CPU Optimization Using A SST Simulator

Group 13: Alexia Arthur, Chenhan Dai, Zhengwei Wang, Haodong Zhao

Introduction

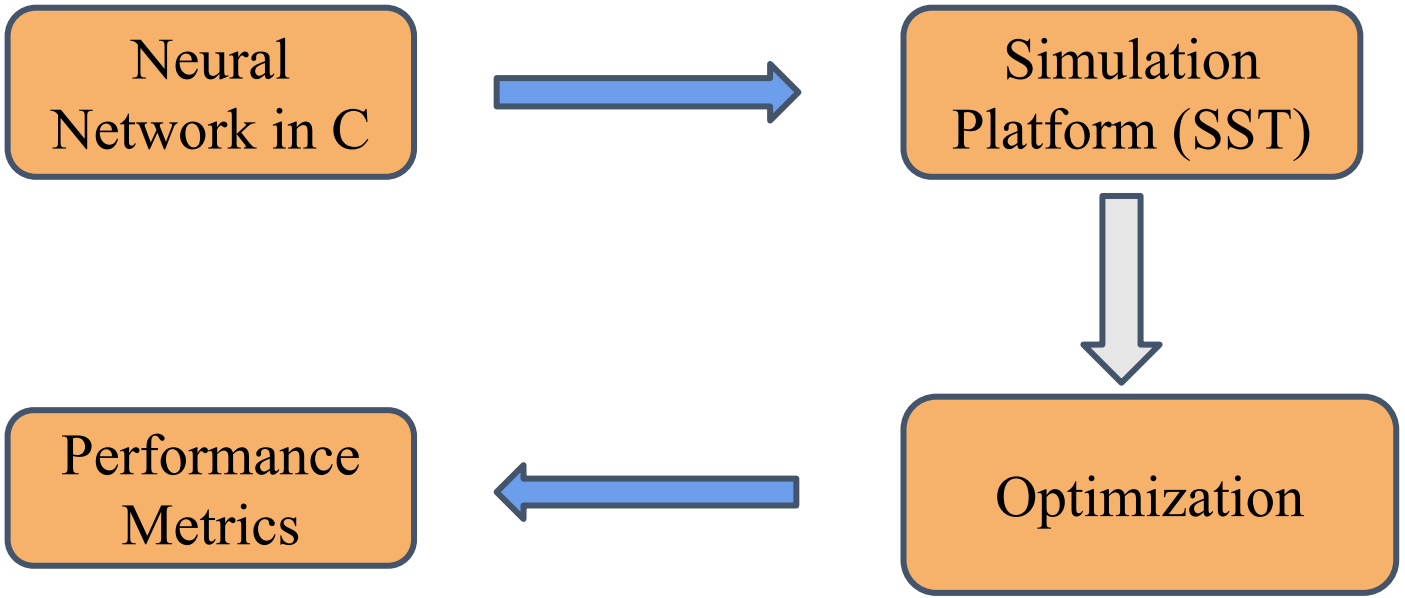
Why CPU Optimization Matters?

