Ramya Bygari

✉ rbygari2@illinois.edu   in linkedin.com/in/ramyabygari   https://ramyabygari.github.io/

Education

University of Illinois Urbana-Champaign **Champaign, IL, USA**
[Master Of Science, Computer Science, **GPA: 4.0/4.0**] August 2022 - May 2024

Relevant Courses: Distributed Systems, Fault-Tolerant Consistent DataCenter Systems, Cloud Storage Systems, Independent Study/Research on distributed systems for GNNs

Teaching Assistant: Introduction to Operating Systems (150+ students), Data Science Discovery (600+ students)

National Institute of Technology Karnataka (NITK) **Surathkal, India**
[Bachelor of Technology - Computer Science] August 2016 - May 2020

Relevant Courses: Advanced DB Systems, OS, Computer Architecture, ML, Data Mining, Big Data, Computer Networks

Experience

Omnistrate | Software Engineer Intern [go, kubernetes] [May 2023 - Aug 2023] [Redwood City, USA]

- Developed end-end horizontal autoscaling feature enabling zero-instance scaling without restarts in a multitenant environment. Accommodates multiple writers, coupled/serverless architectures, stateless/stateful apps. Dynamically adjusting of CPU and memory allocation for efficient resource utilization, scalability, and availability in response to failures.
- Integrated autoscaling with a sidecar component for real-time process-level monitoring, enhancing instance responsiveness tracking. Improved scaling decisions and ensuring high availability through accurate performance assessment.

Razorpay | Software Engineer [Scala, Hadoop, Python, pSQL] [Aug 2020 - Aug 2022] [Bangalore, India]

- Smart Routing : Revamped batch processing system to scalable stream processing ML Random Forest system using Spark and Kafka. Transformed real-time payment transactions, boosting success-rate by 6% and enhancing user engagement while reducing product gap. **https://ieeexplore.ieee.org/document/9671961**
- Deployed distributed batch processing pipelines for seamless data analysis & modeling, ensuring reliable and high throughput user personalization infrastructure. Developed data pipelines for efficient extraction, processing terabytes of data.

Adobe Systems | Product Intern [C++] [May 2019 - July 2019] [Bangalore, India]

- Developed a plugin to support platform-independent, variable frame-rate animation and low latency export through Lottie, an open-source JSON-based animation file. Reduced the animation export time by 60% (5 mins to 2 mins).

Projects

Distributed Serverless GNN Training Framework System [C++, go, CUDA] [May'23]

Developed distributed system for GNN training, leveraging serverless computing for scalable and cost-effective computation. Overcoming limitations of limited memory and expensive GPUs, by easily adjusting distributed compute units and their computing power with this model.

Exploiting Nil Externality in message queues Project Link [Java, Kafka, rpc, storage systems] [May '23]

Developed KSkyros, optimizing message queues through nil-externalizing interfaces. Integrated Skyros replication protocol into Apache Kafka, reducing produce request latency by a minimum of 25% for throughput-optimized Kafka.

Syros - Read Optimized Consensus Protocol Project Report [C++, socket, replication] [May '23]

Incorporated Paxos Quorum Reads and Skyros Replication Protocol to enhance read operation latency and maintain strong consistency. Achieved 50% drop in latency compared to Paxos for mixed read-write workloads and nearly 58% increase in throughput for read-only workloads in geo-replicated environments.

Distributed ML Platform & File System Project Link [C++, go, concurrency] [Dec '22]

Developed Distributed Scalable Fault-tolerant a) ML Platform, ensuring consistent model-serving and prediction with fair-time inference & resource sharing. Effectively handles recovery from n-simultaneous node failures, b) File System, enabling versioned-file storage, retrieval, & updates using consistent hashing and RPC communication among replicas.

Operating Systems Simulator [C, C++, multithreading] [May '20]

Developed a simulation environment, replicated functions and interactions between the kernel components, such as system calls, process and disk scheduling, memory management and page replacement algorithms.

ECN+ and ECN+/Wait Congestion Control Algorithms Project Link [C, C++, open-source] [Dec '21]

Implemented ECN+ and ECN+/Wait algorithms in ns-3 for efficient congestion management in network communication. ECN+ extends ECN to TCP control messages with caution regarding SYN packets. ECN+/Wait, an extension of ECN+, introduces a 1 RTT waiting period upon detecting congestion, optimizing congestion control.

Cancer Diagnostic Systems Project Link [python, keras, tensorflow, ML] [Dec '21]

Developed novel multistage deep neural network method for automated prostate cancer grading (Accuracy@92.38%) providing heatmap visualization explainability. **https://link.springer.com/chapter/10.1007/978-981-19-5868-7_21**
Deployed a modified LadderNet architecture, for molecular subtyping of breast cancer from IHC images, streamlining the pathology procedure and enabling targeted therapy. **https://ieeexplore.ieee.org/abstract/document/10038704**

Technical Skills

Languages: Python, Scala, C/C++, Java, CUDA, JavaScript, GO
Web Technologies: MySQL, NOSQL, Flask, MongoDB, React, REST, AngularJS, Ajax
Tools: GDB, Valgrind, Docker, Kafka, AWS, Airflow, Hadoop, Kubernetes, ElasticSearch, Cassandra, DynamoDB, Apache Spark, PySpark, Tensorflow, Keras, Pytorch, Github, JIRA, Confluence, Grafana, Looker, Prometheus