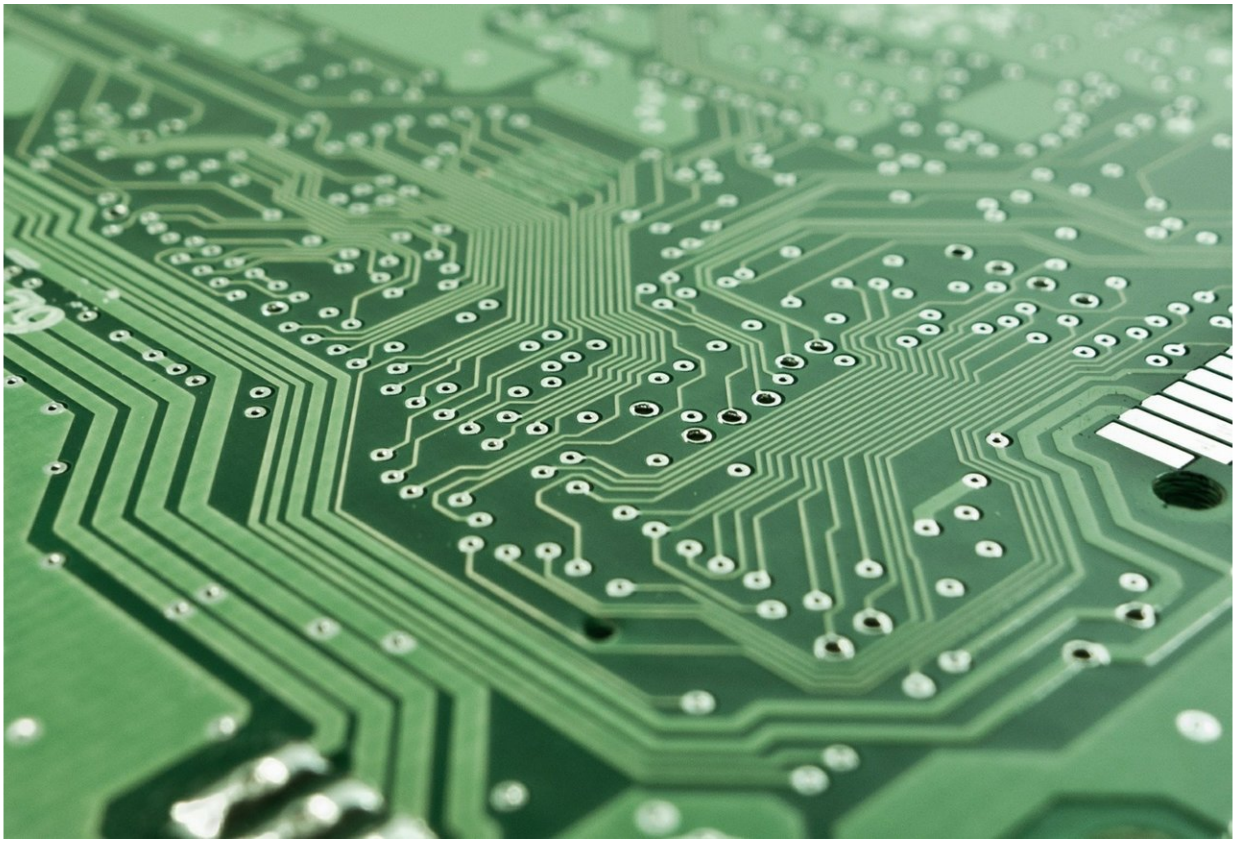
TechXplore

Tool predicts how fast code will run on a chip

January 6 2020, by Rob Matheson



Credit: CC0 Public Domain

MIT researchers have invented a machine-learning tool that predicts how fast computer chips will execute code from various applications.

To get code to run as fast as possible, developers and compilers—programs that translate programming language into machinereadable code—typically use performance models that run the code through a simulation of given chip architectures.

Compilers use that information to automatically optimize code, and developers use it to tackle performance bottlenecks on the microprocessors that will run it. But performance models for machine code are handwritten by a relatively small group of experts and are not properly validated. As a consequence, the simulated performance measurements often deviate from real-life results.

