Characterizing L2 Cache Behavior of Programs on Multi-core Processors: Regression Models and Their Transferability

Jitendra Kumar Rai[1#§] Atul Negi[2#] Rajeev Wankar[3#] K. D. Nayak[4§]

ANURAG[§]
Hyderabad, India
jk.anurag@yahoo.com[1]
anuragdir@satyam.net.in[4]

Department of Computer & Information Sciences[#]
University of Hyderabad, Hyderabad, India
atulcs@uohyd.ernet.in[2]
wankarcs@uohyd.ernet.in[3]

Abstract

In this study we investigate the transferability of trained regression models to estimate solo run L2 cache stress of programs running on multi-core processors. We used machine learning to generate the trained regression models. Transferability of a regression model means how useful is a regression model (which is trained on one architecture) to predict the solo run L2 cache stress on another architecture. The statistical methodology to assess model transferability is discussed. We observed that regression models trained on a given L2 cache architecture are reasonably transferable to other L2 cache architecture and vice versa.

1. Introduction

Multi-core processors generally have level-2 caches (L2 caches) which are shared between cores or hardware threads [1][2][3]. Contention for shared L2 cache between programs running on multi-core processors is one of the performance bottlenecks. The solutions proposed by researchers to reduce the contention for shared L2 caches on multi-core processors [4][5][6][7][8] [9][10] need to know about the L2 cache related characteristics of the programs running on a multi-core processor.

In our previous work [11] we used solo run L2 cache stress of programs to characterize their L2 cache behavior, while running on multi-core processor. We also observed [11] that the regression models generated by training machine learning algorithms can be used to predict solo run L2 cache stress of programs running on multi-core processors. Transferability of a regression model means how useful is a regression model (which is trained on one scenario e.g. architecture) to predict the dependent variable i.e. solo run L2 cache stress on another scenario (e.g. architecture). A regression model trained on architecture A is considered transferable to architecture B if it can be used to accurately predict the solo run L2 cache stress on architecture B. Transferability of regression model is an important property for utilization of the model across various scenarios. It amortizes the efforts involved in training. Here scenario includes interactions between architecture and application with reference to L2 cache.

In this study we investigate the transferability of trained regression models across quad core Xeon X5482 and dual core Core2 6300 processors. Section 2 gives brief account of related work. In section 3 we mention the machine learning algorithms used. Section 4 gives brief of the experimental setup. In section 5 the statistical methodology adopted for assessing the transferability is described along with the results. Finally in section 6 we conclude.

2. Related work

Machine learning algorithms have been used for various purposes like, in [12] neural networks were used for workload characterization of a 3-tier web service in terms of functional characteristics of the application. Kumar and Negi [13] [14] used various attributes from ELF executables and the previous execution history of the processes to characterize the workload and improve scheduling. In [15] [16] machine learning algorithms were used to study the performance in terms of Instructions Per Cycle (IPC) using the event data collected from hardware performance counters. In [17] the transferability of regression models is discussed to predict the performance across various work-loads in term of their IPC. The focus of the present study is to investigate the transferability of trained regression models to predict shared L2 cache related behavior of programs across two multi-core processors having different L2 cache organization.

978-1-4244-5612-3/09/$26.00 ©2009 IEEE

3. Machine learning algorithms used

Different machine learning algorithms correspond to different concept description spaces searched with different biases. Some problems are served well by different description languages and biases, while others are not served well or even served badly. This entails the study of various machine learning algorithms belonging to different families to check their efficacy to solve a given problem across various scenarios [19]. The machine learning algorithms used in the study are: Linear Regression (LR), Artificial Neural Networks (ANN), Model Trees (M5'), K-nearest neighbors classifier (IBK), KStar (K*) and Support Vector Machines (SVM). In this study we used weka-3.6.1 machine learning workbench [19].

4. Experimental setup

This section gives brief of the experimental setup used in the study. All the 55 benchmark programs from SPEC cpu2006 suite [20] with reference input were used as workload. The two platforms used in study were based on quad-core Xeon X5482 and dual-core Core2 6300 processor respectively. The operating system kernel on both platforms was Linux-2.6.28. The reader is referred to our earlier work [11] for methodology used for data collection and training.
It is necessary to note the difference between the two processors with respect to their L2 cache organization to have a view of differences in the scenarios. On both processors there is separate level-1 (L1) instruction and data cache, each of size 32KB per core. Both of the processors have unified level-2 (L2) cache shared between two cores. Table 1 shows the L2 cache related data [1] for both the processors.. There is difference in the L2 cache organization of the two processors in terms of cache size and ways of associativity.

