ISSN 1512-1232

TCT ANALYSIS OF 0.2 NS CORE CLOCK SERIES PROCESSORS

Abdul Kareem. P, R. A. Singh
Department of Physics and Electronics, Dr H S Gour University, Sagar 470003, India
E-mail: kareemskpa@hotmail.com

Abstract:

0.2 ns clock cycle processors are most advanced microarchitecture processors. These processors are useful in advanced scientific computing. Nano Technology lead today's semiconductor technology into new era. In this paper we describe the statistical analysis of TCT (Task Completion Time) of 0.2 ns clock cycle processors. We used recently published SPEC CPUint2006 benchmark scores and classified benchmark programs into four subgroups. High performance processor is a great challenge to all microprocessor manufacturers and scientists. To day we need a high processor which can boost for a broad spectrum of application area. We use statistical analysis techniques, Principal Component Analysis (PCA) and Cluster Analysis (CA) for the study of TCT and benchmark workload classification using recently published SPEC CPUint2006 performance numbers of twenty three most advanced commercial processors. We calculated five most significant PCs, which are retained for 79.9% of the variance, PC2, PC3, PC4 and PC5 covers 13.9%, 3.3%, 1.2% and 0.5% variance respectively. We classified the CINT benchmarks in four sub groups. We found that the 400.perlbench and 483.xalancbmk exhibits higher memory wait time. Our results and analysis can be used by performance engineers, scientists and developers to better understand the benchmark workload and select input dataset for better microarchitecture design of the processors.

Keywords: *SPEC CPU, Performance, TCT.*

1. INTRODUCTION

0.2 ns clock core processors are most advanced hihg performance processors. These processor contain more than one billion transistors with high switching speed. According to moore's law the device density of the processors is doubled in every 18 months. Semiconductor researchers are reducing the gate length of CMOS for getting more performance. International Technology Roadmap for Semiconductor devices (ITRS-2006), the gate length is 32 nm in 2005 and expecting less than13nm up to 2013. SPEC, the Standard Performance Evaluation Corporation released the long awaited SPEC CPU2006 on August 24, 2006. SPEC is a non-profit organization formed in 1988. SPEC's CPU benchmarks have been the worldwide standard for measuring compute-intensive performance since their introduction in 1989. The firstly released SPEC CPU benchmark suite is a collection of ten compute-intensive benchmark programs. Now the recently released SPEC CPU 2006 benchmark suite consists of upgraded previous benchmarks. SPEC CPU 2006 contains two components that focus on two different type of compute-intensive performance. The first suite (CINT 2006) measures compute-intensive performance, second suite (CFP 2006) measures compute-intensive floating point performance. The SPEC CPU2006 benchmark suite comprises of 12 CINT2006 based on real applications and 17 CFP2006 benchmarks written in C, C++, and various FORTRAN versions, as well as C/FORTRAN [1].

In this paper we presented statistical analysis of 0.2 ns clock core processors by using Linear Regression, Principal Component Analysis (PCA) and Cluster Analysis. We analyze the performance characterization of the processors by using recently published SPEC CPU2006 benchmarks scores.

ISSN 1512-1232

The rest of the paper is organized as follows. In section 2 we describe SPEC CPU2006 benchmarks. Section 3 presents the methodology used in this paper. The results of our analysis done in this paper are discussed section 4.

2. SPEC CPU 2006 BENCHMARKS

As stated in section 1 the SPEC CPU2006 suite contains 17 floating point compute-intensive programs (Some programs are written in C and some in FORTRAN) and 12 integer programs (8 written in C and 4 written in C++). Table.1 and Table 2 provide a complete description of the benchmarks in SPEC CPU2006 suite. The SPEC CPU2006 benchmarks replace the SPEC89, SPEC92, SPEC95 and SPEC CPU 2000 benchmarks [2] [3] [4]. SPEC benchmarks cover a broad spectrum of the field. To validate the reduced input sets they propose in MinneSPEC, A.J. KleinOsowski and David Lilja [5] performed a chi-square analysis of each set's function level execution profiles.

