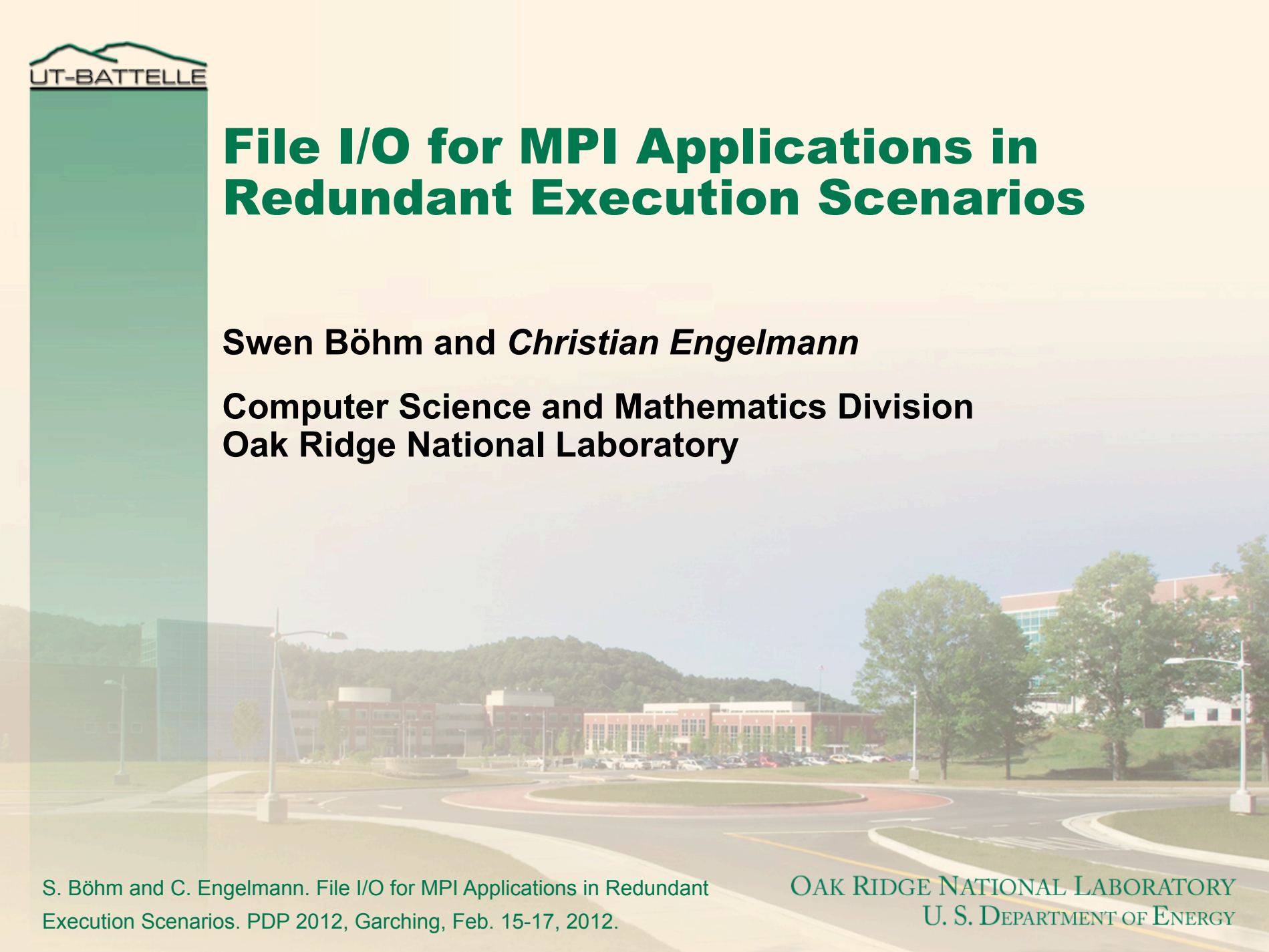
UT-BATTELLE

File I/O for MPI Applications in Redundant Execution Scenarios

Swen Böhm and *Christian Engelmann*

Computer Science and Mathematics Division
Oak Ridge National Laboratory



S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Current-Generation HPC Systems

- Large-scale PFlop/s systems:
  - #1 RIKEN K: 8.162 PFlop/s, 548,352 cores, 93%
  - #2 NSCT Tianhe-1A: 2.566 PFlop/s, 186,368 cores, 55%
  - #3 ORNL Jaguar XT5: 1.759 PFlop/s, 224,162 cores, 75%
  - #4 NSCS Nebulae: 1.271 PFlop/s, 120,640 cores, 43%
  - #5 GSIC Tsubame 2.0: 1.192 PFlop/s, 73,278 cores, 61%
  - #5 LANL Cielo: 1.110 PFlop/s, 142,272 cores, 81%
  - #6 NASA Pleiades: 1.088 PFlop/s, 111,104 cores, 81%
  - #7 LBNL Hopper: 1.054 PFlop/s, 153,408 cores, 82%
- The trend is toward even larger-scale systems
  - End of processor frequency scaling ➔ Node/core scaling
  - New technologies: Chip stacking, NoC, NVRAM, Si photonics

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Discussed Exascale Road Map

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

Systems	2009	2012	2016	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	1-4 days	5-19 hours	50-230 min	22-120 min
Power	6 MW	~10MW	~10 MW	~20 MW

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Resilience Issues in Extreme-scale HPC

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

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Risks of the Business as Usual Approach

