UT-BATTELLE

Resilient Software for ExaScale Computing

Christian Engelmann

Computer Science Research Group
Computer Science and Mathematics Division
Oak Ridge National Laboratory, USA

Discussed Exascale Road Map

Many design factors are driven by the power ceiling of 20MW

Systems	2009	2011	2015	2020
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	days	days	days	O(1 day)
Power	6 MW	~10MW	~10 MW	~20 MW

Discussed Exascale Road Map

Many design factors are driven by the power ceiling of 20MW

Systems	2009	2011	2015	2020
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System size (nodes)		5x	5x	2x

System MTTI based on optimistic assumption that node MTTI goes up

MTTI	days	days	days	O(1 day)

My Exascale Resilience Scenario

MTTI Scales with Node Count

Systems	2009	2011	2015	2020
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System size (nodes)		5x	5x	2x
MTTI	4 days	19 h 4 min	3 h 52 min	1 h 56 min

Vendors are able to maintain current node MTTI

My Scary Scenario

Current MTTI of 1 Day

Systems	2009	2011	2015	2020
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System size (nodes)		5x	5x	2x
MTTI	1 day	4 h 48 min	58 min	29 min

Current system MTTI is actually lower

My Really Scary Scenario

Component MTTI drops 3% Each Year

Systems	2009	2011	2015	2020
System peak	2 Peta	20 Peta	100-200 Peta	1 Exa
System size (nodes)		5x	5x	2x
MTTI	1 day	4 h 31 min	48 min	22 min

Vendors are not able to maintain current node MTTI

Discussed Exascale Road Map

Many design factors are driven by the power ceiling of 20MW

Systems	2009	2011	2015	2020
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)
IO	[illegible]B/s	2 TB/s	10 TB/s	20 TB/s
MTTI	1-4 days	5-19 hours	50-230 min	22-120 min
Power	6 MW	~10MW	~10 MW	~20 MW

MTTI is still the wrong metric for RESILIENCE!

Why are we only talking about MTTI?

- None of these MTTI numbers are accurate ±O(10)
- Looking at MTTI alone doesn't make much sense
- For 90% availability with 1M nodes, each none needs:
  - 7 nines

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1}{1 + \frac{MTTR}{MTTF}}$$

9s	Availability	Annual Downtime
1	90%	36 days, 12 hours
2	99%	87 hours, 36 minutes
3	99.9%	8 hours, 45.6 minutes
4	99.99%	52 minutes, 33.6 seconds
5	99.999%	5 minutes, 15.4 seconds
6	99.9999%	31.5 seconds

Why are we only talking about MTTI?

- None of these MTTI numbers are accurate ±O(10)
- Looking at MTTI alone doesn't make much sense
- For 90% availability with 1M nodes, each none needs:
  - 7 nines without redundancy
  - 4 nines for DMR
  - 3 nines for dynamic DMR
  - 3 nines for TMR
  - 2 nines for dynamic TMR

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1}{1 + \frac{MTTR}{MTTF}}$$

9s	Availability	Annual Downtime
1	90%	36 days, 12 hours
2	99%	87 hours, 36 minutes
3	99.9%	8 hours, 45.6 minutes
4	99.99%	52 minutes, 33.6 seconds
5	99.999%	5 minutes, 15.4 seconds
6	99.9999%	31.5 seconds

Why are we only talking about MTTI?

- None of these MTTI numbers are accurate ±O(10)
- Looking at MTTI alone doesn't make much sense
- For 90% availability with 1M nodes, each none needs:
  - 7 nines without redundancy
  - 4 nines for DMR
  - 3 nines for dynamic DMR
  - 3 nines for TMR
  - 2 nines for dynamic TMR
  - ? nines for checkpoint/restart
  - ? nines for message logging
  - ? nines for algorithm-based fault tolerance

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1}{1 + \frac{MTTR}{MTTF}}$$

9s	Availability	Annual Downtime
1	90%	36 days, 12 hours
2	99%	87 hours, 36 minutes
3	99.9%	8 hours, 45.6 minutes
4	99.99%	52 minutes, 33.6 seconds
5	99.999%	5 minutes, 15.4 seconds
6	99.9999%	31.5 seconds

Why are we only talking about MTTI?

