

# CHANGWOO MIN

Room 455, Durham Hall,  
Blacksburg, VA 24060

*The Department of Electrical and Computer Engineering*  
*Virginia Tech*

(540) 231-4580  
changwoo@vt.edu  
<https://multics69.github.io/>

## Contents

<b>1 Research Interests</b>	<b>1</b>
<b>2 Employment History</b>	<b>1</b>
<b>3 Education</b>	<b>2</b>
<b>4 Honors and Awards</b>	<b>2</b>
<b>5 Research, Scholarship, and Creative Activities</b>	<b>2</b>
5.1 Publication . . . . .	2
5.1.1 Conference Publications . . . . .	2
5.1.2 Journal Publications . . . . .	7
5.1.3 Posters . . . . .	9
5.2 Open Source Softwares . . . . .	10
5.3 Patent . . . . .	10
5.4 Invited Talks and Presentations . . . . .	11
5.5 Work Experience before Starting Ph.D. Study (1998–2009) . . . . .	12
<b>6 Teaching</b>	<b>13</b>
6.1 Courses Taught . . . . .	13
6.1.1 Service on Thesis or Dissertation Committees . . . . .	15
<b>7 Service</b>	<b>16</b>
7.1 Conference Committee Activities . . . . .	16
7.2 Journal Reviewing Activities . . . . .	17
7.3 Funding Agency Panel Activities . . . . .	17
7.4 Memberships and Activities in Professional Societies . . . . .	17

## 1 Research Interests

Operating Systems, Storage Systems, Parallel and Distributed Systems, System Security

## 2 Employment History

08/2017–present	<b>Assistant Professor</b> , The Department of Electrical and Computer Engineering, Virginia Tech . . . . . Blacksburg, VA, USA
11/2016–07/2017	<b>Research Scientist</b> , School of Computer Science, Georgia Tech . . . . Atlanta, GA, USA
11/2014–10/2016	<b>Postdoctoral Fellow</b> , School of Computer Science, Georgia Tech . . Atlanta, GA, USA
09/2014–11/2014	<b>Postdoctoral Fellow</b> , Computer Science, Sungkyunkwan University . . . . Suwon, Korea
12/2005–07/2014	<b>Principal S/W Engineer</b> , Samsung Electronics . . . . . Suwon, Korea
01/1998–11/2005	<b>Staff R&amp;D Engineer</b> , IBM . . . . . Seoul, Korea

### 3 Education

<b>Ph.D.</b>	Mobile Systems Engineering Advisor: Dr. Young Ik Eom Dissertation: DANBI: A Programming Model and Runtime for Dynamic and Scalable Stream Parallelism Sungkyunkwan University, Korea	03/2010 – 02/2014
<b>M.S.</b>	Computer Science Advisor: Dr. Myung Won Kim Thesis: Compact Fuzzy Rule Generation Algorithm for Data Mining Soongsil University, Korea	03/1996 – 02/1998
<b>B.S.</b>	Computer Science Soongsil University, Korea	03/1992 – 02/1996

### 4 Honors and Awards

1.	<b>Faculty Fellow Award.</b> College of Engineering at Virginia Tech	2022
2.	<b>Memorable Paper Award Finalist.</b> NVMW	2021
3.	<b>Distinguished Reviewer.</b> ACM Transactions on Storage (TOS)	2019, 2020, 2021
4.	<b>Best paper award.</b> VEE	2019
5.	<b>100 Future Technologies and Leaders.</b> The National Academy of Engineering of Korea (NAEK)	2017
6.	<b>Best student paper award.</b> EuroSys	2017
7.	<b>Top 10 finalist.</b> CSAW Applied Research Competition	2016
8.	<b>Outstanding Post-Doctoral Research Award.</b> College of Computing, Georgia Institute of Technology	2016
9.	<b>First place best paper, Chester W Sall Memorial Awards.</b> IEEE Consumer Electronics Society	2016
10.	<b>Post-doctoral research fellowship.</b> National Research Foundation of Korea (NRF)	2015
11.	<b>Best papers award.</b> APSYS	2015
12.	<b>Second place, ACM Student Research Competition (SRC).</b> PACT	2013

### 5 Research, Scholarship, and Creative Activities

#### 5.1 Publication

##### 5.1.1 Conference Publications

1. **Write-Light Cache for Energy Harvesting Systems.**  
Jongouk Choi, Jianping Zeng, Dongyoon Lee, Changwoo Min, and Changhee Jung.  
In Proceedings of *International Symposium on Computer Architecture (ISCA-50)*.

2. **Protect the System Call, Protect (most of) the World with BASTION.**  
Christopher Jelesnianski, Mohannad Ismail, Yeongjin Jang, Dan Williams, and Changwoo Min.  
In Proceedings of *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2023)*.
3. **PRISM: Optimizing Key-Value Store for Modern Heterogeneous Storage Devices.**  
Yongju Song, Wook-Hee Kim, Sumit Kumar Monga, Changwoo Min, and Young Ik Eom.  
In Proceedings of *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2023)*.
4. **Odinfs: Scaling PM performance with Opportunistic Delegation.**  
Diyu Zhou, Yuchen Qian, Vishal Gupta, Zhifei Yang, Changwoo Min, and Sanidhya Kashyap.  
In Proceedings of *the 14th annual non-volatile memories workshop (NVMW 2023)*.
5. **TENET: Memory Safe and Fault tolerant Persistent Transactional Memory.**  
R. Madhava Krishnan, Diyu Zhou, Wook-Hee Kim, Sudarsun Kannan, Sanidhya Kashyap, and Changwoo Min.  
In Proceedings of *USENIX Conference on File and Storage Technologies (FAST 2023)*.
6. **CJFS: Concurrent Journaling for Better Scalability.**  
Joontaek Oh, Seung Won Yoo, Hojin Nam, Changwoo Min, and Youjip Won.  
In Proceedings of *USENIX Conference on File and Storage Technologies (FAST 2023)*.
7. **Tightly Seal Your Sensitive Pointers with PACTight.**  
Mohannad Ismail, Andrew Quach, Christopher Jelesnianski, Yeongjin Jang, and Changwoo Min.  
In Proceedings of *the 31st USENIX Security Symposium (Security 2022)*.
8. **Durinn: Adversarial Memory and Thread Interleaving for Detecting Durable Linearizability Bugs.**  
Xinwei Fu, Dongyoon Lee, and Changwoo Min.  
In Proceedings of *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2022)*.
9. **Odinfs: Scaling PM performance with Opportunistic Delegation.**  
Diyu Zhou, Yuchen Qian, Vishal Gupta, Zhifei Yang, Changwoo Min, and Sanidhya Kashyap.  
In Proceedings of *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2022)*.
10. **ReplayCache: Enabling Volatile Caches for Energy Harvesting Systems.**  
Jianping Zeng, Jongouk Choi, Xinwei Fu, Ajay Paddayuru Shreepathi, Dongyoon Lee, Changwoo Min, and Changhee Jung.  
In Proceedings of *the 13rd annual non-volatile memories workshop (NVMW 2022)*.
11. **Fireworks: A Fast, Efficient, and Safe Serverless Framework using VM-level post-JIT Snapshot.**  
Wonseok Shin, Wook-Hee Kim, and Changwoo Min.  
In Proceedings of *ACM European Conference on Computer Systems (EuroSys 2022)*.
12. **Witcher: Systematic Crash Consistency Testing for Non-Volatile Memory Key-Value Stores.**  
Xinwei Fu, Wook-Hee Kim, Ajay Paddayuru Shreepathi, Mohannad Ismail, Sunny Wadkar, Dongyoon Lee, and Changwoo Min.  
In Proceedings of *ACM Symposium on Operating Systems Principles (SOSP 2021)*.
13. **PACTree: A High Performance Persistent Range Index Using PAC Guidelines.**  
Wook-Hee Kim, R. Madhava Krishnan, Xinwei Fu, Sanidhya Kashyap, and Changwoo Min.  
In Proceedings of *ACM Symposium on Operating Systems Principles (SOSP 2021)*.
14. **Birds of a Feather Flock Together : Scaling RDMA RPCs with FLOCK.**  
Sumit Kumar Monga, Sanidhya Kashyap, and Changwoo Min.  
In Proceedings of *ACM Symposium on Operating Systems Principles (SOSP 2021)*.
15. **VIP: Safeguard Value Invariant Property for Thwarting Critical Memory Corruption Attacks.**  
Mohannad Ismail+, Jinwoo Yom+, Christopher Jelesnianski, Yeongjin Jang, and Changwoo Min.  
In Proceedings of *Conference on Computer and Communications Security (CCS 2021)*.

