Evaluate the Prediction Accuracy and Confidence Intervals of Intel Nehalem base on Regression Model

Mahmoud Askari
Belarusian State University of Information and Radioelectronics, Minsk, Belarus
Department of Computer, Damavand Branch, Islamic Azad University, Damavand, Iran

ABSTRACT

In this paper, has been investigated the predicted accuracy and confidence intervals of performance on multi–core processor i5–460M in various modes of processor included: single, parallel and hyper–threading on SPEC CPU2000 with fixed point operations. The experiments have been performed by Intel–vtune 2013 and have been modeled base on two methods of regression analysis that are Multi–linear and Robust regression along with the accuracy of their predictions. Result of this paper is applicable for producers and users of operating systems and applications due to more accurate models have a lower risk in prediction and thus they can contribute to the better scheduling of applications.

Keywords

Nehalem, Performance, SPEC CPU2000, Regression, Prediction accuracy, Confidence interval

1. INTRODUCTION

In this paper, it is calculated the predicted accuracy and confidence interval of performance on a multi–core processor i5–460M. The calculations have been performed in three modes of processor included: single, parallel, hyper–threading modes. The results show the prediction accuracy in hyper–threading mode is better than other modes. It can help to designers that want to perform the scheduling with low risk and higher reliability to predictions.

Efforts of performance analysts are finding more accurately models to predict. It can help to suitable predict to better task scheduling. In this paper, it is analyzed a multi–core processor i5–460M. Intel Core i5–460M micro–architecture Nehalem [1] has two cores and three levels of caches, where L1 and L2 are exclusive and L3 is an inclusive cache with respect to L1 and L2. The L1 cache is divided for instruction and data parts, they are allocated to each core separately, L2 cache is also allocated to each core, instructions and data are stored in L2 cache together. L3 cache is shared between the cores. TLB design is performed in hardware mode on this processor. It has two levels, the first level of the buffer allocated for each core, and then it is divided for instruction and data. Instruction TLB is divided into two modes: 4 kilobyte pages size and 2 (or 4) megabyte pages size. 4 kilobyte mode has 4–way set associative structure and 64 entries line in cache. 4 megabyte mode has fully associative structure with 7 entries line. Data TLB is divided into two modes: 4 kilobyte pages size and 2 (or 4) megabyte pages size. 4 kilobyte mode has 4–way set associative structure and 64 entries line in cache. 4 megabyte mode has 4–way set associative structure with 32 entries line. The second level TLB (STLB) allocated for each core separately. During the execution multi–threading was enabled and pre–fetching was disabled.

To evaluate the performance of a processor, a way is to measure Cycle per Instruction (CPI) [2]. There are several ways to predict the CPI based on independent variables generally base on miss ratio of hierarchy memory components. These methods are mainly statistical techniques, decision trees, neural networks and genetic algorithms [3]. In statistical, several regression methods are used. The accuracy of these methods is evaluated using accuracy parameters [4].

A lot of work has been done for modeling the processor and also to evaluate the accuracy of prediction models. ElMoustapha et al [5] have compared the accuracy of some regression models on Intel Core 2 processor. They have concluded that regression tree and linear regression have best prediction accuracy than other regression methods. Hussam Mousa et al. [6] have used multi–linear regression model and model–tree design to analyze the Cycle per Instruction (CPI) in its various architectural and virtualization events. They have illustrated a path to building a predictive model for workload performance. Rai et al. [7] have suggested regression models by learning the cache L2. They have shown that the processor Intel Core Duo model obtained from a single processor accurately predicts L2 on a different processor. There are also other works in this field [8] [9] to study.

2. EXPERIMENTAL METHODS

Experiments have performed base on 64–bit Intel environment which use features of Performance Monitoring Unit (PMU) [10] to measure various events using Intel–vtune 2013 [11]. The applications that have been used to run and calculating Cycle per Instruction (CPI) and miss ratios of memory hierarchy components are all 12 fixed point benchmarks of SPEC CPU2000 package [12]. For the reliability of the results, each benchmark is performed 50 times and each execution consists of three random repetitions for each working mode of processor including single, parallel, hyper–threading mode and therefore a total of 5400 times have been performed experiments. During the experiments, pre–fetching was disabled and it has no effect on the results. To calculate the CPI and miss ratios have been used Equations 1 to 8 based on related events in Table 1.