- None of these MTTI numbers are accurate ±O(10)
- Looking at MTTI alone doesn't make much sense
- For 90% availability with 1M nodes, each none needs:
  - 7 nines without redundancy
  - 4 nines for DMR
  - 3 nines for dynamic DMR
  - 3 nines for TMR
  - 2 nines for dynamic TMR
  - ? nines for checkpoint/restart
  - ? nines for message logging
  - ? nines for algorithm-based fault tolerance

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1}{1 + \frac{MTTR}{MTTF}}$$

9s	Availability	Annual Downtime
1	90%	36 days, 12 hours
2	99%	87 hours, 36 minutes
3	99.9%	8 hours, 45.6 minutes
4	99.99%	52 minutes, 33.6 seconds
5	99.999%	5 minutes, 15.4 seconds
6	99.9999%	31.5 seconds

- *There is a huge gap in HPC resilience models and metrics*

HPC Resilience Solutions

- "Existing" solutions
  - Application-level C/R, system-level C/R, incremental C/R
  - C/R to memory or SSDs in neighbor or I/O nodes
  - Message logging + C/R
  - Proactive fault tolerance (migration-based fault avoidance)
  - Rejuvenation (reboot to clear latent errors)
  - Process-level redundancy

HPC Resilience Solutions

- "Existing" solutions
  - Application-level C/R, system-level C/R, incremental C/R
  - C/R to memory or SSDs in neighbor or I/O nodes
  - Message logging + C/R
  - Proactive fault tolerance (migration-based fault avoidance)
  - Rejuvenation (reboot to clear latent errors)
  - Process-level redundancy
- Only the C/R solutions are ready for production
- None of the advanced solutions are even close to that

HPC Resilience Solutions

- "Existing" solutions
  - Application-level C/R, system-level C/R, incremental C/R
  - C/R to memory or SSDs in neighbor or I/O nodes
  - Message logging + C/R
  - Proactive fault tolerance (migration-based fault avoidance)
  - Rejuvenation (reboot to clear latent errors)
  - Process-level redundancy
- Only the C/R solutions are ready for production
- None of the advanced solutions are even close to that
- *There is a huge gap between research and production HPC resilience solutions*

What about soft errors?

- Smaller circuit sizes in conjunction with lower circuit voltages will result in higher soft error rates

