Estimation of the Performance of Computer Hardware using Linear Regression Model of Machine Learning

Pulkit Batra, Praneetha Yekkaluru, Ritik Nageshwar, Gopal Pandey, Nidhi Lal

Dept. of Computer Science and Engineering
IIIT Nagpur, India

pulkitbatra34@gmail.com, praneethayekkaluru20@gmail.com, ritiknageshwar17@gmail.com, gopalk.pandey26@gmail.com, nidhi.2592@gmail.com

Abstract: **It is not feasible to compare the older CPUs according to today's benchmarks. Therefore, in order to compare Estimated Relative Performance, the factors taken into consideration for such a comparison would be common characteristics of all the processors of that era such as Machine Cycle Time (MYCT), Cache Memory (CACH) etc. In this paper we present our methodology that relies on extracting the performance levels of a small fraction of the processors and using this we try to develop a linear regression models to predict the performance of any processor .As a result, the configuration resulting in high performance is discovered without any expense and time consumption.**

***Keywords*: Estimated Relative Performance (E.R.P), Cache Memory (CACH), Machine Cycle Time (MYCT)**

I. INTRODUCTION

Performance depends not only on the architecture of the program or on the underlying hardware, but also on values of numerous configuration options. For example, it can be the size of the internal cache for input-output (I/O) operations or the number of working threads etc. [4-8] In order to achieve maximum performance, configuration options must be carefully tuned by the processor. Currently they rely on their professional experience and on basic monitoring tools for analysing various program configurations [5-8]. For example, upgrading a CPU is necessary if the CPU load will exceed a given threshold, or that the number of working threads might be increased if the CPU has multiple cores. But it is not always accurate and also requires a thorough understanding of both hardware and software of the system. Thus, system administrators perform multiple experiments to select an optimal configuration of the processor. [19-24]

The configuration resulting in the highest performance is selected for practical deployment. Unfortunately, such experiments are time-consuming and put a high workload on the processor. In addition to the actual running time of the experiments, we have to set up trial runs, collect data, and analyse results. Altogether, these experiments result in a high cost of trial runs. Moreover, trial runs might require a special hardware or software not available at the present moment, which will effectively render this approach useless. The given model will predict performance of the program for the given set of parameters, including incoming request flow, configuration options, and characteristics of the hardware. A number of different configurations will be tested; the configuration resulting in the highest performance will be selected.[15] The model can find a good configuration automatically by the information about its components.

In this paper we present our methodology to calculate the performance of computer hardware. We rely on a dynamic analysis to collect data for finding the performance. We employ a modular approach from a number of small components; each component performing a simple
operation such as computations, I/O activities, flow control etc.

In summary, in this work we develop:

1) Linear regression to estimate the performance of a processor just by using the information about its components.

2) Show that the performance of a system can be accurately predicted by using information from past systems.

3) Show that the performance of a processor can be accurately predicted by using a small fraction of the overall set of possible simulations.

II. RELATED WORK

There are many factors to be taken into consideration for predicting the Estimated Relative Performance of a given processor. There has been previous usage of the same data set in order to predict the performance of the processors by factoring their technical specifications and previously known performance data. This dataset was created by Phillip Ein Dor and Jacob Feldmesser in October, 1987.Ein Dor is the Faculty of Management at Tel Aviv University in Israel. In the past few people have used this dataset for prediction. The first of them were the creators of the dataset, Ein Dor and Feldmesser(CACM 4/87, pp 308-317) [16] whose aim was to prediction relative CPU Performance. They used the method of linear regression and recorded 34% average deviation from actual values.

The next usage was by Kibler,D. & Aha,D. in 1988 in the Proceedings of the CSCSI (Canadian AI Conference) [17]. It was done to show Instance-Based Prediction of Real-Valued Attributes (CPU relative performance). They got similar results; no transformations required. The last usage was by Mr. Anand Kumar with the aim to predict Estimated Relative Performance with maximum accuracy. This time Coefficient of determination R^2 of the prediction of 0.958 was achieved. The Mean squared error achieved was 0.39 and, the Test variance score achieved was 0.93. The estimated relative performance values were estimated by the authors using a linear regression method. See their article (pp 308-313) for more details on how the relative performance values were set.

III. PROPOSED WORK

We are using multivariate linear regression which considers multiple factors (numerical variables) for prediction of results. Our main aim is to increase the coefficient of determination R^2 of the prediction, and decrease the mean square error.