S. No	Integer Benchmark	Language	Description
1	400.perlbench	C++	PERL Programming Language
2	401.bzip2	C	Data Compression
3	403.gcc	C	C Language Optimizing Compiler
4	429.mcf	C	Combinatorial Optimization
5	445.gobmk	C	Artificial Intelligence : Game Playing
6	456.hmmer	C	Search a Gene Sequence Database
7	458.sjeng	C	Artificial Intelligence : Chess
8	462.libquantum	C	Physics / Quantum Computing
9	464.h264ref	C	Video Compression
10	471.omnetpp	C++	Discrete Event Simulation
11	473.astar	C++	Path – Finding Algorithm
12	483.xalancbmk	C++	XSLT Processor

Table 1: The CINT 2006 Suite Benchmarks

S. No	Floating Point Benchmark	Language	Description
1	410.bwaves	Fortran – 77	Computational Fluid Dynamics
2	416.gamess	Fortran	Quantum Chemical Computations
3	433.milc	C	Physics / Quantum Chromo Dynamics
4	434.zeusmp	Fortran – 77	Physics / Magneto Hydro Dynamics
5	435.gromacs	C/Fortran	Chemistry / Molecular Dynamics
6	436.cactusADM	C / Fortran-90	Physics / General Relativity
7	437.leslie3d	Fortran – 90	Computational Fluid Dynamics
8	444.namd	C++	Scientific, Structural Biology, Classical Molecular Dynamics Simulation.
9	447.dealII	C++	Solution of Partial Differential Equations using the Adaptive Finite Element Method.
10	450.soplex	C++	Simplex Linear Programming Solver
11	453.povray	C++	Computer Visualization / Ray Tracing
12	454.calculix	C/Fortran-90	Structural Mechanics
13	459.GemsFDTD	Fortran-90	Computational Electromagnetic
14	465.tonto	Fortran-95	Quantum Crystallography
15	470.lbm	C	Computational Fluid Dynamics
16	481.wrf	C/Fortran – 90	Weather Processing
17	482.sphinx3	C	Speech Recognition

Table 2: The CFP2006 Suite Benchmarks

ISSN 1512-1232

3. METHODOLOGY

To analyze the performance of 0.2 ns clock cycle processors we have used recently published SPEC CPUint2006 benchmark scores data of 600+ commercial processors and selected twenty three high performance processors in this series for our analysis. Each benchmark runs on these machines three times. There are 12 performance numbers, one per each benchmark for twenty three most advanced commercial machines. We reported the scaling of processor performance in some modern Intel's processors using linear regression analysis [6].

We use statistical data analysis techniques called Principal Component Analysis (PCA) and Cluster Analysis (CA) to analyze the benchmark workload. These results are discussed in section 4. For this analysis we used a commercial software package STATISTICA [7] for statistical computation. Lieven Eeckhout, Hans Vandierendonck and Koen De Bosschere, proposed a method for Designing Computer Architecture Research Workloads [8] using SPEC CPU benchmarks.

3. 1. PRINCIPAL COMPONENT ANALYSIS

Principal components analysis (PCA) is a statistical data analysis technique that builds on the assumption that many variables are correlated and hence measure the same or similar properties of the program-input pairs [9] [10] [11]. PCA computes principal components: new variables that are linear combinations of the original variables such that all principal components are uncorrelated. PCA transforms the p variables X_1, X_2, … , X_p into p principal components Z_1, Z_2, … , Z_p with

$Z_i = \sum_{i}^{p} a_{ij} X_j$, This transformation has the properties

- ❖ $Var[Z_1] > Var[Z_2] > \ldots > Var[Zp]$, which means that Z_1 contains the most information and Z_p the least; and
- ❖ $Cov[Z_i, Z_j] = 0$, $i \neq j$, which means that there is no information overlap between the principal components.

The total variance in the data remains the same before and after the transformation, namely

$$\sum_{i=1}^{p} Var[X_i] = \sum_{i=1}^{p} Var[Z_i]$$

3.2. CLUSTER ANALYSIS

Cluster analysis (CA) is first used by Tryon in 1939 to encompass a number of different classification algorithms. CA aims the number of benchmarks programs exhibits similar behavior. CA is classified in two types, first linkage clustering and second K-means clustering. The graphical representation of each similar and dissimilar benchmarks programs using linkage distance is called dendrogram. We use linkage cluster analysis to identify similar and dissimilar benchmark behavior [12] [13].

4. RESULTS AND DISCUSSIONS

Figure 1 show the important results obtained from our SPEC CPU int 2006 benchmarks analysis. Figure 1 (a) shows the variation in TCT over 0.2 ns clock cycle processors. A tradeline is extended which cuts the TCT axis at 1564.1 Sec. Figure 1 (b) and (c) shows the higher and lower TCT with clock core cycle of 0.2 ns series processors. Their tradeline is fitted with good R^2 value.