Objective: Structural Simulation Toolkit (SST) & CPU performance

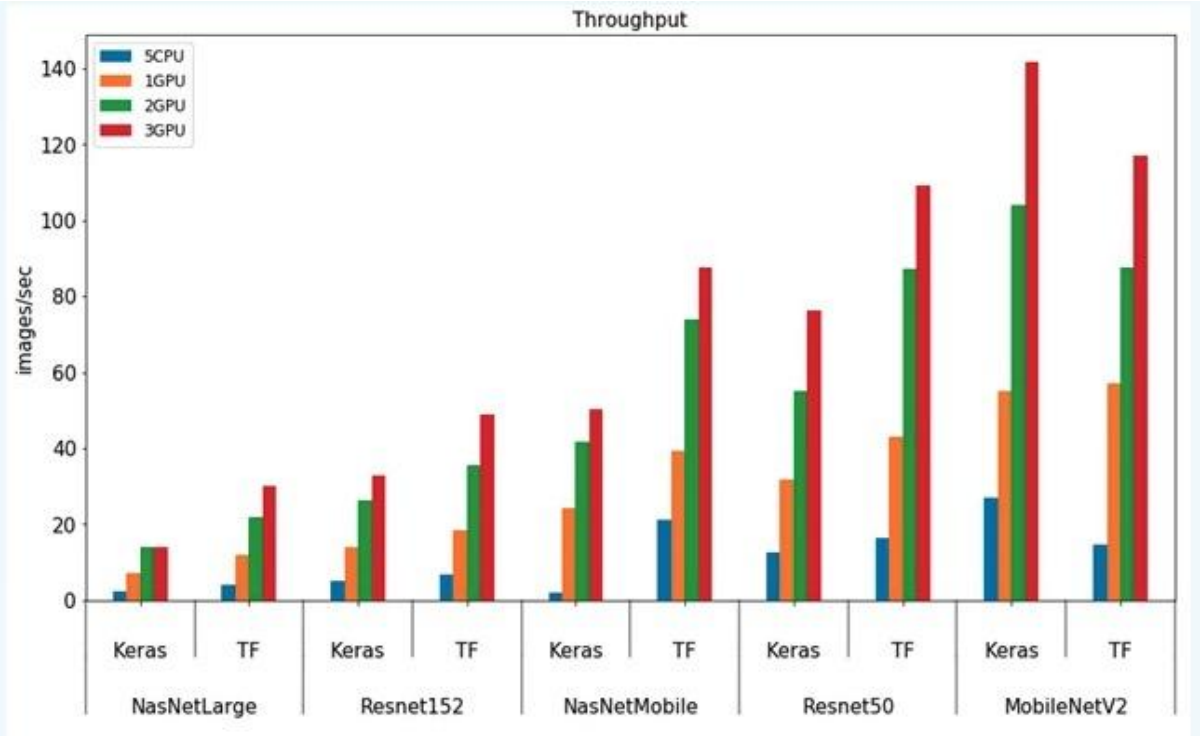
Approach: Vanadis CPU simulation element & parameter tuning, performance evaluation

Key Focus Area: Runtime Performance, Memory efficiency, & Branch Prediction accuracy

Introduction



Problem



Current focus on GPU & Limitations on CPU

What about CPU?

Improve CPU Performance

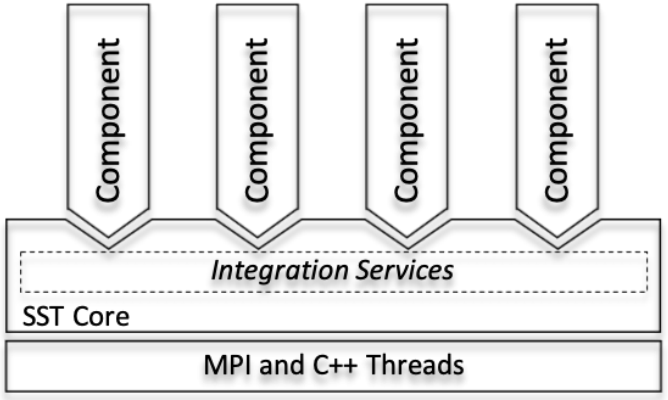
Progress

- Understanding how hardware simulation works
- Concatenating software and hardware to work together
  - Software → Cross-platform Compiler → Binary → memory hierarchy → processor
- Utilizing SST to build up a functional simulation platform
  - Cache → Link → CPU
- Optimizing the structural module of the processor and cache
- Evaluating performance of different sizing module

Internal Implementation

- Structural Simulation toolkits contains two parts:
  - SST Core - Platform(Link, Clock, Event, Parallelism)
  - SST Element - Implementation component(Processor, Memory)

- Vanadis Core:
  - Out of order Processor
  - Accept RISCV64/MIPS32 ISA binary
  - Basic branch predictor
  - Simulated decoder/reservation station/function unit
  - Basic LSQ with speculation
  - Physical register file
  - Parameterized ROB



- MemHierarchy:
  - Basic build-in memory component
  - Including cache, memory block, memory controller

Internal Implementation

Cross-platform compiler

- Riscv64-linux-gnu-gcc

- Alexnet.c
  - A simple convolution neural network program with five layer
  - Generator random number as input data
  - Customizable iteration

Memory/Cache Parameters

L1 Data Cache

- Associativity: 8 & 16
- Cache size (KB): 32 & 64
- Prefetcher: None/cassini.StridePrefetcher
- Prefetcher Reach: 4

