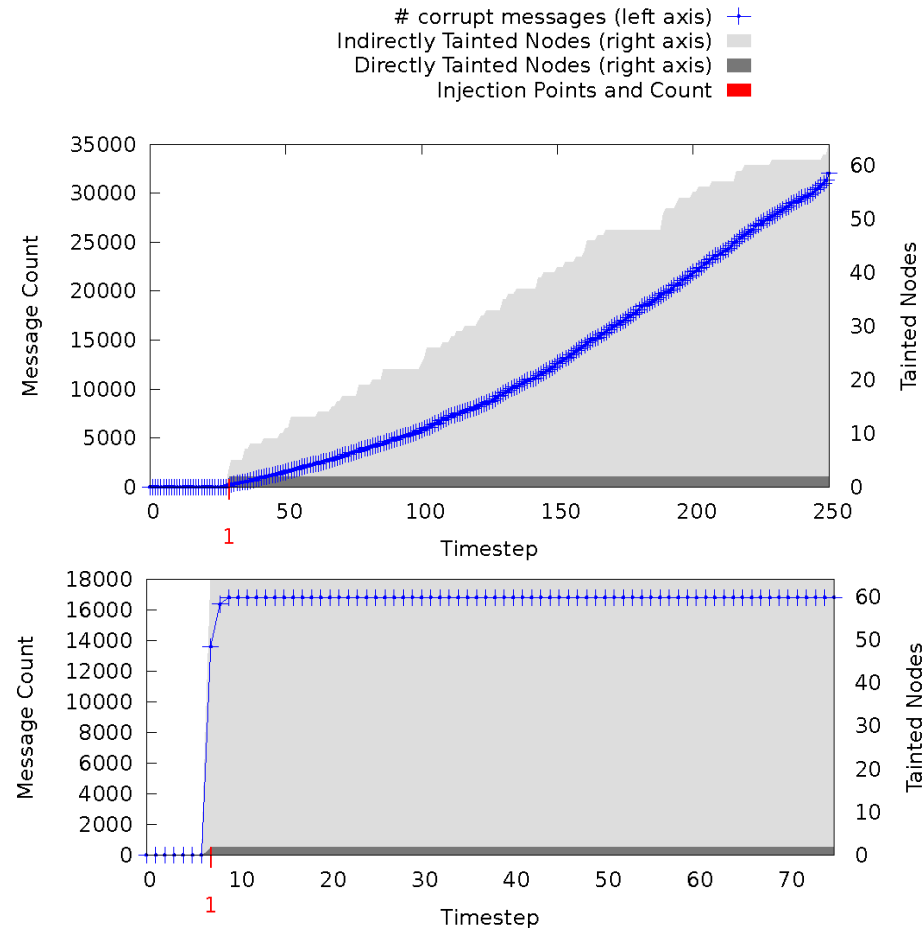
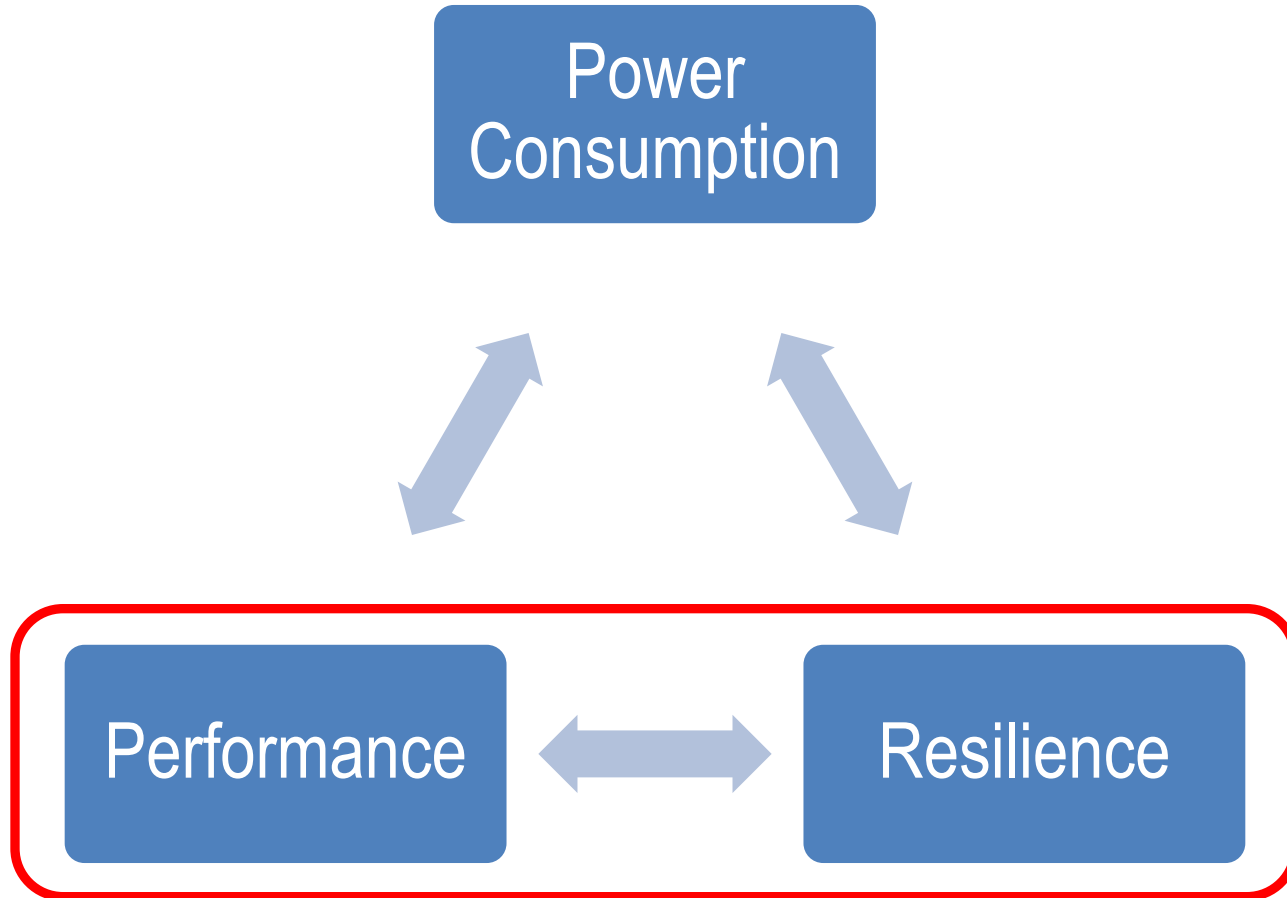


