A Preliminary Performance Model for Optimizing Software Packet Processing Pipelines

Ankit Bhardwaj, Atul Shree, Bhargav Reddy V and Sorav Bansal

Department of Computer Science and Engineering, Indian Institute of Technology Delhi

Motivation and Objectives

- Newer functionalities are being developed over time with increasing popularity of SDN.
- Researchers are developing DSLs to make the task easier. However, ease of programming doesn’t guarantee application performance.

Goal: Develop a compiler that can automatically map a high-level specification to an underlying machine architecture in an optimal way.

Compilation Phases


Packet Processing Pipeline

* Exploit DMA bandwidth between NIC and Main Memory labeled ① in Figure
* Exploit Memory Level Parallelism between CPU and Memory labeled ② in Figure

Compiler Transformations related to Batching, Sub-Batching and Prefetching

Performance Model

- Let the service rate of CPU, CPU-Memory, I/O-Memory interface be c, m, d.
- **Throughput** of the application will be \( \min(c, m, d) \).
- The value of m will vary with change in b and will follow some \( f_{mem}(m, b) \).
- DMA interface is not linearly scalable and d increases based on some function \( f_{dma}(d, B) \).
- **Throughput** = \( \min(c, f_{mem}(m, b), f_{dma}(d, B)) \)

Experiments and Results

Discussion

- Use of Batching(b) to exploit NIC-Memory Parallelism.
- Use of Prefetching to exploit MLP at Memory-CPU interface in the pipeline.
- Use of sub-batch(b) to adjust the prefetch distance.
- Run the applications to find optimal value of b and B.

Future Work

- Perform more fine-grained experiments to gain better understanding of underlying hardware.
- Explore optimizations other than scheduling to the compiler to generate optimized DPDK application.

References