Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

The Dependence of Physical Memory Footprint of Processor on the Applications

Mahmoud Askari[1], Nick Ivanov[2]
Belarusian State University of Informatics and Radioelectronics, Belarus
[1]mahask2004@yahoo.com

ABSTRACT

Recently, with growing the gap between processors and memory speeds, parallel performance on chip multicore processors becomes more attractive for filling up this gap. In this direction, calculating the Cycle Per Instructions (CPI) and its relationship with cache miss ratio is important. In this paper, impact of cache usage on Intel i5–460M processor by SPEC CPU2000 with fixed point operations is investigated. At first, the model of the memory hierarchy is under discussion. Afterwards, dependency of cache memory usage is discussed. In part IV and V, regression analysis of data and results are considered. Experiments exploited VTune 2013 counters demonstrate that switching off particular caches depended on kind of application enhances processor performance.

Keywords— **Intel Nehalem, Processor performance, Software footprint, SPEC2000, VTune2013**

I. INTRODUCTION

The paper investigates the dependency of processor caches on application. Methods of statistical analysis show that the management strategy of caches is largely dependent on the application. It is proposed to change the management of caches, depending on the type of application. For example, by entering into the operating system, it is able to disable some levels of memory at a mass execution of applications of one kind. Dividing applications into these classes and putting these classes in performance leads to more efficient use of computing resources. Using more of an application of one class, in order to optimize the use of caches, can bring significant savings in time.

The effort of computer developers increases the efficiency of the computer by providing new solutions and developing operating systems [1]. Modern personal computers have multiple cores and 2 to 4 levels of cache memories. Cache is a fast memory to store data in order to speed up the operation of exchanging data with RAM. Stored in cache instructions, processes and intermediate information, which are called "data", exist. Memory, different cache levels, and TLB (Translation Look–aside Buffer) are as a hierarchical structure that are largely dependent on the speed of the multithreading in the core of processors. The paper deals with the dual–core processor and three levels of cache including L1, L2 and L3. The highest level of L1 cache is divided into two sections one keeps the instructions (L1I) and the other keeps the data (L1D). In the considered architecture, the L2 cache is divided into two blocks, each being designed to use only one core. The L2 cache instruction and data are stored together. L3 cache in the third level is common to both cores, along with instructions and data. The problem of the intensive use of the graphics

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

processor and cache to speed up this processor is not considered here. With the memory Paging, which is used in almost all modern processors, converting virtual address to physical address is done in TLB. This is a specialized cache of the CPU that is used to accelerate translating virtual memory address into a physical memory address. TLB has a fixed set of entries (8 to 4096) and is organized as an associative memory. Each input contains the corresponding address of the page of virtual memory addresses to physical memory. Multi–core processors usually have two levels of TLB. The first level is divided for instruction (TLB–I) and data (TLB–D); the second level TLBs has common content, as well as common multiple cores. The rational use of caches and their management has begun to explore the emergence of the idea of creating an intermediate buffer between memory and the processor [2]. The vast majority of applications in both instruction and data only use the limited space of memory. If this space is small in size, it can be copied to the buffer that significantly accelerates execution time. Multiple levels of caches, together with RAM memory hierarchy, create modern personal computers. The principles of the memory hierarchy depend on the design of the CPU, RAM, and the interface between them [3]. A detailed description of the principles of memory architecture is given in [4]. For the study of the caches several approaches are used. The classic study with the construction of the analytical probability model is described in [5]. The slowdown in the processing of data is caused by misses when accessing the cache. The authors, on the basis of the available data, have derived the probability of misses from the previous and current state of the cache like the content of the cache when the application starts. It is assumed that during the execution of the program, it goes into a steady state over the probabilities of misses in caches. On the basis of the construction and analysis, the Markov chains transitions from one application to another. Analytical approach to the memory hierarchy also considerably simplifies the real work of a computer system. The increasing complexity of the model causes difficulties in its analysis, especially when trying to optimize memory management. In addition, stability of the model to the unrecorded details of architecture and operating system remains unclear. Another approach to the analysis of management caches is a simulation of a computer system in terms of calls to the caches. A simulation model allows us to analyze any configuration and to compare different configurations. It is required to plan experiments, but in the case of the analytical model, the question of taking all relevant factors into account is there to be considered. Yet if the equipment is subject to change, the importance of the factors may change. Furthermore, it remains unclear whether all significant dependence is included in the model. The third approach is the experimental study of a computer system. There is a widespread testing of computer system applications, specially created for this purpose. In 1988, SPEC (Standard Performance Evaluation Corporation), which has the goal of developing the applications for the evaluation of the performance of computers, was founded. Benchmarking SPEC, its applications became standards to verify the performance of modern processors and system software [6]. SPEC develops benchmarking applications and in the process of their implementation it gathers various options. In particular, these applications are tested for the processor in conjunction with an existing design memory. There is a set of tests for computing the fixed–point and a separate set for floating point. In this paper, by statistical methods, parameters of