The values of the CPI and miss ratios for each benchmark have been used to produce two models of regression by Matlab software in three various modes of processor under test [14]. These models are used to predict the dependent variable CPI from independent variables including miss ratios of hierarchical memory components [15]. In Table 2, the coefficients of 4 core mode for two models of regressions are shown. The columns included: ITLB, DTLB, STLB are shown to enable readers to evaluate how the models have been produced.

Table 1. Events used to calculate CPI and miss ratios on Intel Nehalem i5–460M

Event name	Explanation
CPU_CLK_UNHALTED. THREAD	Total execution cycle of the application under test.
INST_RETIRED.ANY	Number of instructions that retired execution.
ITLB_MISS_RETIRED	Number of retired instruction that miss on ITLB.
DTLB_MISSES.ANY	Number of data requests that miss on DTLB.
DTLB_LOAD_MISSES. STLB_HIT	Number of miss on DTLB that Hit on STLB.
L1I_MISSES	Number of miss on Instruction L1.
L1D_REPL	Number of miss on Data L1 when L1 Data cache line is allocated.
L2_LINES_IN.SELF.ANY	Number of allocated lines to miss on L2.
MEM_LOAD_RETIRED. L3_MISS	Number of Retired loads that miss the L3 cache.

To evaluate the prediction accuracy following common metrics is used [5]:

The Correlation Coefficient: This value measure the amount of linear relationship between predicted (*P*) and actual (*A*) values. Its range is between *-1* to *1* that *1* is ideal correlation. This correlation coefficient *C* is given by Equation 9.
That $Cov\,(P, A\,)$ is covariance between predicted and actual values, σ_P and $\sigma_A$are standard deviation for P and A respectively.
Root Mean Squared Error (RMSE): This value is used to measure of confidence intervals. Its range is from *0* to infinity that *0* is ideal case. This error is calculated by Equation 10.
That p_i and a_i are predicted and actual value of dependent variable in *i*th test and *N* is the number of observations or instances.

Table 2. The coefficients of Multi–linear (M) and Robust (R) regression models for 4 core mode of 460M.

Benchmark		ITLB	DTLB	STLB
gzip	M	1033	1552.7	-58.499
	R	1087	1512	-56.696
vpr	M	717.93	-143.29	5.40880
	R	797.65	-136.33	5.71520
gcc	M	704.49	-65.534	-77.53
	R	698.88	-61.812	-78.62
mcf	M	4253.1	-375.38	112.74
	R	3625.9	-379.2	125.26
Crafty	M	286.43	-310.57	1.59180
	R	318.39	-155.81	3.85370
Parser	M	047.4	-142.92	20.297
	R	1001.8	-126.17	20.595
Eon	M	45.675	-26.58	-60.18
	R	44.901	-29.009	-58.547
Perlbmk	M	818.99	-12.33	-0.7208
	R	797.2	-12.699	2.1188
Gap	M	1041.7	-215.44	5.21240
	R	1026.6	-217.4	5.0958
Vortex	M	824.91	-146.12	4.99820
	R	806.22	-140.75	4.57280
bzip2	M	1310.7	128.5	77.947
	R	1422.4	132.37	72.086
Twolf	M	362.99	-49.822	-6.6632
	R	366.28	-45.893	-6.6385

$$CPI = CPU\_CLK\_UNHALTED.THREAD\ /\ INS\_RETIRED.ANY$$
(1)

$$ITLB\ RATIO = ITLB\_MISS\_RETIRED\ /\ INS\_RETIRED.ANY$$
(2)

$$DTLB\ RATIO = DTLB\_MISSES.ANY /\ INS\_RETIRED.ANY$$
(3)

$$STLB\ RATIO = 1 - (DTLB\_LOAD\_MISSES.STLB\_HIT\ /\ INS\_RETIRED.ANY) \qquad (4)$$

$$L1I\ RATIO = L1I\_MISSES\ /\ INS\_RETIRED.ANY$$
(5)

$$L1D\ RATIO = \ L1D\_REPL\ /\ INS\_RETIRED.ANY$$
(6)

$$L2\ RATIO = L2\_LINES\_IN.SELF.ANY\ /\ INS\_RETIRED.ANY$$
(7)

