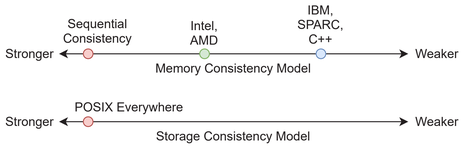
Consistency Models: the Parallel File System's Perspective

Chen Wang[1], Kathryn Mohror[1], Marc Snir[2]
Lawrence Livermore National Laboratory[1], University of Illinois Urbana-Champaign[2]

Motivation

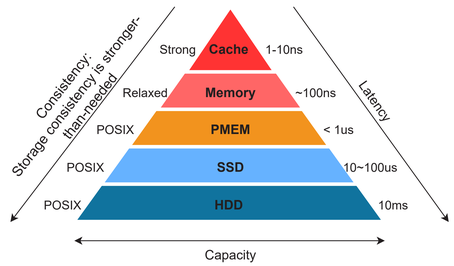
Storage consistency models have received much less attention compared to their counterparts in memory systems, particularly in the field of parallel file systems where consistency models have not evolved much since the universal adoption of the POSIX standard and its strict consistency model.



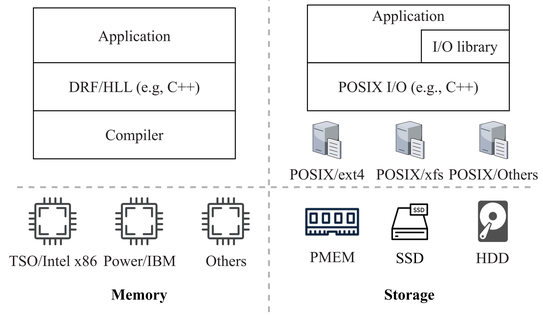
Recent advances in parallel storage techniques, such as fast node-local storage, have presented significant performance challenges to the use of the strict POSIX consistency model as the cost of maintaining it is becoming increasingly unmanageable.

Key Questions

- What are the reasons for the lack of attention to storage models compared to memory models?
- How do existing storage models compare and what commonalities exist among them? Can they be defined in a unified and formal manner?
- What are the performance implications of a storage model? How to effectively evaluate and compare the performance of different storage models?



Memory vs. Storage



Programming Hierarchy: The lack of compiler layer in storage programming hierarchy makes it harder to adopt different consistency models for different hardware.

Software Overhead: The software overhead incurred is minor compared to the slow I/O performance of HDDs. Consequently, less attention has been given to alternative consistency models.

Design Considerations: (1) Program Text (2) Reordering (3) Synchronization Information (4) Atomicity (5) Granularity

	Process 1:	Process 2:
Memory	$x = 100;$	$while(!flag)\{\};$
	$flag = 1$	$y = x;$

	Process 1:	Process 2:
Storage	$write(file);$	$recv(msg);$
	$send(msg);$	$read(file);$

A Unified Framework

Existing relaxed storage models: commit consistency, session consistency, MPI-IO consistency, etc.

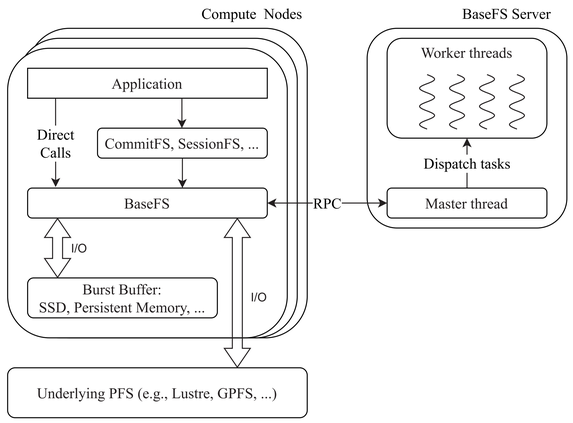
Sequential Consistency Normal Form: A consistency model is in sequential consistency normal form (SCNF) iff it guarantees sequential consistency to a set of formally-characterized programs.