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

processor memory Intel Core Duo i5–460M are investigated. Processed data were obtained from the SPEC tests for fixed–point operations. Similar studies have been described in several papers. ElMoustapha and others built multiple regression coefficients depending on the average number of CPU Cycles Per Ieam (CPI) from the misses while accessing the TLB, L1, L2, L3 [7]. This paper investigates the use of regression models for the analysis of different types of memory dual–core processor. The authors note that the regression tree model gives the best interpretation of the functioning of memory, which corresponds to the intuitive idea of the role of L2 in the use of data. In the paper, J. K. Rai et al [8]. constructed regression models by learning the cache L2. The authors show that the processor Intel Core Duo model obtained from a single processor accurately predicts L2 work on a different processor. P. J. Joseph [9] built a regression by factor analysis for the IBM PowerPC processor and showed that there are three most significant factors: the pipeline depth of the command, the reordering buffer and queue micro–processor. In addition, the size of the L2 cache latency and exchanges with it are important. There are studies on the dependence of tactics cache utilization characteristics of the application. So [10] an analytical model, which recognizes the needs of the practical application at the beginning of the dynamic distribution, is suggested. Then, when the application requests data from the cache, it addresses the needs in the near future. In the paper [11], proposed changing the role of a buffer clearing (victim cache), which temporarily stores rows deleted from the cache. In addition, the authors analyze the size of the L2 cache and note the dependence of its optimal amount of class applications running on the computer.

II. THE MODEL OF THE MEMORY HIERARCHY

Here are analyzed experiments for the mobile processor Intel Duo Core i5–460m microarchitecture Nehalem. The processor has two cores and three levels of caches, in which L1 and L2 are exclusive and L3 cache is inclusive with respect to L1 and L2. The L1 cache is divided into instructions and data for each core, and L2 cache for each core is separated; instructions and data are stored in the shared cache. The TLB–buffer, which has two levels, is constructed by hardware. The first level of the buffer is separated into each core. It is also divided for instructions and data. The buffer of the second level TLBs is common for the two cores. If the address is not found in the TLB of the first layer being accessed, a message will be sent to TLBs buffer to search for the addresses in the TLB of the first level of the other core. If the address of the second core is not found either, the situation will be a miss with a complicated and time–consuming search for the page that contains the desired address. This search is divided into several branches, which are out of this paper. The interaction between levels of memory and the processor and among themselves is shown in Fig.1.

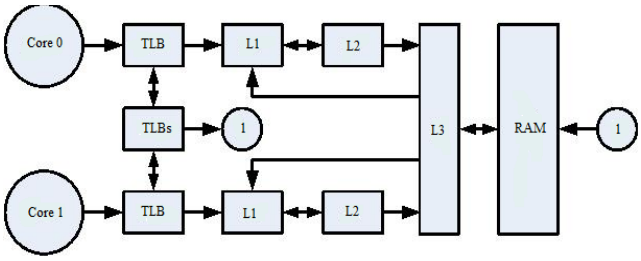

Fig.1. Memory hierarchy of the processor Intel i5–460M

III. THE STUDY OF DEPENDENCE OF CACHE USAGE FROM APPLICATION

Profiling of the program is performance measurement of the whole program and all its individual fragments [12]. In this paper, the

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

processor Intel Duo Core i5–460m investigated tests computing performance for SPEC CPU2000 for fixed point. In this paper, all 12 benchmarks of package CINT2000 are used: 164.gzip, 175.vpr, 176.gcc, 181.mcf, 186.crafty, 197.parser, 252.eon, 253.perlbmk, 254.gap, 255.vortex, 256.bzip2, 300.twolf and 252.eon [13]. For the reliability of the results of the experiments, each application is performed at least 50 times. One execution consists of three repetitions of the application to calculated average parameters and in total 1800 times is performed. The experiments were carried out under the Intel VTune [14], which writes the current values of performance counters in the process of execution of the application. The experiments were performed by testing programs for the collection of the parameters listed in Table I. They are used for calculating five parameters: CPI –the average number of processor cycles to the instructions when the application misses handling proportion to caches L1, L2, L3, and the table TLB, which are calculated by the formulas (1) – (5) from the firm Intel document[15]. In this paper, equal names are used instead event's names.

