

# VIKRAM BHATTACHARJEE

Email: vbhattac@alumni.cmu.edu

Github: <https://github.com/vbhattaccmu>

Contact No: +85293158106

---

## EDUCATIONAL BACKGROUND

### Carnegie Mellon University

**Masters:** Electrical & Computer Engineering, 3.32/4.00

**Relevant Coursework:** Numerical Methods in Engineering Design and Optimization (18660), How to Write Fast Code (18645), Data Structures and Applied Programming (08722), Distributed Systems (15640), Data Warehousing (95797), Web Application Development (15637).

Location: Pittsburgh, PA, USA

Graduation Date: 22<sup>nd</sup> May, 2018

### Jadavpur University

**B.E:** Chemical Engineering, 7.48/10.00

**Relevant Coursework:** Numerical Methods and Programming, Introduction to Computer Programming, Operations Research, Project Engineering & Management.

Location: Kolkata, India

Graduation Date: 26<sup>th</sup> May, 2015

---

## TECHNICAL SKILLS

**Languages:** C++-11/14/17, Rust 1.66.

**Cloud Services:** AWS Lambda, API Gateway, Sagemaker, EFS, S3, EC2.

**Databases:** Postgres, AWS DynamoDB, RocksDB.

**Stream Processing Frameworks:** Apache Kafka, Apache Zookeeper, Kafka Connect, RabbitMQ.

**Containerization Tools:** Docker.

---

## PROFESSIONAL EXPERIENCE

### ParallelChain Lab

#### Staff Engineer

Location: Hong Kong, Hong Kong SAR

Duration: 24th January, 2022-Present

#### Roles: -

- Part of the Mainnet team developing ParallelChain, a turing complete L1 smart contract platform in Rust.
- Development of features related to the ParallelChain Testnet 4.0 and the Mainnet release (mempool gossip, networking layer, pchain-runtime, gas-metering middleware).
- Owning and maintaining the development of the services (back-end for front-end) stack (ChainScanner, Explorer API, Faucet and CPS) for the ParallelChain Mainnet/Testnet ecosystem.

#### Achievements: -

- Developed a caching service, API gateway and load balancing crate in Rust to catchup and serve blocks/transactions to the Explorer (front-end) for Testnet 3 and Mainnet which serves as the core component of user interaction with ParallelChain. The service recorded a net DAU of 10000 users towards the end of Testnet 3 and is still operational on [Mainnet](#).
- Developed a guest memory adjustment layer for ``pchain_runtime`` in order to reduce CPU load on the server during cross-contract/view calls from client side.
- Co-developed a non-determinism filter and methods to assign costs for WASM op-code and storage (world-state) read/writes for gas-metering on ``pchain_runtime``.
- Developed ``pchain_compile``, the smart contract compiler using Docker in Rust for the ParallelChain ecosystem.
- Researched and prototyped SNARK verification circuits for ed25519 signature scheme in ``pchain_runtime`` to reduce gas costs for call and deploy type transactions on Testnet 4.
- **Operating System used:** Linux (Debian).
- **Languages/Libraries used:** Rust 1.66, libp2p, warp, plonky2-ed25519, tokio.
- **Databases used:** Postgres (mobic postgres), RocksDB.
- **Containerization Tools used:** Docker.

### Read-Ink Technologies Pvt. Ltd.

#### Software Engineer

Location: Bengaluru, KA, India

Duration: 10<sup>th</sup> February, 2020-7<sup>th</sup> December, 2021

#### Roles: -

- Responsible for cross platform shared library development for OCR on online and offline platforms.

#### Achievements: -

- Developed and implemented a fast orthographic curve matching module with C++-14 by inheriting concepts of matrix multiplication using macros and SVD from Eigen library into the stack which brought down recognition latency to 80 ms per character on average on a page having 600-700 words.

- Improved library performance by performing memory profiling across different components of the code base and upgrading their design patterns.
- Improved loading time for parsing models into memory of the library by utilizing simdjson for select customers relying on Intel chipsets for daily operations with the library.
- Developed build system generators for shared library generation using CMake/Make for Windows and Ubuntu platforms which improved downtime during production.
- **Operating System used:** Linux (Debian), Windows.
- **Languages/Libraries used:** C++-14/17, boost, CMake, Python 3.6.
- **Cloud Platform used:** AWS.
- **Debuggers/Simulation/Profiling Tools used:** gdb, valgrind.

## **Hyllion Holdings Corp.**

### **Simulation Engineer**

Location: Austin, TX, USA

Duration: 18<sup>th</sup> June, 2018-18<sup>th</sup> October, 2019

#### **Roles: -**

- Worked with the embedded software engineering team for development of a simulation platform using C++ and MATLAB/Simulink for in-house control algorithm testing, fuel-economy validation, and improvement.

#### **Achievements: -**

- Successfully ported Fortran packages for linear interpolation utilized in embedded C++ code to the MATLAB s-function environment for comparing its performance, deviations with solvers like interp, euler, etc.
- Designed and deployed an adaptive weight estimation algorithm in C++-11 running on Linux over J1939 CAN Bus for the Hybrid-Ex product. The project met code review, OTR (over the road testing) requirements and it contributed to improving fuel savings for major fleet operators in northern US region.
- **Operating System used:** Linux (Debian).
- **Languages/Libraries used:** C++-11, STL, CMake.
- **Debuggers/Simulation/Profiling Tools used:** MATLAB/Simulink, MPLAB, gdb, valgrind.

## **RELEVANT OPEN SOURCED GITHUB PROJECTS**

### **XRPL Rate Limiting Hook**

Duration: 1st March, 2023-1st April, 2023

#### **Description:-**

- Developed a hook for implementing a rate limiter for transactions on XRP Ledger which can be built on XRPL Hooks Builder platform for deployment on any of the corresponding Testnets.
- The rate limiter hook allows a maximum spending limit of 100 XRP on a window of 5 minutes for the signer.
- The hook uses custom state management and a custom protocol along with the “hookapi” library to perform rate limiting on a signer account.

### **LiarsLie**

Duration: 1st December, 2022-1st January, 2023

#### **Description:-**

- Presented an approximate solution to the Byzantine Generals’ Problem to determine the true value of an attribute gossiped around in a network of n agents where m ( $m \leq n$ ) agents always lie and the rest always speak the truth.
- Utilized go-libp2p library for peer discovery of these agents and Kadmelia DHT for storing quorums from each agent during each round of consensus.

### **A Minimal Back Tester and Order Management System in C++**

Duration: 1st October, 2022-Present

#### **Description:-**

- Developed a C++ mini back testing application where the user can implement their own strategy to place an order to the order management service.
- Designed an order matching algorithm that on receiving an order from the Strategy module, tries to fill the order.

## **RELEVANT INTERNSHIPS**

- Worked as an intern at University of Hochschule Augsburg, Department of Electrical Engineering under the guidance of Dr. Michael Finkel to develop smart load flow algorithms for assessing power factor control in micro grids over German rural regions using DIGSILENT PowerFactory and C++-11. (May. 2015-July 2015).