$$L3\ RATIO = MEM\_LOAD\_RETIRED.L3\_MISS\ /\ INS\_RETIRED.ANY \qquad (8)$$

$$C = \ Cov\,(P, A\,) /\ (\sigma_P * \sigma_A) \qquad (9)$$

$$RMSE = \sqrt[2]{\frac{\sum_{i=1}^{N}(p_i - a_i)^2}{N}} \qquad (8 \qquad (10)$$

The Correlation Coefficient: This value measure the amount of linear relationship between predicted (*P*) and actual (*A*) values. Its range is between *-1* to *1* that *1* is ideal correlation. This correlation coefficient *C* is given by Equation 9:

3. RESULT and DISCUSSION

Table 3 and 4 shows the calculated values of *Correlation* and *RMSE* for all 12 benchmark in 3 modes of Intel processor i5–*460*M. In this table, *R* is *Robust* and *M* is *Multi–linear* regression. In 3 benchmarks *eon*, *perlbmk* and *mcf* the value of *Correlation* have more difference from 1 core mode to modes 2 and 4 cores. The *mcf* (single–depot vehicle scheduling) is benchmarks that needs to 190 megabyte memory. Because the amount of caches is less than it, therefore in 1 core mode (sequentially execution) they refer repeatedly to the memory. But, if the number of cores is 2, it is discussed using the second level of cache on other core (snoop) that it is predicted (maybe) to find the required value. The caches have a more effective role compared to 1 core mode and the extracted model is more effective to prediction. Similarly, in the case of 4 cores each thread can use the cache of other threads.

But, on 1 core mode *RMSE* shows unsuitable confidence interval for all benchmarks.

Table 3. The Correlation of Multi–linear (M) and Robust (R) regression models

Benchmark		Correlation		
		1 core	2 core	4 core
gzip	M	0.9669	0.9693	0.9707
	R	0.9670	0.9695	0.9707
vpr	M	0.9774	0.9639	0.9792
	R	0.9774	0.9643	0.9792
gcc	M	0.8037	0.9262	0.9729
	R	0.8587	0.9262	0.9729
mcf	M	0.2681	0.8593	0.9321
	R	0.7031	0.8626	0.9331
crafty	M	0.9775	0.8774	0.9776
	R	0.9778	0.8951	0.9777
parser	M	0.9767	0.9714	0.9730
	R	0.9779	0.9714	0.9731
eon	M	0.2090	0.7673	0.8830
	R	0.7514	0.7676	0.8832
perlbmk	M	0.4258	0.8594	0.9780
	R	0.9559	0.8624	0.9781
gap	M	0.7851	0.8996	0.9739
	R	0.8357	0.8997	0.9739
vortex	M	0.9679	0.9512	0.9746
	R	0.9679	0.9517	0.9748
bzip2	M	0.9585	0.9455	0.9726
	R	0.9592	0.9450	0.9726
twolf	M	0.9788	0.9789	0.9796
	R	0.9788	0.9789	0.9796

It can be seen on Table 4 in mode of Core 1. The *eon* is a computer visualization program with a small size about 1.5 megabyte that renders a 150x150 pixel image, in each time of instruction execution, previous calculated values.

Table 4. The RMSE value of Multi–linear (M) and Robust (R) regression models

Benchmark		RMSE		
		1 core	2 core	4 core
gzip	M	2.4867	0.2848	1.4191
	R	0.1887	0.0002	0.0001
vpr	M	0.1257	0.3388	0.2482
	R	0.3018	0.0001	0.0001
gcc	M	0.7217	0.0544	8.4020
	R	0.7478	0.0002	0.0001
mcf	M	0.3571	2.5085	4.5526
	R	1.7042	0.0048	0.0046
crafty	M	0.0282	0.1202	2.0224
	R	0.1176	0.0009	0.0001
parser	M	0.0748	0.0063	3.5534
	R	0.2717	0.0002	0.0001
eon	M	0.5094	0.2269	11.8260
	R	0.7605	0.0004	0.0003
Perlbmk	M	0.3453	0.0877	3.0994
	R	0.2840	0.0001	0.0002
gap	M	0.2184	0.1278	3.2097
	R	0.1297	0.0000	0.0002
vortex	M	0.2511	0.0970	0.8684
	R	0.2976	0.0001	0.0000
bzip2	M	0.6084	0.1785	0.1764
	R	0.6317	0.0002	0.0001
twolf	M	0.3396	0.1070	0.8541
	R	0.3108	0.0000	0.0001

This value is not use to another step in cache. But, the number of cores is suitable for it to run as parallel mode. The extracted model has not an accurate prediction compared to most other benchmarks. The *RMSE* for cases 2 and 4 core modes shows especially for the Multi–linear model values equal to zero.