ISSN 1512-1232

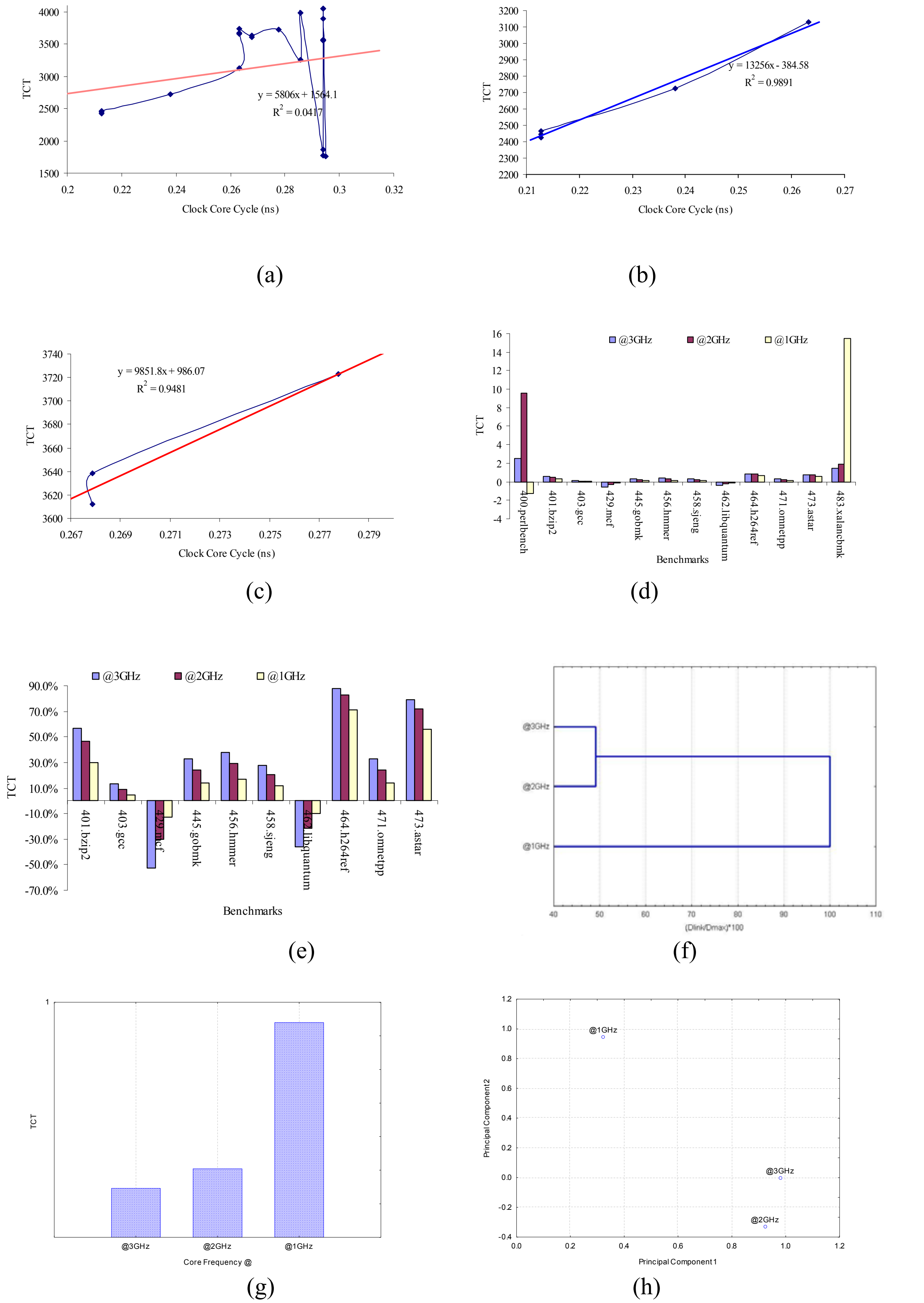


Figure 1: (a) Variation of TCT of 0.2 ns clock cycle processors. **(b)** and **(c)** represents high and low TCT of 0.2 ns clock cycle processors with tradeline respectively. **(d)** and **(e)** represents variation of TCT of 0.2 ns clock cycle processors @1GHz, 2GHz and 3 GHZ respectively. **(f)** Dendogram showing the relation between execution of program @1GHz, 2GHz and 3 GHZ. **(g)** and **(h)** represents TCT and principal component plot of @1GHz, 2GHz and 3 GHZ frequency respectively.

