

PRABHUDATTA MISHRA

206-581-9647 | pmishra4@ncsu.edu | [Github](#) | [Linkedin](#)

EDUCATION

North Carolina State University, NC USA

Aug 2023-Dec 2024

Master of Computer Science

CGPA 3.67/4

Relevant Courses: Operating Systems, Linux Networking, Cloud Computing, Multi-cloud and Hybrid-cloud architecture, Design and Analysis of Algorithms. **Other Courses:** Distributed Systems

PROFESSIONAL EXPERIENCE

Research Assistant | North Carolina State University, Raleigh

Feb 2025 -

- **Design and manage multi-cloud Kubernetes clusters** for machine learning and computer vision applications, ensuring **scalable, fault-tolerant** deployments across **AWS, Azure, and GCP**.
- Optimize **container orchestration pipelines** for **large-scale data processing and training workloads**, improving system performance and resource utilization, while collaborating with cross-functional teams to implement cutting-edge **ML frameworks** and drive advancements in **computer vision solutions**.

Software Engineer | Glosity, Bhubaneswar, India

Jan 2023-July 2023

- Orchestrated the deployment of a **scalable profile recommender system** on **AWS and DigitalOcean**, achieving a **44% increase** in hiring match accuracy, and designed a **multipurpose chatbot on AWS** for 24/7 academic support, **reducing response times by 50%** and boosting user engagement.
- Strengthened security protocols **using AWS KMS** while optimizing **load balancing** and **auto-scaling** strategies for a **31% reduction in infrastructure costs**, and integrated **AWS CloudWatch and Azure Monitor** for comprehensive **monitoring**, resulting in a **25% decrease in downtime** and faster issue resolution.

CERTIFICATIONS

- **AWS SOLUTIONS ARCHITECT | ONGOING**
- **CERTIFIED KUBERNETES ADMINISTRATOR | ONGOING**

PROJECTS

Hybrid Load Balancer | Dynamic Traffic Management, Fault Tolerance, Golang | [Github](#)

- Constructed a **hybrid Layer 4/Layer 7 load balancer** in Golang, dynamically managing traffic across distributed systems with high **fault tolerance** and **scalability**. Leveraged key metrics—**CPU utilization, memory usage, response time, error rate, and ping status**—to compute server health and optimize intelligent traffic distribution using **weighted round-robin, IP hashing, and sticky sessions**.
- Integrated advanced **circuit breaker** mechanisms to **isolate failing nodes**, ensuring **continuous availability**. Developed a **real-time monitoring dashboard** with server health and traffic metrics, enabling rapid diagnostics and seamless scaling in distributed environments.

Age-Based and Linux-Like Scheduler | C, OS Kernels, Process Scheduling, Real-Time Monitoring | [Github](#)

- Implements an **epoch-based scheduler** where each process receives a **CPU quantum** computed as either its **base priority** (if fully used previously), ensuring efficient and **fair time allocation**, and integrates an **age-based boosting mechanism** that raises the scheduling “goodness” of **long-waiting processes**, preventing perpetual delays under heavy loads.
- Combines **dynamic quantum recalculation** with **age-based priority** adjustments to ensure efficient, fair, and **starvation-free CPU allocation**.

Fault-Tolerant Primary-Backup Service | View Server, Server Failover Mechanism, Java | [Github](#)

- Integrated a ViewServer to monitor **server health**, promote **backups to primary during failures**, and maintain seamless client-server communication during network partitions, **improving system availability by 99.99%**.
- Engineered robust failover with primary-backup replication and exactly-once RPC delivery, **ensuring 100% data consistency** and **reducing recovery time during server outages by 40%**. Synchronized state replication enhanced reliability and supported seamless failover for uninterrupted service.

Paxos-Based Replicated State Machine | Consensus Algorithm, Fault Tolerance, Java | [Github](#)

- Developed a **fault-tolerant** state machine leveraging the **Paxos Consensus Algorithm** for consistent, reliable command ordering in **distributed key-value stores**, resilient to network and server failures.
- Enhanced efficiency through **multi-instance Paxos, leader election optimization, dynamic quorum adjustments**, and garbage collection mechanisms for effective log and memory management.

Resilient Peer-to-Peer File Sharing System | Decentralized Architecture, Network Optimization, Python | [Github](#)

- Innovated a highly **scalable P2P system** inspired by BitTorrent, enabling **efficient and fault-tolerant** file distribution with **dynamic peer coordination** and **secure data exchange**.
- Improved network performance using advanced strategies like **optimistic unchoking** and peer prioritization, **boosting data throughput by 35%**, while incorporating **file integrity verification** and **real-time tracking** for enhanced **reliability and scalability**.

Scalable Distributed Pub-Sub System with Real-Time Monitoring | Go, Distributed Messaging system, Prometheus, Grafana | [Github](#)

- Developed a scalable distributed Pub-Sub system in **Go**, leveraging goroutines for concurrency, real-time monitoring with **Prometheus and Grafana (reducing memory usage by 30% and latency by 50% vs. Java)**, and enhancing fault tolerance/persistence via message partitioning, broker replication, and **distributed key-value stores**.

TECHNICAL SKILLS AND CERTIFICATIONS

- **Languages and Databases:** Python, Java, Go, C
- **Databases:** MySQL, PostgreSQL, MongoDB, Cosmos DB
- **Cloud Technologies :** AWS(EC2, S3, Lambda, CloudWatch, KMS), AZURE(Key Vault, Azure Monitor, Azure Functions)
- **DevOps:** Terraform, Ansible, Jenkins, GitLab CI/CD, Prometheus, Grafana, Docker, Kubernetes
- **Networking:** TCP/IP routing, Vxlan, DNS, DHCP, VPN, CDN, Firewalls