Using Machine Learning to Guide Architecture Simulation

Greg Hamerly, Erez Perelman, Jeremy Lau, Brad Calder, Timothy Sherwood
7th Journal of Machine Learning Research
2006

Bhaskar Gautam, 16CS04F

I. Keywords and Definitions

1) **Interval** An interval is a section of contiguous execution (a time slice) of a program's execution
2) **Phase** A set of intervals within program's execution that all have similar behaviour (CPI, Cache Miss, Branch Prediction)
3) **Basic Block Vector** It is the One Dimension Array whose each element points to the distint basic block of a program
4) **Frequency Vector** Basic Block Vector are one type of Frequency Vector such that it has ability to represent basic block, branch edge or any other type of program related structure which provides a representative summary of a program's beahavior for each interval of execution
5) **Unsupervised Learning** Learning in the absence of a teacher
6) **Basic block** It is a contiguous block of instructions that have a single entry point and single exit point
7) **Simulation Point** SimPoint intelligently chooses a set of samples called Simulation Points to perform targeted program analysis.
8) **Warmup** When using simulation points, an approach is needed for warming up the architecture state (e.g., the caches, TLBs, and branch predictor)..
9) **Profile Granularity** It is measure of the how many of the instructions in a computer program can be executed simultaneously.
10) **No-Warmup** If a large enough interval size is used (e.g., 100 million instructions), no warmup may be necessary for many programs.

II. Paper Summary

A. Abstract

An essential step in designing a new computer architecture is the careful examination of different design options. At present researchers use very detailed simulators to estimate processor performance, which models every cycle of an executing program (simulating every cycle of a real program) can take weeks or months. To address this problem authors present a tool SimPoint that uses k-means clustering algorithms to automatically find repetitive patterns in a programs execution. By simulating one representative of each repetitive behavior pattern, simulation time can be reduced to minutes instead of weeks for standard benchmark programs, with very little cost in terms of accuracy.

B. Methedology

Inorder to estimate the cycle level behaviour of a program, authors proposed a tool called SimPoint. SimPoint can able to reduce the simulation time to a few minutes instead of computing over week even months. This task of predicting cycle level behaviour is complicated because different programs, and even different parts of the same program, may have distinct behaviors also interacts with the hardware in different ways. But author's determines that there exist a repetitive patterns in a program's execution such that programs exhibit a few unique behaviors which are interleaved with one another through time. By using unsupervised k-means clustering algorithm, SimPoint able to find the repetitive patterns in a program's execution.

The annotation of SimPoint preceeds by determining the phase behaviour in an architecture independent fashion using Basic Block Vector Similarity Matrix then propose a automated way of extracting the phase information from program using SimPoint Clustering algorithm and passing frequency vector of a program as a parameter to the algorithm. Summarization of algorithm discussed in following subsection.

C. Algorithm

- Profile the program by dividing the program's execution into contiguous intervals of fixed length (say 1M, 10M, 100M instruction) and for each interval collects the frequency vector tracking the program's use of some program structure. Finally, Each frequency vector is normalized so that the sum of all the elements equals to 1
- Reduce the dimensionality of the frequency vector data to a much smaller number of dimensions using ”Random Linear Projection”
- Run the k-means clustering algorithm on the projected data with values of k in the range from 1 to K, where K is a user-prescribed maximum number of phases that can be detected.
- To compare and evaluate the different clusters formed for different k, we use the Bayesian Information Criterion (BIC)
- The final step is to choose the clustering with a small k such that its BIC score is nearly as good as the best observed. The chosen clustering is the final grouping of intervals into phases.

D. Selection of Simulation Point from Phase Classification

SimPoint chooses one representative interval that will be simulated in detail to represent the behavior of the whole phase. Therefore, by simulating only one representative interval per phase, we can extrapolate and capture the behavior of the entire program.

To choose a representative for a cluster, SimPoint picks the interval that is closest (Euclidean distance) to the clusters k-means center. The selected interval is called a simulation point for that phase and we can then perform detailed simulation on the set of simulation points.

As part of its output SimPoint also gives a weight for each simulation point. Each weight is a fraction (the total number of instructions represented by the intervals in the cluster from which the simulation point was taken divided by the number of instructions in the program). With the weights and the detailed simulation results of each simulation point, we can compute a weighted average for the architecture metric of interest (CPI, cache miss rate, etc.) for the entire programs execution. Selecting and simulating only a handful of intelligently picked sections of the full program provides an accurate picture of the complete execution of a program, which gives a highly accurate estimate of performance.

E. Results

For the industry-standard SPEC programs, SimPoint has less than a 6% error rate (2% on average) for the results in this paper, and is 1,500 times faster on average than performing simulation for the complete programs execution. Because of this time savings and accuracy, our approach is currently used by architecture researchers and industry companies (e.g. Patil et al. (2004) at Intel) to guide their architecture design exploration.

III. Ideas

- Apply different Machine Learning algorithms to improve the accuracy.
- Try to reduce computation cost using Cloud Computing Architecture.

References

Greg et. al, ”Using Machine Learning to Guide Architecture Simulation”