Low Design-Risk Checkpointing Storage Solution for Exascale Supercomputers

Nilmini Abeyratne and Trevor Mudge, University of Michigan, Ann Arbor, Michigan USA

Background & Introduction

Building exascale supercomputers require overcoming the **resilience challenge**. The standard resilience technique of checkpoint/restart is becoming more difficult because the number of failing components keep increasing and the amount of data to checkpoint is getting bigger.

To improve the speed of checkpointing, emerging non-volatile memory (phase change, magnetic, resistive RAM) has been proposed. But these unproven memories only increase the risk of designing exascale memory systems. This thesis shows that exascale memories can be constructed for **low design risk** using **commodity DRAM and SSD flash memory** and that newer non-volatile memories are unnecessary, at least for the next generation.

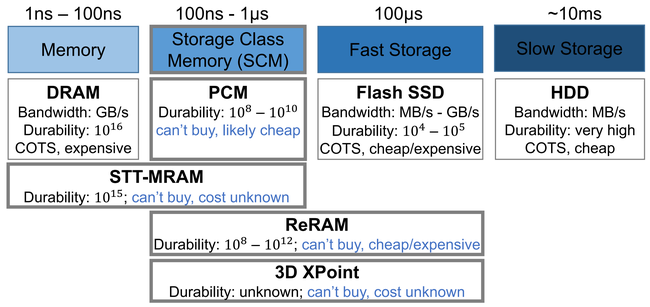


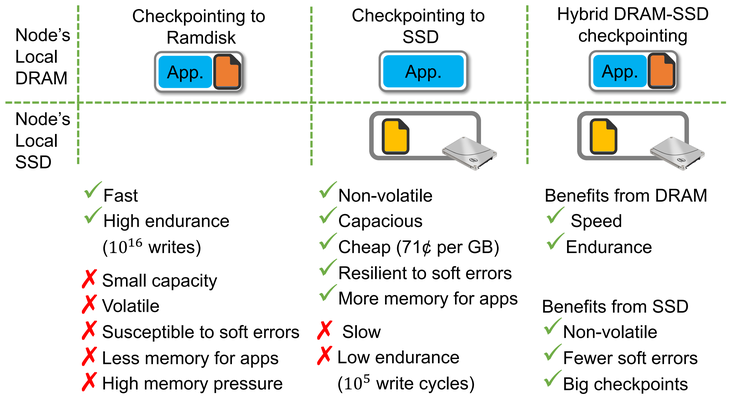
Figure: The landscape of memory technologies are organized according to their speed and type. Storage class memories (SCM) are non-volatile like flash SSDs and HDDs, but have properties similar to DRAM such as byte addressability. However, SCM's commonly share big unknowns such as

i) whether they can be economically manufactured,
ii) when they will be available on the market,
iii) how much they will cost per device,
iv) what types of risks they will pose once in the hands of consumers.

Existing Checkpointing Storage

This work presents a checkpointing framework based on commercial off-the-shelf (COTS) devices, namely DRAM and SSD.

Using only DRAM or only SSD in local checkpointing falls short of meeting the requirements of making a fast and reliable checkpoints. Rather than checkpointing to either device, this work proposes a hybrid scheme that checkpoints to both platforms.



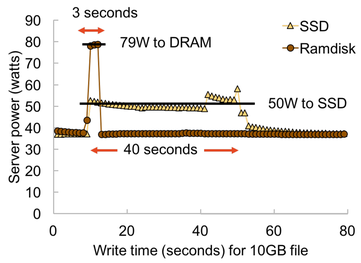


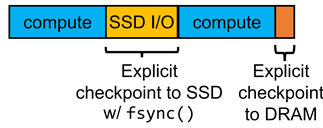
Figure: Writing a large checkpoint to the SSD device increases the server power to 50W and takes 40 seconds to complete. On the other hand, writing the same checkpoint to DRAM increases the server power to 79W, but only takes 3 seconds. Although DRAM write power is higher, it is more energy efficient because it is faster.