Table 1. L2 cache related data

Processor	No of cores sharing L2 cache	Size (MB) shared between two cores	Ways of associativity
Xeon X5482	2	12	24
Core2 6300	2	4	8

5. Transferability of trained regression models

The transferability of regression models is assessed across processors i.e. regression models trained using data from Intel Xeon X5482 were used to make predictions on Intel Core2 6300 and vice versa. Please note that each data i.e. an instance refers to a benchmark paired with a unique benchmark form 55x55 pairs. We used statistical tests and prediction accuracy metrics to assess the transferability of trained regression models. Statistical tests try to make inference on population from a given sample, while prediction accuracy metrics are confined to sample only. Here population refers to the data generated from interactions of all real world applications with architecture. While number of instances representing the pairs formed by SPEC cpu 2006 programs is 55X55 (i.e. 3025), which represents a sample from population of the instances formed from the pairings of real world programs. These methods are discussed in next subsections.

5.1. Statistical tests

We followed the statistical methods [18] [21] to compare two alternatives for assessing the transferability of trained regression models. These methods fall in two categories parametric and non-parametric. Parametric methods include t-test, where the data are assumed to be normally distributed. If there is any reason to doubt the assumption of normality of data, then we can use a distribution free test i.e. Wilcoxon test, which falls under non-parametric methods. We used R [21] for performing statistical computations.

The significance testing is done using p-values, where p-values less than the threshold (0.05) indicate towards rejection of null hypothesis [18] [22].

Testing for Normality of data:

We test the normality of class variable solo run L2 cache stress by three methods: quantile-quantile (Q-Q) plot, Kolmogorov-Smirnov test and Shapiro-Wilk test [21]. The Q-Q plot is shown in Fig. 1, where the sampled data is shown as circles against theoretical quantiles (i.e from normal distribution) shown as straight line. The calculated p-values from tests for normality are shown below:

Kolmogorov-Smirnov test	:	1.050e-13
Shapiro-Wilk test	:	1.542e-09

The p-values are much below threshold (0.05). The Q-Q plot (i.e. points v/s straight line) also indicates towards rejection of the null hypothesis (i.e. the data follows normal distribution). Hence we observe that there is doubt to assume that the data come from normal distribution.

The number of pairs of benchmarks i.e. instances of data used in study is 418. The 418 instances collected from XeonX5482 and Core2 6300 are used to generate regression models for XeonX5482 and Coere2 6300 respectively. The trained model for XeonX5482 was used to predict solo run L2 cache stress for Core2 6300 and vice versa. To assess transferability we perform the statistical tests to check the difference between actual values and the predicted values.

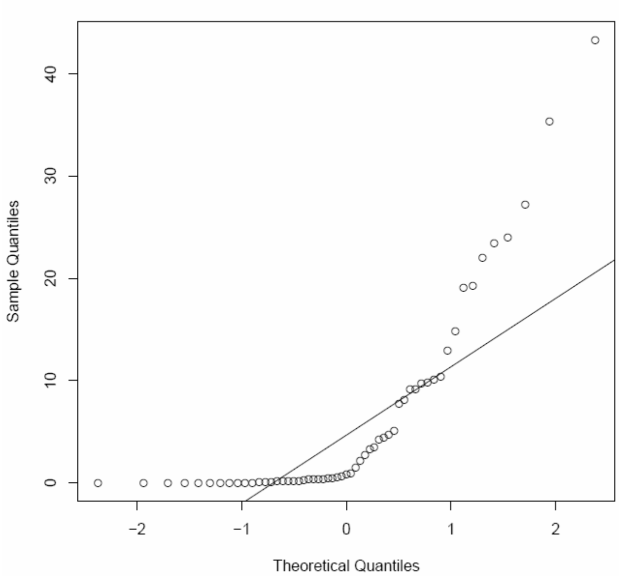

Figure1. Q-Q plot for class variable

Testing for difference between actual and predicted values:

We have two samples to compare against each other for presence of any significant difference between them. First sample consists of predicted and the other sample consists of actual values of solo run L2 cache stress on a given processor. The parametric method (t-test) assumes the data to have normal distribution. The number of instances 418 is quite large, whereby the parametric method (t-test) becomes robust enough to be used for non-normal data [22]. Hence to assess the transferability we use both parametric method (t-test) as well as non-parametric method (Wilcoxon test).

