Electrical & Computer ENGINEERING

Carnegie Mellon University

Graduate Seminar

Hiding the Long Latency of Persist Barriers Using Speculative Execution

Yan Solihin

Program Director, Computer and Network Systems at NSF
Professor, ECE Department
North Carolina State University

Thursday, August 31st 4:30 PM Scaife Hall 125

ECE Seminar Committee

Aswin Sankaranarayanan
saswin@ece.cmu.edu

Maysam Chamanzar
mchamanz@andrew.cmu.edu

Swarun Kumar
swarun@cmu.edu

Abstract:

Byte-addressable non-volatile memory technology is emerging as an alternative for DRAM for main memory. This new Non-Volatile Main Memory (NVMM) allows programmers to store important data in data structures in memory instead of serializing it to the file system, thereby providing a substantial performance boost. However, computer systems reorder memory operations and utilize volatile caches for better performance, making it difficult to ensure a consistent state in NVMM. Intel recently announced a new set of persistence instructions, clflushopt, clwb, and pcommit. These new instructions make it possible to implement fail-safe code on NVMM, but few workloads have been written or characterized using these new instructions.

In this talk, I will discuss my two recent works in the area. First, we describe how these instructions work and how they can be used to implement write-ahead logging based transactions. We implement several common data structures and kernels and evaluate the performance overhead incurred over traditional non-persistent implementations. In particular, we find that persistence instructions occur in clusters along with expensive fence operations, they have long latency, and they add a significant execution time overhead, on average by 20.3% over code with logging but without fence instructions to order persists. To deal with this overhead and alleviate the performance bottleneck, we propose to speculate past long latency persistency operations using checkpoint-based processing. Our speculative persistence architecture reduces the execution time overheads to only 3.6%.

Second, I will discuss a new logging approach for durable transactions that achieves the favorable characteristics of both prior software and hardware approaches. Like software, it has no hardware constraint limiting the number of transactions or logs available to it, and like hardware, it has very low overhead. Our approach introduces two new instructions: one that indicates whether a load instruction should create a log entry, and a log flush instruction to make a copy of a cache line in the log. We add hardware support, primarily within the core, to manage the execution of these instructions and critical ordering requirements between logging operations and updates to data. We also propose a novel optimization at the memory controller that is enabled by a battery backed write pending queue in the memory controller. We implemented our design on a cycle accurate simulator, MarssX86, and compared it against state-of-the-art hardware logging (ATOM) and a software only approach. Our experiments show that Proteus improves performance by 1.48×, on average, compared to a system without hardware logging and 10.5% faster than ATOM. A significant advantage of our approach is dropping writes to the log when they are not needed. On average, ATOM makes 2.11× more writes to memory than our design.

Bio:

Yan Solihin obtained his B.S. degree in computer science from Institut Teknologi Bandung in 1995, B.S. degree in Mathematics from Universitas Terbuka Indonesia in 1995, M.A.Sc degree in computer engineering from Nanyang Technological University in 1997, and M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1999 and 2002. His research interests include computer architecture, memory hierarchy design, non-volatile memory architecture, programming models, and workload cloning.

He is a Program Director at the Division of Computer and Network Systems (CNS) at the National Science Foundation. His responsibilities include managing the Computer Systems Research (CSR) cluster, Scalable Parallelism in the eXtreme (SPX), and Secure and Trustworthy Cyberspace (SaTC), among others. He is also a Professor of Electrical and Computer Engineering at North Carolina State University. He has written two graduate-level textbooks, including Fundamentals of Parallel Multicore Architecture, CRC Press, 2015. He has published more than 50 papers in computer architecture and performance modeling. He has released several software packages to the public: ACAPP - a cache performance model toolset, HeapServer - a secure heap management library, Scaltool - parallel program scalability pinpointer, and Fodex - a forensic document examination toolset.

He is a recipient of 2010 and 2005 IBM Faculty Partnership Award, 2004 NSF Faculty Early Career Award, and 1997 AT&T Leadership Award. He is listed in the HPCA Hall of Fame. He is also a senior member of the IEEE.

SEMINAR NOTES: (REFRESHMENTS SERVED AT 4 PM)