16. **ReplayCache: Enabling Volatile Caches for Energy Harvesting Systems.**  
Jianping Zeng, Jongouk Choi, Xinwei Fu, Ajay Paddayuru Shreepathi, Dongyoon Lee, Changwoo Min, and Changhee Jung.  
In Proceedings of *Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-54)*.
17. **TIPS: Making Volatile Index Structures Persistent with DRAM-NVMM Tiering.**  
R. Madhava Krishnan, Wook-Hee Kim, Xinwei Fu, Sumit Kumar Monga, Hee Won Lee, Minsung Jang, Ajit Mathew, and Changwoo Min.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2021)*.
18. **LODIC: Logical Distributed Counting for Scalable File Access.**  
Jeoungahn Park, Taeho Hwang, Jongmoo Choi, Changwoo Min, and Youjip Won.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2021)*.
19. **CrossFS: A Cross-layered Direct-Access File System.**  
Yujie Ren, Changwoo Min, and Sudarsun Kannan.  
In Proceedings of *the 12th annual non-volatile memories workshop (NVMW 2021)*.
20. **POSEIDON : Safe, Fast and Scalable Persistent Memory Allocator.**  
Wook-Hee Kim, Anthony Demeri, R. Madhava Krishnan, Jaeho Kim, Mohannad Ismail, and Changwoo Min.  
In Proceedings of *the 12th Annual Non-Volatile Memories Workshop (NVMW 2021)*.
21. **Making Volatile Index Structures Persistent using TIPS.**  
R. Madhava Krishnan, Wook-Hee Kim, Hee Won Lee, Minsung Jang, Sumit Kumar Monga, Ajit Mathew, and Changwoo Min.  
In Proceedings of *the 12th annual non-volatile memories workshop (NVMW 2021)*.
22. **ExpRace: Exploiting Kernel Races through Raising Interrupts.**  
Yoochan Lee, Changwoo Min, and Byoungyoung Lee.  
In Proceedings of *the 30th USENIX Security Symposium (Security 2021)*.
23. **POSEIDON: Safe, Fast and Scalable Persistent Memory Allocator.**  
Anthony Demeri, Wook-Hee Kim, R. Madhava Krishnan, Jaeho Kim, Mohannad Ismail, and Changwoo Min.  
In Proceedings of *the 21st ACM/IFIP International Middleware Conference (Middleware 2020)*.
24. **An OpenMP Runtime for Transparent Work Sharing Across Cache-Incoherent Heterogeneous Nodes.**  
Robert Lyerly, Changwoo Min, Christopher J. Rossbach, and Binoy Ravindran.  
In Proceedings of *the 21st ACM/IFIP International Middleware Conference (Middleware 2020)*.
25. **CrossFS: A Cross-layered Direct-Access File System.**  
Yujie Ren, Changwoo Min, and Sudarsun Kannan.  
In Proceedings of *the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2020)*.
26. **Exploiting Kernel Races through Taming Thread Interleaving.**  
Yoochan Lee, Byoungyoung Lee, and Changwoo Min.  
*BlackHat USA (2020)*.
27. **MARDU: Efficient and Scalable Code Re-randomization.**  
Christopher Jelesnianski, Jinwoo Yom, Changwoo Min, and Yeongjin Jang.  
In Proceedings of *the 13th ACM International Systems and Storage Conference (SYSTOR 2020)*.
28. **DEX: Scaling Applications Beyond Machine Boundaries.**  
Sang-Hoon Kim, Ho-Ren Chuang, Robert Lyerly, Pierre Olivier, Changwoo Min and Binoy Ravindran.  
In Proceedings of *the 40th IEEE International Conference on Distributed Computing Systems (ICDCS 2020)*.
29. **HydraList: A Scalable In-Memory Index Using Asynchronous Updates and Partial Replication.**  
Ajit Mathew and Changwoo Min.  
In Proceedings of *Endowment, 13(9) (VLDB 2020)*.
30. **Durable Transactional Memory Can Scale with TimeStone.**  
R.Madhava Krishnan, Jaeho Kim, Ajit Mathew, Xinwei Fu, Anthony Demeri, Changwoo Min, and Sudarsun Kannan.  
In Proceedings of *the 11th Annual Non-Volatile Memories Workshop (NVMW 2020)*.