Fig. 8. NPB CG Overview of Corrupt Nodes and Messages

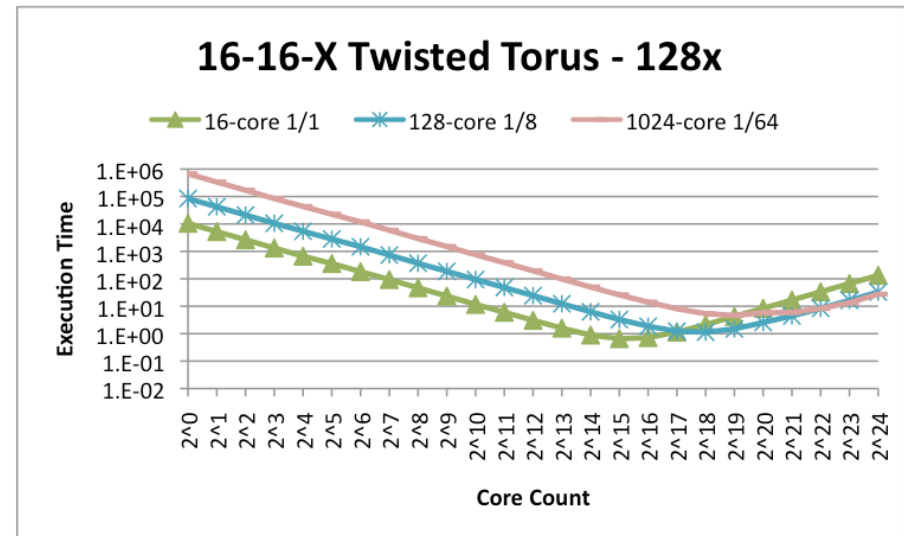
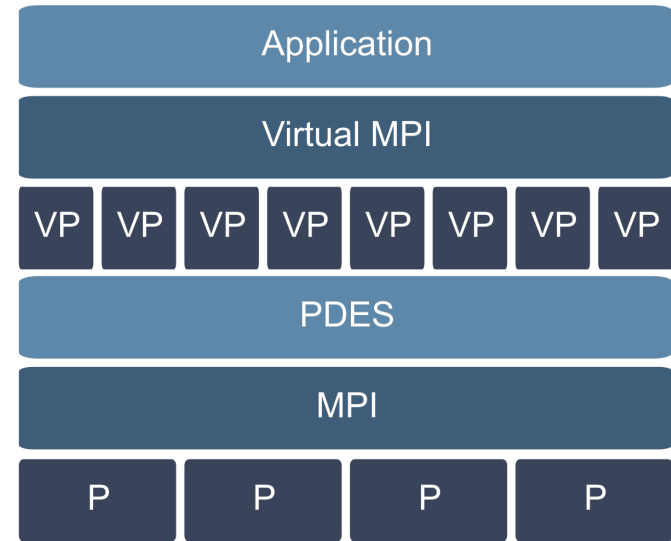
HPC System Hardware/Software Co-Design: Optimizing Trade-offs



