

# SDI/ISTC Seminar



**Thursday**  
**July 25, 2019**  
**GHC 8102**  
**12:00 - 1:00 pm**

## Persistent Buffer Management With Optimistic Consistency



**Lucas Lersch**  
*TU Dresden*

Lucas Lersch is a fourth year PhD student at TU Dresden, working in close collaboration with SAP SE. His PhD work consists in exploring opportunities to leverage non-volatile memory technologies in modern data systems. Through his work he gained in-depth insights into the challenges of NVM and investigated key-value store designs, log-structured storage, persistent data structures, transactional recovery, and enabling low and predictable tail-latencies in key-value stores. He received his MSc in Computer Science from the TU Kaiserslautern (Germany) in 2015 and BSc in Computer Science from the UFRGS (Brazil) in 2013.

Non-volatile memory technologies (NVM) enable persistent media to be directly accessed by the CPU through its caches. The biggest challenge introduced by NVM is the little control the application has when persisting data. This stems from the fact that it is not possible to prevent data from being evicted from the CPU cache to NVM at arbitrary points in time, possibly leading to corruption. To deal with this problem, work so far treat NVM either like storage (by always writing to DRAM and prohibiting direct writes to NVM) or treat NVM like memory (by pessimistically enforcing fine-granular consistency at all times through SFENCE and CLFLUSH). In this talk we make a few observations. First, allowing direct writes to NVM is essential to avoid unnecessary traffic between DRAM and NVM in the memory bus. Second, system failures are the exception not the norm. Third, if no consistency is neurotically enforced, the amount of corrupted data is bound by the size of the CPU cache (i.e., stores that were not persisted). Fourth, modern database systems already offer most of the building blocks for recovering corrupted media. Based on these observations, we explore the memory-storage duality of NVM by integrating it into the buffer pool of database systems. The core idea is to embrace corruption and directly write to NVM like memory during runtime while treating NVM like storage during recovery time. That is achieved by combining the use of checksums, for detecting page corruptions, with ARIES-style recovery, for repairing these corruptions. The main benefit of our approach is an easy integration into existing database systems while enabling a gradual trade-off between higher performance (more DRAM) and faster peak-performance recovery and lower costs (more NVM).

Partially funded by:  
**YAHOO! LABS**



VISITOR HOST: Andy Pavlo, [pavlo@cs.cmu.edu](mailto:pavlo@cs.cmu.edu)  
For more information or questions:  
Karen Lindenfelser, 8-6716, [karen@ece.cmu.edu](mailto:karen@ece.cmu.edu)  
<http://www.pdl.cmu.edu/SDI/>

