Akshay Gopalakrishnan

Curriculum Vitae

"Silence is the best teacher.."

Current- PhD Computer Science

Shared memory semantics, also known as memory consistency models, specify how a concurrent program behaves. While sequential interleaving is considered as the de-facto way of reasoning with such programs, current hardware as well as software drift away from it significantly. Semantics for them are termed as weak consistency models, allowing for more concurrent behaviors that result due to optimizations mainly understood as those done by hardware. While hardware models reflect desired optimizations, the same cannot be said for software models. Software (compiler) optimizations have historically been designed for sequential programs, and thus, optimization choices sometimes interact badly with concurrent software. This has resulted in software exhibiting unexpected concurrent behaviors, which makes software correctness a hard task to address. Existing solutions to this are too complex to be feasible in practice. They tend to either make code development difficult or limit the range of potential optimizations. This results in a trade-off between performance, as enabled by aggressive optimization, and the ability to ensure correctness of the optimized piece of software. The goal of this PhD is to unveil a precise and practically useful correlation between memory consistency models and compiler optimizations. In this endeavor, we propose a new design methodology of memory models for software, one that places emphasis on the compiler optimizations desired to be performed. We intend to establish a system of formally constructing models that allow compilers to also do much of the performance-giving optimizations, without sacrificing correct functioning of software.

Education

2021-Today **PhD Computer Science**, *McGill University*, Montreal.
Courses Taken

1. High Level Synthesis of Digital Systems - Winter 2022.
2. Mathematical Tools for Computer Science - Fall 2022.
3. Advanced Computer Systems - Fall 2023 (ongoing).

2018–2021 **Masters of Computer Science** - **Thesis**, *McGill University*, Montreal.
Courses taken

1. Machine Learning - Fall 2018.
2. Teaching in Computer Science - Fall 2018.
3. Compiler Design - Winter 2019.
4. Epistemic & Sociocultural Understanding of Computer Science - Winter 2019.
5. Semantics of Programming Languages - Seminar course at University of Montreal - Fall 2019.
6. Meta-Programming - Winter 2020.

2014–2018 **Bachelor of Information Technology**, *College of Engineering*, Pune.
Relevant Courses Taken

1. Data Structures and Algorithms - 2015.
2. Digital Systems - 2015.
3. Discrete Structures and Graph Theory - 2015.
4. Microprocessor Techniques - 2016.
5. Principles of Programming Languages - 2016.
6. Theory of Computer Science -2016.
7. Operating Systems - 2016.
8. Algorithms and Complexity - 2017.
9. Assemblers and Compilers - 2017.

Masters Thesis

Title *Analysis of the ECMAscript memory model: a program transformation perspective*

Supervisors Professor Clark Verbrugge

Abstract Concurrent programs have been shown to give us tremendous performance benefits compared to their sequential counterparts. With the addition of several hardware features such as read/write buffers, speculation, etc., more efficient forms of concurrent memory accesses are introduced. Known as relaxed memory accesses, they are used to gain substantial improvement in the performance of concurrent programs. A relaxed memory consistency model specifically describes the semantics of such accesses for a particular programming language. Historically, such semantics are often ill-defined or misunderstood, and have been shown to conflict with common program transformations essential for the performance of programs. In this thesis, we give a formal declarative (axiomatic) style description of the ECMAScript relaxed memory consistency model. We analyze the impact of this model on two common program transformations, viz. instruction reordering and elimination. We give a conservative proof under which such an optimization is allowed for relaxed memory accesses. We use this result to reason about the validity of reordering accesses outside loops under the same model. We conclude this thesis by eliciting the limitations of our approach, critique on the semantics of the model, possible future work using our results, and pending foundational questions that we discovered while working on this thesis.

Publications

2023 **Memory Consistency Models for Program Transformations: An Intellectual Abstract**, *International Symposium on Memory Management (ISMM) 2023*, Orlando, Florida, USA, Conference.
Paper

2022 **Reordering Under the ECMAScript Memory Consistency Model**, *Languages and Compilers for Parallel Computing (LCPC) 2020*, New York, USA, Conference.
Paper

2021 **Analysis of the ECMAScript Memory Model: A program transformation perspective**, *McGill University*, Montreal, Thesis.
Link

Invited Talks

Feb 2023 **Memory Consistency and Program Transformations**, *Programming Languages and Systems (PLAS) Seminar*, University of Kent, Canterbury, United Kingdom.

Nov 2022 **Memory Consistency and Program Transformations**, *Compiler Driven Performance (CDP) 2022*, Toronto, Canada.