In series of conference papers, the researchers describe a novel machine-learning pipeline that automates this process, making it easier, faster, and more accurate. In a paper presented at the International Conference on Machine Learning in June, the researchers presented Ithemal, a neural-network model that trains on labeled data in the form of "basic blocks"—fundamental snippets of computing instructions—to automatically predict how long it takes a given chip to execute previously unseen basic blocks. Results suggest Ithemal performs far more accurately than traditional hand-tuned models.

Then, at the November IEEE International Symposium on Workload Characterization, the researchers presented a benchmark suite of basic blocks from a variety of domains, including machine learning, compilers, cryptography, and graphics that can be used to validate performance models. They pooled more than 300,000 of the profiled blocks into an open-source dataset called BHive. During their evaluations, Ithemal predicted how fast Intel chips would run code even better than a performance model built by Intel itself.

Ultimately, developers and compilers can use the tool to generate code that runs faster and more efficiently on an ever-growing number of diverse and "black box" chip designs. "Modern computer processors are opaque, horrendously complicated, and difficult to understand. It is also incredibly challenging to write computer code that executes as fast as possible for these processors," says co-author Michael Carbin, an assistant professor in the Department of Electrical Engineering and Computer Science (EECS) and a researcher in the Computer Science and Artificial Intelligence Laboratory (CSAIL). "This tool is a big step forward toward fully modeling the performance of these chips for improved efficiency."

Most recently, in a paper presented at the NeurIPS conference in December, the team proposed a new technique to automatically generate compiler optimizations. Specifically, they automatically generate an algorithm, called Vemal, that converts certain code into vectors, which can be used for parallel computing. Vemal outperforms hand-crafted vectorization algorithms used in the LLVM compiler—a popular compiler used in the industry.

Learning from data

Designing performance models by hand can be "a black art," Carbin says. Intel provides extensive documentation of more than 3,000 pages describing its chips' architectures. But there currently exists only a small group of experts who will build performance models that simulate the execution of code on those architectures.

"Intel's documents are neither error-free nor complete, and Intel will omit certain things, because it's proprietary," Mendis says. "However, when you use data, you don't need to know the documentation. If there's something hidden you can learn it directly from the data."

To do so, the researchers clocked the average number of cycles a given microprocessor takes to compute basic block instructions—basically, the sequence of boot-up, execute, and shut down—without human intervention. Automating the process enables rapid profiling of hundreds of thousands or millions of blocks.

Domain-specific architectures

In training, the Ithemal model analyzes millions of automatically profiled basic blocks to learn exactly how different chip architectures will execute computation. Importantly, Ithemal takes raw text as input and does not require manually adding features to the input data. In testing, Ithemal can be fed previously unseen basic blocks and a given chip, and will generate a single number indicating how fast the chip will execute that code.

The researchers found Ithemal cut error rates in accuracy—meaning the difference between the predicted speed versus real-world speed—by 50 percent over traditional hand-crafted models. Further, in their next paper, they showed that Ithemal's error rate was 10 percent, while the Intel performance-prediction model's error rate was 20 percent on a variety of basic blocks across multiple different domains.

The tool now makes it easier to quickly learn performance speeds for any new chip architectures, Mendis says. For instance, domain-specific architectures, such as Google's new Tensor Processing Unit used specifically for neural networks, are now being built but aren't widely understood. "If you want to train a model on some new architecture, you just collect more data from that architecture, run it through our profiler, use that information to train Ithemal, and now you have a model that predicts performance," Mendis says.

Next, the researchers are studying methods to make models interpretable. Much of machine learning is a black box, so it's not really clear why a particular model made its predictions. "Our model is saying it takes a processor, say, 10 cycles to execute a basic block. Now, we're trying to figure out why," Carbin says. "That's a fine level of granularity that would be amazing for these types of tools."

They also hope to use Ithemal to enhance the performance of Vemal even further and achieve better performance automatically.

More information: Ithemal: Accurate, Portable and Fast Basic Block Throughput Estimation using Deep Neural Networks: proceedings.mlr.press/v97/mendis19a/mendis19a.pdf

BHive: A Benchmark Suite and Measurement Framework for Validating x86-64 Basic Block Performance Models: groups.csail.mit.edu/commit/pa … emal-measurement.pdf

Compiler Auto-Vectorization with Imitation Learning: papers.nips.cc/paper/9604-comp … itation-learning.pdf

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: Tool predicts how fast code will run on a chip (2020, January 6) retrieved 9 April 2024 from https://techxplore.com/news/2020-01-tool-fast-code-chip.html