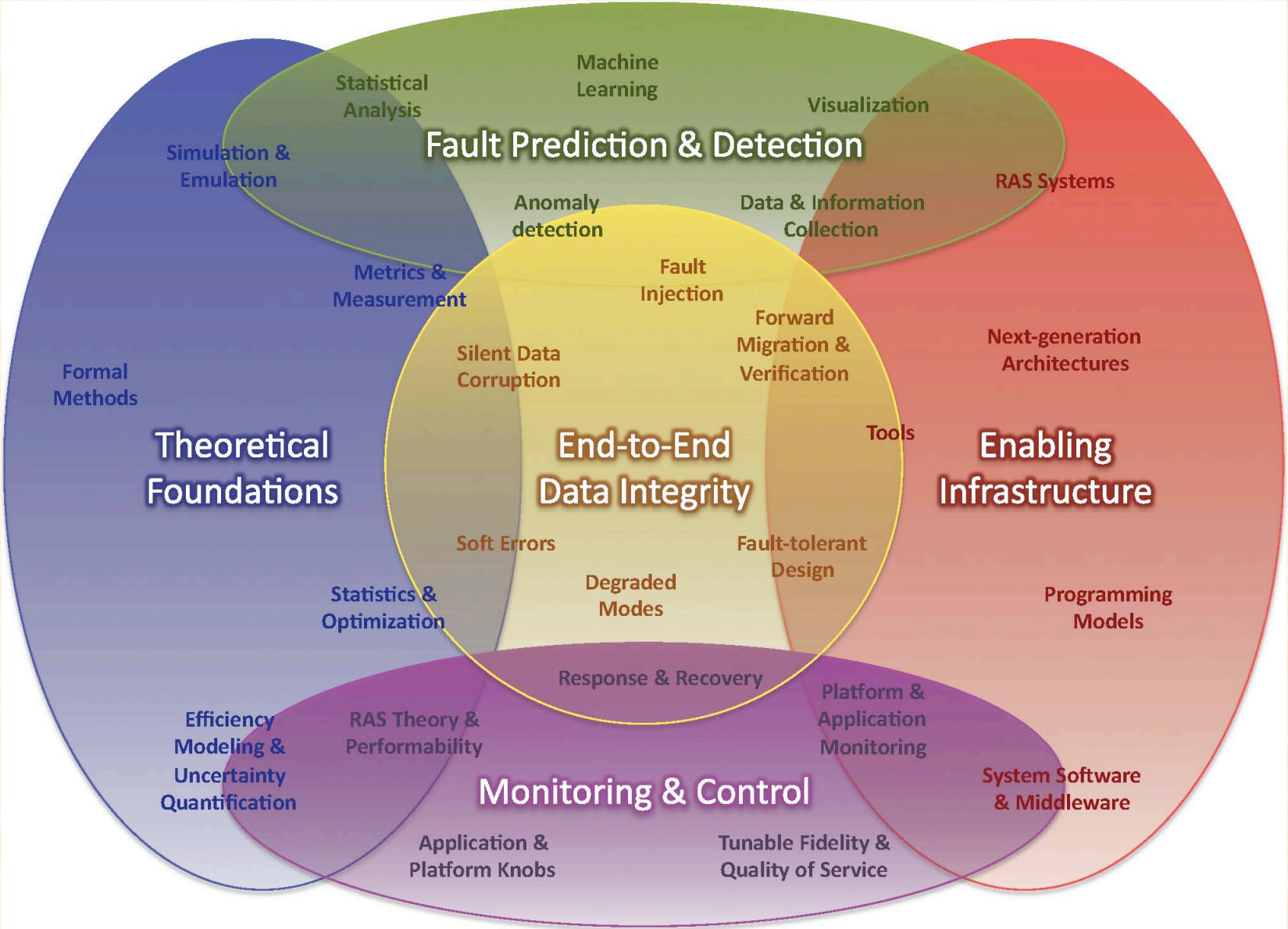
What about soft errors?

- Smaller circuit sizes in conjunction with lower circuit voltages will result in higher soft error rates
- Missing strategy for silent data/code corruption will cause applications to produce erroneous results, hang or crash
- Initial work focuses on redundancy and online correction

What about soft errors?

- Smaller circuit sizes in conjunction with lower circuit voltages will result in higher soft error rates
- Missing strategy for silent data/code corruption will cause applications to produce erroneous results, hang or crash
- Initial work focuses on redundancy and online correction
- *There is a huge gap in understanding and dealing with soft errors*

Key Areas for Future Resilience Research, Development, and Standards Work



Proactive Fault Tolerance (PFT) Framework

- Objectives
  - Facilitate fault resilience at extreme scale
  - Offer migration-based fault avoidance for 24x7 RAS
- Impact
  - Permit avoiding the impact of predictable failures
  - Help in understanding failure causes and propagation
- Novelty
  - Establishes a new area of research in ***HPC resilience***
  - System-software approach for anticipating failures
  - Offers process migration and scalable system monitoring
  - Builds theoretical foundations:
    - System and component reliability analysis
    - System logging and monitoring requirements analysis

