

Amogh Akshintala

324 6th Avenue S, Apt 1, Kirkland WA 98033

☎ +1 (631)-371-4820 | ✉ amoghakshintala@gmail.com | 🌐 aakshintala.com | 📱 aakshintala

Education

Doctor of Philosophy (ABD) | Computer Science

UNIVERSITY OF NORTH CAROLINA - CHAPEL HILL

Aug 2016 — Dec 2020 (ABD)

STONY BROOK UNIVERSITY

Jan 2014 — Aug 2016

Master of Science | Computer Science

STONY BROOK UNIVERSITY

Aug 2012 — Dec 2013 (issued May 2016)

Bachelor of Engineering | Computer Science and Engineering

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

2008 — 2012

Experience

Software Engineer

REALITY LABS | META (FORMERLY FACEBOOK REALITY LABS)

Dec 2020 — Present

- Building custom IDL & RPC framework for in-house micro-kernel OS.
- *Skills exercised:* Compiler construction (IDL compiler, code generators), API design, Mentorship.
- *Technical Concepts:* code generation, data serialization, async programming (on custom Async runtime).
- *Tools used:* Rust, Scala

Lecturer (limited-term)

UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

[Computer Architecture](#)

Jan 2020 — May 2020

Research Assistant (projects described below)

UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

Aug 2016 — Dec 2020

UNIVERSITY OF TEXAS AT AUSTIN

Fall 2017; Calendar Year 2019

STONY BROOK UNIVERSITY

Jan 2014 — Aug. 2016

Research Intern (projects described below)

VMWARE RESEARCH GROUP

Summer 2017; Summer 2018

Member of Technical Staff - Intern

VMWARE INC.

Summer 2014

TINTRI INC.

Summer 2013

Publications

CONFERENCES

Accelerated Virtualization of Accelerators

HANGCHEN YU, ARTHUR PETERS, **AMOGH AKSHINTALA**, CHRISTOPHER J. ROSSBACH

ASPLOS '20

Trillium: The code is the IR

AMOGH AKSHINTALA, HANGCHEN YU, ARTHUR PETERS, CHRISTOPHER J. ROSSBACH

VIRT '19

x86-64 Instruction Usage among C/C++ Applications

AMOGH AKSHINTALA, BHUSHAN P. JAIN, CHIA-CHE TSAI, MICHAEL FERDMAN, DONALD E. PORTER

SYSTOR '19

Optimizing Every Operation in a Write-optimized File System

[Awarded Best Paper](#)

JUN YUAN, YANG ZHAN, WILLIAM JANNEN, PRASHANT PANDEY, **AMOGH AKSHINTALA**, KANCHAN CHANDNANI,

POOJA DEO, ZARDOSHT KASHEFF, LEIF WALSH, MICHAEL BENDER, MARTIN FARACH-COLTON, ROB JOHNSON,

FAST '16

BRADLEY C. KUSZMAUL, AND DONALD E. PORTER.

BetrFS: A Right-Optimized Write-Optimized File System

[Best Paper Runner-up](#)

WILLIAM JANNEN, JUN YUAN, YANG ZHAN, **AMOGH AKSHINTALA**, JOHN ESMET, YIZHENG JIAO, ANKUR MITTAL,

PRASHANT PANDEY, PHANEENDRA REDDY, LEIF WALSH, MICHAEL BENDER, MARTIN FARACH-COLTON, ROB

FAST '15

JOHNSON, BRADLEY C. KUSZMAUL, AND DONALD E. PORTER.

JOURNALS

Writes Wrought Right, and Other Adventures in File System Optimization.

JUN YUAN, YANG ZHAN, WILLIAM JANNEN, PRASHANT PANDEY, **AMOGH AKSHINTALA**, KANCHAN CHANDNANI, POOJA DEO, ZARDOSHT KASHEFF, LEIF WALSH, MICHAEL A. BENDER, MARTIN FARACH-COLTON, ROB JOHNSON, BRADLEY C. KUSZMAUL, AND DONALD E. PORTER.