The *perlbmk* is a cut–down version of Perl programming language that has less value of *Correlation* and *RMSE*. Table 5 shows three benchmarks with low *Correlation* than other benchmarks. This value is occurred on Multi–linear regression method and on one Core mode of system.

Table 5. The low Correlations of Multi–linear

Benchmark	Correlation		
	1 core	2 core	4 core
mcf	0.2681	0.8593	0.9321
eon	0.2090	0.7673	0.8830
perlbmk	0.4258	0.8594	0.9780

4. CONCLUSION

In this paper, it is modeled 3 working mode of Intel Nehalem i5–460M by two regression models included Multi–linear and Robust regression. The models have been evaluated for predicted accuracy and confidence intervals. The best features of predicted accuracy and confidence intervals can be seen on hyper–threading mode of processor. The authors suggest as future work to evaluate the predicted accuracy and confidence interval by other regression models like Tree–regression to compare the results due to these results is applicable for producers and users of operating systems and applications to select the best predict models.

5. REFERENCES

[1] Thomadakis, E. M. 2011 The architecture of the Nehalem processor and Nehalem-EP SMP platforms. Texas A&M University.

[2] Hennessy, J. L. and Patterson, D. A. 2011 Computer Architecture: A Quantitive Approach. Morgan & Kaufmann Publishers.

[3] Y.S. Kim, "Comparison of the decision tree, artificial neural network, and linear regression methods based on the number and types of independent variables and sample size", Expert System with Application: An International Journal, 2008, No 2. pp. 1227-1234.

[4] Lee, B. C. and Brooks, D. M. 2006. Accurate and efficient regression modeling for microarchitectural performance and power prediction. In Proceeding of the 12th international conference on Architectural support for programming languages and operating systems. pp. 185-194.

[5] Ould, E., Woodlee, J., Yount, C. et al. 2008. On the Comparison of Regression Algorithms for Computer Architecture Performance Analysis of Software Applications. In International Symposium on Performance Analysis of Systems and Software. pp. 179-190.

[6] Mousa, H., Doshi, K., Sherwood, R. et al. 2010. VrtProf: Vertical Profiling for System Virtualization. In Hawaii International Conference on System Science. pp. 1-10.

[7] Rai, J. K., Negi, A., Wankar, R. et al. 2010. Characterizing L2 cache behavior of programs on multi-core processors: Regression models and their transferability. In International Journal of Computer Information Systems and Industrial Management Applications. pp. 212-221.

[8] Joseph, P. J., Vaswani, K., Thanzhuthaveetil, M. J. et al. 2006. Construction and Use of Linear Regression Models for Processor Performance Analysis. In Twelfth International Symposium on High-Performance Computer Architecture. pp. 99-108.

[9] Xu, Z., Sohani, S., Min, R. et al. "An Analysis of Cache Performance of Multimedia Applications", IEEE Transactions on Computers, 2004, pp. 20-38.

[10] Choi, Y. 2001. Design and Experience: Using the Intel Itanium 2 Processor Performance Monitoring Unit to Implement Feedback Optimizations. In Proceedings of the 34th Annual International Symposium on Microarchitecture. pp. 182-191.

[11] Reinders, J. 2005 Vtune Performance Analyzer Essentials. Intel.

[12] Gfroerer, D., Tricket, N., Nakagawa, T. et al. 2003 Understanding IBM eServer pSeries Performance and Sizing. IBM.

[13] Intel Corporation: Intel 64 and IA-32 Architectures Optimization Manual. [Online]: www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html.

[14] Solka, M. 2011 Exploratory Data Analysis with MATLAB. Chapman & Hall.

[15] Askari, M., Ivanov, N. N., "The Dependence of Physical Memory Footprint of Processor on the Applications", Asian Journal of Computer Science and Technology, No 2. Vol. 2. pp. 4-10.