L1 Instruction Cache

- Associativity: 8 & 16
- Cache size (KB): 32 & 64
- Prefetcher: None/cassini.NextBlockPrefetcher
- Prefetcher Reach: 2 & 4

Memory/Cache Parameters

L2 Cache

- Associativity: 8 & 16
- Cache size (MB): 1 & 2
- Prefetcher: cassini.StridePrefetcher
- Prefetcher reach: 4 & 8

Memory (xbar_bw): 1 & 2GB/s

Integer arithmetic units: 2 & 4

Floating point arithmetic units: 2 & 4

Memory/Cache Parameters

CSV File Name	l1dcache	l1icache	l1icache	l2cahe	CPU	Mem
stats.csv	none	no changes	no changes	no changes	no changes	no changes
stats1.csv	associativity 16, cache size 64kb	no changes	no changes	no changes	no changes	no changes
stats2.csv	associativity 16, cache size 64kb	prefetcher reach 2	prefetcher reach 2	no changes	no changes	no changes
stats3.csv	associativity 16, cache size 64kb	prefetcher reach 2	prefetcher reach 2	no changes	no changes	no changes
stats4.csv	associativity 16, cache size 64kb	prefetcher reach 4, cache size 64, associativity 16	prefetcher reach 4, cache size 64, associativity 16	cache size 2MB	no changes	no changes
stats5.csv	associativity 16, cache size 64kb	no changes	no changes	no changes	no changes	no changes
stats6.csv	associativity 16, cache size 32kb	no changes	no changes	associativity 16	no changes	no changes
stats7.csv	associativity 16, cache size 32kb	no changes	no changes	associativity 16, prefetcher reach 8, prefetcher cassani.StridePrefetcher	no changes	no changes
stats8.csv	associativity 16, cache size 64kb	cache size 64	cache size 64	associativity 16, prefectcher reach 8, prefecter cassani.StridePrefetcher	no changes	no changes
stats9.csv	associativity 16, cache size 64kb, prefetcher reach 4, prefetcher cassani.StridePrefetcher	cache size 64	cache size 64	associativity 16, prefetcher reach 8, prefetcher cassani.StridePrefetcher	no changes	no changes
stats10.csv	associativity 16, cache size 64kb, prefetcher reach 4, prefetcher cassani.StridePrefetcher	cache size 64	cache size 64	associativity 16, prefetcher reach 8, prefetcher cassani.StridePrefetcher, cache size 2MB	integer_arith_units 4, fp_arith_units 4	xbar_bw 2GB
stats11.csv	associativity 16, cache size 64kb, prefetcher reach 4, prefetcher cassani.StridePrefetcher	cache size 64	cache size 64	associativity 16, prefetcher reach 8, prefetcher cassani.StridePrefetcher, cache size 1MB	integer_arith_units 4, fp_arith_units 4	xbar_bw 2GB