31. **Durable Transactional Memory Can Scale with TimeStone.**  
R.Madhava Krishnan, Jaeho Kim, Ajit Mathew, Anthony Demeri, Xinwei Fu, Changwoo Min, and Sudarsun Kannan.  
In Proceedings of *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2020)*.
32. **Scalable and Practical Locking with Shuffling.**  
Sanidhya Kashyap, Irina Calciu, Xiaohe Cheng, Changwoo Min, and Taesoo Kim.  
In Proceedings of *ACM Symposium on Operating Systems Principles (SOSP 2019)*.
33. **Alleviating Garbage Collection Interference Through Spatial Separation in All Flash Arrays.**  
Jaeho Kim, Kwanghyun Lim, Youngdon Jung, Sungjin Lee, Changwoo Min, and Sam H. Noh.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2019)*.
34. **A Binary-Compatible Unikernel.**  
Pierre Olivier, Daniel Chiba, Stefan Lankes, Changwoo Min, and Binoy Ravindran.  
In Proceedings of *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2019)*.
35. **MV-RLU: Scaling Read-Log-Update with Multi-Versioning.**  
Jaeho Kim+, Ajit Mathew+, Sanidhya Kashyap, Madhava Krishnan Ramanathan, and Changwoo Min.  
In Proceedings of *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2019)*.
36. **When Address Remapping Techniques Meet Consistency Guarantee Mechanisms.**  
Dong Hyun Kang, Gihwan Oh, Dongki Kim, In Hwan Doh, Changwoo Min, Sang-Won Lee, and Young Ik Eom.  
In Proceedings of *Workshop on Hot Topics in Storage and File Systems (HotStorage 2018)*.
37. **Scaling Guest OS Critical Sections With eCS.**  
Sanidhya Kashyap, Changwoo Min, and Taesoo Kim.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2018)*.
38. **A Scalable Ordering Primitive for Multicore Machines.**  
Sanidhya Kashyap, Changwoo Min, Kangnyeon Kim, and Taesoo Kim.  
In Proceedings of *ACM European Conference on Computer Systems (EuroSys 2018)*.
39. **SOLROS: A Data-Centric Operating System Architecture for Heterogeneous Computing.**  
Changwoo Min, Woonhak Kang, Mohan Kumar, Sanidhya Kashyap, Steffen Maass, and Taesoo Kim.  
In Proceedings of *ACM European Conference on Computer Systems (EuroSys 2018)*.
40. **FLSCHED: A Lockless and Lightweight Approach to OS Scheduler for Xeon Phi.**  
Heeseung Jo and Woonhak Kang and Changwoo Min and Taesoo Kim.  
In Proceedings of *Asia-Pacific Workshop on Systems (APSys 2017)*.
41. **Designing New Operating Primitives to Improve Fuzzing Performance.**  
Wen Xu, Sanidhya Kashyap, Changwoo Min, and Taesoo Kim.  
In Proceedings of *ACM Conference on Computer and Communications Security (CCS 2017)*.
42. **Scalable NUMA-aware Blocking Synchronization Primitives.**  
Sanidhya Kashyap, Changwoo Min, and Taesoo Kim.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2017)*.
43. **Mosaic: Processing a Trillion-Edge Graph on a Single Machine.**  
Steffen Maass, Changwoo Min, Sanidhya Kashyap, Woonhak Kang, Mohan Kumar, and Taesoo Kim.  
In Proceedings of *ACM European Conference on Computer Systems (EuroSys 2017)*.
44. **APISan: Sanitizing API Usages through Semantic Cross-checking.**  
Insu Yun, Changwoo Min, Xujie Si, Yeongjin Jang, Taesoo Kim, and Mayur Naik.  
In Proceedings of *USENIX Security Symposium (Security 2016)*.
45. **Instant OS Updates via Userspace Checkpoint-and-Restart.**  
Sanidhya Kashyap, Changwoo Min, Byoungyoung Lee, Taesoo Kim, and Pavel Emelyanov.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2016)*.
46. **Understanding Manycore Scalability of File Systems.**  
Changwoo Min, Sanidhya Kashyap, Steffen Maass, Woonhak Kang, and Taesoo Kim.  
In Proceedings of *USENIX Annual Technical Conference (ATC 2016)*.

47. **Cross-checking Semantic Correctness: The Case of Finding File System Bugs.**  
Changwoo Min, Sanidhya Kashyap, Byoungyoung Lee, Chengyu Song, and Taesoo Kim.  
In Proceedings of *ACM Symposium on Operating Systems Principles (SOSP 2015)*.
48. **Scalability in the Clouds! A Myth or Reality?**  
Sanidhya Kashyap, Changwoo Min, and Taesoo Kim.  
In Proceedings of *Asia-Pacific Workshop on Systems (APSys 2015)*.
49. **Lightweight Application-Level Crash Consistency on Transactional Flash Storage.**  
Changwoo Min, Woon-Hak Kang, Taesoo Kim, Sang-Won Lee, and Young Ik Eom.  
In Proceedings of *Annual Technical Conference (ATC 2015)*.
50. **Effective SSD Caching for High-performance Home Cloud Server.**  
Dongwoo Lee, Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference on Consumer Electronics (ICCE 2015)*.
51. **Reducing Excessive Journaling Overhead in Mobile Devices with Small-Sized NVRAM.**  
Junghoon Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference on Consumer Electronics (ICCE 2014)*.
52. **Dynamic-prelink: An Enhanced Prelinking Mechanism without Modifying Shared Libraries.**  
Hyungjo Yoon, Changwoo Min, and Young Ik Eom.  
In Proceedings of *The 2014 International Conference on Embedded Systems and Applications (ESA 2014)*.
53. **Understanding Implications of Trim, Discard, and Background Command for eMMC Storage Device.**  
Byungjo Kim, Dong Hyun Kang, Changwoo Min, and Young Ik Eom.  
In Proceedings of *IEEE Global Conference on Consumer Electronics (GCCE 2014)*.
54. **Page Allocation Scheme for Anti-Fragmentation on Smart Devices.**  
Jaewon Kim, and Changwoo Min, Jeehong Kim, Dong Hyun Kang, Inhyeok Kim, and Young Ik Eom.  
In Proceedings of *IEEE Global Conference on Consumer Electronics (GCCE 2014)*.
55. **An Efficient Buffer Replacement Algorithm for NAND Flash Storage Devices.**  
Dong Hyun Kang, Changwoo Min, and Young Ik Eom.  
In Proceedings of *IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS 2014)*.
56. **X-FTL: transactional FTL for SQLite databases.**  
Woon-Hak Kang, Sang-Won Lee, Bongki Moon, Gi-Hwan Oh, and Changwoo Min.  
In Proceedings of *ACM SIGMOD International Conference on Management of Data (SIGMOD 2013)*.
57. **User-aware Power Management for Mobile Devices.**  
Geunsik Lim, Changwoo Min, Dong Hyun Kang, and Young Ik Eom.  
In Proceedings of *IEEE Global Conference on Consumer Electronics (GCCE 2013)*.
58. **Ballooning Memory Trap Dynamic Memory Management in Virtualized Smart TV Environments.**  
Taehun Kim, Junghoon Kim, Keonwoo Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *on Computer Applications and Information Processing Technology (CAIPT 2013)*.
59. **Experimental Evaluations for the Relationship between Program Performance and Lifetime of NAND Flash Memory.**  
Son Yoo Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *on Computer Applications and Information Processing Technology (CAIPT 2013)*.
60. **Content-Based Chunk Placement Scheme for Decentralized Deduplication on Distributed File Systems.**  
Keonwoo Kim, Jeehong Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *international conference on Computational Science and Its Applications - Volume Part III (ICCSA 2013)*.
61. **Enhancing application performance by memory partitioning in Android platforms.**  
Geunsik Lim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference on Consumer Electronics (ICCE 2013)*.

62. **DANBI: Dynamic Scheduling of Irregular Stream Programs for Many-Core Systems.**  
Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference on Parallel Architectures and Compilation Techniques (PACT 2013)*.
63. **Optimized Lightweight Thread Framework for Mobile Devices.**  
Geunsik Lim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference of the IET Brunei Darussalam (IETBIC 2012)*.
64. **Load-Balancing for Improving User Responsiveness on Multicore Embedded Systems.**  
Geunsik Lim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *Ottawa Linux Symposium (OLS 2012)*.
65. **EIMOS: enhancing interactivity in mobile operating systems.**  
Sunwook Bae, Hokwon Song, Changwoo Min, Jeehong Kim, and Young Ik Eom.  
In Proceedings of *international conference on Computational Science and Its Applications - Volume Part III (ICCSA 2012)*.
66. **Usage pattern-based prefetching: quick application launch on mobile devices.**  
Hokwon Song, Changwoo Min, Jeehong Kim and Young Ik Eom.  
In Proceedings of *international conference on Computational Science and Its Applications - Volume Part III (ICCSA 2012)*.
67. **SFS: random write considered harmful in solid state drives.**  
Changwoo Min, Kangnyeon Kim, Hyunjin Cho, Sang-Won Lee, and Young Ik Eom.  
In Proceedings of *USENIX conference on File and Storage Technologies (FAST 2012)*.
68. **Resource Redundancy Elimination by Bridging the Semantic Gap in Virtualized Systems.**  
Inhyeok Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference on Ubiquitous Information TEchnologies & Applications (CUTE 2011)*.
69. **Kernel-level dynamic binary instrumentation method using binary translation.**  
Dongwoo Lee, Inhyuk Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *International Conference on Internet (ICONI 2010)*.
70. **MAS: Malware Analysis System Based on Hardware-Assisted Virtualization Technology.**  
Taehyoung Kim, Inhyuk Kim, Changwoo Min, and Young Ik Eom.  
In Proceedings of *Security Technology, Disaster Recovery and Business Continuity (SecTech 2010)*.
71. **Efficient fuzzy rule generation based on fuzzy decision tree for data mining.**  
Myung Won Kim, Joong Geun Lee, and Changwoo Min.  
In Proceedings of *International Fuzzy Systems Conference (FUZZ-IEEE 1999)*.