ISSN 1512-1232

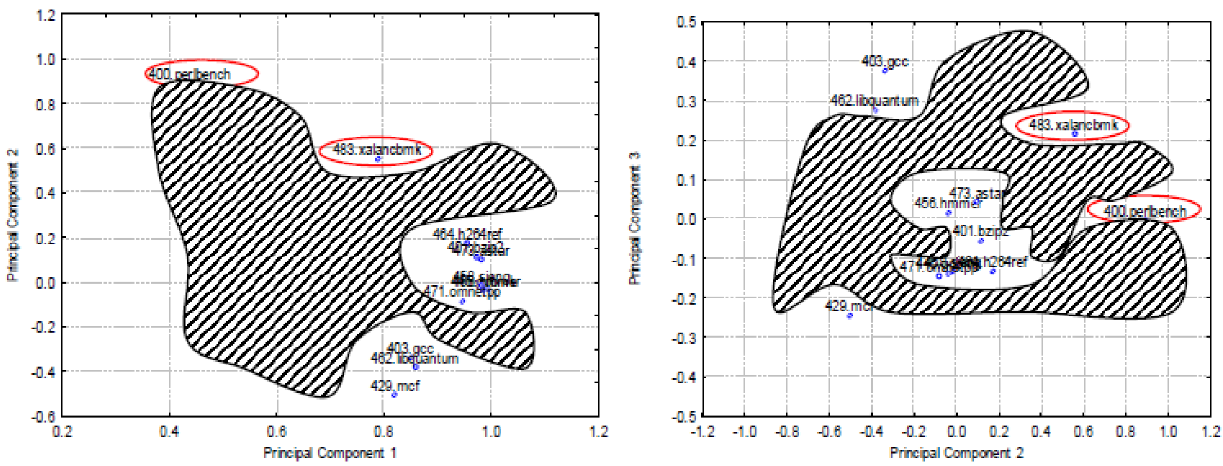
We calculated the variation of task completion time @1GHz, 2GHz and 3 GHZ processor frequency of 0.2 ns clock cycle series processors, which is shown in figure 1.(d) & (e). The benchmarks 400.perlbench and 483.xalancbmk shows high TCT as compare to other benchmarks. The complete variation of task completion time in this series is shown in table 2. Figure 1 (f) shows the dendogram of processor frequency with linkage distance, from (g) it is clear that 1GHz shows high task completion time. Figure 1 (h) shows the frequency in principal component space. Depending on TCT we classified benchmark workload into four subgroups as mentioned in table 4.

Benchmarks	**Memory wait time, % of TCT**		
	@3GHz	**@2GHz**	**@1GHz**
401.bzip2	56.4%	46.3%	30.2%
403.gcc	13.2%	9.2%	4.8%
429.mcf	-52.6%	-29.8%	-13.0%
445.gobmk	32.6%	24.4%	13.9%
456.hmmer	37.9%	28.9%	16.9%
458.sjeng	28.0%	20.6%	11.5%
462.libquantum	-35.9%	-21.4%	-9.6%
464.h264ref	87.9%	83.0%	70.9%
471.omnetpp	32.5%	24.3%	13.8%
473.astar	79.3%	71.8%	56.0%

Table 3: The complete description of Benchmarks Memory wait time @1GHz, 2GHz and 3GHz frequency in 0.2 clock cycle series

Sub Groups	**Benchmarks**
Sub Group 1	403.gcc,429.mcf and 462.libquantum
Sub Group 2	445.gobmk. 456.hmmer, 458.sjeng and 471.omnetpp
Sub Group 3	401.bzip2, 464.h264ref and 473.astar
Sub Group 4	400.perlbench and 483.xalancbmk

Table 4: Classification of Benchmarks into sub groups depending on TCT.