Oct 2022 **Memory Consistency and Program Transformations**, *Strategic Research Network: Computing Hardware for Emerging Intelligent Sensing Applications (COHESA) NSERC 2022*, Virtual.

Aug 2020 **Analysis of the ECMAScript Memory Model**, *Strategic Research Network: Computing Hardware for Emerging Intelligent Sensing Applications (COHESA) NSERC 2020*, Virtual.

Research Internships

Jan 2023 - April 2023	**Research Assistant**, *University of Kent*, Canterbury, Supervisor: **Professor Mark Batty**. Transformational Specification of Out-of-Thin-Air memory models.
July 2020 - Oct 2020	**Research Fellow**, *Max Planck Institute for Software Systems (MPI-SWS)*, Virtual, Supervisor: **Dr. Viktor Vafeiadis**. Symmetry Reduction for Model Checking Relaxed Memory programs.

Projects

Personal

Summer 2017 (solo)	**Generic Lex (C++)**, Bachelors, College of Engineering Pune. Basic generic lexical analyzer to define any syntax whose grammar is given by the user in the form of regular expressions. ○ Experience in practical implementation of parsing. ○ Project Link : GenLex
Summer 2016 (solo)	**Automata Generator (C++)**, Bachelors, College of Engineering Pune. Automata generator used to define grammars in the form of regular expressions given by the user. Functionalities to convert from deterministic to non-deterministic automata and vice versa and verify if given input belongs to the grammar. ○ Experience in C++ coding, C++ templates and recursion style programming. ○ Project Link : AutomataGen
Summer 2016 (solo)	**Assembler for 8086 (C)**, Bachelors, College of Engineering Pune. A full fledged assembler for a subset of 8086. ○ Experience in practical implementation of assembler pipeline. ○ Successfully designed assembler for 10 instructions of 8086.

Course-based

Winter 2022 (solo)	**Constraint driven Scheduling of fine-grained C concurrency for Reconfigurable Hardware**, *High level Synthesis of Digital Systems*, PhD, McGill University. Proposed to design an optimization addressing resource constraints while synthesizing designs for concurrent hardware. Addresses the scheduling problem of HLS designs for fine grained concurrent constructs. ○ Experience in high level design specification for VHDL circuits. ○ Experience in using Intel Quartus hardware design tool. ○ Project link: HLS-Project.
Winter 2020 (solo)	**Kripke Style Interpretation of lambda circle**, *MetaProgramming*, Masters, McGill University, Montreal. Proposed to represent the "next" temporal logic operator using Kripke Style semantics. ○ Experience in analyzing the corelation between modal logic and temporal logic. ○ Project report link: KripkeLambdaCircle.
Fall 2019 (solo)	**Extending Typer with Linear Types**, *Semantics of Programming Languages*, Masters, University of Montreal. Proposed to extend an existing language with Linear Types. ○ Experience reading and implementing extension of compiler code designed in OCAML. ○ Experience working with Functional programming languages. ○ Project report link:TyperExtLinearTypes.
Winter 2019 (solo)	**Vocabulary and its Influence on Computer Science Research**, *Epistemic and Sociocultural Understanding of Computer Science*, Masters, McGill University. Wrote a research paper that discusses the role of vocabulary in CS research and its influence on CS education across different communities. ○ Experience understanding Pierre Bourdieu's work on "Reproduction in Education, Society and Culture". ○ Link to paper : Vocabulary in CS research and its Impact

Winter 2019 (group of 2)	**Compiler for GoLite (OCAML)**, *Compiler Construction*, Masters, McGill University. Designed a full fledged compiler for compiling a subset of GO language to Python.

- Experience building every phase of compiler (Lexical, Parsing, Semantics and Code gen).
- Experience in using OCAML language to build a fully functional compiler.
- Project Link with report: GoLite Compiler.

Fall 2018 (group of 4)	**Foundations of Programming for Grade 6 students**, *Course: Teaching in Computer Science*, Masters, McGill University.

Designed an entire course on programming foundations, including a subset of lectures, assignments and rubrics for grade 6 students.

- Experience designing syllabus catering towards first time programming students.
- Experience in teaching concepts of geometry using programming.

Fall 2018 (solo)	**Predicting Compiler Optimizations in C (bash, C, Python)**, *Machine Learning*, Masters, McGill University.

Designed and trained a machine learning model to predict the optimal gcc compiler optimization level for a given program.

- Experience using autoencoders.
- Experience data mining for machine learning purposes.
- Project Link with report : Compiler Optimization Prediction

2017-2018 (group of 3)	**Ontology Based Intrusion Detection System (SPARQL, Python)**, *Final Year project*, Bachelors, College of Engineering Pune.