Properly-synchronized Program: A program is properly synchronized iff for every execution of the program, all storage operations can be distinguished by the system as either data or synchronization, and there are no data races in the execution.

Properly-synchronized SCNF System: A system is said to be a properly-synchronized SCNF system iff the result of every run of a properly-synchronized program on the system is the result of a sequentially consistent execution of the program.

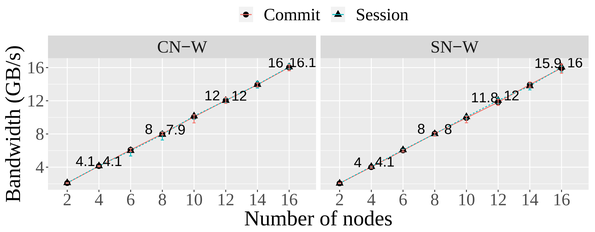
The Impact of Consistency Models on Performance

We present a "layered" implementation that allows for an easy performance comparison of different consistency models. We design a "base-layer" PFS, called BaseFS, which supports the basic functionalities of a PFS with essentially zero optimization. BaseFS provides a very *minimum consistency guarantee*, but it exposes a set of flexible primitives that can be used to implement custom consistency models. On top of BaseFS, we can implement PFSs providing different consistency models using these primitives. Since these PFSs use the set of primitives and thus the same underlying implementation, we can limit the impact of other components of the PFS to a very low level. Comparing the performance of these PFSs thus can give us a good understanding of the impact of different consistency models.

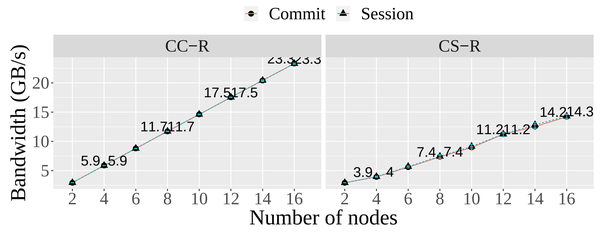


Conclusion: Consistency models can have a significant performance impact, especially for small access sizes and on fast devices.

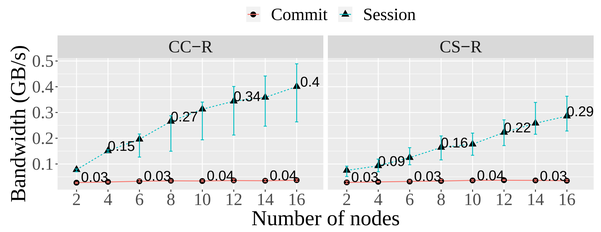
Benchmark: Write-only Workload (Large)



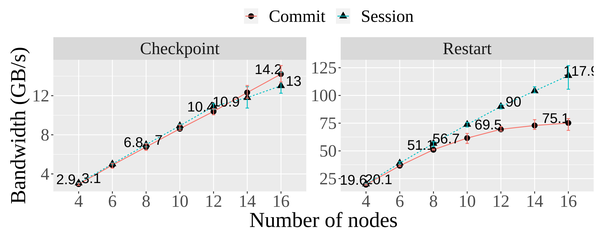
Benchmark: Read-after-write Workload (Large)



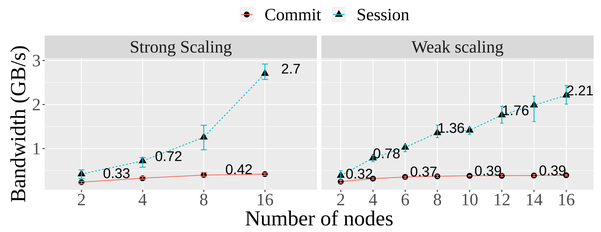
Benchmark: Read-after-write Workload (Small)



Case Study: Scalable Checkpoint/Restart



Case Study: Distributed Deep Learning



This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

LLNL-POST-847872