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

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Redundancy: A HPC Resilience Alternative

- Instead of using rollback recovery, focus on redundancy
- Addresses the deficiencies of recovery-oriented HPC, such as scalability and soft-error coverage
- Centers on a software-only approach at the system software layer that is completely transparent
- Initial prototypes provided redundancy at the Message Passing Interface layer via state-machine replication
- None of the initial prototypes support file I/O in redundant parallel application scenarios

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

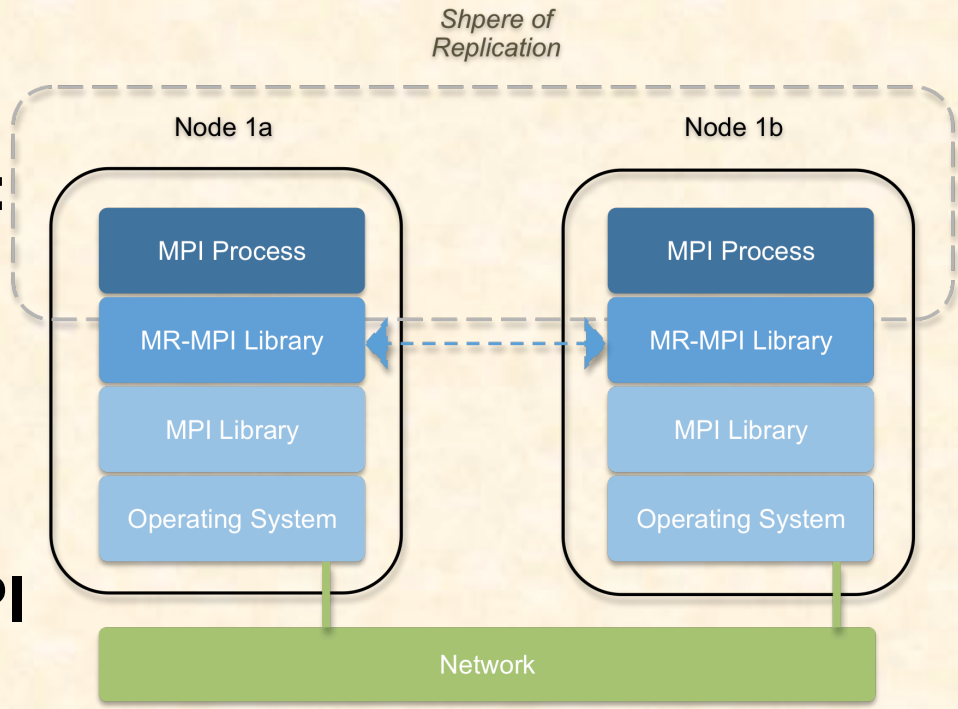
Related Work and Prior Accomplishments

- Redundancy has been used to ensure availability and reliability in information technology, aerospace and command & control systems for decades
- Redundant execution with rMPI, MR-MPI and redMPI
  - Using the MPI profiling layer PMPI to intercept all MPI calls from an application and to hide all redundancy mechanisms (presented work utilizes MR-MPI)
- Redundant execution with VolpexMPI
  - MPI library implemented from scratch that supports redundancy in its communication layer
- Redundant Execution with MMPI
  - Recent study of MPI redundancy protocols

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

MR-MPI: Technical Approach

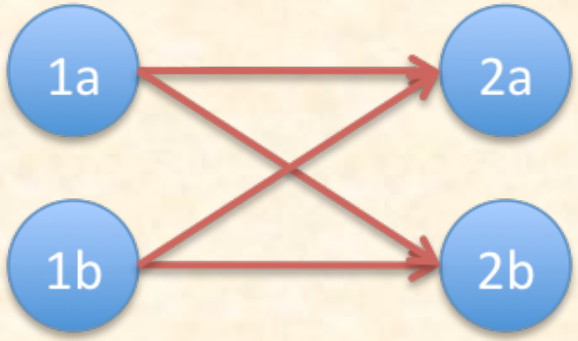
- Aim at MPI process-level redundancy
- Transparent redundant execution of MPI processes:
  - On the same processor
  - On different processors
  - On different compute nodes
- Input replication and output comparison between the MPI library and the application
- The fault model is fail-stop
- MPI and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