The acceptance of null hypothesis indicates the absence of significant difference between the predicted and actual values. In other words we can say that the given regression model used to predict solo run L2 cache stress is transferable across the two processors viz. one on which it was trained and the other on which it was used to make predictions.

The results of p-values for t-test as well as Wilcoxon test are shown in table-2. We use ^{x}ALGO-NAME^{y} to denote the trained regression model, where ALGO-NAME is the machine learning algorithm used, x and y are the processor used for training and testing respectively. For most of the cases p-values for both tests is greater than threshold (0.05), indicating the acceptance of the null hypothesis, which says there is not significant difference between actual and predicted values of class variable solo run L2 cache stress. There are few cases where one of the test gives p-values lower than threshold, we plan to investigate it in our future work.

Table 2. p-values and Prediction accuracy metrics

Regression Models	p-value for t-test	p-value for Wilco-xon test	C	MAE	RMSE
^{Xeon}LR^{Core2}	0.7646	0.1947	0.9928	1.1549	1.869
^{Core2}LR^{Xeon}	0.7893	0.3319	0.9922	0.7285	1.1699
^{Xeon}ANN^{Core2}	0.541	0.0002	0.9718	1.8035	3.7943
^{Core2}ANN^{Xeon}	0.9977	0.3938	0.9844	0.8603	1.6337
^{Xeon}M5’^{Core2}	0.7022	0.9486	0.9766	1.3619	3.4289
^{Core2}M5’^{Xeon}	0.8162	0.8084	0.9844	0.8603	1.6337
^{Xeon}IBK^{Core2}	0.4535	0.9556	0.9247	2.5865	5.7782
^{Core2}IBK^{Xeon}	0.2067	0.0592	0.9485	1.6731	5.0481
^{Xeon}K*^{Core2}	0.0388	0.2088	0.8754	2.4519	7.3883
^{Core2}K*^{Xeon}	0.7028	0.0143	0.9524	1.0873	2.7824
^{Xeon}SVM^{Core2}	0.6147	0.0974	0.9938	1.0508	1.6873
^{Core2}SVM^{Xeon}	0.928	0.5135	0.9930	0.6135	1.1400

5.2. Prediction accuracy metrics

The prediction accuracy of the trained models can be expressed in terms of following prediction metrics [19]: *Correlation Coefficient (C)* (value of 1 in ideal case), *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)* (for both value of 0 in ideal case). The observed values of prediction accuracy metrics are given in table 2. The prediction accuracy metrics shown indicate that most of the trained regression models perform reasonably well across two different architectures, where one is used for training and other is used for testing. The instance based classifiers IBK and K* seem to be on lower side with respect to performance accuracy metrics. In their case a new regression equation is fitted each time, for making prediction on a new instance. This may be attributed to lower performance metrics across different architectures.

6. Conclusions and future work

In this study we assessed the transferability of trained regression model across two multi-core processors viz. Intel quad-core Xeon X5482 and Intel dual-core Core2 6300. It was observed that at 0.05 level of confidence, most of the regression models under study are transferable across the L2 cache architecture of these two processors. The prediction accuracy metrics look reasonable for the given samples and indicate that trained regression models are transferable across the two processors used in the study.

In our future work we plan to use machine learning generated regression models to assist scheduling decisions on multi-core processors.

7. References

[1] Intel® 64 and IA-32 Architectures Software Developer's Manuals. http://www.intel.com/products/processor/manuals.

[2]] Poonacha Kongetira, Kathirgamar Aingaran, Kunle Olukotun, "Niagara, a 32-way Multithreaded Sparc Processor" in *IEEE Micro*, March-April 2005 pp. 21-29.

[3] Niagara2:A Highly Threaded Server-on-a-Chip http://www.opensparc.net/pubs/preszo/06/04-Sun-Golla.pdf.

