Efficient MART-Aided Modeling for Microarchitecture Design Space Exploration and Performance Prediction

Bin Li
Department of Experimental Statistics
Louisiana State University
Baton Rouge, LA 70803
bli@lsu.edu

Lu Peng
Electrical & Computer Engineering
Louisiana State University
Baton Rouge, LA 70803
lpeng@lsu.edu

Balachandran Ramadass
Electrical & Computer Engineering
Louisiana State University
Baton Rouge, LA 70803
bramad2@lsu.edu

ABSTRACT

Computer architects usually evaluate new designs by cycle-accurate processor simulation. This approach provides detailed insight into processor performance, power consumption and complexity. However, only configurations in a subspace can be simulated in practice due to long simulation time and limited resource, leading to suboptimal conclusions which might not be applied in a larger design space. In this paper, we propose an automated performance prediction approach which employs state-of-the-art techniques from experiment design, machine learning and data mining. Our method not only produces highly accurate estimations for unsampled points in the design space, but also provides interpretation tools that help investigators to understand performance bottlenecks. According to our experiments, by sampling only 0.02% of the full design space with about 15 millions points, the median percentage errors, based on 5000 independent test points, range from 0.32% to 3.12% in 12 benchmarks. Even for the worst-case performance, the percentage errors are within 7% for 10 out of 12 benchmarks. In addition, the proposed model can also help architects to find important design parameters and performance bottlenecks.

Categories and Subject Descriptors

C.4 [PERFORMANCE OF SYSTEMS] - Measurement techniques; Modeling techniques.

General Terms

Measurement, Performance, Design, Experimentation.

Keywords

Design Space Exploration; Performance Prediction; MART-Aided Models.

1. INTRODUCTION

Computer architects usually evaluate new designs by employing cycle-accurate processor simulators which provide detailed insight into processor performance, power consumption and complexity. A huge design space is composed by the product of the choices of many microarchitectural design parameters such processor frequency, issue width, cache size/latency, branch predictor settings, etc. To achieve an optimal processor design, a wide configuration spectrum of the design space has to be tested before making a final decision. However, only configurations in a subspace can be simulated in practice due to long simulation time and limited resource, leading to suboptimal conclusions which might not be applied in the whole design space. In addition, more parameters brought by chip-multiprocessors make this problem more urgent [2][3].

In this paper, we propose to use the state-of-the-art tree-based predictive modeling method combining with advanced sampling techniques from statistics and machine learning to explore the microarchitectural design space and predict the processor performance. This bridges the gap between simulation requirements and simulation time/resource costs. The proposed method includes the following four components: (1) the *maximin space-filling sampling* method that selects the initial design representatives from among a large amount of design alternatives; (2) the state-of-the-art predictive modeling method *Multiple Additive Regression Trees* (MART) [1] that builds a nonparametric model with exceptional accuracy while remaining remarkably robust; (3) an *active learning* method that selects the most informative design points needed to improve the prediction accuracy sequentially; (4) interpretation tools for MART-fitted models that shows the importance and partial dependence of design parameters.

According to our experiments on 12 SPEC benchmarks, by sampling 3000 points drawn from a microarchitecture design space with nearly 15 million configurations (sampled up to 0.02 percent of the full design space) for each program, we can summarize the following results:

1. **Performance Prediction**: Application-specific models predict performance, based on 5000 independent sampled design points, with median percentage error ranges from 0.32% to 3.12% (average percentage error ranges from 0.41% to 4.18%).
2. **Worst-Case Performance**: the worst percentage errors are within 7% for 10 out of 12 benchmarks. The largest worst case percentage error of our proposed method is 22.55% for art. This is still much better than that of linear regression model which has a worst case percentage error 87.69%.
3. **Model Interpretation**: The proposed model shows that several design factors are more important than others: Fetch/Issue/Commit width and the number of ALU units, L2 cache size and branch predictor types and sizes. It also finds a performance bottleneck resulting from a relative small number of LSQ entries.

SIGMETRICS' 08, June 2-6, 2008, Annapolis, Maryland, USA.
ACM 978-1-60558-005-0/08/06.

2. METHODOLOGY

In experiment design, the distance-based space-filling sampling methods are popular, especially, when we believe that interesting features of the true model are just as likely to be in one part of the experimental region as another. Among them, the maximin distance design is commonly used. However, since some of the architectural design parameters are nominal (no intrinsic ordering structure) and the others are discrete (having a small number of values), we use the following defined distance before applying the maximin distance criterion. Let wt_j be the weight for the jth design parameter.

