Llvm-cm: A static Cost Modeling Tool

Dayann D’Almeida

Host(s): Mircea Trofin, Kazu Hirata

Background

The MLGO project seeks to integrate ML techniques into LLVM, in order to replace existing optimization heuristics.

MLGO needs a ground-truth way to evaluate performance of ML-based optimizations.

llvm-cm provides a training signal for MLGO models, providing a ground-truth means of evaluating the performance of optimizations.

The Beginning: uiCA

Originally, llvm-cm was meant to serve as a port to uiCA.

What is uiCA?

- Machine basic block throughput predictor, similar to llvm-mca, IACA, or Ithemal*.
- Boasts results within ~1% of BHive benchmarks (for all microarchitectures between 2011-2021).
- Uses intelXED to get information about individual instructions, before matching them up with latency and throughput data collected on uops.info.

Why couldn’t it be ported?

- Requires a frequent amount of benchmark updates in order to remain accurate, obtained from uops.info.
- Other issues regarding licensing the code.

Ithemal

- Another throughput predictor—uses an LSTM (long short-term memory) approach.

- Creating a performance model by hand is an error-prone and lengthy process.
  - A throughput estimator capable of capturing microarchitecture-specific intricacies and handling corner cases without a tremendous amount of human investment is ideal.
  - Being able to get an estimation at steady state is invaluable for speed.
  - Ithemal uses training data and ISA specifications to generate its predictions.

GRANITE

Another machine learning model that estimates throughput of basic blocks on several microarchitectures.

Many of the same benefits of Ithemal.

Uses a graph neural network to process data dependencies across basic blocks.

Still seeks to resolve many of the issues with analytical models needing domain expertise to be properly generated.

llvm-cm

- Utilizes ML models such as GRANITE to perform latency estimation at the machine basic block and function level.
- Disassembles input files to obtained individual instruction information and processes machine basic block profile information obtained from profiles alongside the aforementioned ML models.
  - Produces a value that can be used as a performance metric for optimizations.
- Microarchitecture agnostic; can handle any architecture supported by the input model.

Questions?