Supplemental Material: Ithemal: Accurate, Portable and Fast Basic Block Throughput Estimation using Deep Neural Networks

Charith Mendis [1] Alex Renda [1] Saman Amarasinghe [1] Michael Carbin [1]

A. Canonicalization Scheme

We demarcate memory operands (consisting of a base address, and an optional offset and displacement) by surrounding them with `<M>` and `</M>` delimiter tokens.

The following is the full grammar for the token strings, as described in Section 3.1.

$$\begin{aligned} \langle \textit{block} \rangle &::= \langle \textit{instr} \rangle + \\ \langle \textit{instr} \rangle &::= \textit{opcode}\ \texttt{<S>}\ \langle \textit{opnd} \rangle *\ \texttt{<D>}\ \langle \textit{opnd} \rangle *\ \texttt{<E>} \\ \langle \textit{opnd} \rangle &::= \textit{register} \mid \texttt{<M>}\ \textit{register}+\ \texttt{</M>} \mid \texttt{CONST} \end{aligned}$$

B. Training Hyperparameters

Here we present the hyperparameters for the model as described in Section 3. All vectors, including the embedding width, hidden, and output states have width 256. We train our models using asynchronous SGD, with a batch size of 4, and 6 parallel trainers. The initial learning rate is 0.1, and after the first 2 epochs, it decreases by a geometric factor of 1.2 every epoch. We use the default PyTorch formulation for momentum (i.e. not Nesterov momentum) with $\beta = 0.9$. Each parallel trainer samples without replacement from the dataset until all training data is exhausted. If a trainer hits a `NaN` gradient, that trainer is halted for the remainder of the epoch, and the elements in that trainer's batch are dropped for the epoch. At the beginning of the next epoch, all trainers are restarted. Training halts once all trainers are halted and an epoch cannot be completed.

C. Heatmaps of Different Prediction Methods

Figure 1 shows all prediction heatmaps for Ithemal, llvm-mca and IACA under the Intel Ivy Bridge, Haswell and Skylake microarchitectures. Note that the latest IACA version does not support Ivy Bridge and hence its prediction heatmap is not available.