Core Parameters

Threads: 1 - 6

CPUs: 1 - 6

Branch Instructions: 32 & 64

ROB: 16 - 2048

ALU: 1 - 32

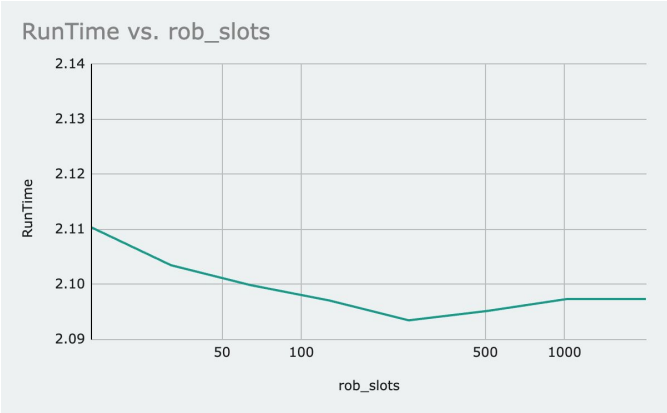
Lsq Load/Store Entries: 1 - 32

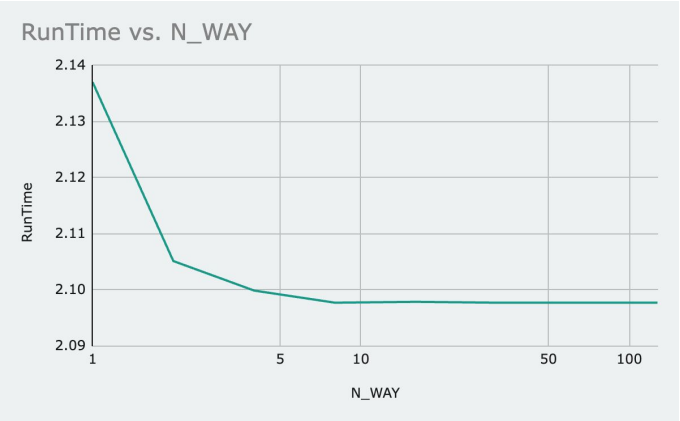
Prefetcher: 2 & 4

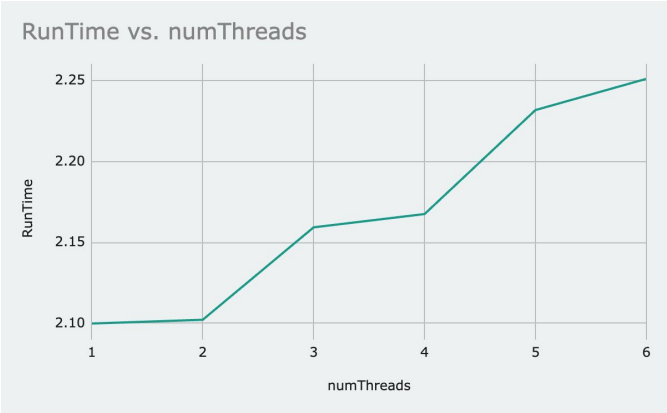
Evaluation Metrics

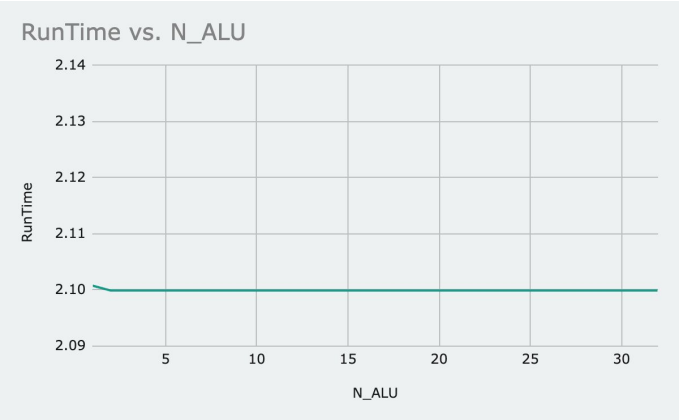
- L1 Data Cache
- L1 Instruction Cache
- L2 Cache
- Branch mispredictions
- Runtime

Results(Core)