$$CPI = CCT/IRA \quad (1)$$

$$L1_M = (MLL * 12 + MLP * 26 + MLX * 43 + MLS * 60 + MLM * 180)/CCT \quad (2)$$

$$L2_M = (MLP * 26 + MLX * 43 + MLS * 60)/CCT \quad (3)$$

$$L3_M = (MLM * 180)/CCT \quad (4)$$

$$TLB_M = (DLS * 7 + DLW)/CCT \quad (5)$$

TABLE I
LIST OF MONITORED EVENTS

Event	Equal name
CPU_CLK_UNHALTED. THREAD :Number of cycles completed in the current moment of time.	CCT
INST_RETIRED.ANY :The number of instructions that are completed in current time.	IRA
MEM_LOAD_UOPS_ RETIRED.L2_HIT :The number of hits at reversal to L2.	MLL
MEM_LOAD_UOPS_MISC_ RETIRED.LLC_MISS_PS :The number of misses at reversal to L3.	MLM
MEM_LOAD_RETIRED.L3_HIT _PS :Number of hits when referring to L3.	MLP
MEM_LOAD_UOPS_LLC_ HIT_RETIRED.XSNP_HIT_ PS :Number of requests sent to another kernel if the data is not found in the L3 of the core, and another kernel memory is available.	MLX
MEM_LOAD_UOPS_LLC_HIT_RETIRED.XSNP_HITM_PS :Number of requests sent to another kernel if the data is not found in the L3 of the core and the memory of another core is in modifies state.	MLS
DTLB_LOAD_MISSES. STLB_HIT :Number of hits when accessing the TLBs.	DLS
DTLB_LOAD_MISSES. WALK_DURATION :Number of near misses when accessing the TLBs.	DLW

IV. REGRESSION ANALYSIS OF DATA

The analysis the experimental data can be carried out by regression method. Linear regression analysis solves the problems associated with building relationships between groups of numerical variables; one of the groups contains the independent variables, that is, predictors affecting the values of the dependent variables. According to the results of experiments required to construct a function which approximately describes the change in the dependent variable from the independent variables of the linear function with a minimum standard deviation of the

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

results of the experiment. Linear regression that used in the study parameters, functions, and relationships between them are unknown. In the process of the study, when it turns out kind of dependency, the linear dependence in the regression is replaced by a more suitable non–linear function. A special case of linear regression is a common situation when a group of dependent variables contain only one variable. In this case, a dependence of the form.

$$y = b_1x_1 + b_2x_2 + K + b_nx_n + a \qquad (6)$$

Where y –the dependent variable, x_i –the independent variable, b_i –coefficients, a –is a free member. The parameters a, b_i are estimated from the results of the experiment, the method of least squares. Since the results of the experiment are randomized, then the parameter estimates are also random. Typically, the accuracy of the estimates of the parameters is verified by the hypothesis of zero values of the coefficients of the independent variables.

$$H_0: b_1 = b_2 = \cdots = b_n = 0 \qquad (7)$$

Hypothesis can be tested using Fisher's exact test, the null hypothesis is accepted if the numerical value of the Fisher statistics (F–statistic) in an experiment more than a predetermined threshold value. Otherwise hypothesis H1 is acceptable.

For the multivariate analysis with several independent variables, it is not recommended to leave the method of incorporating of all the variables established by default. This approach corresponds to the simultaneous processing of all the independent variables selected for analysis. If the coefficients of multiple regressions used as indicators of the influence of parameters and factors, it should be remembered that the regression coefficients are generally not comparable. Their numerical values depend on the units of measurement for each factor. Because regression coefficients are comparable, they are normalized. To do this, all the variables are expressed as dimensionless, so–called standardized units by converting Gaussian random variable to a standard normal distribution. While standardizing, the constant term of the regression (6) equation vanishes, and the equation itself with normal and adjusted variables will be as follows:

$$y = \beta_1x_1 + \beta_2x_2 + \cdots + \beta_nx_n + \varepsilon \qquad (8)$$

The coefficients of β_1, β_2, ..., β_m are the regression coefficients on a standardized scale. All of the variables expressed in equations are comparable units. Now more than β_i, the stronger influence factor corresponding setting on the dependent variable.