PFT: Reactive vs. Proactive Fault Tolerance

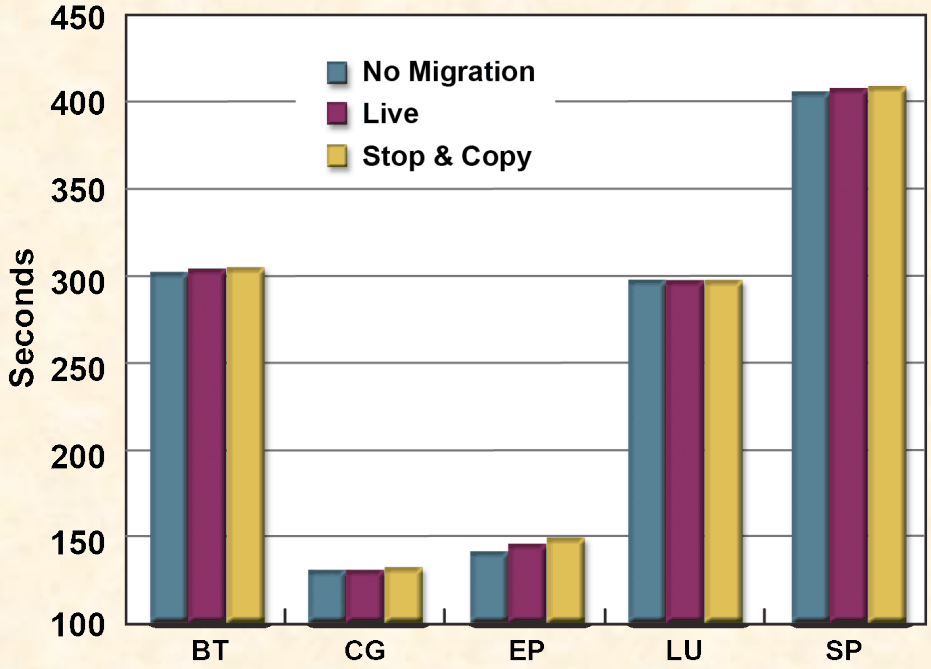
- Reactive fault tolerance
  - Keeps parallel applications alive through recovery from experienced failures
  - Employed mechanisms react to failures
  - Examples: Checkpoint/restart and message logging/replay
- Proactive fault tolerance
  - Keeps parallel applications alive by avoiding failures through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

PFT: Approach

- 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

PFT: BLCR Process-Level Migration

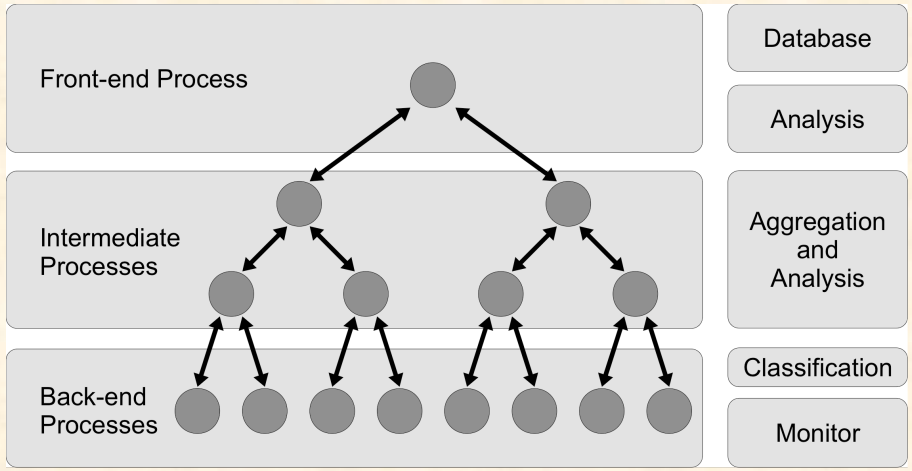
- Single migration overhead
  - Stop & copy : 0.09-6.00%
  - Live : 0.08-2.98%
- Single migration duration
  - Stop & copy : 1.0-1.9s
  - Live : 2.6-6.5s
- Application downtime
  - Stop & copy > Live
- Node eviction time
  - Stop & copy < Live