D. Prediction Errors for Throughput Ranges

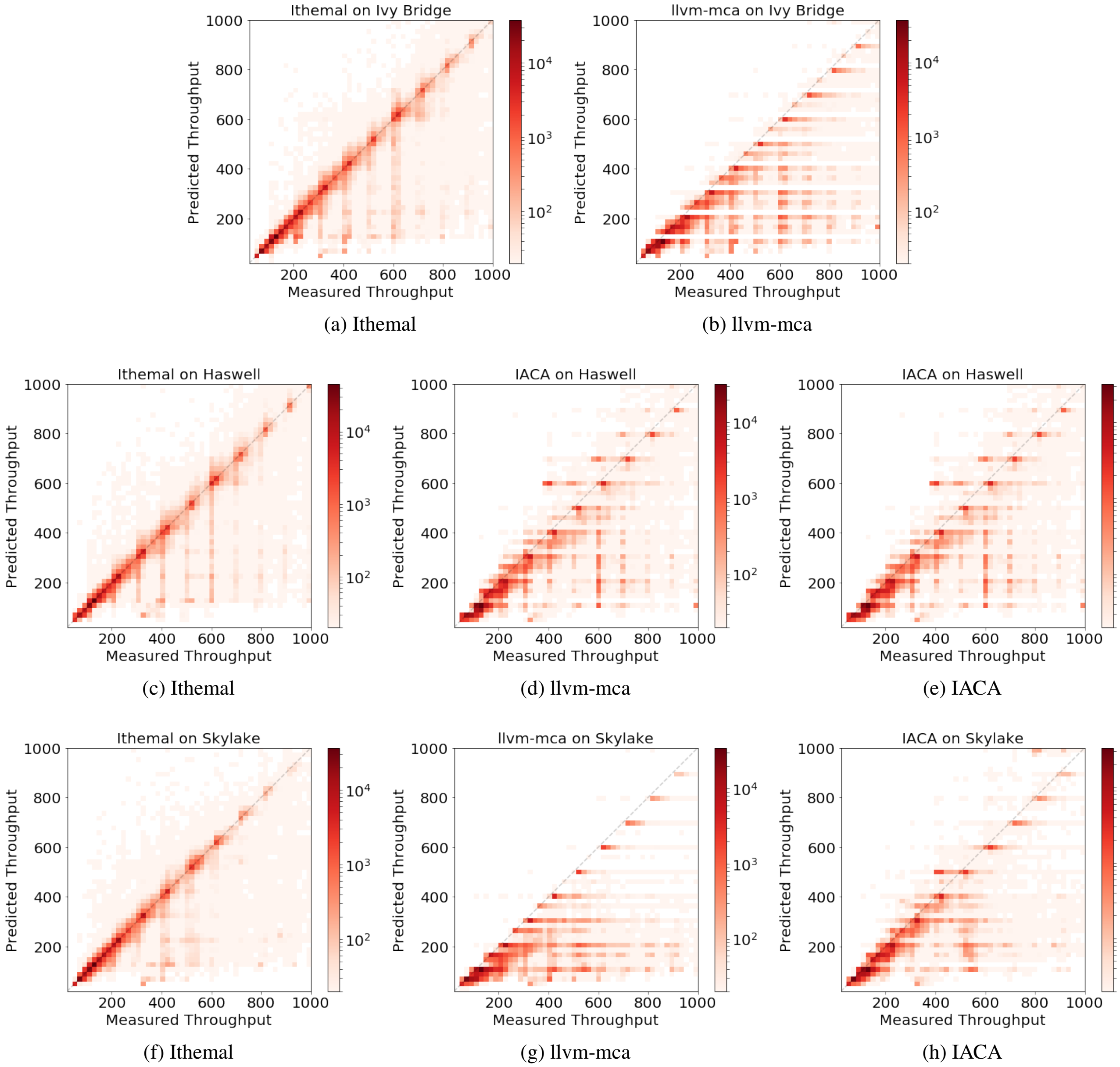
Figure 2 shows how the average error changes between various throughput ranges for each prediction method under different microarchitectures for basic blocks with throughput values under 1000 cycles. Throughput values are broken up in to bins of length and width 20 cycles on each axis. It also shows the throughput distribution of the basic blocks, and the average error across different measured throughput ranges. Ithemal consistently predicts throughput values with lower average errors compared to llvm-mca and IACA. Overall, Ithemal is more robust in its prediction across all throughput ranges compared to llvm-mca and IACA which show higher fluctuations.

E. Token RNN Architecture

The full architecture for the Token RNN as presented in Section 6 is shown in Figure 3.

[1] MIT CSAIL. Correspondence to: Charith Mendis <charithm@mit.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).