### 5.1.2 Journal Publications

1. **Securely Sharing Randomized Code that Flies.**  
Christopher Jelesnianski, Jinwoo Yom, Changwoo Min, and Yeongjin Jang.  
*ACM Digital Threats: Research and Practice*.  
March 2022.
2. **An OpenMP Runtime for Transparent Work Sharing Across Cache-Incoherent Heterogeneous Nodes.**  
Robert Lysterly, Carloc Bilbao, Changwoo Min, Christopher J. Rossbach, and Binoy Ravindran.  
*ACM Transactions on Computer Systems*.  
December 2021.
3. **A Syscall-Level Binary-Compatible Unikernel.**  
Pierre Olivier, Hugo Lefeuvre, Daniel Chiba, Stefan Lankes, Changwoo Min, and Binoy Ravindran.  
*IEEE Transactions on Computer*.  
October 2021.
4. **Making Application-level Crash Consistency Practical on Flash Storage.**  
Dong Hyun Kang, Changwoo Min, Sang-Won Lee, and Young Ik Eom.  
*IEEE Transactions on Parallel and Distributed Systems, Volume 31-5*.  
May 2019.

5. **Opportunistic Spinlocks: Achieving Virtual Machine Scalability in the Clouds.**  
 Sanidhya Kashyap, Changwoo Min, and Taesoo Kim.  
*ACM SIGOPS Operating Systems Review (OSR), Volume 50-1.*  
 January 2016.  
[LWN: qspinlock in Linux](#)
6. **vCanal: Paravirtual Socket Library towards Fast Networking in Virtualized Environment.**  
 Dongwoo Lee, Changwoo Min, and Young Ik Eom.  
*IEICE Transactions on Information and Systems, Volume E99-D.*  
 February 2016.
7. **Dynamic Scheduling of Irregular Stream Programs toward Many-Core Scalability.**  
 Changwoo Min and Young Ik Eom.  
*IEEE Transactions on Parallel and Distributed Systems, Volume 26-6.*  
 2015.
8. **Integrating Lock-free and Combining Techniques for a Practical and Scalable FIFO Queue.**  
 Changwoo Min and Young Ik Eom.  
*IEEE Transactions on Parallel and Distributed Systems, Volume 26-7.*  
 2015.
9. **Static Dalvik Bytecode Optimization for Android Applications.**  
 Jeehong Kim, Inhyeok Kim, Changwoo Min, Hyung Kook Jun, Soo Hyung Lee, Won Tae Kim, and Young Ik Eom.  
*ETRI Journal, Volume 37-2.*  
 October 2015.
10. **Effective flash-based SSD caching for high performance home cloud server.**  
 Dongwoo Lee, Changwoo Min, and Young Ik Eom.  
*IEEE Transactions on Consumer Electronics, Volume 61-2.*  
 2015.
11. **Block Utilization-aware Buffer Replacement Scheme for Mobile NAND Flash Storage.**  
 Dong Hyun Kang, Changwoo Min, and Young Ik Eom.  
*IEICE Transactions on Information and Systems, Volume E97-D.*  
 September 2014.
12. **Symbiotic Dynamic Memory Balancing for Virtual Machines in Smart TV Systems.**  
 Junghoon Kim, Taehun Kim, Changwoo Min, Hyung Kook Jun, Soo Hyung Lee, Won Tae Kim, and Young Ik Eom.  
*ETRI Journal, Volume 36-5.*  
 October 2014.
13. **Design and Implementation of a Log-Structured File System for Flash-Based Solid State Drives.**  
 Changwoo Min, Sang-Won Lee, and Young Ik Eom.  
*IEEE Transactions on Computer, Volume 63-9.*  
 September 2014.
14. **Reducing Excessive Journaling Overhead with Small-Sized NVRAM for Mobile Devices.**  
 Junghoon Kim, Changwoo Min, and Young Ik Eom.  
*IEEE Transactions on Consumer Electronics, Volume 60-2.*  
 2014.
15. **Supporting Transactional Atomicity in Flash Storage Devices.**  
 Woon-Hak Kang, Sang-Won Lee, and Bongki Moon, and Gi-Hwan Oh, and Changwoo Min.  
*IEEE Data Eng. Bull., Volume 37-2.*  
 2014.
16. **Zero-Sum Defender: Fast and Space-Efficient Defense against Return-Oriented Programming Attacks.**  
 Jeehong Kim, Inhyeok Kim, Changwoo Min, and Young Ik Eom.  
*IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Volume E97.A.*  
 2014.



17. **Virtual memory partitioning for enhancing application performance in mobile platforms.**  
Geunsik Lim, Changwoo Min, and Young Ik Eom.  
*IEEE Transactions on Consumer Electronics, Volume 59-4.*  
2013.
18. **VMMB: Virtual Machine Memory Balancing for Unmodified Operating Systems.**  
Changwoo Min, Inhyeok Kim, Taehyoung Kim, Young Ik Eom.  
*Journal of Grid Computing, Volume 10-1.*  
March 2012.
19. **Scalable Cache-Optimized Concurrent FIFO Queue for Multicore Architectures.**  
Changwoo Min, Hyung Kook Jun, Won Tae Kim and Young Ik Eom.  
*IEICE Transactions on Information and Systems, Volume E95.D-12.*  
2012.
20. **Hardware assisted dynamic memory balancing in virtual machines.**  
Changwoo Min, Inhyeok Kim, Taehyoung Kim, and Young Ik Eom.  
*IEICE Electronics Express, Volume 8-10.*  
May 2011.
21. **Weight Perturbation for Efficient Learning of Neural Networks.**  
Samkeun Kim, Changwoo Min, and Myungwon Kim.  
*Journal of Electrical Engineering and Information Science, Volume 3-5.*  
1998.