S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

MR-MPI: Design

- *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

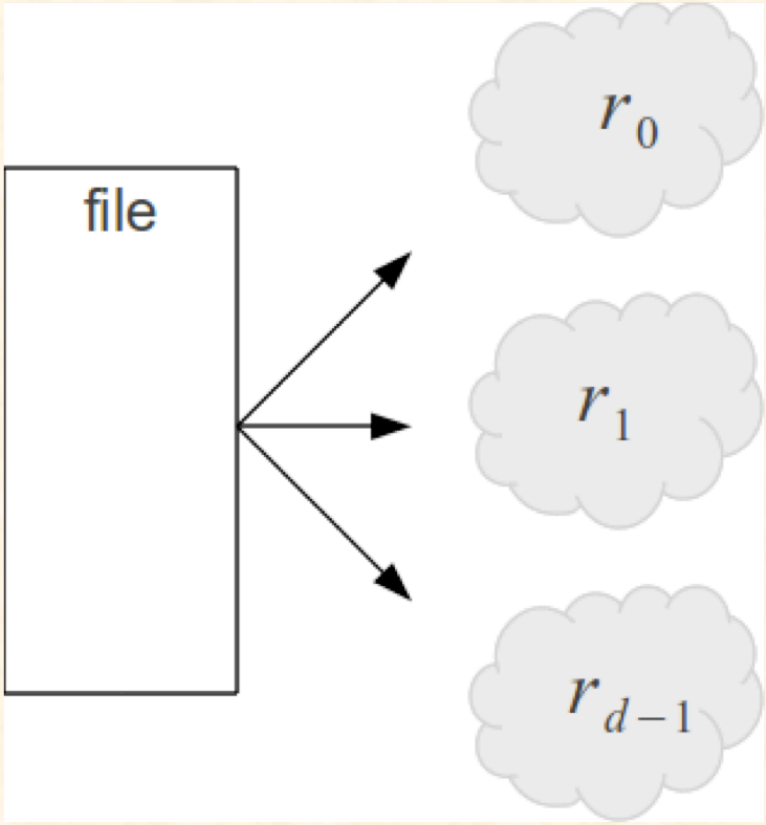
Technical Approach

- File I/O to a node-local file system, e.g., temporary files
  - Redundancy-oblivious file I/O protocol
- File I/O to a shared file system, e.g., input/output data
  - Redundancy-aware file I/O protocol
- Redundancy-aware file I/O protocol features
  - Fully unified file I/O, i.e., only one file is read/written
  - Fully redundant file I/O, i.e., multiple files are read/written
  - Fault tolerance to compensate for the loss of a replica
- Focus on Portable Operating System Interface for Unix (POSIX) file I/O for an initial prototype
- Develop and evaluate different file I/O protocols

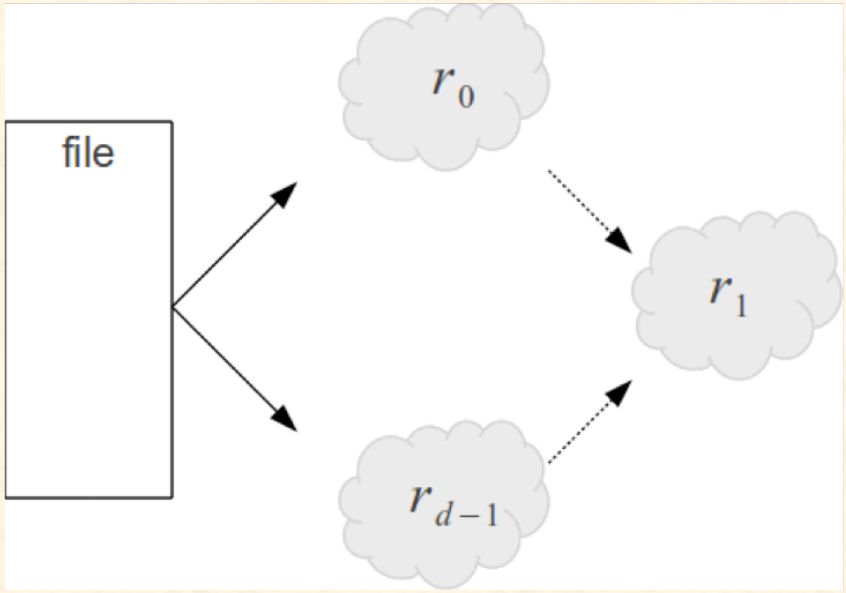
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Redundancy-aware Protocols: Read (1/3)

Simple Read



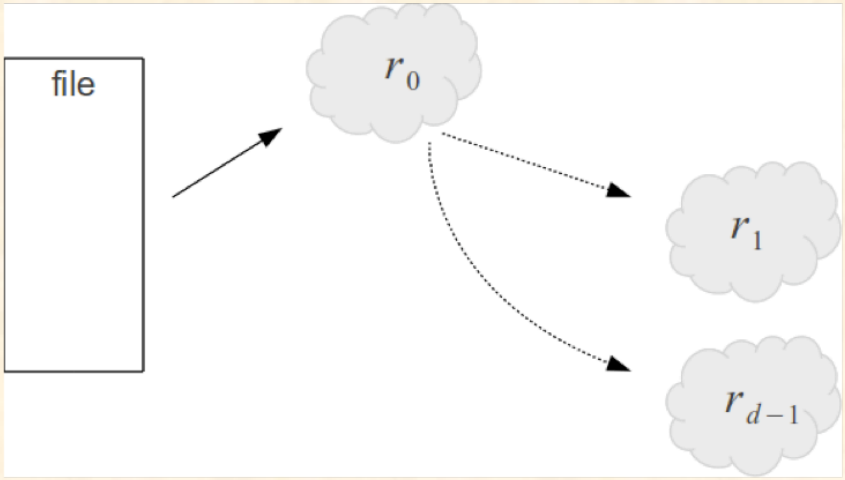
Extended Simple Read



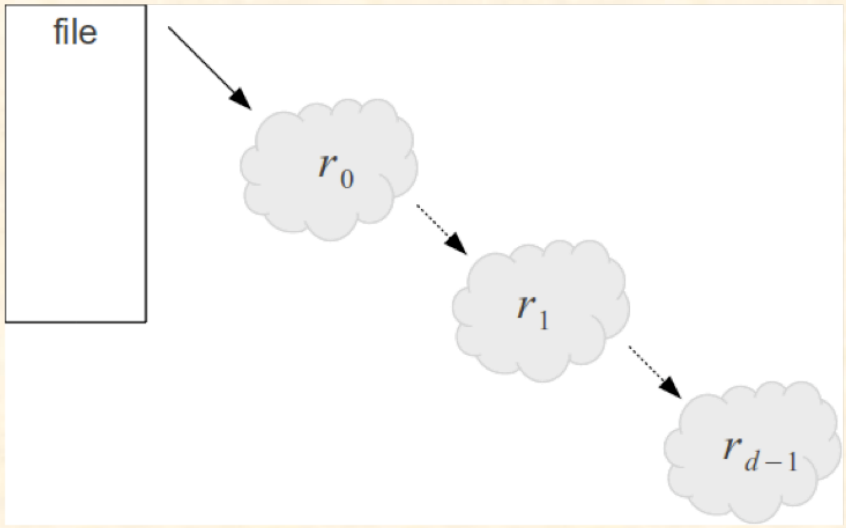
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Redundancy-aware Protocols: Read (2/3)