Diverting checkpoints to DRAM saves power!

Hybrid DRAM-SSD Checkpointing

The hybrid scheme selectively writes checkpoints to both DRAM and SSD.

When writing to an SSD storage device, the OS hides the I/O latency by writing the data to a temporary DRAM page cache. There is no guarantee the checkpoints are persisted to flash because the OS flushes the page cache at unknown times. This puts the checkpoint in danger of DRAM errors and failures.



Balance reliability and speed!

Figure: The hybrid scheme always flushes the page cache for the SSD checkpoints. The performance loss is balanced out writing the remaining checkpoints to DRAM.

Contributions

1. **A low-risk exascale memory system.** Uses mature COTS devices rather than waiting for newer newer non-volatile memory technologies are ready.
2. **Hybrid DRAM-SSD checkpointing.** Uses both local DRAM and local SSD flash memory to create fast and reliable checkpoints.
3. **SSD-lifetime-aware checkpoint controller.** An intelligent Checkpoint Location Controller (CLC) decides when to checkpoint to the SSD considering its endurance decay and the application's performance degradation.
4. **Dual-ECC memory.** Uses a normal ECC to protect regular application data and a strong ECC to protect the DRAM checkpoint. ECC-protected checkpoints ensure error-free restarts at recovery.

Checkpoint Location Controller (CLC)

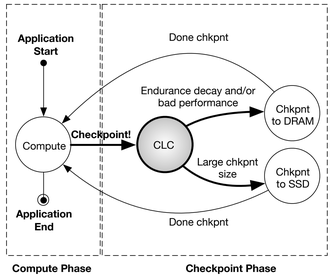


Figure: The CLC is evoked each time the application enters the checkpoint phase. The CLC dynamically decides the checkpoint location on each iteration.

1 SSD Endurance Decay Estimate:

If $\frac{PBW_{rating}}{B_{SSD}} < 5\ years$ then, choose ramdisk**

** PBW_{rating} is the SSD's endurance in petabytes written and B_{SSD} is the write bandwidth to the SSD

2 Performance Loss Estimate:

If $\frac{T_{chk}}{T_{compute} + T_{chk}} > tolerance$ then, choose ramdisk

3 Size Check:

If $sizeof(chkpnt) > RamdiskSz$ then, choose SSD

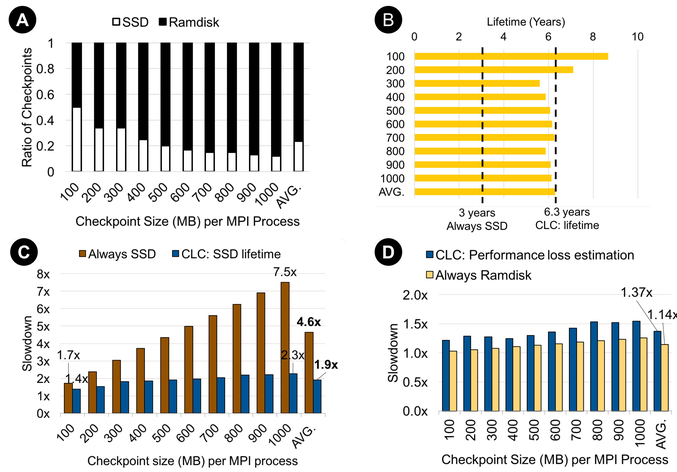


Figure: Microbenchmark-driven results sweeping checkpoint sizes from 100MB – 1GB per MPI process. (A) The hybrid scheme diverts select checkpoints to DRAM in order to reduce the number of writes to the SSD and increase its lifetime. (B) The hybrid scheme is able to increase SSD lifetime from 3 to 6.3 years for this microbenchmark. (C) They hybrid scheme only results in 1.9x application slowdown as opposed to 4.6x when always checkpointing to the SSD. (D) The CLC can further reduce performance loss to 1.3x. Always checkpointing to DRAM will still incur 1.1x slowdown, therefore slowdown due to checkpointing is unavoidable.