### 5.1.3 Posters

1. **Scheduling HPC Workloads on Heterogeneous-ISA Architectures.**  
Mohamed L Karaoui, Anthony Carno, Rob Lysterly, Sang-Hoon Kim, Pierre Olivier, Changwoo Min, and Binoy Ravindran.  
*Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming (PPoPP 2019).*
2. **TS-CLOCK: Temporal and Spatial Locality Aware Buffer Replacement Algorithm for NAND Flash Storages.**  
Dong Hyun Kang, Changwoo Min, and Young Ik Eom.  
*Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2014).*
3. **Flash-Friendly Buffer Replacement Algorithm for Improving Performance and Lifetime of NAND Flash Storages.**  
Dong Hyun Kang, Changwoo Min, and Young Ik Eom.  
*Proceedings of the 12th USENIX conference on File and Storage Technologies (FAST 2014).*
4. **Can Lock-free and Combining Techniques Co-exist? A Novel Approach on Concurrent Queue.**  
Changwoo Min, and Young Ik Eom.  
*Proceedings of the 22st International Conference on Parallel Architectures and Compilation Techniques (PACT 2013).*
5. **NUMA-aware Scheduler: Taking Both Data Locality and Caching Effectiveness into Account on NUMA Platforms.**  
Junghoon Kim, Changwoo Min, and Young Ik Eom.  
*Proceedings of 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2012).*
6. **Towards Bare-metal Network Performance via Para-virtualized Socket Library and Exitless I/O.**  
Dongwoo Lee, Changwoo Min, Junghan Kim, and Young Ik Eom.  
*Proceedings of 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2012).*
7. **Accelerating Virtual Machine Storage I/O for Multicore Systems.**  
Dongwoo Lee, Junghan Kim, Junghoon Kim, Changwoo Min, and Young Ik Eom.  
*Proceedings of the 10th USENIX conference on File and Storage Technologies (FAST 2012).*

8. **DANBI: Dynamic and Scalable Stream Parallelism for Many-core Systems.**  
Changwoo Min, and Young Ik Eom.  
*Proceedings of 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI '12).*
9. **Guest Transparent Dynamic Memory Balancing in Virtual Machines.**  
Changwoo Min, Inhyuk Kim, Taehyoung Kim, and Young Ik Eom.  
*9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010).*

## 5.2 Open Source Softwares

<b>PRISM</b>	<i>Prism: Optimizing Key-Value Store for Modern Heterogeneous Storage Devices (<a href="#">code</a>)</i>	11/2022
<b>PACTight</b>	<i>Tightly Seal Your Sensitive Pointers with PACTight (<a href="#">code</a>)</i>	08/2022
<b>Durinn</b>	<i>Durinn: Adversarial Memory and Thread Interleaving for Detecting Durable Linearizability Bugs (<a href="#">code</a>)</i>	07/2022
<b>OdinFS</b>	<i>Odinfs: Scaling PM performance with Opportunistic Delegation (<a href="#">code</a>)</i>	07/2022
<b>VIP</b>	<i>VIP: Safeguard Value Invariant Property for Thwarting Critical Memory Corruption Attacks (<a href="#">code</a>)</i>	11/2021
<b>Witcher</b>	<i>Witcher: Systematic Crash Consistency Testing for Non-Volatile Memory Key-Value Stores (<a href="#">code</a>)</i>	09/2021
<b>PACTree</b>	<i>PACTree: A High Performance Persistent Range Index Using PAC Guidelines (<a href="#">code</a>)</i>	09/2021
<b>TIPS</b>	<i>Framework to Make a Volatile Index Persistent (<a href="#">code</a>)</i>	07/2021
<b>TimeStone</b>	<i>Scalable Durable Transactional Memory (<a href="#">code</a>)</i>	06/2021
<b>CrossFS</b>	<i>Cross-layered Direct-Access File System (<a href="#">code</a>)</i>	11/2020
<b>HydraList</b>	<i>Scalable In-Memory Sorted Index (<a href="#">code</a>)</i>	11/2020
<b>Shuffle Lock</b>	<i>Scalable and Practical Locking with Shuffling (<a href="#">code</a>)</i>	09/2019
<b>HermiTux</b>	<i>HermiTux: A Binary Compatible Unikernel (<a href="#">code</a>)</i>	05/2019
<b>MV-RLU</b>	<i>MV-RLU: Scaling Read-Log-Update with Multi-Versioning (<a href="#">code</a>)</i>	05/2019
<b>Solros</b>	<i>SOLROS: A Data-Centric Operating System Architecture for Heterogeneous Computing (<a href="#">code</a>)</i>	11/2018
<b>CST Lock</b>	<i>Scalable NUMA-aware blocking synchronization primitives (<a href="#">code</a>)</i>	07/2017
<b>Mosaic</b>	<i>Graph processing engine on a single machine for a trillion-edge graph (<a href="#">code</a>)</i>	04/2017
<b>APISan</b>	<i>A tool to find API misuse bugs through semantic cross-checking (<a href="#">code</a>)</i>	08/2016
<b>FxMark</b>	<i>Benchmark to measure filesystem multicore scalability (<a href="#">code</a>)</i>	06/2016
<b>Juxta</b>	<i>A tool to find filesystem-specific semantic bugs (<a href="#">code</a>)</i>	12/2015
<b>VBench</b>	<i>Multicore scalability application benchmark (<a href="#">code</a>)</i>	07/2015
<b>DANBI</b>	<i>Dataflow Parallel Runtime for Manycore Systems (<a href="#">code</a>)</i>	08/2013

## 5.3 Patent

1. **Method and apparatus for managing memory in virtual machine environment.**  
Changwoo Min, Inhyeok Kim, Taehyoung Kim, Young Ik Eom.  
[US9229651B2](#) 2016
2. **Method and apparatus for encrypting and processing data in flash translation layer.**  
Changwoo Min, and Jin-Ha Jun.  
[US8261098B2](#) 2012

3. **Analyzing user viewership of advertisements appearing in a screen display in a user terminal.**  
SangChul Kang and Changwoo Min.  
[US8205167B2](#) 2012
4. **Memory and method for data compression and management.**  
Soon-Yong Jeong and Changwoo Min.  
[US8037255B2](#) 2011
5. **Apparatus for developing a transfer dictionary used in transfer-based machine translation system.**  
Seong Mook Kim, Changwoo Min, SangChul Kang and Jeong In Cha.  
[US7487082B2](#) 2009

## 5.4 Invited Talks and Presentations

<b>Sungkyunkwan University</b>	<i>Designing Storage Systems in the World of Extreme Heterogeneity</i>	Virtual, 09/2022
<b>SDC</b>	<i>RETINA: Exploring Computational Storage (SmartSSD) Usecase (<a href="#">video</a>)</i>	CA, USA, 09/2022
<b>Samsung SDS Research</b>	<i>Understanding the Evolution of Storage Stack from a Storage Hardware Perspective</i>	Seoul, Korea, 07/2022
<b>SAIT</b>	<i>Designing High-Performance, Scalable Storage Stack with Computational Storage and Byte-addressable Non-Volatile Memory</i>	Suwon, Korea, 06/2022
<b>Sungkyunkwan University</b>	<i>Taming Data Movement Challenges with Optimized RDMA RPC</i>	Suwon, Korea, 05/2022
<b>KAIST</b>	<i>Fast, Scalable Storage Stack for Emerging Storage Devices: Exploring the Boundaries of Operating System</i>	Daejeon, Korea, 05/2022
<b>Samsung Semiconductor</b>	<i>Taming Data Movement Challenges with Optimized RDMA RPC and Computational Storage</i>	Virtual, 04/2022
<b>POSTECH</b>	<i>Designing High-Performance, Scalable, and Bug-Free Non-Volatile Memory Based Storage Systems</i>	Virtual, 11/2021
<b>NVRAMOS</b>	<i>Designing High-Performance and Bug-Free Key-Value Stores for Non-Volatile Memory</i>	Busan, Korea, 10/2021
<b>KAIST</b>	<i>Designing and Optimizing Storage Systems in the Era of Extreme Heterogeneity</i>	Virtual, 09/2021
<b>Samsung Semiconductor</b>	<i>Designing and Optimizing Storage Systems in the Era of Extreme Heterogeneity</i>	Virtual, 09/2021
<b>KIISE Computer System Society</b>	<i>Scaling Persistent Memory Program for Manycore Systems</i>	Virtual, 02/2021
<b>Eta Kappa Nu ECE Honor Society</b>	<i>Computer System and System Software Challenges and (a Little Bit of) Solutions for 2020s</i>	Blacksburg, Virginia, 12/2020
<b>NVRAMOS</b>	<i>ShflLocks: Scalable and Practical Locking for Manycore Systems</i>	Jeju, Korea, 10/2019
<b>Samsung Research</b>	<i>Designing Operating Systems for 2020s</i>	Seoul, Korea, 07/2018
<b>UNIST</b>	<i>Designing Operating Systems for 2020s</i>	Ulsan, Korea, 07/2018

