Dipesh Kafle

dipesh.kaphle111@gmail.com | github.com/dipeshkaphle | linkedin.com/in/dipeshk111 | dipeshkaphle.github.io

Curious Software Engineer with a strong interest in Programming Languages, Formal Verification and Systems Programming.

Education

National Institute of Technology Tiruchirappalli 2019-2023
B.Tech in Computer Science and Engineering CGPA: 8.84/10

- Studied algorithms and data structures, discrete mathematics, computer architecture, operating systems, computer networks, databases, theory of computation, and compilers.

Work Experience

Uber 07/2023 – Present
Software Engineer Bengaluru, India

- Primarily a backend engineer in the Trip Operations Platform team responsible for HITL(Human In The Loop) workflow orchestration and building a platform for knowledge workers.
- Working mostly with **Java**, **gRPC**, in-house dependency injection framework(based on **Spring Boot**), **Cadence**(A durable workflow orchestration engine), **Kafka** and **distributed databases** in my day to day work.
- Contributing to the frontend side of things as well using **Typescript**, **React** and **GraphQL**.

IIT Madras 07/2022 – 02/2024
Research Intern Remote

- Worked with Dr. KC Sivaramakrishnan and Dr. Kartik Nagar alongside a PhD student on a project that aimed to verify an OCaml style garbage collector with F*/Low*.
- Helped with the integration of the extracted verified code with the OCaml bytecode interpreter, ran real-world OCaml programs and ran benchmarks to analyze performance.
- Wrote a next-fit allocator in Rust which would then be hooked with the generated verified stop-the-world mark and sweep code. Analyzed performance using this before the bytecode interpreter integration.
- Helped with refactors in F*/Low* code.

Tarides 05/2023 – 07/23
Software Engineering Intern Remote

- Worked on developing Par_incr, a library for incremental computation with support for freshly introduced parallelism constructs in OCaml.

CDAC Bangalore 02/2023 – 05/23
Research Intern Remote

- Developed proof-of-concept GCC plugins for program transformation, tuning heuristics, etc. and evaluated performance.
- Developed tool to visualize GCC's AST and filter out unnecessary information, to help with our program transformation experiments.
- Suggested potential ARM specific optimizations for future exploration.

Uber 06/2022 – 07/2022
Software Engineering Intern Bengaluru, India

- Worked on improving reliability and observability of a service, involved setting up alerts and dashboards, integrating and collecting metrics, and error analysis.

Technical Projects

Par_incr

- A library for incremental computation with support for parallelism in **OCaml**. Other similar libraries lack parallelism constructs. The work is based on the paper Efficient Parallel Self-Adjusting Computation. [Slides]
- Wrote the library from scratch, thoroughly tested it,
- Identified performance bottlenecks through profiling and applied various optimization techniques in OCaml.
- Wrote benchmarks, compared the performance with other similar libraries, and achieved similar if not better performance on average.

Code Character

- A strategy-based programming game where you control troops in a turn-based game with the code you write in one of the multiple programming languages (C++, Python, Java) available in the game.
- Worked on the implementation of the simulator(**C++**)
- Worked on the game driver(**Rust**). Implemented the process orchestration, communication among the game processes, concurrent execution of games. Leveraged different system programming concepts, such as inter-process communication, unix processes, epoll, pipes, SPMC channels, etc in the implementation.

Enma

- A toy programming language written in **C++** and **OCaml**.
- The language has a uni-directional type checker and can be transpiled to readable C++ code or compiled to bytecode. The bytecode interpreter is written in OCaml.

BF JITs

- Implemented Just In Time compilers for Brainfuck language using Dynasm crate and Inkwell crate (provides LLVM bindings) in **Rust**.

Pragyan CTF

- Prepared challenges for Binary Exploitation/Reversing category, involving a small custom memory allocator, reversing SIMD instructions, and other common vulnerabilities.

Talks and Writings

Understanding Memory Management

- Slides, Video

Personal Blog

- What is a Fixed Point Combinator?
- Non Local Jumps with setjmp and longjmp

Skills

Programming: C, C++, Rust, OCaml, Java, Typescript, Python
Areas: Programming Languages, Systems Programming, Back-End Development, Databases

Languages

- **Nepali**: Native proficiency
- **Hindi**: Native proficiency
- **English**: Fluent (Professionally)