ARUNIT BAIDYA

Boston, MA, 02130 (Ready to Relocate)
Github LinkedIn ☎+16179355115 ✉ baidyaarunit@gmail.com

Education

Northeastern University, Khoury College of Computer Sciences — Aug 2024 – April 2026

Master of Science, Computer Science — Boston, MA

Anna University, PSG College Of Technology — Aug 2018 – July 2022

Bachelor of Engineering, Computer Science and Engineering — Coimbatore, India

Skills

- **Technical Skills**: Java, Python, C, Go, API Development, object-oriented programming, object-oriented design, PostgresDB, RestAPIs, SQL, Spark, relational databases, MVC, design patterns, Operating systems, Git, Backend, troubleshooting, AWS, AWS SNS, AWS S3, OpenSearch, Docker, Redis, Kafka
- **Software Development:** CI/CD, Agile Development, Test Driven Development, Code Reviews, Problem-solving
- **Collaboration:** Excellent communication skills, cross-functional collaboration, domain knowledge interest

Work Experience

Arcesium

Software Engineer — **Jul 2022 - Jul 2024, Bangalore**

- **Met client SLOs** (max 10 min time to output for datasets with 1,000–100,000 records) by optimizing SQL operations, debugging **Python & Java** code using **Spark**, ***and improving process times by 40%.***
- **Designed partitioning strategy** for constant read/write performance in Delta-Tables with historical data.
- Developed a **config-driven data-filtering** module in Java and Spark using partition pruning for **static/dynamic filters**, ***reducing compute resource usage and improving workflow efficiency by 30%.***
- **Optimized resource provisioning buckets** using ArgoWorkflow, Kubernetes & AWS, ***adopted in 10+ workflows.***
- Implemented **SQL** for SCD type 2 merge, enabling bi-temporality & historical querying, state of data at point in time.
- **Created LLD documentation** with business logic translation, ***reducing*** cross-functional team ***inquiries by 50%.***

Software Engineering Intern — **Feb 2022 - May 2022, Remote**

- ***Increased ETL throughput by 70%*** via preprocessing stage to split heterogeneous data for parallel processing.
- **Built data ingestion module** supporting multiple input types with schema validation for data consistency.
- **Boosted ETL reliability** by engineering module in python to extract malformed data to global kickouts delta-table.
- **Enhanced ETL observability** with efficient run history logging and storing execution stats in Delta-Tables.

Software Engineering Intern — **May 2021 - Jul 2021, Remote**

- **Automated PDF Parsing** in **python** to identify and parse tabular information using OpenCV and Camelot.
- Implemented **CRON processing** to parse PDFs mapped against filters & annotations, ***saving 4 hours work/week***.

Projects

Distributed Overengineered TODO API (Go, Python, Redis, Kafka)(Project) — December 2024, Boston

- Designed a TODO API with CRUD functionality using **PostgreSQL & PostgresDB**, employing factory & strategy patterns to achieve **SOLID** design principles and extensibility to new data storage methods & query strategies.
- Deployed application using **Docker** & **Kubernetes**, with JWT authentication, rate limiting, **Kafka** for streaming events from API requests and consuming events, and **Redis** for caching and performing **batch database updates.**

Image Processing Application (Java) (Course Project) — October 2024 – November 2024, Boston

- Developed an image processing application in **Java** supporting CLI, script, and GUI inputs, demonstrating SOLID principles, higher order functions, MVC, design patterns, and test-driven development.

Qthreads – Cooperative user space thread library (C) (Course Project) — November 2024, Boston

- A **user-space thread library** in **C**, like POSIX threads library, with functions to create threads, yield threads, sleep threads, provide mutexes & condition variables for managing concurrency in a **cooperative threading** model.

Simple Linux Shell (C) (Course Project) — October 2024, Boston

- A Linux shell in **C** with command execution for internal & external commands, piping, & I/O redirection using **fork**.