ACM Transactions on Storage (TOS)
Mar'17

BetrFS: Write-Optimization in a Kernel File System

WILLIAM JANNEN, JUN YUAN, YANG ZHAN, **AMOGH AKSHINTALA**, JOHN ESMET, YIZHENG JIAO, ANKUR MITTAL, PRASHANT PANDEY, PHANEENDRA REDDY, LEIF WALSH, MICHAEL BENDER, MARTIN FARACH-COLTON, ROB JOHNSON, BRADLEY C. KUSZMAUL, AND DONALD E. PORTER.

ACM Transactions on Storage (TOS)
Nov'15

WORKSHOPS

USETL: Unikernels for Serverless Extract Transform and Load. Why should you settle for less?

Awarded Best Paper

HENRIQUE FINGLER, **AMOGH AKSHINTALA**, CHRISTOPHER J. ROSSBACH

APSys '19

Automatic Virtualization of Accelerators

HANGCHEN YU, ARTHUR PETERS, **AMOGH AKSHINTALA**, AND CHRISTOPHER J. ROSSBACH.

HotOS '19

Talk to My Neighbors Transport: Decentralized Data Transfer and Scheduling Among Accelerators.

AMOGH AKSHINTALA, VANCE MILLER, DONALD E. PORTER, AND CHRISTOPHER J. ROSSBACH.

SFMA '18

Projects

Accelerator Virtualization

Jan '18 — Dec '20

- Accelerators seem to operate as horizontally and vertically isolated silos — i.e., the only exposed surfaces that can be interposed are either the user-space API or the HW interface. Leveraging this key insight, we explored an accelerator-virtualization framework that automatically generates much of the code required to forward user-space accelerator APIs through the hypervisor in order to provide the desirable properties of virtualization.
- Role: Instigator, Wordsmith, Mercenary coder.
- Outcome: 1 workshop paper. 2 conference papers.

TMNT: accelerating data movement among accelerators

Jan '18 — May '19

- Data movement is a first-order determinant of performance when programming high-throughput accelerators. When you throw in the additional challenge of co-ordination among multiple accelerators, the problem is compounded because of the synchronous nature of control in the single-task offload model that most accelerators operate under.
- I explored the idea of a capability-based hardware structure that provides the necessary primitives to express data placement/movement, manage co-ordination and scheduling of computation on accelerators, and enforce capabilities for processes running on accelerators. We believe these extensions naturally fit the data-flow programming paradigm, which should greatly ease programmability.
- Outcome: 1 Workshop paper.

GPGPU Virtualization

Summer '17 — May '18

- Many methods have been proposed to virtualize general purpose compute on GPUs. However, none of them hit the right spot. We observed that there are actually two separate elements that must be virtualized when dealing with GPGPUs: device control, and compute. We extended the para-virtual model used by VMware to investigate our hypothesis that handling these two elements separately is the key to achieving good GPGPU performance.
- I built an LLVM backend for TGSI (VMware and Linux (Mesa) vISA for graphics).
- Outcome: 1 conference paper.

Instruction Popularity <http://x86instructionpop.com>

Summer '16 — Fall '19

- Overlapping-ISA multi-core computers have been actively studied in the past decade. In most of these studies, the data used to select the various ISA subsets is collected from a small number of applications: usually popular but outdated benchmarks.
- I built a data collection tool that statically analyzed 9000 C/C++ binaries from the Ubuntu 16.04 repositories to synthesize a higher fidelity data set of static instruction distribution among applications, and a visualization tool.
- Outcome: 1 Conference paper. Visualization tool: x86instructionpop.com

BetrFS (<http://www.betrfs.org/>)

Spring '14 — Spring '16

- The B^ε-tree File System, or BetrFS, is an in-kernel file system that uses B^ε-trees to organize on-disk storage. B^ε-trees are a form of write-optimized dictionaries, and offer the same asymptotic behavior for sequential I/O and point queries as a B-tree. The advantage of a B^ε-tree is that it can also ingest small, random writes 1-2 orders of magnitude faster than B-trees and other standard on-disk data structures.
- I was the Benchmark Czar for the project — I was responsible for measuring and understanding the performance of BetrFS.
- Outcome: 2 Conference and 2 Journal papers. Project still ongoing; my involvement is limited.

Service

- ACM SOCC '18 (External Reviewer)
- USENIX ATC '21 (Program Committee Member)