Linear regression:

Multivariate linear regression model is one where there is one continuous dependent variable (E.R. P in our case) which is dependent on multiple factors and predictor. These factors are independent variables in a formula to calculate the dependent variable.

$$Y_i = \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+...+\beta_p x_{ip}+\epsilon$$

where, for i=n observations:

y_i = dependent variable (Estimated Relative Performance)

x_i = explanatory variables

β_0 = y-intercept (constant term)

β_p = slope coefficients for each explanatory variable

ϵ = the model's error term
(also known as the residuals)

The dataset contains 209 instances of different processors with their specifications and Estimated relative performance.

Attrib utes of each processor in dataset are as follows:

Vendor names (adviser, 2ixdor,2ixdor, basf, bti, burroughs, c.r.d, cambex, cdc, dec, dg, formation, four-phase, gould, 2ixdorf2l, hp, ibm, ipl, magnuson, microdata, nas, ncr, 2ixdorf, perkin-elmer, prime, siemens, sperry, sratus, wang)

Model Name: many unique symbols

MYCT: machine cycle time in nanoseconds (integer)

MMIN: minimum main memory in kilobytes (integer)

MMAX: maximum main memory in kilobytes (integer)

CACH: cache memory in kilobytes (integer)

CHMIN: minimum channels in units (integer)

CHMAX: maximum channels in units (integer)

PRP: published relative performance (integer)

ERP: estimated relative performance from the original article (integer)

As linear regression only works with numerical variables, we are going to take only the integer attributes into consideration for the training and testing. Packages used: numpy, pandas, sklearn and matplotlib. We import the data and normalize it for training and testing. Then we train it with different random states for maximizing the coefficient of R^2 score. We used a test split of 0.5 for training.

After the training we received a maximum coefficient of determination R^2 of 0.984502…, mean squared error of 0.05 and a test variance score of 0.81.

IV. RESULT

The Previous maximum coefficient of determination R^2 was 0.965929…, previous mean squared error was 0.37 and the previous test variance score was 0.92. After testing and training we get the following results corresponding to random state 172. The proposed maximum coefficient of determination R^2 is 0.984502…, proposed mean squared error is 0.05 and the proposed test variance score is 0.8. The improvement in maximum coefficient of determination R^2 is 1.922812% and the improvement in mean squared error is 86.486486%.