(a) (b)

ISSN 1512-1232

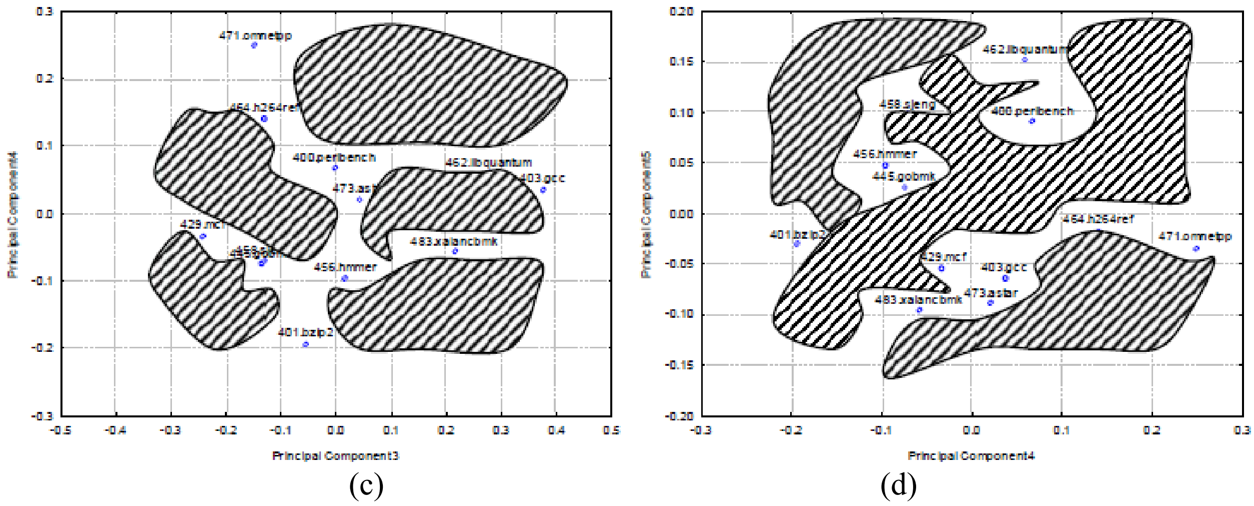
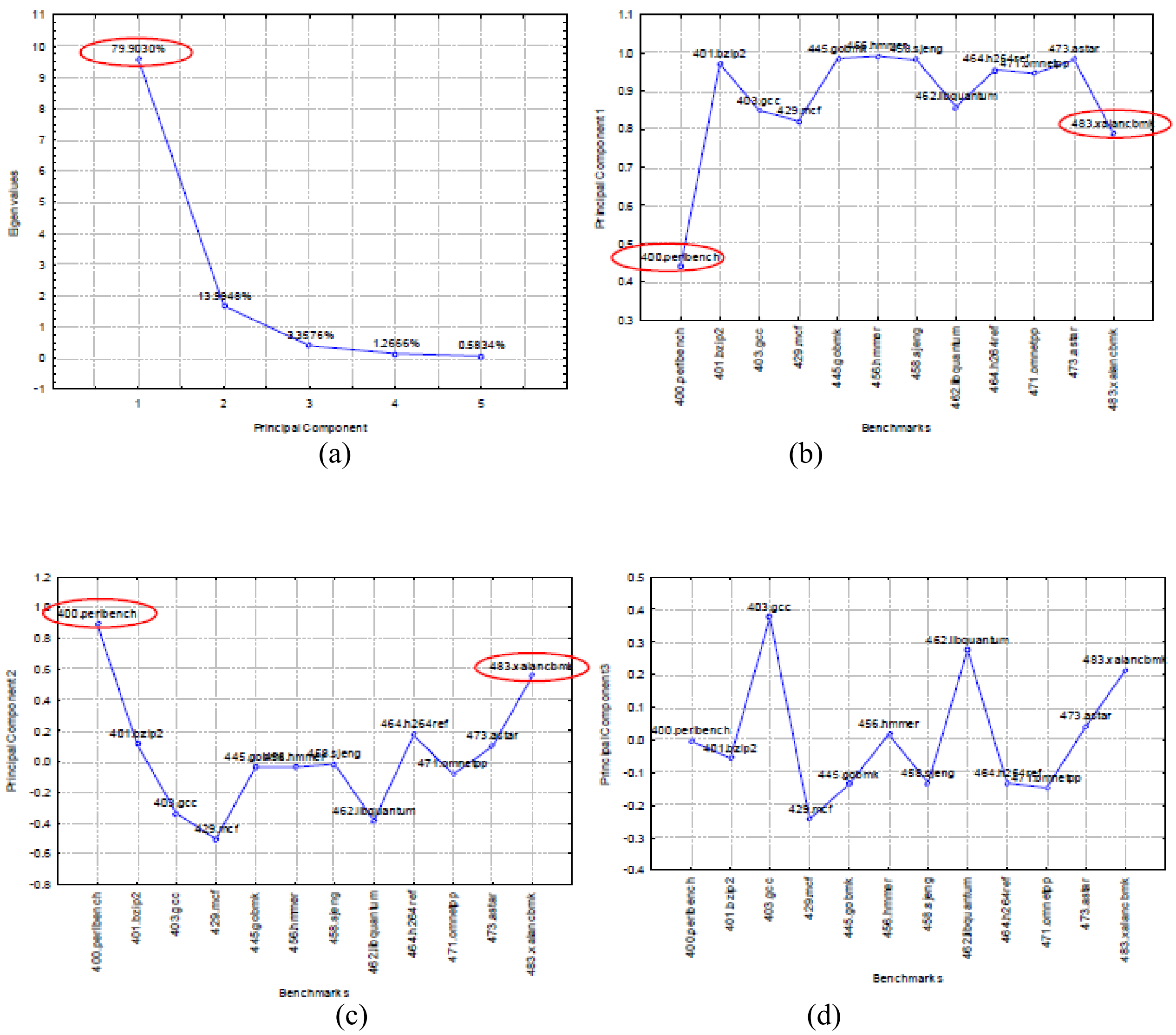
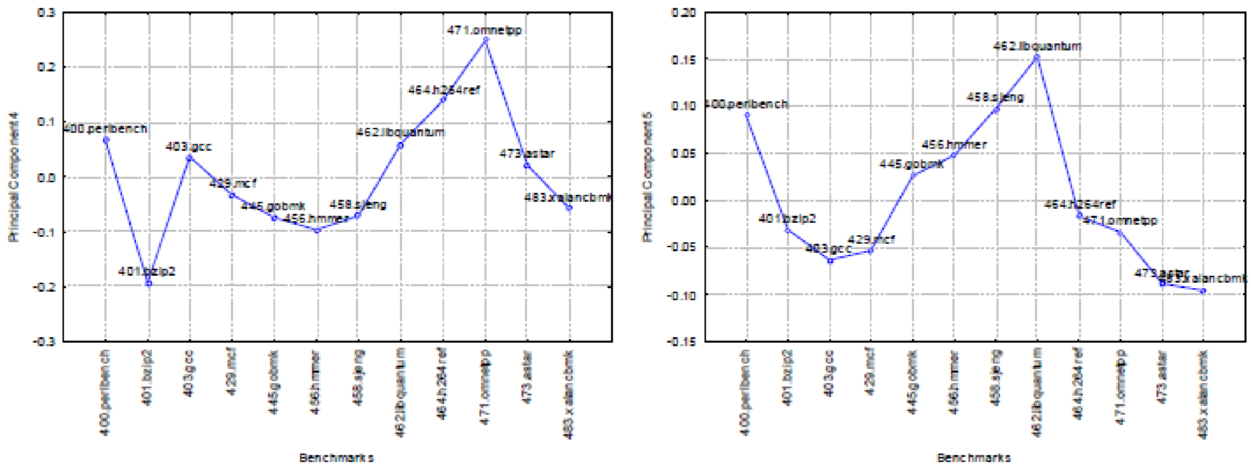