Examples: ECC memory, checkpoint storage, data redundancy, computational redundancy, algorithmic resilience

xSim – The Extreme-Scale Simulator

- Combining highly oversubscribed execution, a virtual MPI, & a time-accurate PDES
 - Execution on native processor
 - Processor and network model
- Support for C/Fortran MPI
- Easy to use:
 - Compile with xSim header
 - Link with the xSim library
 - Execute: `mpirun -np <np> <application> -xsim-np <vp>`

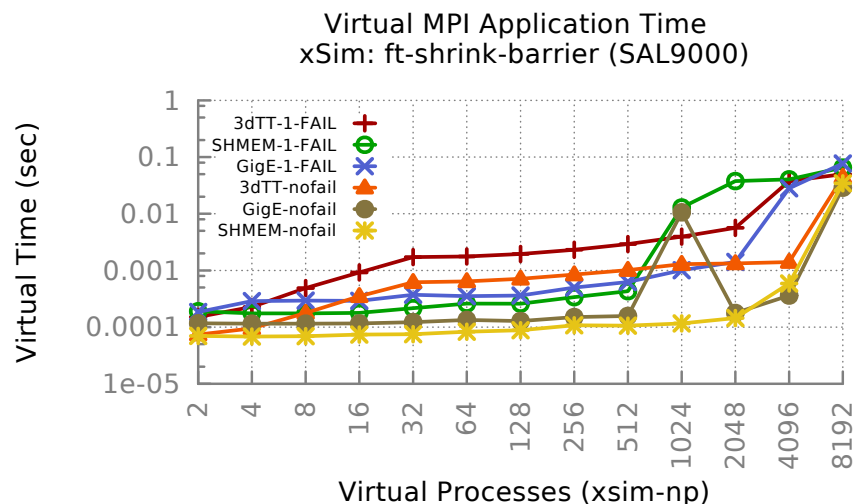


Resilience Simulation Features

- Simulated MPI process failure
 - Injection, propagation and detection in modeled architecture
- Simulated MPI application checkpoint, abort, and restart
 - Support for checkpoint/restart cycles until completion
- Simulated fault tolerant MPI
 - Support for resilient solvers using fault tolerant MPI (ULFM)
- Used in MCREX RX-Solvers project (Trilinos-based solver)

Table 1: Varying the checkpoint interval and system MTTF

$MTTF_S$	C	E_1	E_2	F	$MTTF_A$
—	1,000	5,248 s	—	0	—
6,000 s	500	5,258 s	7,957 s	1	3,978 s
6,000 s	250	6,377 s	7,074 s	1	3,537 s
6,000 s	125	6,601 s	6,750 s	1	3,375 s
3,000 s	500	5,258 s	10,584 s	2	3,528 s
3,000 s	250	6,377 s	8,618 s	2	2,872 s
3,000 s	125	6,601 s	7,948 s	2	2,649 s



Conclusion

- Resilience is a critical challenge for extreme-scale HPC
- Solutions have been and are being developed
- There are still many open questions
 - There is no HPC fault model
 - Performance/resilience/power trade-off is not understood holistically
 - Coordination across the stack is missing
 - End-to-end management of resilience is needed
 - Resilient parallel programming models do not exist
 - Operational resilience policies do not go beyond checkpoint/restart

Acknowledgements

- **Oak Ridge National Laboratory (ORNL):** S. L. Scott, C. Engelmann, G. Vallée, T. Naughton, C. Wang, A. Tikotekar, S. Vazhkudai, Y. Kim, G. Shipman, S. Boehm, I. Jones, and F. Lauer
- **Sandia National Laboratory:** K. Ferreira and Ron Brightwell
- **North Carolina State University:** F. Mueller, A. Nagarajan, J. Varma, X. Ma, F. Meng, and J. Elliott
- **Virginia Tech University:** M. Li, A. Butt
- **Funding sources:** DoE OASCR, U.S. NSF, and ORNL/SNL LDRD



NC STATE UNIVERSITY



Questions



C. Engelmann. Resilience Challenges and Solutions for Extreme-Scale Supercomputing, SOS Workshop, Park City, UT, 3/2-5, 2015.