Read & Distribute



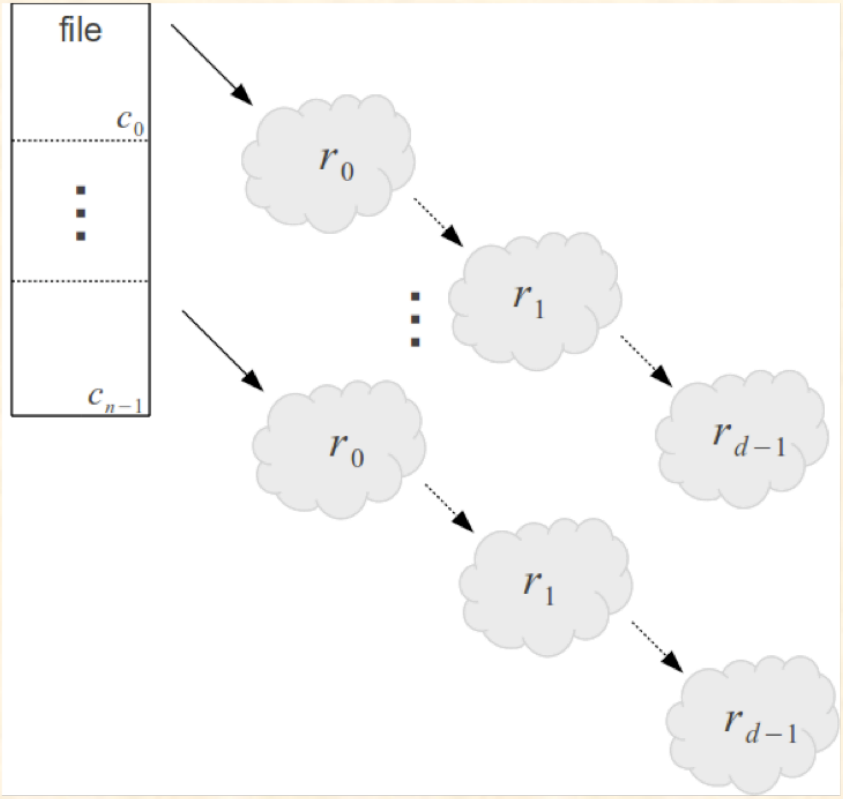
Chained Read



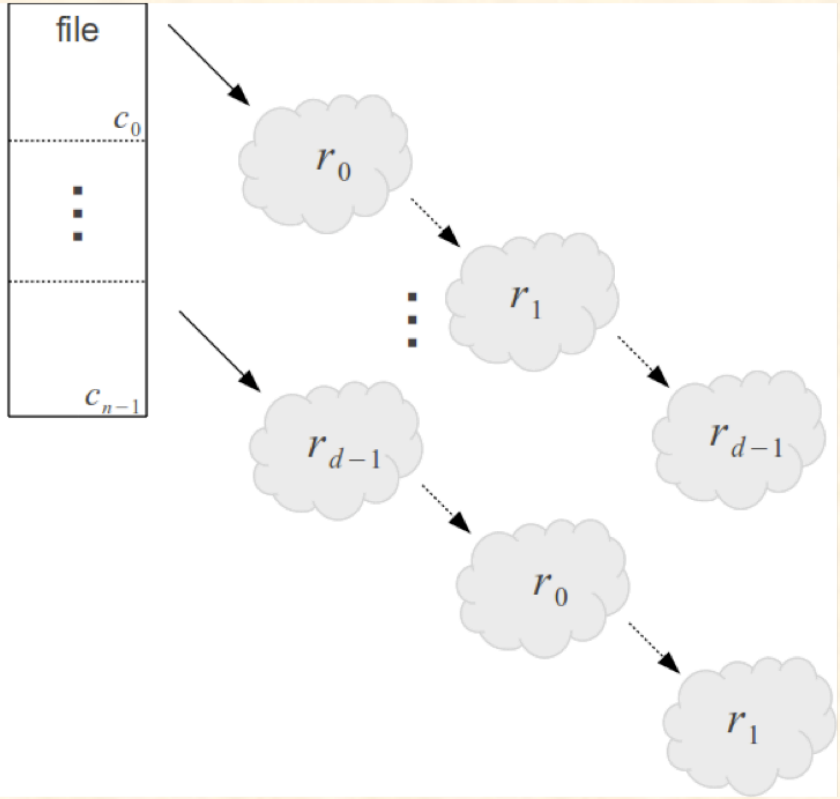
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Redundancy-aware Protocols: Read (3/3)