Proposed building an proof of concept Intrusion Detection System (IDS) for a specific application layer based Denial of Service (DoS) attacks using HTTP protocol called as SlowDos.

- Experience in reading research papers, conducting exhaustive literature review and concretely defining research statement for our problem.
- Research work link : Ontology-based IDS.

2016 (group of 3)	**Interactive Debugger and Interpreter for 8086 (Python)**, *Principles of Programming Languages*, Bachelors, College of Engineering Pune.

Designed an interactive debugger for Assembly Language incorporated with GUI crafted using PyQT4 library.

- Experience in debugging 20 instructions of the 8086 instruction set.
- Project Link : Visualemu 8086

2015 (solo)	**Project Othello (C)**, *Data Structures*, Bachelors, College of Engineering Pune.

Designed a turn based game Othello (Reversi) with AI opponents and GUI (using Ncurses).

- Experience in first major programming project.
- Experience in effective strategies in Othello integrating as an AI opponent.
- Project Link : Project Othello.

Teaching Experience

Winter 2019, 2020, 2021, 2022	**Teaching Assistant: Concurrent Programming**, *McGill University*. Duties: Office hours, Grading, Assignment solutions, Invigilation.
Fall 2018, 2021, 2022, 2023	**Teaching Assistant: Operating Systems**, *McGill University*. Duties: Office hours, Tutorials, Grading, Assignment solutions, Admin, Invigilation.
Fall 2019	**Teaching Assistant: Foundations of Programming**, *McGill University*. Duties: Office hours, Tutorials, Grading.
Fall 2020	**Teaching Assistant: Advanced Algorithms**, *McGill University*. Duties: Office hours, Grading, Assignment solutions.

Academic Responsibilities Held

Fall 2019	**Programming Languages and Compilers Reading Group**, *Co-ordinator*, McGill University, Montreal. Duties: Room bookings, finding presenters, scheduling biweekly meetings.

Fall 2019 - Winter 2022	**Lab meetings**, *Co-ordinator*, McGill University, Montreal. Duties: Room bookings, scheduling biweekly meetings.

Awards Received

Research Funds (merit-based)

Fall 2023	**Mitacs Globalink Research Award (GRA) - UK Research and Innovation (UKRI) 2023**, *Mitacs*. The Mitacs Globalink Research Award - UKRI is an initiative that provides the opportunity for faculty members and students at Canadian academic institutions and UKRI institutions to build an international research network and undertake research abroad.
2021-ongoing	**Graduate Excellence Award**, *PhD*, McGill University. Award granted while pursuing PhD at McGill university.
2021-2022	**Murata Family Fellowship**, *PhD*, McGill University. Established in 2010 by Taketo Murata (McGill Alumni). It is awarded by the Faculty of Science to outstanding students in the Faculty at McGill University.

Conference Travel Awards

June 2023	**ISMM/PLDI 2023 SIGPLAN PAC**, *ACM SIGPLAN*.
Jan 2020	**VMCAI 2020 Winter School**, *VMCAI*.
Jan 2020	**PLanQC 2020**, *ACM SIGPLAN*.
Sept 2019	**PLMW SPLASH 2019**, *ACM SIGPLAN*.

Attended Conferences/ Winter Schools

2023	**International Symposium on Memory Management (ISMM)**, *Orlando*, Florida, USA.
2023	**Programming language Design and Implementation (PLDI)**, *Orlando*, Florida, USA.
2023	**Verified Trustworthy Software Systems (VeTSS) Inaugural Meeting**, *London*, UK.
2022	**Isaac Newton Institute (INI) Concurrency Workshop**, *Virtual*.
2021	**Midlands Graduate School (MGS)**, *Virtual*.
2021	**Programming language Design and Implementation (PLDI)**, *Virtual*.
2020	**Languages and Compilers for Parallel Computing (LCPC)**, *Virtual*.
2020	**Heidelberg Laureate Forum (HLF)**, *Virtual*.
2020	**Principles of Programming Languages (POPL)**, *New Orleans*, Louisiana, USA.
2020	**Verification Model Checking and Abstract Interpretation (VMCAI) Winter School**, *New Orleans*, Louisiana, USA.
2019	**Programming Languages Mentoring Workshop (PLMW)**, *Athens*, Greece.
2019	**Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH)**, *Athens*, Greece.
2019	**CS-CAN Student Symposium**, *McGill University, Montreal*, Quebec, Canada.