Pratik Pramod Fegade

CONTACT INFORMATION

555, W Middlefield Rd
Apt. B307
Mountain View, CA 94043

Voice: 412-352-7529
E-mail: pratikfegade@gmail.com
Webpage: pratikfegade.github.io

EDUCATION

Carnegie Mellon University, Pittsburgh, PA — Aug, 2016 - Dec, 2022
PhD in the Computer Science Department
Dissertation Title: Auto-batching Techniques for Dynamic Deep Learning Computations
Advisors: Todd C. Mowry, Phillip B. Gibbons and Tianqi Chen

Indian Institute of Technology, Bombay, India — Jul, 2012 - May, 2016
Bachelors of Technology in Computer Science and Engineering
Honours in Computer Science
Minor in Electrical Engineering
GPA: 9.53/10.0

PROFESSIONAL EXPERIENCE

Software Engineer, Google — Jan, 2023 - present
Working on optimizing deep learning performance on TPUs at Google, with an emphasis on compiler-based techniques.

Research Intern, Oracle, Inc — May - Aug, 2019
Scalable Pointer Analysis of Data Structures Using Semantic Models:
We adapted and simplified previous work on semantically modelling data structures implementations for Andersen's pointer analysis to obtain more precise results, with minimal rise in analysis costs. Implementing this in the Graal Native Image compiler for Java, useful rise in precision (1.35X rise in the number of checkcast statements) was demonstrated with a 19% rise in analysis cost on an average.

RESEARCH PROJECTS

Optimizing Dynamism in Deep Learning Models — Nov, 2019 - Dec, 2022
Graduate Research Assistant, Carnegie Mellon University
Advisors: Prof. Todd C. Mowry, Prof. Phillip B. Gibbons, Prof. Tianqi Chen
Designing compilation and execution techniques for efficient batching in the presence of dynamism in deep learning.
Deep learning models often exhibit control flow (for eg., search procedures such as beam search) and shape dynamism (for eg., ragged tensors in transformer models). We are developing new techniques to efficiently and automatically perform batching in the presence of such dynamism.
This ongoing work has lead to two publications describing tensor compilers for recursive models and ragged tensors, respectively. A third paper is expected to be submitted later this year.

Daedalus: Data Structure Aware Distinctness Analysis — Aug, 2016 - Aug, 2017
Graduate Research Assistant, Carnegie Mellon University
Advisors: Prof. Todd C. Mowry, Prof. Phillip B. Gibbons
Assisted Chris Fallin with his work on an innovative data structure aware static analysis with applications to parallelization and other optimizations.
Contributed to the design of distinctness analysis, a compiler analysis to more precisely infer memory dependences across loop iterations.
Assembled a benchmark suite of irregular, CPU intensive java programs for evaluating Daedalus.
Generally helped with infrastructure development.

Static Resource Bounds Inference for Functional Programs May - Jul, 2015
Research Intern, École Polytechnique Fédérale De Lausanne
Advisor: Prof. Viktor Kuncak
Extended previous work on inferring time bounds of functional Scala programs to add increased capabilities for inference of non linear bounds. Worked also on inferring bounds on stack usages.
Worked on Leon, an automated system for verification and synthesis of functional Scala programs built at EPFL.
Added support for inferring non linear time bounds of recursive functions by a using composition of bounds on number of recursive calls and time per recursion for recursive functions.
Developed an empirical model of stack usage of Scala programs through a survey of the generated bytecode for Scala programs. Evaluated the results of stack bounds inference by measuring the stack usage by actually executing the programs under consideration.

Concurrent Program Verification May - Jul, 2014
Research Intern, Institute of Science and Technology, Austria
Advisor: Prof. Thomas Henzinger
Developed a system using ordering predicates on executions of statements of concurrent programs with the aim of verifying them.
Developed an extension to an existing framework based on the CEGAR (CounterExample-Guided Abstraction Refinement) approach to include ordering predicates.
Created a set of sound and complete inference rules for these predicates.
Implemented a proof of concept in OCaml and proved the correctness of Peterson's algorithm.