Chunked & Chained Read



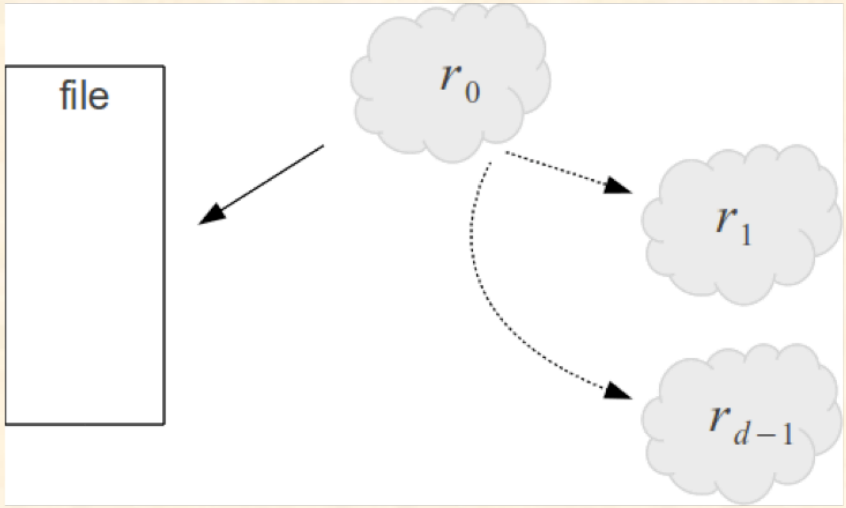
Distributed Chunked Read



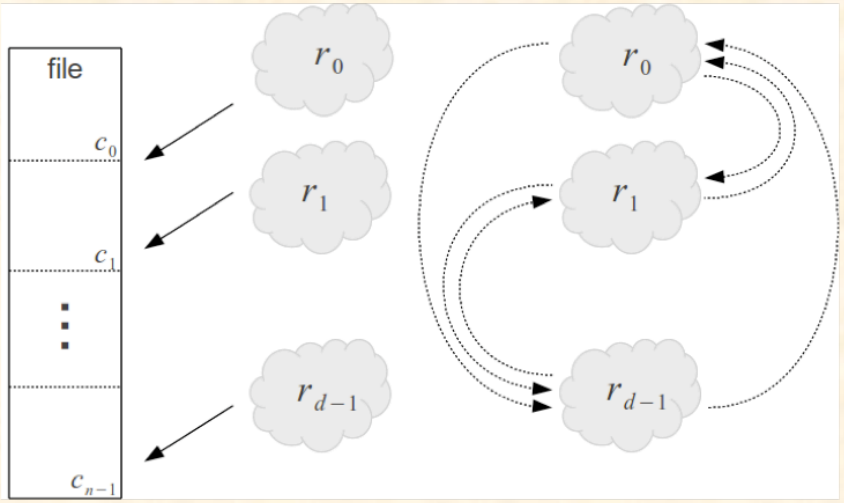
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Redundancy-aware Protocols: Write (1/2)

Simple Write



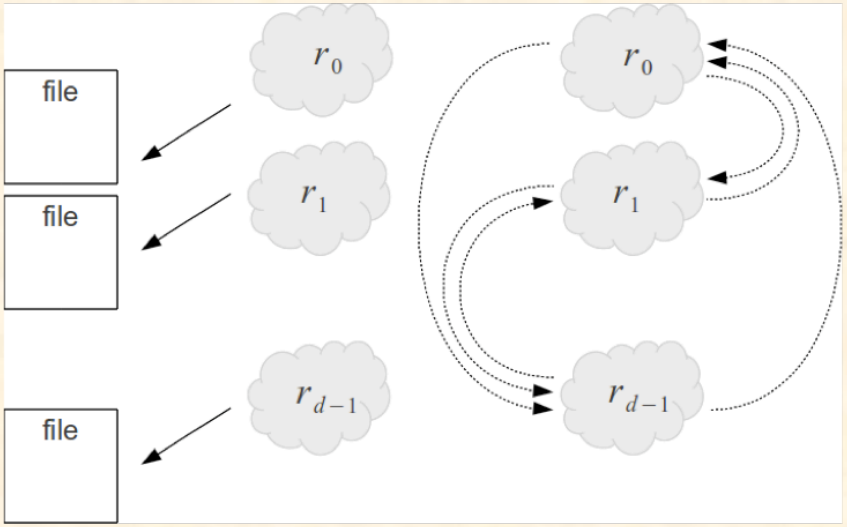
Distributed Write



S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Redundancy-aware Protocols: Write (2/2)