<b>Sungkyunkwan University</b>	<i>Designing Operating Systems for 2020s</i>	Suwon, Korea, 07/2018
<b>KAIST</b>	<i>Designing Operating Systems for 2020s</i>	Daejeon, Korea, 07/2018
<b>EuroSys</b>	<i>SOLROS: A Data-Centric Operating System Architecture for Heterogeneous Computing (video)</i>	Porto, Portugal, 04/2018
<b>MICS, Virginia Tech</b>	<i>System Software for Many and Specialized Core Era</i>	Blacksburg, VA, 12/2017
<b>UNC Charlotte</b>	<i>Finding Software Bugs through Semantic Cross-checking</i>	Charlotte, NC, 05/2017
<b>Virginia Tech</b>	<i>Improving Filesystems for Reliability and Scalability</i>	Blacksburg, Virginia, 02/2017
<b>Stony Brook University</b>	<i>Improving Filesystems for Reliability and Scalability</i>	New York, 03/2017
<b>KAIST</b>	<i>Improving Filesystems for Reliability and Scalability</i>	Daejeon, Korea, 11/2016
<b>POSTECH</b>	<i>Improving Filesystems for Reliability and Scalability</i>	Pohang, Korea, 09/2016
<b>Seoul National University</b>	<i>Improving Filesystems for Reliability and Scalability</i>	Seoul, Korea, 06/2016
<b>Sungkyunkwan University</b>	<i>Improving Filesystems for Reliability and Scalability</i>	Suwon, Korea, 06/2016
<b>UNIST</b>	<i>APISan: Finding API Misuse Bugs through Semantic Cross-checking</i>	Ulsan, Korea, 09/2016
<b>USENIX ATC</b>	<i>Understanding Manycore Scalability of File Systems (video)</i>	Denver, CO, 06/2016
<b>SOSP</b>	<i>Cross-checking Semantic Correctness: The Case of Finding File System Bugs (video)</i>	Monterey, CA, 10/2015
<b>USENIX ATC</b>	<i>Lightweight Application-Level Crash Consistency on Transactional Flash Storage (video)</i>	Santa Clara, CA, 07/2015
<b>ETRI</b>	<i>Tiny Little Things for Manycore Scalability: Scalable Locking and Lockless Data Structures</i>	Daejeon, Korea, 08/2014
<b>PACT</b>	<i>DANBI: Dynamic Scheduling of Irregular Stream Programs for Many-Core Systems</i>	Edinburgh, UK, 09/2013
<b>PACT</b>	<i>Can Lock-free and Combining Techniques Co-exist? A Novel Approach on Concurrent Queue</i>	Edinburgh, UK, 09/2013
<b>ETRI</b>	<i>Virtualization Technology: Overview, Memory Management and I/O Virtualization</i>	Daejeon, Korea, 09/2012
<b>USENIX FAST</b>	<i>SFS: Random Write Considered Harmful in Solid State Drives (video)</i>	San Jose, CA, 02/2012

## 5.5 Work Experience before Starting Ph.D. Study (1998–2009)

- **Linux Mobile Platform (2007–2009)** I was one of the founding members of Samsung’s Linux-based mobile platform for consumer electronics devices, also known as Tizen today. I was involved in the design and development of the Linux platform. In particular, I optimized the core components, such as the registry service

(gconf), inter-process communication (IPC) library (D-Bus), and graphics server (X server). I optimized gconf for read-mostly workloads to reduce the boot-up time of the device. The new gconf backed used memory mapping (mmap) to minimize latency. I re-architected D-Bus, an open-source communication system, from a centralized architecture to a distributed peer-to-peer architecture to reduce communication latency. Also, I contributed to developing the X server, a compositing window manager, and GPU acceleration on the platform. Two smartphones were successfully commercialized for the largest European telecommunication company. I was also involved in the standardization of the mobile platform in LiMo (Linux Mobile) Foundation as an architecture council (AC) representative of Samsung Electronics.

- **Demand Paging for RTOS-based Embedded System (2005–2006)** I developed the demand paging for Samsung’s in-house RTOS (real-time operating system) for 3G feature phone models. The demand paging allows executing a program beyond its DRAM size, reducing the DRAM size (from 64MB to 32MB) and lowering the hardware cost. The main technical challenges lie in meeting the hard real-time requirement of the 3G network protocols with a wimpy ARM7 processor and slow NAND flash memory as backing storage. Besides developing the demand paging code on the ROTS, I optimized the code placement in the compiled binary to reduce costly veneer code execution in ARM architectures. The 3G feature phone models with the demand paging were hugely successful in the market.
- **Distributed Compilation System (2005–2006)** I developed the distributed compilation system, which performs compilation in parallel over tens of build machines in a network. It reduces the build time of the feature phone source tree from three hours to 20 minutes. I designed and developed the dynamic work-stealing scheduling to efficiently utilize the build machines whose performance is heterogeneous.
- **IBM Java Virtual Machine (J9) (2003–2005)** J9 is IBM’s clean room implementation of Java virtual machine (now open-sourced). I worked on poring and optimizing the JVM and class libraries for embedded devices such as mobile phones and set-top boxes. In particular, I worked on porting and optimizing J9 on ARM architecture. I contributed to porting and optimizing the Just-In-Time (JIT) compiler for the ARM architecture, developing a virtual machine hibernation technique, which mitigates class loading and JIT compilation overhead in run-time. Also, I developed a space-efficient memory allocator for memory-limited embedded systems.
- **WebSphere Translation Server (1999–2003)** WebSphere Translation Server (WTS) is a machine translation product of IBM, which is designed for high-performance servers and supports multiple operating systems, including Linux, Solaris, AIX, and Microsoft Windows. I designed and developed English-to/from-Korean machine translation engines, especially core language processing components, such as morphological analyzer, syntactic parser, and semantic analyzer. Besides the core language processing components, I also developed key-value stores for language dictionaries using a perfect hash function and the memory allocator to optimize performance for highly multi-threaded servers.
- **OS/2 Kernel (1998–1999)** I worked on the internalization and localization of the OS/2 desktop operating system. I contributed to developing one major service pack release, whose bug patches range from OS/2 kernel to applications, one major release of video device drivers, and Korean Input Method Editor (IME).