The degree of the influence of the factor can be estimated using partial correlation coefficients. Partial factors are used in the selection of factors: the desirability of placing a factor in the model is determined by the partial correlation index. Partial correlation coefficient indicates a close relationship between the relevant factors in the elimination of other factors in the regression equation. Indicators of partial correlation are the ratio of the reduction of residual variance, which occurs as a result of additionally including a new factor in the analysis, to the residual variance, which takes place before the introduction of a new factor in the model.

For multi–factor analysis, you can select one of the turn–based methods. In the next step in the directly method with the independent variable in regression equation, the dependent variable has the highest correlation coefficient. The reverse method starts with the result that contains all the independent variables, and then the

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

independent variables with the smallest partial correlation coefficients are eliminated incrementally as long as the corresponding regression coefficient with specified degree of importance is significant.

The most common step by step method that combines direct and inverse method [16]. It alternates at each step of the forward and reverse motion. Thus there are two important parameters, p and p–inclusion removal, while p–turn must be less than p–removal. In typical applications, the method is believed p–switching=0.5; p–remove=0.10, is used exactly these values . The inclusion of executed test of the hypothesis about the importance of zero partial correlation factors with the dependent variable. The hypothesis is rejected, and then there is a factor considered significant if the value of the F–statistic is less than the Fisher partial correlation given probability p–inclusion. Otherwise, a factor not included in the set of meaningful and the algorithm proceeds to the verification of the following factors.

If the factor is included in the number of significance, the situation will be changed and perform the step of the inverse method. In this step, the factors included in the number of significance are checked. For these factors, hypotheses about the importance of zero partial correlations, which are calculated for the F–statistic of partial correlations, are tested sequentially. If the resulting statistics are more than a given p–removal probability, the main hypothesis is accepted and the factor is removed from the significant number.

V. RESULTS AND ANALYSIS

Experiments were conducted to test the impact of the use of caches in the average number of execution cycles of the same instructions, mean the macro processor Intel. As test applications used all 12 programs package CINT2000 testing applications SPES CPU2000. Event counters accounted system VTune Amplifier 2013. The results are in a summary table II.

Table II contains L1, L2, L3 caches as well as TLB addresses. In Table II, row with #1 means K–regression, #2 means K–factors, #3 means Factor, #4 means p–value and #5 means Standard-coef. The B is related to row #1. In addition, for the multiple linear regressions with the four mentioned variables, the free member “a” is calculated. These coefficients and the constant term of the formula (6) are included in the lines "K–regression". In the following line, the coefficients indicate the coefficient of determination R2. The coefficient of determination indicates how much of the variance resultant variable (CPI) is explained by the regression equation. The higher R2 is, the more the variance explains resultant variable regression equation, and the better the regression equation describes the original data. In the absence of a variable depending on the CPI four independent variables, the coefficient of determination R2 is close to zero. R2 can range from 0 to 1. In all experiments, except for EON and GAP, the value of R2 is close to the unit.

Along with R2, the probability p of the null hypothesis is given in the same line (see Application). For the null hypothesis, it is accepted that all the regression coefficients are equal to 0, i.e. the regression equation is not meaningful. If we accept the null hypothesis when confidence level equals 0.95, when $p<0.05$, this hypothesis will be rejected. In all experiments, with the p quantity much below 0.05, the coefficients are significant. In the line "K–factor," there are coefficients of the formula (8). The next line of "Factor" shows the importance of the calculated factors, the most important factor in isolated

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

digit 1. If the factor is not significant, it will be marked with the symbol X. The string "p–value" indicates the amount of p–turn at the last step. In the line "Standard-Coef", the standardized deviation is obtained from the regression equation (8), the lower the number is, the better the value of the predictor. For non–significant factors, it is stated that the experiment was unable to identify the importance of the reliable factor. This may cause excessive variability numeric factor in the experiment. In some cases, this could cause a significant increase in the number of cross experiments. In others, even a considerable amount of experimentation is not possible to estimate the importance of the factor. In particular, this applies to the test 181.mcf –a program for solving the scheduling of warehouse car park maintenance (single–depot vehicle scheduling). The problem is solved by the modified simplex method and requires 192 MB of memory for 64–bit architecture. The experiment showed that the volume of 3 MB L3 cache is not enough to improve the performance of the application.