Distributed Separate Write



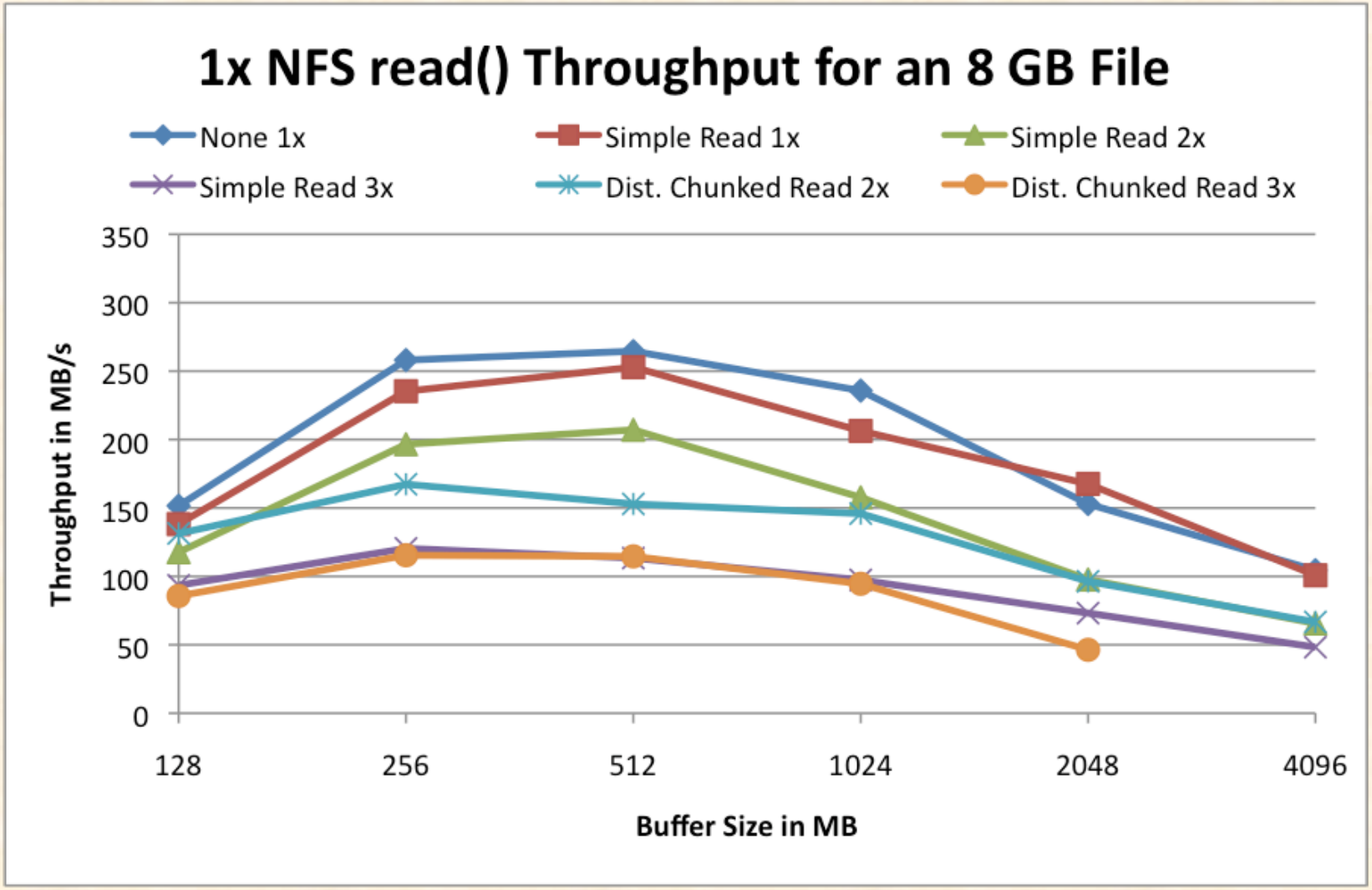
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Performance: Single/Multiple NFS Read

- Single (non-)replicated MPI process
  - Reading an 8 GB file from a single NFS server
  - Reading an 8 GB file from multiple redundant NFS servers
- Using None, Simple Read and Chunked & Chained Read
- Varying buffer (block) sizes from 128MB to 4GB
- Without (1x), with double (2x) & with triple (3x) redundancy
- Over Gigabit Ethernet with non-blocking switch

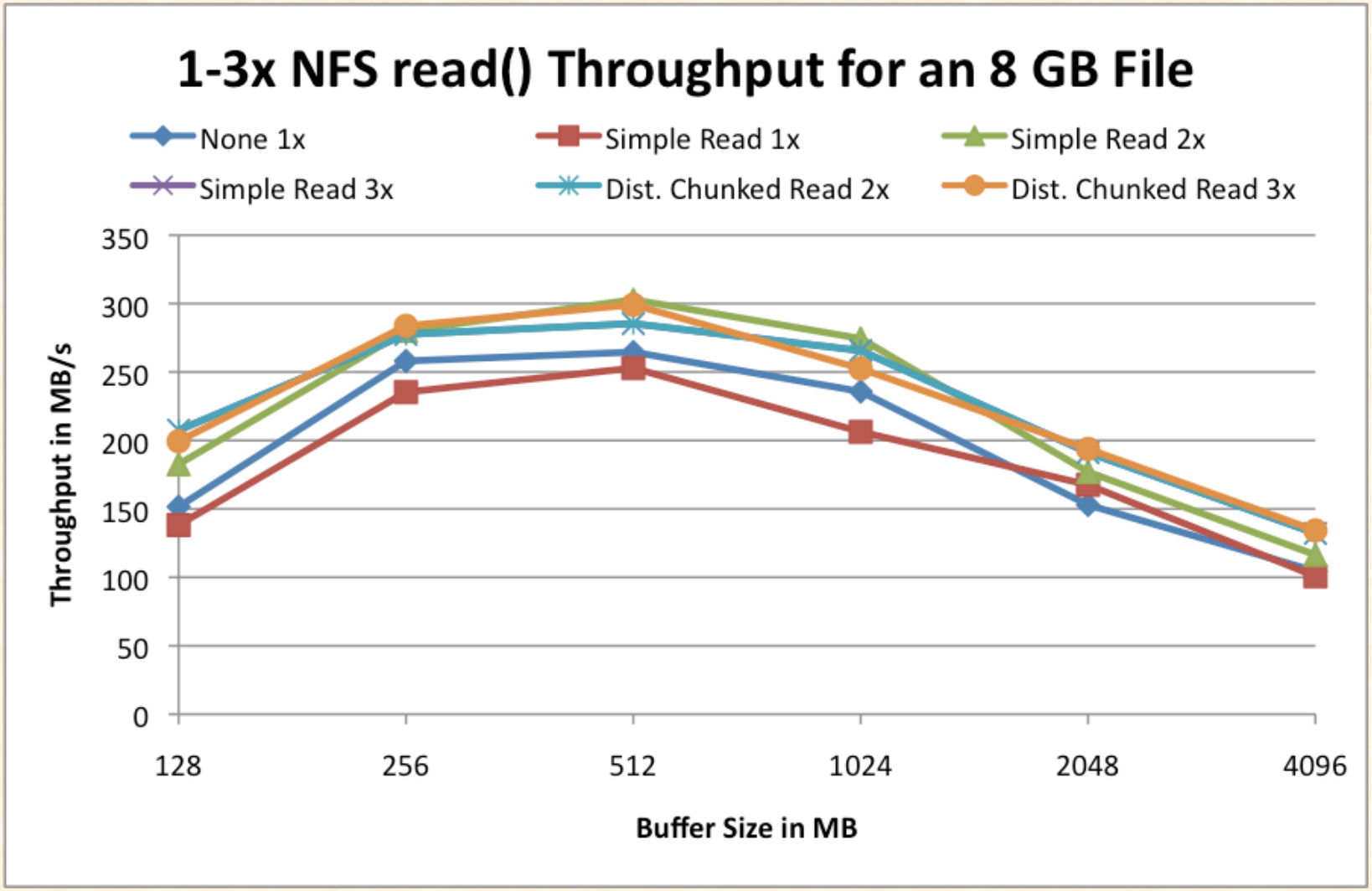
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Performance: Single NFS Server Read