Figure 2: (a), (b), (c) and **(d)** represents Principal Components of the benchmark in memory space





(e) (f)

Figure 3: (a) Represents the variance in the five significant Principal Components. **(b)-(f)** Presents the variation of individual Principal component score corresponding to each benchmark

Figure 2 shows the variation of benchmarks in principal components memory space, which shows the dissimilar behavior of the benchmarks. In the PC workload space of first two principal components i.e. PC1 and PC2, there are some weak spots. These points are highlighted through a mesh shape, similarly in figure 2.(b)-(d). these weak spots is provide a valuable information for performance analyst and computer architects. Figure 3 (a) explains the variance in most significant principal components. PC1 retained for 79.9% variance. In figure3 (b)-(f) shows the variation of individual benchmark in five significant PCs. Dendogram in figure 4 shows the similarities and dissimilarities of the benchmark in workload space, which is useful to reduce the benchmark workload for future architecture design, by drawing a line linkage distance at 600 we can select seven benchmarks for microarchitecture design, so we can reduce the simulation time. Improper selection of benchmarks programs may not accurately describe the true performance of a processor design. It is observed that some benchmark programs have same behavior as other benchmarks; they only increase the simulation time without providing any extra information. Some benchmarks are connected to the other benchmarks trough a long linkage distance, i.e. 462.libquantum. This benchmark is much larger than the linkage distance for more strongly clustered input pairs. As compared to previous generation processors the 0.2 ns clock cycle processors are more useful to 403.gcc, 429.mcf and 462.libquantum application areas. Our results and analysis can be used by performance engineers, scientists and developers to better understand the benchmark workload and select input dataset for better microarchitecture design of the processors.

