Aneesh Durg

Email: aneeeshdurg17@gmail.com | Website: aneeshdurg.me | Github: github.com/aneeshdurg

EDUCATION

University of Texas at Austin
Aug 2025 - Present

PhD student in **Computer Science**

University of Illinois at Urbana-Champaign
Aug 2015 - May 2019

Recieved BS in **Computer Science & Mathematics** with **High Distinction**

RESEARCH EXPERIENCE

Research Assistant
Apr 2024 - Present

University of Washington (Prof. Simon Peter)

- Investigating the role **runtime reconfigurable networking** will play in large scale **distributed graph applications**
- Benchmarking the effect of changing network topology on real world distributed graph databases
- Built a framework to enable running existing applications (such as distributed graph databases) in customizable network topologies

WORK EXPERIENCE

AI Software Engineer
Feb 2025 - Present

Corvic AI — remote

- Designing and building an agentic platform to build and customize agents connected to a repository of data.
- Helping design and build an **LLM**-assisted data pipeline generation tool
- Converted operators to work with incremental streams of data, **reducing costs by an order of magnitude** in some cases.
- Added internal tools to write and safely execute migrations of the system database.
- Added support for deleting input data and purging any system generated artifacts related to a deleted input.

Senior Software Engineer
Jul 2023 - Nov 2024

Bodo.ai — remote

- Developing the core engine - an optimizing compiler and scalable distributed runtime (using **MPI**) for **SQL** and python/pandas workflows.
- Designed and implemented a distributed streaming external sort operator for a **2x** speedup in some benchmarks.
- Expanded **Iceberg** support by implementing **DDL** operations and adding integrations with the **AWS Glue** catalog
- Helped redesign and implement orchestrator/worker compilation model to hide distributed semantics from users.

Senior Software Engineer/Team Lead
Feb 2021 - Jun 2023

KatanaGraph — Austin, TX

- Worked on building a distributed graph compute engine that provides AI, analytics, and a graph database.
- Lead a team of **5** to implement and support graph database querying and ingest.
- Implemented compiler and runtime support for the **Cypher** query language.
- Designed and implemented novel high performance algorithms for distributed subgraph pattern matching (tested on ~**20B** nodes, **44B** edges)
  - Improved performance by **100x** in queries against the **LDBC-SNB** datasets and reduced memory usage by over **95%** on benchmarks simulating specific client workloads.
- Designed and implemented hotswap mechanism to enable testing new code on existing **kubernetes** deployments - reduced iteration time by **30x**
- Built infrastructure for benchmarking the query engine in isolation from the rest of the product using **slurm**

Member of Technical Staff
Aug 2019 - Feb 2021

Qumulo — Seattle, WA

- Worked on building a distributed scale-out filesystem, supporting both on-prem and cloud.
- Designed and implemented a solution for reducing server downtime during upgrades by **10x** in a team of four
- Implemented **SMB3.1** support and features, and extended platform support for two new hardware configurations
- Lead migration of **python2** code to **python3**, and introduced enforced type checking via **mypy**

PROJECTS

rainbow
https://github.com/aneeshdurg/rainbow

- Static analysis tool for **C/C++** to reject semantically invalid callgraphs, powered by **clang** and **Cypher**
- Provides an ergonomic way for users to label functions and lambdas and to define relationships between those labels that should be considered invalid. Some example usecases are:
  - Prevent functions that assume locks are held from being called without a lock
  - Prevent functions using collective **MPI** operations from being called during another collective operation

spycy
https://github.com/aneeshdurg/spycy

- An in-process graph database library for python that implements a **openCypher** frontend
- Provides implementable interfaces for data sources to enable querying real world graphs.
  - Wrote a demo that uses **spycy** and **WASM** to filter HTML nodes in a browser using **openCypher**
  - Wrote a demo that uses **spycy** and **LLVM** to implement compiler passes in **openCypher**

What Is a Filesystem?
https://aneeshdurg.me/what_is_a_filesystem

- An online interactive book/visualization for students learning filesystem concepts.
- Implements a interactive **ext2**-esque filesystem simulator with animations to illustrate disk accesses
- Features a terminal simulator demonstrating how standard **GNU/Linux coreutils** might interact with the disk.