

Chang Hyun Park

changhyunpark@calab.kaist.ac.kr

<https://iamchanghyunpark.github.io/>

(cell) +82 10 6473 1280

RESEARCH INTERESTS

Interested in architectural support for system softwares: both native and virtualized.

Interested in the virtual memory system and address translation, cache and memory systems.

Looking for opportunities to expand research (virtual memory system & address translation) to emerging memory systems such as stacked DRAM and NVRAM.

Recent interest in TLB improving TLB shoot downs on state of art systems, and TLB organization on GPUs/Accelerators.

EDUCATION

KAIST, Daejeon, Korea - Advisor: [Professor Jaehyuk Huh](#)

M.S. in Computer Science

February 2015

Ph.D., in Computer Science

(Expected) August 2019

Hanyang University, Seoul, Korea

B.S. in Computer Science and Engineering

Graduated Summa Cum Laude

February 2013

PUBLICATIONS

Published

Jungi Jeong, Chang Hyun Park, Jaehyuk Huh, Seungryul Maeng, "Efficient Hardware-assisted Logging with Asynchronous and Direct Update for Persistent Memory", The 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), October 2018.

Jeongseob Ahn, Chang Hyun Park, Taekyung Heo, Jaehyuk Huh, "Accelerating Critical OS Services in Virtualized Systems with Flexible Micro-sliced Cores", The European Conference on Computer Systems (EuroSys), April 2018.

Chang Hyun Park, Taekyung Huh, Jungi Jeong, Jaehyuk Huh, "Hybrid TLB Coalescing: Improving TLB Translation Coverage under Diverse Fragmented Memory Allocations" The 44th International Symposium on Computer Architecture (ISCA), June 2017

Daehoon Kim, Chang Hyun Park, Hwanju Kim, and Jaehyuk Huh, "Virtual Snooping Coherence for Multi-core Virtualized Systems", IEEE Transactions on Parallel and Distributed Systems (TPDS), July 2016

Chang Hyun Park, Taekyung Heo, and Jaehyuk Huh, "Efficient Synonym Filtering and Scalable Delayed Translation for Hybrid Virtual Caching", the International Symposium on Computer Architecture (ISCA), June 2016

Jeongseob Ahn, Chang Hyun Park, and Jaehyuk Huh, "Micro-sliced Virtual Processors to Hide the Effect of Discontinuous CPU Availability for Consolidated Systems", The 47th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), December 2014

Submission

Chang Hyun Park, Jungi Jeong, Jaehyuk Huh, "Trusted SSD Architecture for Data Availability Protection", Under peer review at the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Submitted September 2018.

Jinglei Ren, Chang Hyun Park, Mengxing Liu, Samira Khan, Thomas Moscibroda, "FAUR Protocol: When Crash Consistency Meets Fault Tolerance", Will be submitted to USENIX Annual Technical Conference (ATC), June 2019.

RESEARCH EXPERIENCE (BY IMPORTANCE)

Providing a TLB with adjustable coverage for diverse system/application mapping requirements

- **Improve TLB performance for heterogenous memory systems (DRAM, Stacked DRAM, NVM, etc.) with varying characteristics**
- Proposed a Hardware-Software joint effort to provide a TLB with adjustable TLB coverage
- Prior work are not prepared for such heterogenous memory systems
 - Rise of non-uniform memory, memory heterogeneity (3D-stacked memory, Non-volatile memory, etc.) require differing memory mappings per system memory configuration.
 - For example NUMA does not work well with large pages (Baptiste ATC '14)
 - Inter/Intra application memory mappings show significant difference of memory mapping
- A TLB scheme that performs well across the diverse memory mapping is necessary.
- We propose a hardware-software hybrid TLB coalescing, where:
 - The HW offers a TLB that can provide adjustable TLB coverage using the contiguity information encoded into the page table by the System SW
 - The SW chooses the best TLB coverage for each application, configures the HW to abide by the coverage, and writes contiguity information onto the unused portions of the page table entry
- **Published and Presented at ISCA 2017**

Speeding up virtual address translation whilst lowering power consumption

- **Improve Address translation scalability by delaying address translation to after cache miss, for the majority translations.**
- Proposed the use of virtually and physically addressed caches to reduce the majority of TLB lookups for every memory access to the cache hierarchy. Entire cache hierarchy is used as a hybrid addressed cache.
- Proposed a system which filters less common synonym causing memory accesses using simple filters. Every memory request accesses the filter prior to accessing cache or TLB.
 - For any synonym causing access, the TLB is looked up and cache is accessed in physical address (to prevent synonym problems).
 - For the majority of memory accesses which are synonym free, virtual address is used to access the cache hierarchy.
- Any last level cache misses of virtually addressed memory requests translate to physical address prior to accessing the main memory. A TLB like structure or proposed scalable translation structure buffer translations.
- Proposed system reduces power consumption and improves system performance.
- **Published and Presented at ISCA 2016.**

Providing Fault tolerance and Crash consistency to Storage Class Memory: Extending RAID

- **Leverage the fault tolerance of RAID to minimize logging write overhead in distributed NVM systems.**
- Made a case for the requirement of a system that provides fault tolerance and crash consistency, when employing storage class memory (SCM) with Non-volatile memories (NVM).
- Proposed adding fault tolerance to SCMs
 - NVM provides persistence, data is expected to be safe. However memory errors and failures result in possible loss of data.
 - Made a case of reusing RAID for SCMs to provide tolerance of random NVM failures.
- Proposed extending RAID abstraction to include crash consistency.
 - All NVM employing systems need to provide crash consistency, preventing any data inconsistency from a failure that occurs in the middle of a write.
 - Most trivial solution is journaling which backs up all data to be written, then updates the actual data.