## 6 Teaching

### 6.1 Courses Taught

Semester Year	Course Number	Course Title	Enrollment
Fall 2022	ECE-4414/5414G(CS-4224/5264G)	(Advanced) Linux Kernel Programming	51
Spring 2022	ECE 3574	Applied Software Design	89
Fall 2021	ECE-4414/5414G(CS-4224/5264G)	(Advanced) Linux Kernel Programming	63
Spring 2021	ECE 3574	Applied Software Design	63
Fall 2020	ECE-4414/5414G(CS-4224/5264G)	(Advanced) Linux Kernel Programming	43
Spring 2020	ECE 3574	Applied Software Design	69
Fall 2019	ECE 4984/5984	(Advanced) Linux Kernel Programming	34
Spring 2019	ECE 3574	Applied Software Design	53
Fall 2018	ECE 4984/5984	(Advanced) Linux Kernel Programming	19
Spring 2018	ECE 3574	Applied Software Design	54
Fall 2017	ECE 4984/5984	(Advanced) Linux Kernel Programming	26

- *“It is a difficult course and is coding intensive but one of the best courses I have taken in 6 years at VT. The professor is invested in the student’s success but it is tough love :) I personally enjoyed taking the class!”* – in ECE-4414/5414G(CS-4224/5264G), Fall 2022
- *“This course really helped to clear the technical interviews. The final project topics suggested by professor are hot research topics which help us learn current trends. This is the one of the best courses in Virginia Tech relevant to my professional career.”* – in ECE-4414/5414G(CS-4224/5264G), Fall 2022
- *“Dr. Min was honestly one of the best professors I have ever had. Amazing explanations, teaching style, and I could tell he truly cared and put effort into helping us understand the material. No complaints at all with the professor. The course, however, did end with kind of a lot of jam packed information during the last 2/3 weeks of classes. We learned several different architectures/design patterns during these lectures and they were quite complicated to learn in just one class.”* – in ECE 3574, Spring 2022
- *“Min was a great professor in terms of help and responding to students to clarify any misunderstandings. He made sure to keep us updated at all times and was overall a very nice instructor to have.”* – in ECE 3574, Spring 2022
- *“Thank you so much for teaching us the Advance Linux Kernel Programming course in Fall 2021. I enjoyed every aspect of the course, projects, assignments, paper readings, everything. I gained so much knowledge in the class and in after class discussions. At the time of my internship hustle this is the only course really helped me to crack an internship. I got the internship which converted to Full-time solely because of this course and I forever thankful to you.”* – in ECE-4414/5414G(CS-4224/5264G), Fall 2021
- *“Min was a phenomenal professor and instructor. He really inspires the class to be more curious about the material. The way he breaks down highly complex topics to be understandable to even the undergrads in the course is really amazing. I cannot recommend this professor enough!”* – in ECE-4414/5414G(CS-4224/5264G), Fall 2021
- *“Certainly one of the best professors that I have had. His explanations are always clear and concise and seems to be able to make difficult concepts simple to understand.”* – in ECE 3574, Spring 2021
- *“Dr. Min was fantastic this semester, I had significant issues this semester including ... and Dr. Min’s consistent compassion and support made all the difference in my capacity to succeed this semester. ... Overall, this was a fantastic course and Dr. Min did an excellent job giving students all the tools they needed for success. Really a fantastic course and I wish I would have the opportunity to take more of Dr. Min’s courses.”* – in ECE 3574, Spring 2021
- *“Thank you for the semester. The final kernel project was exactly what I was hoping to get out of this class, and I am no longer intimidated by the thought of diving into kernel code.”* – in ECE-4414/5414G(CS-4224/5264G), Fall 2020
- *“The instructor incorporated examples from the actual Linux codebase into his lectures, thus helping to reify the concepts covered in the lecture. Few courses bother to incorporate actual production code, but its presence here was a significant and somewhat unique benefit.”* – in ECE-4414/5414G(CS-4224/5264G), Fall 2020

- *“Professor Min presented concepts in a way that was very easy to understand, and the provided sample code was always very helpful. I really appreciate all the examples and reference links provided to me as a student in this course. My programming skills have definitely grown as a result of taking this course and that is in large part thanks to Professor Min’s teaching style. Additionally, the way the course was altered as part of the transition to online format made it much easier to keep up with and I’m incredibly grateful for the effort the professor put into changing things so quickly. I would have struggled immensely to finish this semester if the class hadn’t been altered to be better suited to online learning.”* – in ECE 3574, Spring 2020
- *“He was very fun to have in class. Very engaging with the class, allowed for question and answer time. Willing to help students during and after class. Projects were fair and well designed.”* – in ECE 3574, Spring 2020
- *“There is not any course like this course. This is because teaching about Linux at kernel level is hard, so it is precious course to me. Even if the understanding Linux kernel is hard, the course guide and draw the overall view of the kernel. From the class, I can build the view of code in Linux kernel, so after that, I can approach the system as a code.”* – in ECE 4984/5984, Fall 2019
- *“This is one of my favorite classes I’ve taken at VT. I can best describe it as an applied OS class - exactly the kind of thing I love to learn as a CPE major. This course is packed with content. Dr. Min did an excellent job breaking it down and bringing everything back together. There is no way I will have mastered the subject by the end of the semester because the kernel is massive. However, my understanding and ability to learn about systems is far better than it was at the beginning of the semester. That’s why I rate my abilities as "good" instead of "excellent". This is a must have course for anybody wanting to get into systems.”* – in ECE 4984/5984, Fall 2019
- *“Professor Min is an extremely knowledgeable and capable teacher. He has probably forgotten more about programming than I will ever know. He is also an entertaining lecturer, a fair teacher, and a good person. I wish I had more teachers like him. Also it really helps when teachers have a lot of industry experience. Relating concepts to actual things he’s done is very helpful.”* – in ECE 3574, Spring 2019
- *“I loved the instructor, Professor Min was very approachable and willing to help.”* – in ECE 3574, Spring 2019
- *“I came to VT in search of knowledge taught by people with amazing experience and the ability to convey that information. Dr Min hits both these points wonderfully on top of juggling that with his research.”* – in ECE 4984/5984, Fall 2018
- *“The lectures and presentations were well prepared and planned such that the topic was explained properly. Home works and assignments were extremely effective and let us explore the subject intensively.”* – in ECE 4984/5984, Fall 2018
- *“Everything best professor I’ve ever had”* – in ECE 3574, Spring 2018
- *“Dr. Min is one of the best instructors I’ve ever had. He went over every important material in class and asked many questions during the lecture to make students to think and to participate.”* – in ECE 3574, Spring 2018
- *“One of my favorite teachers in my collage career.”* – in ECE 4984/5984, Fall 2017
- *“The course is fantastic and I got to learn a lot. I definitely have a better insight in linux kernel as compared to what I started off with. The projects are very interesting and your teaching style/ hints and support for the projects make the overall experience better.”* – in ECE 4984/5984, Fall 2017