VI. CONCLUTION AND FUTURE WORK

Experiments were carried out on real processors. The authors of the article were not able to change the amount of cache, so the results of the experiments are only valid for the specified fixed amount. Conclusions about the local change of volume of one of the caches are valid only at constant other characteristics of the processor, in particular, the preservation of the volumes of other caches. On the other hand, the conclusions of the article for the majority of test programs are supported by the known amount of memory required to run the algorithms implemented in the tests. Since 186.crafty test – chess computer software – has the most dependence volume cache L1, as a positive value in a regression coefficient (1), its volume brings about more decrease in test performance. The second most important factor – the cache L3 – affects the performance of the processor positively; it reduces the increase in the average number of cycles per instruction. L2 cache and TLB are too volatile; the experiment could not reveal the extent of their impact on processor performance. From this we can conclude that the L1 cache of the application as well as all applications using the same memory must be released. After this, it is necessary to repeat the experiments, then compare the results and draw further conclusions.

TABLE II
LINEAR REGRESSION AND FACTOR ANALYSIS OF CACHES

	TLB	L1	L2	L3
0	1	2	3	4
CRAFTY, R2=0.9935; p=7.064E-39, B=0.6102				
#1	-0.3648	6.1501	-1.1860	-4.150
#2	-0.3029	5.4992	-0.4814	-3.5601
#3	X	1	X	2
#4	0.13	1.9E-33	0.69	7.63E-28
#5	0.1947	0.131	1.2201	0.1203
GZIP, R2=0.9894; p=8.911E-44, B=2.280				
#1	-0.9853	-2.105	2.384	2.978
#2	-0.9853	-2.105	2.384	2.978
#3	4	3	2	1
#4	2.47E-3	9.59E-9	1.22E-21	7.94E-22
#5	0.3073	0.2999	0.1368	0.1690
PARSER, R2=0.9968; p=4.959E-53, B=-0.3767				
#1	0.8874	-0.9797	10.37	4.728
#2	0.8640	-0.6643	9.786	4.011
#3	X	X	2	1
#4	6.87E-2	7.21E-1	1.89E-29	2.37E-46
#5	0.4630	1.851	0.3623	0.06102
VORTEX, R2 =0.9825; p=3.7422E-37, B=-1.488				
#1	2.323	-10.26	22.89	13.50
#2	2.323	-10.26	22.89	13.50
#3	1	4	2	3
#4	9.37E-11	3.46E-3	8.27E-10	4.54E-5

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

#5	0.2733	3.316	2.924	2.976
TWOLF, R2=0.9989; p=6.1990E-65, B=-1.554				
#1	2.561	-10.03	13.22	13.02
#2	2.561	-10.03	13.22	13.02
#3	4	1	3	2
#4	2.19E-4	4.36E-28	5.18E-12	7.34E-22
#5	0.6358	0.3914	1.415	0.7257
EON, R2=0.4870; p=3.7571E-11, B=3.892				
#1	-1.027	-8.365	-8.416	9.291
#2	-1.083	-8.819	-1.365	3.921
#3	X	X	1	2
#4	1.70E-1	1.67E-1	2.84E-4	3.24E-2
#5	0.7820	6.318	3.602	1.802
0	1	2	3	4
GAP, R2=0.8570; p=3.4602E-17, B=-1.175				
#1	1.934	9.573	-2.384	-1.126
#2	1.930	-1.660	6.969	-1.630
#3	2	X	1	X
#4	1.22E-2	2.91E-1	1.67E-7	2.86E-1
#5	0.7318	1.549	1.084	1.506
VPR, R2=0.9973; p=1.7058E-53, B=-1.509				
#1	3.278	-30.38	36.78	32.46
#2	3.278	-30.38	36.78	32.46
#3	4	1	3	2
#4	2.43E-3	1.96E-26	8.09E-15	4.09E-22
#5	1.016	1.245	3.140	1.721
PERLBMK, R2=0.9724; p=2.2952E-31, B=-1.707				
#1	2.240	7.075	11.50	-4.965
#2	2.231	1.694	17.08	-4.965
#3	3	2	1	X
#4	9.45E-7	1.54E-17	2.30E-19	1.97E-1
#5	0.3889	0.1200	1.075	3.783
GCC, R2=0.8309; p=1.1463E-15, B=3.034				
#1	-0.9356	-11.67	3.241	11.46
#2	-1.973	-0.4808	-13.36	-0.1456
#3	2	X	1	X
#4	4.56E-2	3.56E-1	1.95E-14	7.83E-1
#5	0.9592	0.5158	1.196	0.5253
BZIP2, R2=0.9792; p=4.5876E-33, B=3.019				
#1	-0.6081	-42.07	40.28	41.25
#2	-0.6081	-44.24	41.81	43.12
#3	X	1	3	2
#4	2.66E-1	4.62E-23	1.66E-17	1.40E-22
#5	0.5390	2.172	2.928	2.180
MCF, R2=0.9935; p=6.6573E-40, B=-0.1367				
#1	3.862	-48.42	41.50	51.41
#2	3.652	-1.771	-18.63	-0.8507
#3	2	X	1	X
#4	5.19E-4	1.74E-1	6.83E-31	5.54E-1
#5	9.9649	1.279	0.5384	1.423