S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Performance: Redundant NFS Server Read



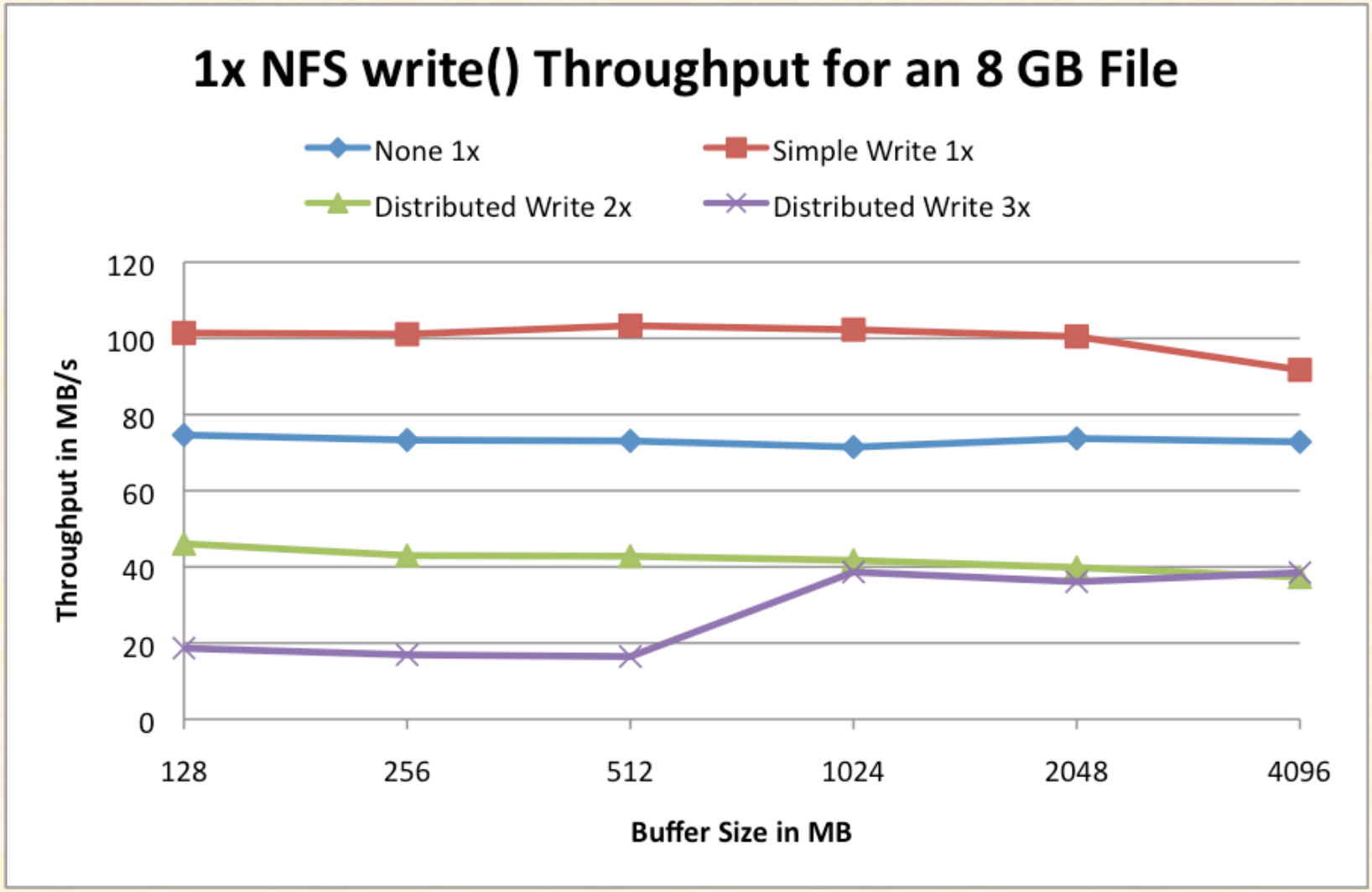
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Performance: Single/Multiple NFS Write

