Tim Baer

630-841-2533 — tbaer2@illinois.edu

Education

University of Illinois at Urbana-Champaign *May 2022*

Bachelor of Science in Mathematics & Computer Science *CS GPA: 3.83*

Selected coursework: parallel numerical algorithms, tensor computations, randomized algorithms, approximation algorithms, numerical analysis, parallel programming

Research Projects

Parallel Minimum Spanning Forest *August 2019 — October 2021*

Advisor: Edgar Solomonik *UIUC*

Generalized matrix algebra has emerged as an expressive and powerful framework for designing parallel graph algorithms. We develop the first formulation of the Awerbuch-Shiloach parallel minimum spanning forest (MSF) algorithm using matrix algebra primitives. To achieve this, we introduce a novel multilinear primitive that updates graph vertices by simultaneously using information from both adjacent edges and vertices. We demonstrate scalability of our MSF algorithm on up to 256 nodes (16K cores) on some of the largest publicly available graphs. I was responsible for the design of the algorithm and implementation with the Cyclops Tensor Framework, a distributed-memory library for generalized sparse tensor algebra. In addition, I developed two optimizations to accelerate the shortcutting step.

Parallel Single Source Shortest Path *September 2020 — current*

Advisor: Edgar Solomonik *UIUC*

We explore connections between a recent theory paper on parallel SSSP in nearly optimal work and depth (Andoni, Stein, Zhong, STOC 20) and fast iterative solvers like algebraic multigrid. I developed a highly scalable algorithm for the main subroutine: given a vertex and a number k, find the k closest neighbors to that vertex. I expressed this calculation as a sequence of sparse matrix-vector multiplications (SpMV) with the adjacency matrix and a vector $\boldsymbol{b}$ containing a sorted list for each vertex's tentative k closest neighbors. I found experimentally that it is faster to express this subroutine with the multilinear primitive described above. With this efficient subroutine, I then implemented the algorithm's main data structure called a low hop emulator.

Papers

Parallel Minimum Spanning Forest Computation using Sparse Matrix Kernels *October 2021*

Tim Baer, Raghavendra Kanakagiri, and Edgar Solomonik

In submission to PP22

Talks

Graph Algorithms with Sparse Linear Algebra *July 2021*

C3.ai Platform Engineering (slides)

Low Hop Emulators and Uncapacitated Min-Cost Flow *October 2020*

UIUC Parallel Algorithms Reading Group (slides)

Awards

Franz Hohn and J.P. Nash Scholarship *May 2021*

Awarded to one undergraduate in recognition of outstanding scholarship and promise in applied mathematics, computational science, or scientific computing

James Scholar

Selection is based upon academic achievement as well as diversity of identity, geography, and major/area of study

Teaching

Software Design Studio *August 2019 — May 2020*

Moderated and graded weekly code review sessions focusing on best practices

Introduction to Computer Science *January 2019 — August 2019*

Facilitated a lab section and held office hours

Internships

C3.ai — Software Engineering Intern *June 2021 — August 2021*

Machine Learning Engineering

- Researched open-source distributed deep learning frameworks & designed end-to-end integrations with the platform
- Developed a transpiler for Python code generation to avoid serialization of large arrays across languages, improving performance up to 4000x
- Presented hour-long talks on internship work & undergraduate research
- Selected for a filmed 1:1 intern testimonial interview published to C3.ai's website

IBM — Software Engineering Intern *June 2020 — December 2020*

Db2z Tools

- Integrated static/dynamic code analysis into CI pipeline to detect security issues
- Automated API & unit tests to improve code coverage by 85%
- Developed a new internal documentation tool

All Information Services — Technology Intern *May 2019 — August 2019*

- Automated client workflow for file syncing with Microsoft Graph API
- Developed Flask app to integrate Slack with our service desk platform
- Leveraged NLP to automate classification of support ticket issue types