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

PFT: System Monitoring Tool using MRNet

- 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

PFT: Current Status and Future Work

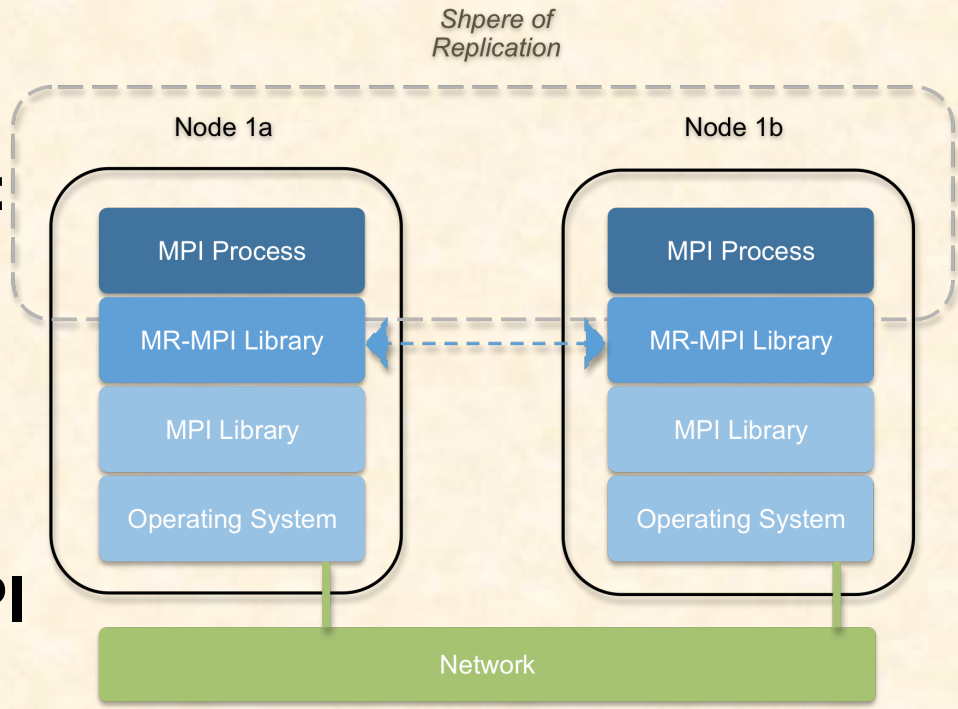
- Developed prototypes for proactive fault tolerance
  - Core framework that plugs individual components together
  - Interfaces to job and resource manager, system monitoring, message logging and runtime environment (MPI)
  - Process-level migration solution (BLCR)
  - Scalable system monitoring solution (MRNet-based)
  - Statistical analyses of system log and monitoring data

MR-MPI: Process-Level Redundancy for MPI

- Objectives
  - Facilitate hard and soft error resilience at extreme scale
  - Offer process-level redundancy for seamless resilience
- Impact
  - Survive hard and soft errors without the need for recovery
  - Detect and optionally correct silent data corruption
- Novelty
  - Establishes a new area of research in HPC resilience
  - Software-only approach, i.e., no need for expensive hardware
  - Redundancy at the Message Passing Interface (MPI)
  - Transparent: No application modification needed
  - Provides resilience on-demand on a job-by-job basis
  - Only solution to support file I/O under redundancy

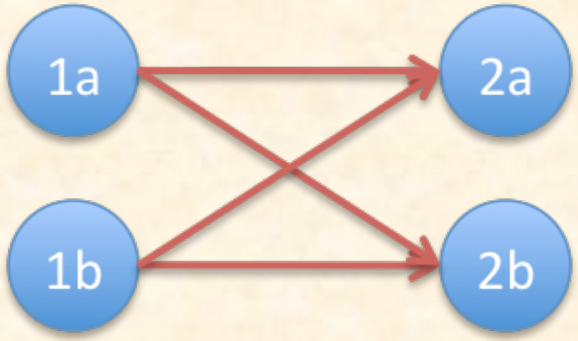
MR-MPI: Technical Approach

- Aim at MPI process-level redundancy
- Transparent redundant execution of MPI processes:
  - On the same processor
  - On different processors
  - On different compute nodes
- Input replication and output comparison between the MPI library and the application
- The fault model is fail-stop
- MPI and platform independent



MR-MPI: Design (1/2)

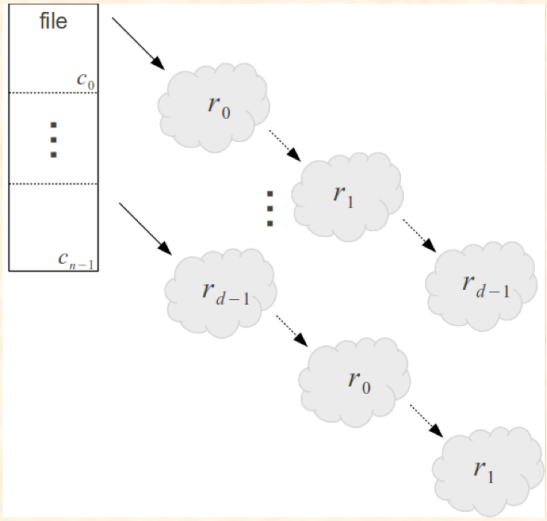
- *r* × *m* native MPI ranks:
  - *r* ranks visible to the application
  - *m* is the replication degree
  - <np> = r*m
  - <vnp> = r