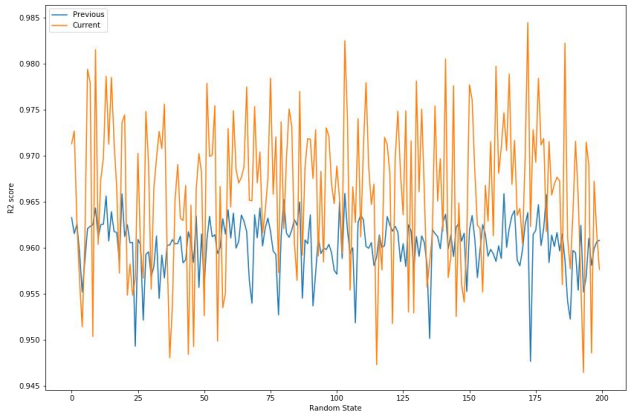

Fig 1. Random state vs Coefficient of R^2 Score

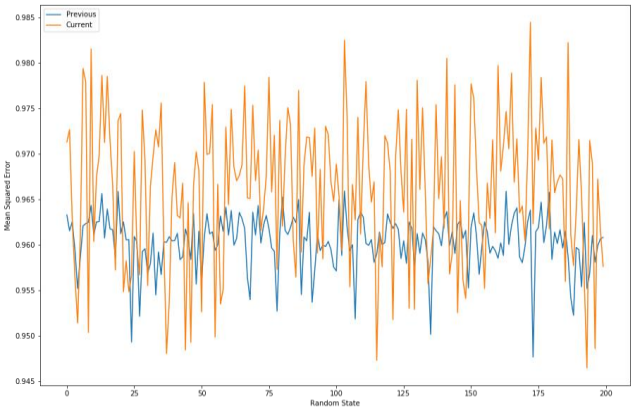

Fig 2. Random State vs Mean Squared Error

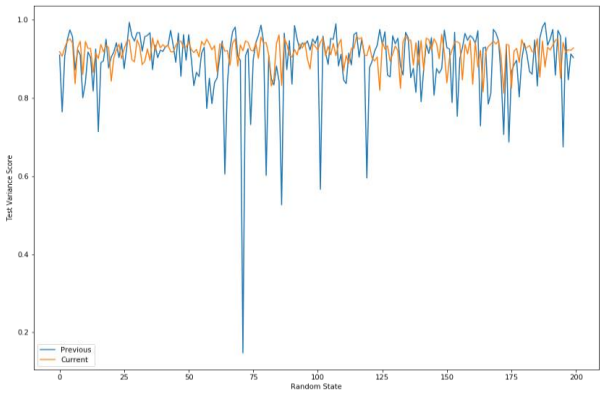

Fig 3. Random State vs Test Variance Score

V. CONCLUSION

In this paper, we presented our methodology on extracting performance of the processor and using this information, we tried to develop a linear regression model to predict the performance of any processor. We developed an extensive end to end simulation framework, which includes tools for data collection and also on metrics for measuring the accuracy of our performance. Although, our experiment has shown good results, an extensive experimentation is required to verify all the aspects of our work. Besides conducting additional experiments, we are actively working on improving our methodology.

Main directions of our work are:

Increasing the coefficient of R^2 of prediction.

Decreasing the mean square error.

Our models are also successful for estimating future system performance when limited data is available for already built system solely on the basis of the specifications of the processors.

References

[1] https://cs.brown.edu/research/pubs/theses/ masters/ 2011/tarvo.pdf

[2] http://cucis.ece.northwestern.edu/projects/ DMS /publications/OziMem08B.pdf

[3] Standard Performance Evaluation Corporation (SPEC). http://www.spec.org.

[4] Don Heller. Rabbit: A performance counters library for Intel/AMD processors and Linux.http://www.scl.ameslab.gov/projects/ rabbit/

[5] B. Aslam, M. Akhlaq, S. A. Khan, IEEE 802.11 wireless network simulator using Verilog, 11th WSEAS International Conference on

Communications,Greece,2007.
http://www.omnetpp.org

[6] E. Lee, R. Katz, An analytic performance model of disk arrays, Conference on Measurement and modeling of computer systems, p.98-109, Santa Clara, CA, 1993.

[7] M. Wang, K. Au, A. Ailamaki, A. Brockwell, C. Faloutsos, G. Ganger, Storage device performance prediction with CART models, 12th International Symposium on Modeling Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS-2004), Volendam, The Netherlands, 2004.

[8] http://sourceforge.net/projects/tinyhttpd/

[9] B. Lee et al, Methods of inference and learning for performance modeling of parallel applications, 12th ACM SIGPLAN symposium on Principles and practice of parallel programming, San Jose, CA, 2007.

[10] C. Gupta, A. Mehta, U. Dayal, PQR: Predicting Query Execution Times for Autonomous Workload Management, International Conference on Autonomic Computing, p.13-22, 2008.

[11] Eno Thereska, Bjoern Doebel, Alice X. Zheng, Peter Nobel, Practical performance models for complex, popular applications, International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'10), New York, NY, 2010 11. T. Kelly, I. Cohen, M. Goldszmidt, and K. Keeton.Inducing models of black-box storage arrays. Technical Report HPL-2004-108, HP Labs, 2004.

[12] S. Li, H. Huang, Black-Box Performance Modeling for Solid-State Drives, 18th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp.391-393, 2010.

[13] L. Yin, S. Uttamchandani, R. Katz. An empirical exploration of black-box performance models for storage systems, 14th IEEE International Symposium on Modeling, Analysis, and Simulation, pages 433-440, Washington, DC, USA, 2006.

[14] E. Thereska, D. Narayanan, and G. R. Ganger. Towards self-predicting systems: What if you could ask, what if ?, 3rd International Workshop on Self-adaptive and Autonomic Computing Systems, August 2005.

[15] D. Narayanan, E. Thereska, A. Ailamaki. Continuous resource monitoring for self-predicting DBMS, International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Atlanta, GA, September 2005.

[16] Kibler, Dennis, David W. Aha, and Marc K. Albert. "Instance-based prediction of real-valued attributes." *Computational Intelligence* 5, no. 2 (1989): 51-57.

[17] Maurya, Vijay, and Suresh C. Gupta. "Comparative Analysis of Processors Performance Using ANN." In *2015 5th International Conference on IT Convergence and Security (ICITCS)*, pp. 1-5. IEEE, 2015.