

Lance Bergeron

Phone: (804) 754-6265 Email: bergeron.lance6@gmail.com Github: github.com/lanceberge

Experience

Amazon

June 2022 - March 2023

Software Development Engineer

Seattle, WA

- Contributed to a team which developed a comprehensive graph model of the Amazon Transportation Network which captures real-time data on cost, capacity, and time restraints
- Applied optimization techniques to identify improvements to the network, resulting in fewer late deliveries and reduced costs
- Created an integration test verification stage on the team's CI/CD pipeline in order to automatically test that new changes pass integration tests before they are pushed to production
- Conducted profiling and optimization of the model, leading to a 15% decrease in its runtime
- Developed a command-line tool which creates smaller graphs representing subsets of the full model
 - This enabled teammates to run many of the team's integration tests in minutes, which previously took hours
- Created a data pipeline to retrieve information on package and warehouse types, then implemented a package routing restriction based on this information
- Leveraged various AWS services, including EC2, S3, Lambda, SQS, Dynamo DB, and Cloudwatch

Education

Virginia Tech

Aug 2019 - May 2022

Bachelor of Science in Computational Modeling and Data Analytics

3.65 GPA

Minor in Computer Science, Minor in Math

- Graduated Magna Cum Laude

Skills

Languages Java, Python, C, C++, Bash, Matlab

Technologies AWS, Git, SQL, JUnit, CUDA, openMP, Unix, PyTorch

- Parallel programming and algorithm design in Java, C, and with CUDA
- Unit testing, integration testing, test-driven development
- Object-oriented programming and design patterns
- Machine learning algorithms including: linear, logistic, and penalized regression, time series, bootstrapping and cross validation
- Convolutional, fully-connected, and recurrent neural network structures
- Agile development methodologies

Projects

CUDA Newton Fractals [link](#)

Nov 2021 - Jan 2022

- Used CUDA and C++ code to asynchronously perform the convergence of Newton's iteration to find the roots of a randomly-generated polynomial
- Implemented a parallel hash algorithm to find the unique roots of that polynomial
- Created a command-line tool that outputs Newton fractals by coloring points on the complex plane based on which root they converged to