- Single (non-)replicated MPI process
  - Writing an 8 GB file from a single NFS server
  - Writing an 8 GB file from multiple redundant NFS servers
- Using None, Simple Write and Distributed Write
- Varying buffer (block) sizes from 128MB to 4GB
- Without (1x), with double (2x) & with triple (3x) redundancy
- Over Gigabit Ethernet with non-blocking switch

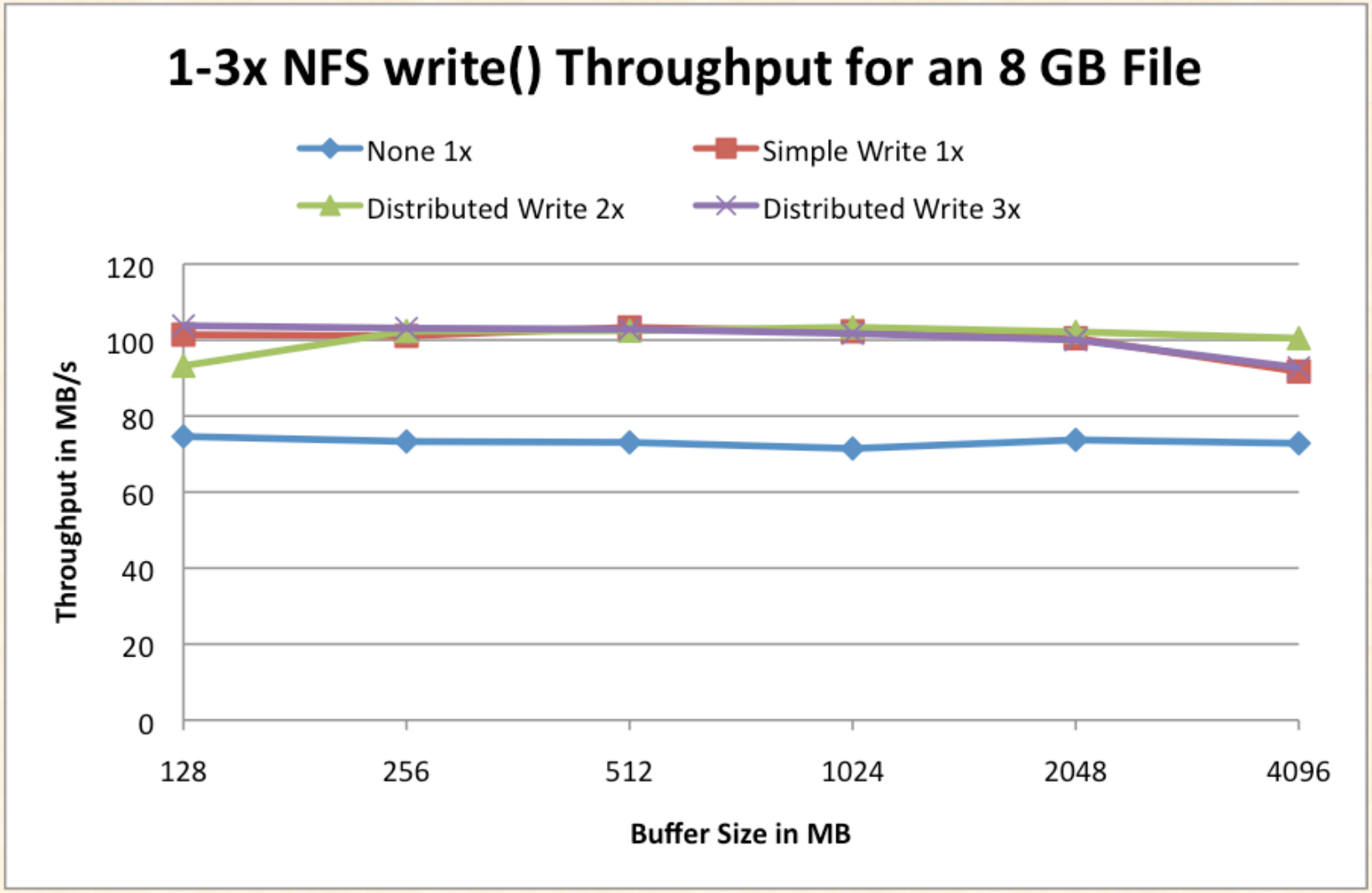
S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Performance: Single NFS Server Write



S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Performance: Redundant NFS Server Write



S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Conclusions

- Developed a proof-of-concept prototype intercepts file I/O calls from the MPI application and employs the needed coordination protocols to enable redundancy-oblivious or redundancy-aware file I/O
- Implementation is largely independent from the employed redundant MPI solution
- Results clearly indicate the performance impact under redundancy when accessing a single NFS server
- They also demonstrate the ability to offset this performance impact using parallel file I/O and MPI communication between replicas for data distribution

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.

Future Work

- A more comprehensive evaluation using HPC applications and larger-scale systems is needed
- A more mature implementation that can be used in production offering dynamic protocol selection based on application demands and system capabilities
- Built-in support for redundant applications in advanced parallel file systems, such as Lustre or PVFS

UT-BATTELLE

Questions?

S. Böhm and C. Engelmann. File I/O for MPI Applications in Redundant Execution Scenarios. PDP 2012, Garching, Feb. 15-17, 2012.