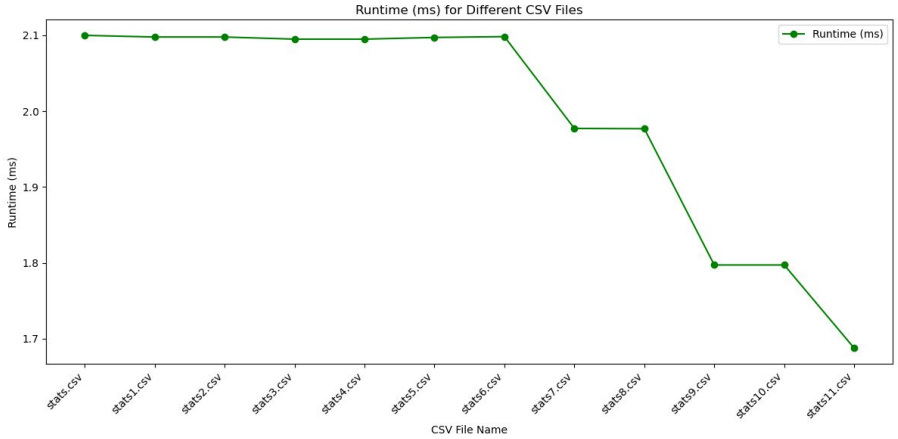


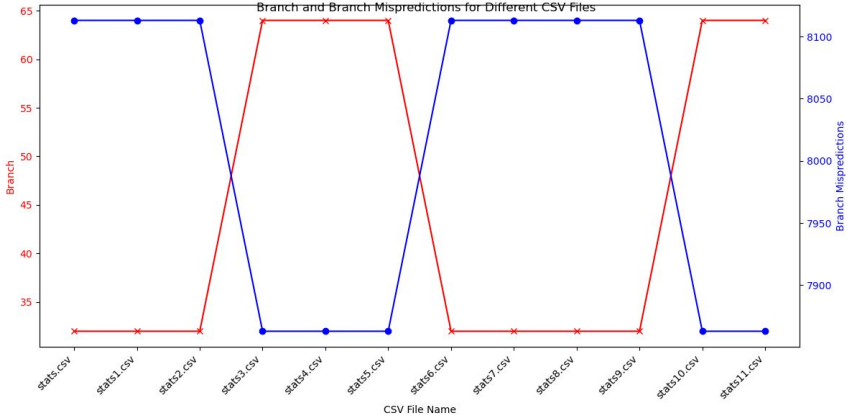


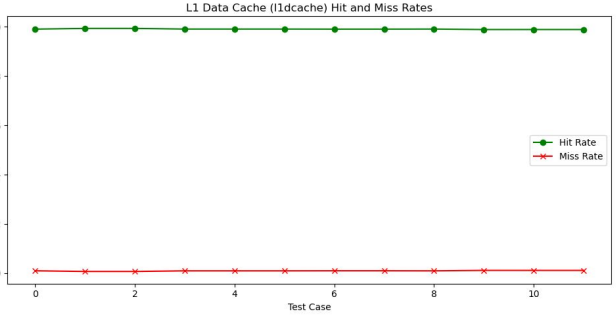


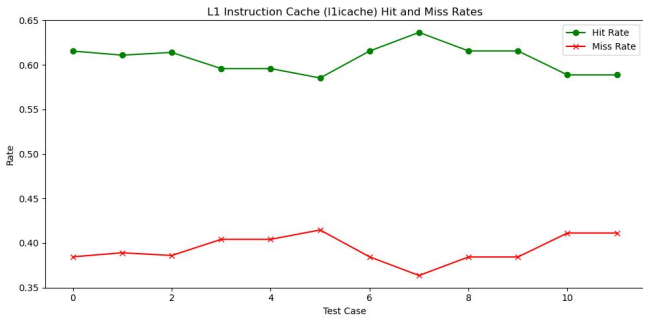


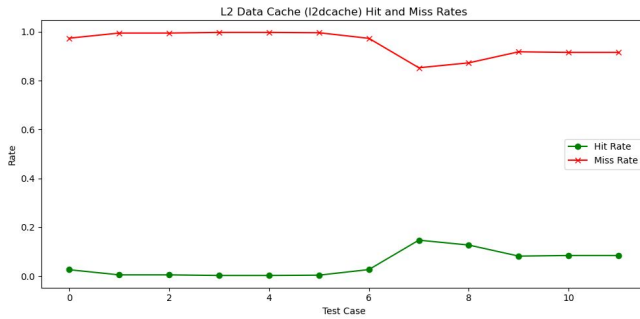
Results(Memory/Cache)











Results(Memory/Cache)

Best Parameter Combination → stats7.csv

- **L1 Data Cache** → associativity:16, cache size 32KB
- **L1 Instruction Cache** → cache size 32KB
- **L2 Cache** → cache size: 32KB, prefetcher reach 8
- **Branch Size** → 32
- **ROB** → 64

Metrics

- L1 Data Cache hit rate: **99.08%**
- L1 Instruction Cache hit rate: **63.63%**
- L2 Cache hit rate: **14.72%**
- Branch Mispredictions: **8113**
- Runtime: **1.97721**

Conclusion

- Superscaler
- Multiple Threading
- Number of FU
- ROB
- Cache Associativity
- Cache Prefetcher (CNN)
- Cache Size

Future Work

- SST Simulator Toolkit → balar GPU
- vanadis vs balar
  - Better Performance (runtime, cache hits/misses)
- Benchmarks
  - RNNs & Vectorization

Q&A

References

1. https://www.intel.com/content/www/us/en/developer/articles/technical/comparing-cpus-gpus-and-fpgas-for-oneapi.html
2. https://azure.microsoft.com/en-us/blog/gpus-vs-cpus-for-deployment-of-deep-learning-models/