(a) Ithemal

(b) llvm-mca

(c) Ithemal

(d) llvm-mca

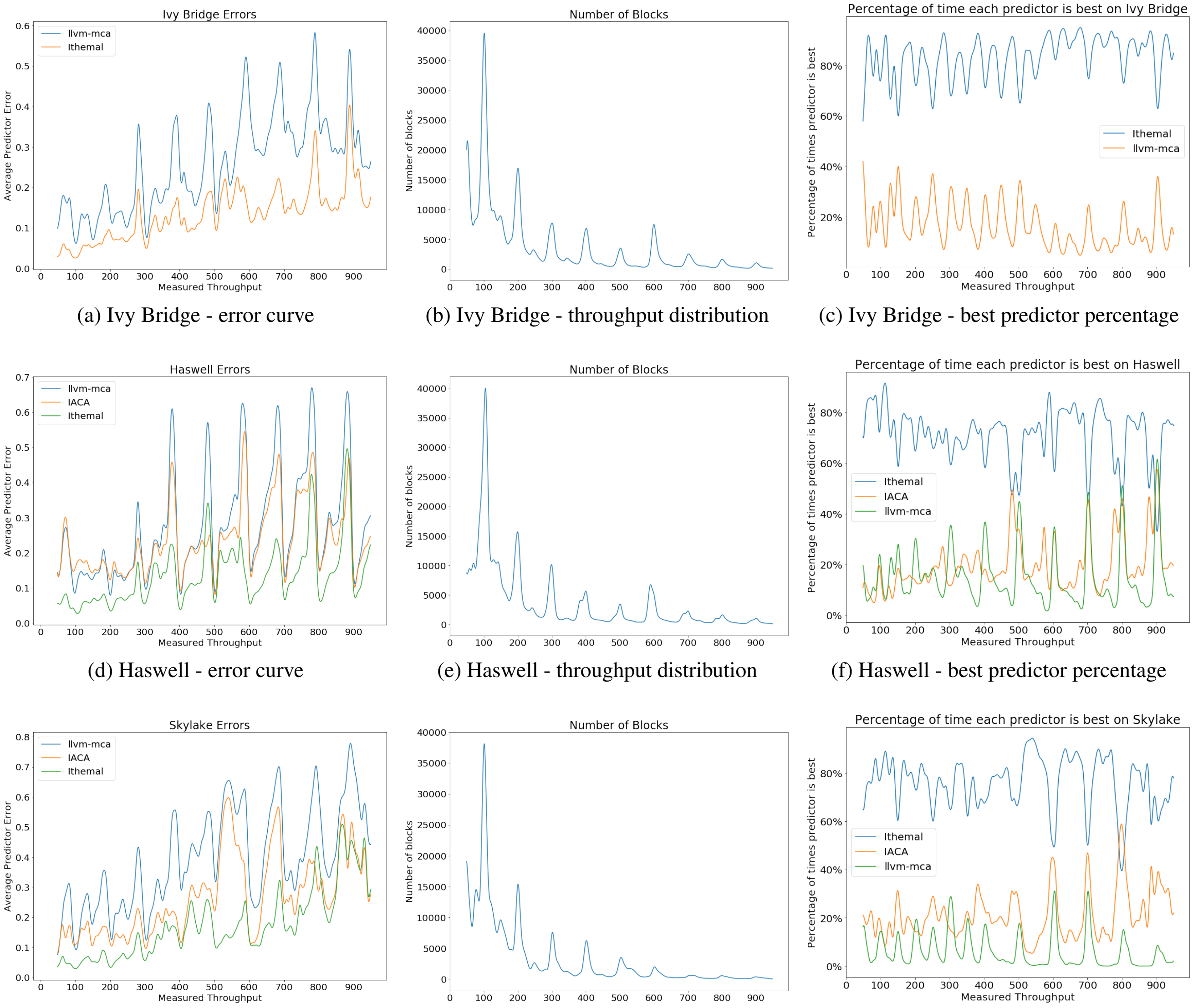
(e) IACA

(f) Ithemal

(g) llvm-mca

(h) IACA

Figure 1. Heatmaps for measured and predicted throughput values under different models for basic blocks with measured throughput values less than 1000 cycles for the Intel Ivy Bridge, Haswell and Skylake microarchitectures



(a) Ivy Bridge - error curve

(b) Ivy Bridge - throughput distribution

(c) Ivy Bridge - best predictor percentage

(d) Haswell - error curve

(e) Haswell - throughput distribution

(f) Haswell - best predictor percentage

(g) Skylake - error curve

(h) Skylake - throughput distribution

(i) Skylake - best predictor percentage

Figure 2. Average error curves and throughput distributions for different models for basic blocks with measured throughput values less than 1000 cycles under different microarchitectures for the test set

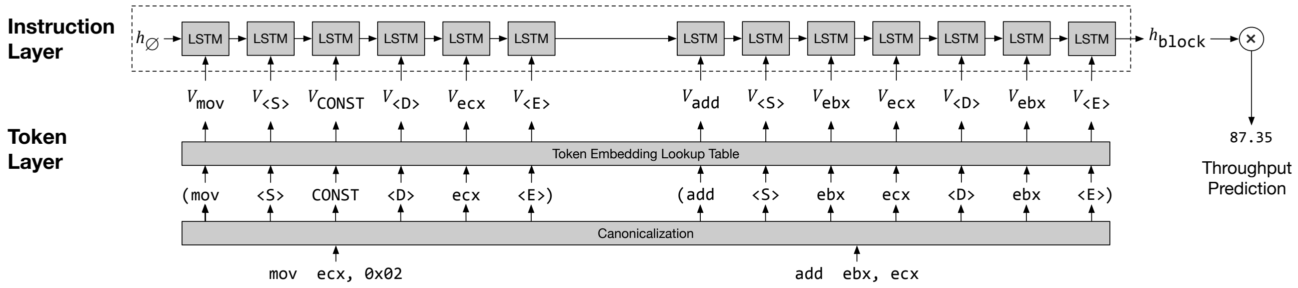


Figure 3. Token RNN Architecture