#### 6.1.1 Service on Thesis or Dissertation Committees

1. Shashwat Jain, *"ProcPIPE: Offloading processing pipelines on Computational Storage"*, April 2023
2. Yuqing Liu, *"Prototyping Hardware-compressed Memory for Multi-tenant Systems"*, February 2023
3. Hadeel Albahar, *"Optimizing Systems for Deep Learning Applications"*, February 2023
4. Uday Kiran Podugu, *"Acoustic Injection Attack on MEMS Accelerometers"*, December 2022
5. Komal Gupta, *"Acoustic Injection Attack on MEMS Gyroscope"*, December 2022
6. Monami Dutta Gupta, *"Protecting data privacy in deep-learning-based recommendation models using SGX"*, December 2022
7. Spandan Mahadevegowda, *"Secure Communication Networks for Connected Vehicles"*, December 2022

8. Christopher Jelesnianski, "*Practical Exploit Mitigation Design Against Code Re-Use and System Call Abuse Attacks*", December 2022
9. Yue Zhan, "*Natural Language Programming for Controlled Object-Oriented English*", May 2022
10. Naga Sanjana Bikonda, "*Retina: Cross-Layered Key-Value store using Computational Storage*", February 2022
11. Xinwei Fu, "*Detecting Persistence Bugs from Non-volatile Memory Programs by Inferring Likely-correctness Conditions*", February 2022
12. Garvit Goel, "*Real-Time Computed Tomography-based Medical Diagnosis Using Deep Learning*", January 2022
13. Moein Borghei, "*Partial Discharges: Experimental Investigation, Model Development, and Data Analytics*", January 2022
14. Burhanuddin Bharmal, "*Real-Time GPU Scheduling with Preemption Support for Autonomous Mobile Robots*", December 2021
15. Mohannad Ismail, "*PACTIGHT: Tightly Seal Sensitive Pointers with Pointer Authentication*", December 2021
16. Sumit Kumar Monga, "*Scaling RDMA RPCs with FLOCK*", November 2021
17. Chandler Jearls, "*Open-Source Parameterized Low-Latency Aggressive Hardware Compressor and Decompressor for Memory Compression*", May 2021
18. Angel Tehillah Isiadinso, "*Attribute Based Student Mappings to Facilitate University Operations*", May 2021
19. FNU Sachin, "*Input Data Pipeline Analysis of TensorFlow Models*", May 2021
20. Wonseok Shin, "*FIREWORKS: A Fast, Efficient and Safe Serverless Framework*", May 2021
21. Sanidhya Kashyap, "*Scaling Synchronization Primitives*", June 2020
22. Saumya Sardana, "*Vision-based Color Detection with the DJI Tello Drone*", May 2020
23. Akhil Ahmed Rafeeq, "*A Development Platform to Evaluate UAV Runtime Verification Through Hardware-in-the-loop Simulation*", May 2020
24. Lakshman Theyyar Maalolan, "*Trusted Unmanned Aerial System Operations*", May 2020
25. Anthony Demeri, "*POSEIDON: Safe and Scalable Persistent Memory Allocator*", May 2020
26. Jinwoo Yom, "*HyperSpace: Data-Value Integrity for Securing Software*", April 2020
27. Ajit Mathew, "*Multicore Scalability through Asynchronous Work*", December 2019
28. Ahmed Helal, "*Automated Runtime Analysis and Adaptation for Scalable Heterogeneous Computing*", December 2019
29. Yihan Pang, "*Leveraging Processor-diversity for Improved Performance in Heterogeneous-ISA Systems*", September 2019
30. Akshat Malik, "*Monitoring and Preventing Data Exfiltration in Android-hosted Unmanned Aircraft System Applications*", July 2019
31. Mihir Sagar Kulkarni, "*Determination of Optimal Technique for Ocean Wave Simulation and Prediction*", July 2019
32. Ashish Malpani, "*Tweets Clustering and Visualization*", May 2019
33. A K M Fazia Mehrab, "*Cross-ISA Execution Migration of Unikernels: Build Toolchain, Memory Alignment, and VM State Transfer Techniques*", November 2018
34. Mohit Garg, "*Generalized Consensus for Practical Fault Tolerance*", Aug 2018
35. Qingrui Liu, "*Compiler-Directed Error Resilience for Reliable Computing*", June 2018
36. Daniel Chiba, "*Accelerating the Adoption of Unikernels through Optimised Hypervisor Boot Times and Enhanced Binary Compatibility with Linux Applications*", May 2018

## 7 Service

### 7.1 Conference Committee Activities

1. Program Committee, *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 2024, 2023
2. Program Committee, *USENIX Annual Technical Conference (ATC)*. 2022-2018



- |     |   |      |
|-----|---|------|
| 3.  | Program Committee, <i>USENIX Conference on File and Storage Technologies (FAST)</i> . 2024, 2022, 2020              |      |
| 4.  | Program Committee, <i>USENIX Symposium on Operating Systems Design and Implementation (OSDI)</i> . 2021             |      |
| 5.  | Program Committee, <i>IEEE International Conference on Distributed Computing Systems (ICDCS)</i> . 2023, 2020, 2018 |      |
| 6.  | Program Committee, <i>Non-Volatile Memory Workshop (NVMW)</i> .   | 2023 |
| 7.  | Program Committee, <i>ACM Workshop on Hot Topics in Storage and File Systems (HotStorage)</i> . 2022-2021           |      |
| 8.  | External Review Committee, <i>European Conference on Object-Oriented Programming (ECOOP)</i> .                      | 2023 |
| 9.  | Program Committee, <i>ACM SIGOPS Asia-Pacific Workshop on Systems (APSys)</i> .                                     | 2020 |
| 10. | Program Committee, <i>ACM International Systems and Storage Conference (SYSTOR)</i> .                               | 2016 |

## 7.2 Journal Reviewing Activities

- |    |  |                      |
|----|--|----------------------|
| 1. | Journal Reviewer, <i>ACM Transactions on Storage (TOS)</i> .                             | 2023-2017, 2014-2013 |
| 2. | Journal Reviewer, <i>ACM Transactions on Computer Systems (TOCS)</i> .                   | 2023, 2022, 2019     |
| 3. | Journal Reviewer, <i>ACM Transactions on Architecture and Code Optimization (TACO)</i> . | 2023, 2021-2020      |
| 4. | Journal Reviewer, <i>IEEE Transactions on Computers (TC)</i> .                           | 2021-2020, 2017      |
| 5. | Journal Reviewer, <i>IEEE Transactions on Cloud Computing (TCC)</i> .                    | 2022                 |
| 6. | Journal Reviewer, <i>The Journal of Systems and Software (JSS)</i> .                     | 2020                 |
| 7. | Journal Reviewer, <i>Journal of Parallel and Distributed Computing (JPDC)</i> .          | 2022, 2019           |
| 8. | Journal Reviewer, <i>Concurrency and Computation: Practice and Experience</i> .          | 2016                 |
| 9. | Journal Reviewer, <i>IEEE Transactions on Parallel and Distributed Systems (TPDS)</i> .  | 2014                 |

## 7.3 Funding Agency Panel Activities

- |    |  |            |
|----|--|------------|
| 1. | Proposal Panelist, <i>National Science Foundation (NSF)</i> .                                  | 2021, 2019 |
| 2. | External Proposal Reviewer, <i>Natural Sciences and Engineering Research Council (NSERC)</i> . | 2018       |

## 7.4 Memberships and Activities in Professional Societies

- Member, Association for Computing Machinery (ACM)
- Member, The Advanced Computing Systems Association (USENIX)
- Member, Institute of Electrical and Electronics Engineers (IEEE)