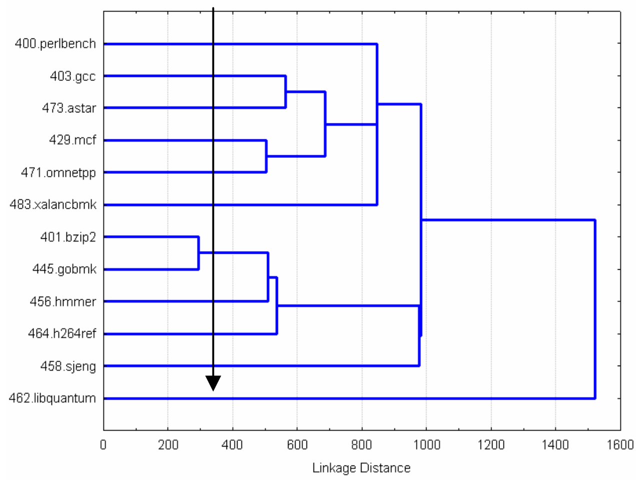


Figure 4: Dendogram showing the similarities and dissimilarities in workload space of 0.2 ns core cycle processors.

ISSN 1512-1232

5. DISCLAIMER

All the observations and analysis done in this paper on SPEC CPUint2006 Benchmarks are the author's opinions and should not be used as official or unofficial guidelines from SPEC in selecting benchmarks for any purpose. This paper only provides guidelines for performance engineers, academic users, scientists and developers to better understand the benchmark workloads and selection of input data sets for computer architecture simulation research.

6. ACKNOWLEDGEMENTS

One of the authors (A.K.P.) gratefully acknowledges financial support of UGC, New Delhi, INDIA, for a Meritorious Research Fellowship.

ISSN 1512-1232

7. REFERENCES

[1]. Standard performance evaluation corporation (SPEC), Benchmarks, http://www.spec.org. http://www.spec.org/cpu2006/results/

[2]. John L. Henning. SPEC CPU2000: Measuring CPU performance in the new millennium. IEEE Computer, July 2000.

[3]. SPEC CPU2000 Press Release FAQ, available at http://www.spec.org/osg/cpu2000/press/faq.html

[4]. A. KleinOsowski and D. Lilja, MinneSPEC: A new SPEC benchmark workload for simulation based computer architecture research. Computer Architecture Letters, Volume 1, June 2002.

[5]. A.J. KleinOsowski and D.J. Lilja, "MinneSPEC: A New SPEC Benchmark Workload for Simulation Based Computer Architecture Research," Computer Architecture Letters, June 2002, pp. 10-13.

[6]. Abdul Kareem P and R. A. Singh, "Performance Scaling of Individual SPECint2006 results for Intel Xeon", e-JST, vol.4, no.2, pp.55-60, April 2009.

[7]. StatSoft, Inc. Commercial statistical software, STATISTICA for Windows, 2006. http://www.statsoft.com

[8]. Lieven Eeckhout, Hans Vandierendonck and Koen De Bosschere, "Designing Computer Architecture Research Workloads," IEEE Computer Society, February 2003, pp.65-71.

[9]. B.F.J. Manly, Multivariate Statistical Methods: A Primer, 2nd ed., Chapman & Hall, 1994.

[10]. I.T. Jolliffe, Principal Component Analysis, 2nd ed., Springer, 2002.

[11]. NIST/SEMATECH e-Handbook of Statistical methods, in http://www.itl.nist.gov/div898/handbook/

[12]. B. F. J.Manly. Multivariate Statistical Methods: A primer. Chapman & Hall, second edition, 1994.

[13]. StatSoft, Inc. STATISTICA for Windows. Computer program manual. 1999. http://www.statsoft.com

Article received: 2009-06-29