Refereed Publications

ACRoBat: Compiler and Runtime Techniques for Efficient Auto-Batching of Dynamic Deep Learning Computations
Pratik Fegade, Tianqi Chen, Phillip B. Gibbons and Todd C. Mowry
To appear at the Seventh Conference on Machine Learning and Systems, 2024

ED-batch: Efficient Automatic Batching of Dynamic Neural Networks via Learned Finite State Machines
Siyuan Chen, Pratik Fegade, Tianqi Chen, Phillip B. Gibbons and Todd C. Mowry
International Conference on Machine Learning. PMLR, 2023

The CoRa Tensor Compiler: Compilation For Ragged Tensors With Minimal Padding
Pratik Fegade, Tianqi Chen, Phillip B. Gibbons and Todd C. Mowry
Fifth Conference on Machine Learning and Systems, 2022

Cortex: A Compiler for Recursive Deep Learning Models
Pratik Fegade, Tianqi Chen, Phillip B. Gibbons and Todd C. Mowry
Fourth Conference on Machine Learning and Systems, 2021
One of five Outstanding Papers in the Conference

Scalable Pointer Analysis of Data Structures Using Semantic Models
Pratik Fegade and Christian Wimmer
ACM SIGPLAN 2020 International Conference on Compiler Construction, San Diego, California, USA, 2020

Other Projects

Improvements in Container based Virtualisation Aug, 2015 - Apr, 2016
Undergraduate Thesis Project, Indian Institute of Technology, Bombay
Advisors: Prof. Umesh Bellur, Prof. Purushottam Kulkarni
Surveyed and experimented with ways to impose limits on usage of resources like CPU and IO, specifically in Docker containers.

Load Generator Scalability Improvement Jan - April, 2015
Research and Development Project, Indian Institute of Technology, Bombay
Advisor: Prof. Varsha Apte
Studied the operation and implementation of a load generator and suggested optimisations to improve its scalability and capacity.
Profiled and instrumented the load generator code to identify possible code to optimize.
Optimized the execution of individual worker threads to improve the single core load generation capacities by about 6X.
Improved multicore scalability by reducing synchronization between the worker threads.

Service

Served as an External Reviewer for the Conference on Object-Oriented Programming Systems, Languages, and Applications, 2022 Jun, 2022 - Aug, 2022

Served on the Artifact Evaluation Committee for Fifth Conference on Machine Learning and Systems, 2022 Feb, 2022 - Mar, 2022

Served on the Artifact Evaluation Committee for IEEE/ACM International Symposium on Code Generation and Optimization, 2023 Dec, 2022

Member of the SCS Dean's PhD Advisory Committee at CMU Dec, 2020 - Present
Carnegie Mellon University

Master of Science in Computer Science Admissions Committee Dec, 2018 - Feb, 2019
Carnegie Mellon University

Teaching and mentorship

15-300: Research and Innovation in Computer Science Aug - Nov, 2018
Carnegie Mellon University, Teaching Assistant

15-745: Optimizing Compilers for Modern Architectures Jan - May, 2018
Carnegie Mellon University, Teaching Assistant

CS 213 (minor): Data Structures and Algorithms Jan - Apr, 2016
Indian Institute of Technology, Bombay, Teaching Assistant

CS 296: Software Systems Laboratory Aug - Nov, 2015
Indian Institute of Technology, Bombay, Teaching Assistant

Signals and Systems MOOC on edX run by IIT Bombay Dec - Jun, 2015
Indian Institute of Technology, Bombay, Teaching Assistant

Department Academic Mentor Aug, 2014 - Apr, 2015
Mentored 5 sophomores in academic and general matters at Indian Institute of Technology, Bombay.

Skills

Proficient in Java, C++. Familiar with Python, Datalog, LLVM, TVM.

Academic Honours and Achievements

Secured **All India Rank 16** in **IIT JEE** and **All India Rank 38** in **AIEEE**.
Invited for the **ITCSC-INC Winter School** held at the Chinese University of Hong Kong, Hong Kong in January 2014.
Offered **KVPY**, **NTSE** and **INSPIRE** fellowships.