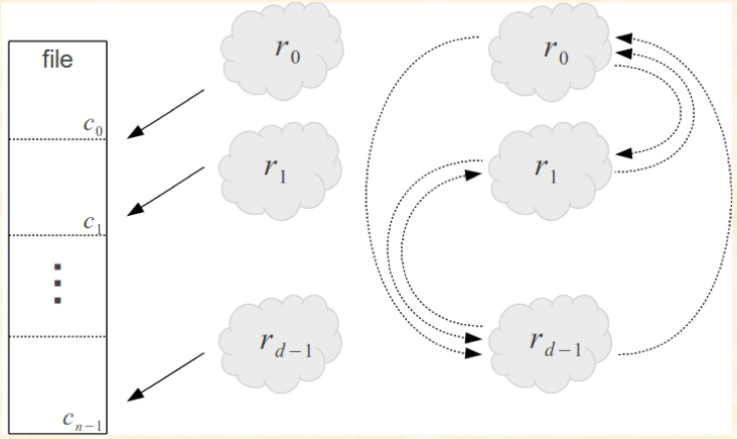


- Full message replication
- Master-failover for non-determinism, e.g., for MPI_Wtime()
- Partial replication is supported statically
  - For example, execute with a replication degree of 1.5

MR-MPI: Design (2/2)

- Intercepts POSIX file I/O calls from the application
- New file I/O calls employ coordination protocols for
  - Redundancy-oblivious node-local file system access
  - Redundancy-aware shared networked file system access
- File read and write protocols with different features
  - Utilizing parallel I/O and inter-node communication





Acknowledgements

- Investigators at Oak Ridge National Laboratory (ORNL):
  - S. L. Scott, C. Engelmann, G. Vallée, T. Naughton, C. Wang, A. Tikotekar, G. Ostrouchov, S. Vazhkudai, Y. Kim, G. Shipman, S. Boehm, I. Jones, F. Lauer
- Investigators at Louisiana Tech University:
  - C. Leangsuksun, N. Naksinehaboon, R. Nassar, M. Paun
- Investigators at North Carolina State University:
  - F. Mueller, A. Nagarajan, J. Varma, X. Ma, F. Meng
- Investigators at Virginia Tech
  - M. Li, A. Butt
- Funding sources:
  - U.S. Department of Energy, Office of Science, FASTOS 2
  - U.S. National Science Foundation, CCF-0937827, CCF-0746832, CCF 0621470, CCF-0937690
  - ORNL LDRD and NCSU/ORNL Joint Appointment programs

Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA)

- @ SC'10, SC'11
- Topics of interest include, but are not limited to:
  - Novel scientific algorithms that improve performance, scalability, resilience and power efficiency
  - Porting algorithms to many-core and heterogeneous architectures
  - Performance and resilience limitations of algorithms at scale
  - Crosscutting approaches (system software and applications)
  - Scientific algorithms that can exploit extreme concurrency
  - Naturally fault tolerant, self-healing or fault oblivious algorithms
  - Programming models & system software for scalability/resilience
- Chairs: V. Alexandrov (BSC), J. Dongarra (UT), A. Geist (ORNL)
- URL: http://www.csm.ornl.gov/srt/conferences/Scala

Workshop on Resiliency in HPC (Resilience) in Clusters, Clouds, and Grids

- @ Euro-Par'11, CCGrid'10, HPDC'09, CCGrid'08
- Topics of interest include, but are not limited to:
  - Reports on current HPC system and application resiliency
  - HPC resiliency metrics, standards, and analysis
  - HPC system and application-level fault handling and anticipation
  - HPC system and application health monitoring
  - Resiliency for HPC file and storage systems
  - Algorithm-based resiliency fundamentals for HPC (not Hadoop)
  - Fault tolerant MPI concepts and solutions
  - Soft error detection and recovery in HPC systems
- Chairs: S. Scott (TTU/ORNL) and C. Leangsuksun (LA Tech)
- URL: http://xcr.cenit.latech.edu/resilience20XX/