Similarly, one can evaluate all the other tests. For example, in the test 300.twolf – computer aided design of integrated circuits – the most dependent volume cache L1, by increasing its volume test performance increases, as confirmed by a negative value in a regression coefficient (1).

The rest of cache and TLB affect performance negatively, reducing their productivity decline, talking about these positive regression coefficients (1).

The output for this application is as follows: increase L1 and try to find the optimal value of the cache with the objective function representing the performance of the system.

On the basis of experiments, it can be concluded that various classes of applications require different approaches to use caches.

While performing the application, if it doesn't change the capacity of cache, at least it should stop some caches which lower performance and efficiency.

Early models of Intel allowed doing so, in all probability, return to this practice.

In the future, the authors suggest, it is needed to experiment with other contemporary models of Intel to get a fuller picture of the usefulness of the cache.

REFERENCES

[1] A. Tanenbaum, *Modern Operating Systems*, 2nd ed., New Jersey, USA: Prentice Hall, 2001.

Asian Journal of Information Science and Technology (AJIST)
Vol1.No.2 2013 pp 1-9.
available at: www.goniv.com
Paper Received :15-10-2013
Paper Published:18-12-2013
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

[2] J. S. Liptay, "Structural aspects of the system/360 model 85," *IBM Systems Journal, Part II: The Cache*, vol. 7, pp. 15–21, 1968.

[3] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 5nd ed., Philadelphia, USA: Elsevier, 2011.

[4] A. J. Smith, "Cache memories," *Journal ACM Computing Surveys*, vol. 14, no. 3, pp. 473–530, 1982.

[5] A. Agraval and J. Hennessy, *ACM Transactions on Computer Systems*, 7nd ed., vol. 7, pp. 864–871, N2, 1989.

[6] A. Phansalkar, A. Joshi, L. Eeckhout and L. K. John, "Measuring program similarity: Experiments with SPEC CPU benchmark suites," *Proc.IEEE International Symposium on Performance Analysis of Systems and Software*, 2005, Austin, TX, pp. 10–20.

[7] O. A. V. ElMostapha, J. Woodli, C. Yount and K. A. Doshi, "On the comparison of regression algorithms for computer architecture performance analysis of software applications," *Proc. IEEE International Symposium on Performance Analysis of Systems and Software*, 2008, pp. 179–190.

[8] J. K. Rai, A. Negi, R. Wankar and K. D. Nayak, "Characterizing L2 cache behavior of programs on Multi–core processors: Regression models and their transferability," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 2, pp. 212–221, 2010.

[9] P. J. Joseph, K. Wasvani and M. J. Thazhuthaveetil, "Construction and use of linear regression models for processor performance analysis," *The Twelfth International Symposium on High–Performance Computer Architecture*, 2006, pp. 99–108.

[10] G. E. Suh, S. Devadas and L. Rudolph, "Analytical cache models with applications to cache partitioning," *International Conference on Supercomputing* ,Sorrento, Italy, 2001, pp. 1–12.

[11] Y. Zheng, T. Brian and D. M. Jordan, "Performance evaluation of exclusive cache hierarchies," *IEEE International Symposium on Performance Analysis of Systems and Software*, 2004, pp. 89–96.

[12] K. Kasperski, *Technique of Program Optimization*, 1st ed., Saint Petersburg: A–List Publishing, 2003.

[13] (2000) The SPEC website. [Online].Available: http://www.spec.org/.

[14] (2013) The Intel website. [Online].Available: http://software.intel.com/ en–us/intel–vtune–amplifier–xe.

[15] (2013) The Intel website. [Online].Available: http://www.intel.com/content/www/us/en/architecture–and–technology/64–ia–32–architectures–software–developer–vol–3b–part–2–manual.html.

[16] A. A. Afifi and S. P. Azen, *Statistical Analysis: A Computer Oriented Approach*, New York, USA: Academic Press, 1979.