- Currently filesystems provide crash consistency for disks; when memory is a storage class memory, a new journaling abstraction is required for ease of programming.
- NVMs have write endurance problems, thus proposed reducing journal writes via theorem.
- Made a case to use RAID technique in a distributed storage network over high performance network (Infiniband).
 - RAID allows distributing data and parity to different disks.
 - Proposed using the feature to distribute data to different nodes over Infiniband.
- Participated as a second author, Main developer of the SW system.
- **Will be submitted to ATC 2019.**

Proposing a HW logging mechanism that supports atomic durability for programmability and performance for transactions in NVMs

- Proposed a HW logging mechanism based on both redo logging.
- HW logging of transactions for NVMs provides three benefits
 1. Ease of programming: Programmer just specifies which data to log. HW does the logging.
 2. Performance benefits: HW approach optimizes the logging and does not pollute the d-cache or i-cache with unnecessary data/instructions.
 3. Provides crash consistency: Non-volatile memory (NVMs) require special attention to keep data consistent, even in the event of a power failure. To make sure data is recoverable during a power failure, crash consistency is required.
- The contribution of this work is that we propose a direct-asynchronous approach to logging. Direct meaning that the data is persisted into the actual data in the NVM from the cache hierarchy, and asynchronous meaning that the actual persisting of the data is not on the critical path of the transaction commit.
- To achieve such properties, we introduce a small DRAM-cache that buffers writes to prevent early write-backs, and we track updated cache-lines to flush when the related transaction is being committed.
- Participated as a second author, providing valuable intuitions and advices.
- **To be published and presented in MICRO 2018**

Restricting cache coherence snooping domain for virtualized systems

- Snooping protocol has scalability and power issues in large systems. Virtualized systems isolate memory among virtual machines providing an opportunity to isolate snooping requests among virtual machine domains.
- Snoop requests only sent to physical cores which hosted the virtual CPUs of the current snoop domain (virtual machine domain).
- Studied the snooping effects on spatial scheduling of virtual CPUs: restricting relocation of virtual CPUs to subset of physical cores.
- Participated as a second author, providing valuable intuitions and advices.
- Published in TPDS 2016.

Improving execution of workloads suffering virtualization induced anomalies

- Introduced the virtual time discontinuity problem, which encompasses anomalies induced by virtualization such as lock preemptions or delayed interrupt problems.
- Proposed a holistic solution to the virtual time discontinuity problem by shortening the hypervisor scheduling time.
- Proposed a HW cache preservation and restoration mechanism to account for performance suffered by cache sensitive workloads which benefit from longer time slices.
- Participated as a second author, providing valuable intuitions and advices.
- Published in MICRO 2014.

INDUSTRY EXPERIENCE

- Microsoft Research Asia, Beijing** Summer of 2016
Proposed fault tolerance and crash consistency abstraction to Storage Class Memories
Submission of work to SOSP 2017 (See Research Experience)
- Samsung Electronics, Korea** Summer of 2012
Refactored and shortened bootup code of *Uboot* bootloader used by Tizen based phones
Tizen bootloader team of Mobile Comm. Business

AWARDS & SCHOLARSHIPS

KAIST:

National Scholarship: Full scholarship March 2013 - February 2019 (expected)

Hanyang University:

Academic Scholarship: Full scholarship (3 semesters), Partial scholarship (3) semesters
Honorable mention at the Capstone Design Fair 2012

Korea Foundation for Advanced Studies:

[EE and CS Graduate Student Scholarship](#) 3 Years beginning April 2016

Microsoft Research Asia:

[Microsoft Research Asia PhD Fellowship 2017](#) October 2017

Naver:

Naver PhD Fellowship 2017 October 2017

TEACHING EXPERIENCE

KAIST Teaching Assistant:

- CS230 System Programming
- CS311 Computer Organization
- CS510 Graduate Computer Architecture
- CC510 Introduction to Computer Application (Graduate, non-CS students)

TECHNICAL SKILLS

Programming Languages: C/C++, Python, Java, Bash shell

System Software:

- Xen hypervisor: modified credit scheduler code and profiled vCPU scheduling
- Linux kernel: modified memory management code

Simulators & Tools:

- Marssx86: implemented address translation after last level cache misses
- Intel Pintool: used to create an in house simulator of TLB and cache

REFERENCES

Available upon request