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{p} \left[wt_j \times I\left(x_{1j} \neq x_{2j}\right) \right]$$

where $wt_j = \log_2(\text{numberof levelsfor } j^{th} \text{ designparameter})$ and $I(A)$ is an indicator function, equal to one when *A* holds, otherwise zero. Note that the weight for each design parameter is equal to its information entropy with uniform probability for each of its possible values.

In our method, a small number of initial design points are selected based on the Maximin distance criterion (maximize the shortest distance among selected points). The processor performance is measured via benchmark simulations on the selected design points. Then, MART is applied 20 times on the sampled points with random perturbation. The reason to use MART, an ensemble of trees, is the following: (1) trees are inherently nonparametric and can handle mixed-type of input variables naturally, i.e. no assumptions are made regarding the underlying distribution of values of the input variables, as well as categorical predictors with either ordinal or non-ordinal structure; (2) trees are adept at capturing non-additive behavior, i.e. complex interactions among predictors are routinely and automatically handled with relatively little input required from the analyst; (3) MART improves the prediction performance from a single tree by using an ensemble of trees.

Adaptive sampling, also known as active learning in machine learning literature, involves sequential sampling schemes that use information gleaned from previous observations to guide the sampling process. Studies have shown that adaptively selecting samples in order to learn a target function can outperform conventional sampling schemes. In our method, for each of the MART-fitted model, it predicts the rest of the points in the design space. Sort these points according to the coefficient of variance (CoV, the ratio of standard deviation to mean) for the model prediction. Selected the points with maximal CoV (under minimal pairwise-distance constrain) and measure their performance. Repeat the underlined adaptive sampling process above until some stopping criterion is met (e.g. time limit and user pre-specified number of iterations).

3. EXPERIMENTAL RESULTS

We modified sim-outorder, the out-of-order pipelined simulator in SimpleScalar, to be an eight-stage Alpha-21264 like pipeline. Twelve (eight integer and four floating point) CPU and memory intensive programs from SPEC2000 were selected. To show the typical behavior, we skipped a number of instructions for each SPEC program based on a previous work [4]. Then we collected the number of execution cycles for the next 100 million instructions. The total design space for each workload is about 15 million configurations composed of the cross product of 13 design parameter choices. For each workload, 500 initial design points are sampled based on the maximin distance criterion described in Section 2. Then another 500 points are sampled according to the adaptive sampling scheme described in Section 2. Repeat the sampling process until 3000 design points were sampled for each benchmark. Notice that for 3000 points, we only explored approximately 0.02% of the total 15 million points in the design space. An independent test set which consists of 5000 points is used to evaluate the prediction performance of fitted models. The following table shows the average percentage errors (PE) on twelve benchmarks with roughly 0.0067%, 0.0133% and 0.02% space sampled. The mean PE ranges from 0.41% to 4.18% for the 12 benchmarks. For the worst-case performance, the percentage errors are within 7% for 10 out of 12 benchmarks. The results indicate that our model achieves highly accurate prediction and robustness under the worst-case situation.

Bench mark	0.0067%		0.0133%		0.020%	
	Mean (%)	Max (%)	Mean (%)	Max (%)	Mean (%)	Max (%)
art	6.299	42.79	4.633	24.95	4.179	22.55
bzip	0.734	4.496	0.460	3.328	0.406	3.165
crafty	1.623	13.10	1.018	7.171	0.865	5.529
equake	2.654	18.69	2.260	15.77	2.130	15.04
fma3d	0.912	5.426	0.704	3.362	0.625	2.964
gcc	0.740	4.044	0.491	3.024	0.426	2.256
mcf	0.668	4.988	0.501	4.217	0.456	4.236
parser	0.831	4.905	0.515	3.649	0.420	2.305
swim	1.442	9.588	0.905	5.937	0.659	4.627
twolf	1.826	10.35	1.380	7.533	1.227	6.315
vortex	1.359	13.07	0.925	7.112	0.800	6.885
vpr	0.983	6.929	0.616	4.533	0.529	4.323

Table 1: Summary of performance prediction error with specified error and percentage of full space sampled. Max PE is the maximum percentage error among the 5000 test points.

4. REFERENCES

[1] J. Friedman, “Greedy function approximation: a gradient boosting machine,” *The Annals of Statistics*, 29: 1189-1232, 2001.

[2] E. İpek, S.A. McKee, B.R. Supinski, “M. Schulz and R. Caru-ana. Efficiently exploring architectural design spaces via predictive modeling,” *ASPLOS XII*, Oct. 2006.

[3] B. Lee and D. Brooks, “Accurate and efficient regression modeling for microarchitectural performance and power prediction,” *ASPLOS XII*, Oct. 2006.

[4] S. Sair, M. Charney, "Memory Behavior of the SPEC2000 Benchmark Suit," Tech. Report, IBM Corp. Oct. 2000.