Dual-ECC Memory

Normal ECC – protects regular data with low overhead

- Access one x4 DIMM, retrieve 32 data + 4 ECC symbols
- RS(36,32)
- 4 ECC symbols gives several possible decoder options
  i. correct 2 symbols (bad because it only corrects a single-chip failure)
  ii. detect 4 symbols (bad because it has no correction capability)
  iii. correct 1 symbol, detect 3 symbols (good because it corrects simple bit/word errors, but also strongly detects double chip failures) → our choice

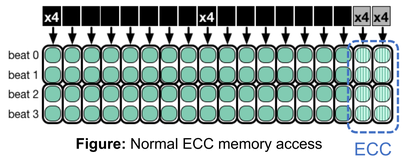


Figure: Normal ECC memory access

Strong ECC – protects checkpoint with Chipkill strength

- Access one x4 DIMM, retrieve 16 data + 2 ECC symbols
- RS(18,16) for detection-only up to 2 failed chips
- If-and-only-if a failure is detected, trigger a 2nd access to another x4 DIMM
  - Retrieve 1 additional parity symbol
  - RS(19,16) when combined with previous 2 symbols, correct 1 chip failure
  - Essentially Chipkill-correct, i.e. single-chip correction, double-chip detection

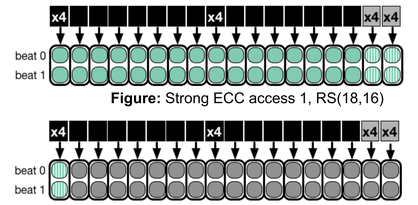


Figure: Strong ECC access 1, RS(18,16)

Figure: Strong ECC access 2, RS(19,16)

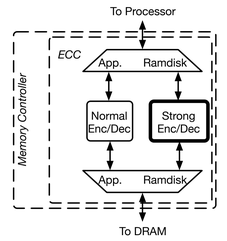


Figure: Modified Memory Controller with two decoders for normal and strong ECC.

Table: Synthesis results for proposed RS codes

	Normal ECC RS(36,32)	Strong ECC RS(18,16)	Strong ECC RS(19,16)
Syndrome calculation	0.48 ns	0.41 ns	0.41 ns
Single symbol correction, Double symbol detection	N/A	N/A	0.47 ns
Single symbol correction, Triple symbol detection	0.47 ns	N/A	N/A

Table: Error coverage of dual-ECC modes compared to Chipkill-Correct

	Failure mode	Normal RS(36,32)	Strong ECC RS(18,16)	Strong ECC RS(19,16)	Chipkill Correct
Single chip failures	1 bit, 1 word	100% correct	100% detect	100% correct	100% correct
	1 pin, 1 chip	100% detect			
Double chip failures	1 bit + {1 bit/word/pin/chip}	100% detect	100% detect	100% detect	100% detect
	1 word + {1 bit/word/pin/chip}	100% detect			
	1 pin + 1 pin	99.9999% detect, 0.0001% silent			
	1 chip + {1 pin, 1 chip}	99.9969% detect, 0.0031% silent			

Conclusion

- Fast and reliable local checkpointing will be imperative for the next generation exascale system.
- This work demonstrates a low-risk checkpointing storage solution that doesn't rely on speculative, emerging non-volatile memories. Instead, it uses COTS DRAM and SSD.
- This work presented a checkpoint controller, CLC, that preserves SSD endurance, and lowers performance loss due to checkpointing. In addition, a strong ECC is recommended to protect the checkpoints in DRAM.
- This framework will be valid for future hierarchical checkpointing storages solutions, including the eventual incorporation of emerging non-volatile memories.

References

1. N. Abeyratne, H-M. Chen, B. Oh, R. Dreslinski, C. Chakrabarti, T. Mudge, "Mitigating Risk in Designing Exascale Memory Systems. Memsys 2016, Washington, DC
2. U.S. Department of Energy Office of Science and National Nuclear Security Administration. "Preliminary Conceptual Design for an Exascale Computing Initiative," November 2014.