[4] Yuejian Xie Gabriel H. Loh, "Dynamic Classification of Program Memory Behaviors in CMPs", in *Proceedings of CMP-MSI: 2nd Workshop on Chip Multiprocessor Memory Systems and Interconnects* in conjunction with the 35th International Symposium on Computer Architecture (ISCA-35) Beijing, China, June 22nd, 2008.

[5] D. Chandra, F. Guo, S. Kim, and Y. Solihin, "Predicting Inter-Thread Cache Contention on a Multi-Processor Architecture", in *Proceedings of the 12th International Symposium on High Performance Computer Architecture HPCA*, 2005.

[6] N. Rafique, W. T. Lim, and M. Thottethodi, "Architectural Support for Operating System-driven CMP Cache Management", in *Proceedings of the 15th International Conference on Parallel Architectures and Compilation Techniques*, Seattle, Washington, Sept. 2006.

[7] G. E. Suh, S Devadas, and L. Rudolph, "A New Memory Monitoring Scheme for Memory-Aware Scheduling and Partitioning", in *Proceedings of the 8th International Symposium on High Performance Computer Architecture HPCA*, 2002.

[8] Lisa R., Steven K., Ravishankar I. and Srihari M., "Communist, Utilitarian, and Capitalist Cache Policies on CMPs: Caches as a Shared Resource", in *Proceedings of the 15th International Conference on Parallel Architectures and Compilation Techniques*, Seattle, Washington, Sept. 2006.

[9] A. Fedorova, M. Seltzer and M. D. Smith, "Improving Performance Isolation on Chip Multiprocessors via an Operating System Scheduler", in *Proceedings of the Sixteenth International Conference on Parallel Architectures and Compilation Techniques*, Brasov, Romania, Sept. 2007.

[10] A. Fedorova, M. Seltzer, C. Small and D. Nussbaum, "Performance Of Multithreaded Chip Multiprocessors And Implications For Operating System Design", in *Proceedings of USENIX 2005 Annual Technical Conference* Anaheim, CA, April 2005.

[11] J. K. Rai, A. Negi, R. Wankar and K. D. Nayak, "On Prediction Accuracy of Machine Learning Algorithms for Characterizing Shared L2 Cache Behavior of Programs on Multicore Processors", in *Proceedings of The 2009 IEEE International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN2009)*, July 23-25 2009, Indore, India.

[12] Richard M. Y., Han L., Kingsum C. and Hsien-Hsin S. L., "Constructing a Non-Linear Model with Neural Networks for Workload Characterization", in *Proceedings of the IISWC*, 2006.

[13] K. K. Pusukuri and A. Negi, "Applying machine learning techniques to improve GNU/Linux process scheduling", in *Proceedings of IEEE Tencon Conference*, Australia*,* Dec., 2005.

[14] K. K. Pusukuri, A. Negi, "Characterizing process execution behavior using machine learning techniques" in *Proceedings of HiPC International Conference, DpROM'4 workshop,* Bangalore, India*,* Oct., 2004.

[15] ElMoustapha O. and James W., Charles Y. and Kshitij A. D., "On the Comparison of Regression Algorithms for Computer Architecture Performance Analysis of Software Applications", in *Proceedings of the First Workshop on Statistical and Machine learning approaches applied to ARchitectures and compilaTion (SMART'07)***,** Ghent, Belgium, January 27 2007.

[16] ElMoustapha O. and James W. and Charles Y., Kshitij A. D. and Seth A., "Using Model Trees for Computer Architecture Performance Analysis of Software Applications", in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems & Software, ISPASS 2007*. pp. 116-125.

[17] ElMoustapha O.., Doshi, K.A., Yount, C., Woodlee, J., "Characterization of SPEC CPU2006 and SPEC OMP2001: Regression Models and Their Transferability",. in *Proceedings of the 2008 IEEE International Symposium on Performance Analysis of Systems and Software. ISPASS* 2008

[18] R. E. Walpole, R. H. Myers, S. L. Myers and K. Ye, *Probability and statistics for engineer and scientists*, 8^{th} Edition, Pearson Education, Delhi, 2007. pp. 413-467.

[19] Ian H. Witten and Eibe Frank (2005) *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[20] Standard performance evaluation corporation. SPEC CPU2006. http://www.spec.org/cpu2006/.

[21] R Development Core Team.: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. (2004)

[22] Non parametric test with small and large samples. http://www.graphpad.com