College of Computing

Department of Computer Science

RQE • Zhiyuan Lu

Thursday, March 4 | 4 to 5 pm

Improving the Performance of NVM Crash Consistency under Multicore

Non-Volatile Memory (NVM) systems require log schemes to ensure crash consistency, introducing severe performance overhead. Recently, LAD was proposed to remove log operations for transactions whose total amount of updated cachelines is smaller than the Asynchronous DRAM Refresh (ADR) buffer, without affecting crash consistency. However, on multicore, concurrent transactions tend to exhaust the ADR resource and hence log operations have to be conducted in LAD.

In this study, we observe that there are a significant number of log operations that could be avoided if a transaction run alone. To remove these unnecessary log operations, this paper presents a new transaction execution scheme, called two-stage transaction execution (TSTE), which allows the write requests of a transaction to be in both the ADR buffer and the staging SRAM buffer. Our new scheme performs log operations for a transaction's write requests in the SRAM buffer and executes in-place update operations for this transaction's write requests in the ADR buffer.

To further improve the ADR resource utilization, this paper also proposes virtual ADR buffers to decouple buffering from the ADR's reliable writing data; the proposed virtual ADR buffers only allow logless operations to access ADR resources. Additionally, this paper proposes to adopt redo log with DRAM cache to speed up the transaction commit speed. Our evaluation results demonstrate that our proposed schemes can efficiently reduce log operations up to 94.9% and improve the transaction throughput up to 78.6%.

1885

Michigan Tech

College of Computing

Because the future is computational.

mtu.edu/computing