Principal Component and Cluster Analysis of SPEC CPUint2006 Benchmarks: Input Data set Selection

Abdul Kareem P and R. A. Singh

Department of Physics and Electronics, Dr H S Gour University, Sagar, India, 470003
Email: kareemskpa@hotmail.com

***Abstract* – Technological scaling of processor parameters has a critical limit. The Scaling advanced CMOS technology to the next generation effects improves performance, increases transistor density, and reduces power consumption of the processor. In this paper we describe the statistical analysis of SPEC CPUint2006 benchmarks workload and input data selection for microarchitectural research. Today we need a processor which can provide high performance boost for a broad spectrum. We use statistical analysis techniques, Principal Component Analysis (PCA) and Cluster Analysis (CA) for the study of benchmark workload classification using recently published SPEC CPUint2006 performance numbers of thirty Intel's commercial processors. We calculated five most significant PCs, which are retained for 85% of the variance, PC2, PC3, PC4 and PC5 covers 11.1%, 2.9%, 0.6% and 0.1% variance respectively. We classified the CINT benchmarks in two sub groups. We found that the benchmarks 471.omnetpp, 462.libquantum 403.gcc, and 429.mcf 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:* PCA, SPEC CPU2006, Processor Performance, Moore's Law**

1. INTRODUCTION

Processors perennially become more powerful in their performance. Today the processors are shifted from 32-bit to 64-bit environment. 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. On June 30, 2000, SPEC retired the CPU95 benchmark suite. Its replacement is CPU2000, a new CPU benchmark suite with 19 applications that have never before been in a SPEC CPU suite. 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 [1]. 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 study we have used thirty commercial processors of Intel. These processors are having IA-32s new microarchitectural features including a 400MHz system bus, hyper pipelined technology, advanced dynamic execution, rapid execution engine, advanced transfer cache, execution trace cache, and Streaming Single Instruction, Multiple Data (SIMD) Extensions 2 (SSE2).

1.1 SCOPE OF THE STUDY

The statistical analysis presented in this paper examines the scaling of performance in some Intel series processors which are fabricated for the requirement of the modern generation utility. Furthermore, contrary to prior work we not only quantify the performance prediction of the processors, but also have evaluated scalability of the Memory Wait Time which degraded the performance of the processor by using a simple statistical correlation technique. This analysis is useful to performance engineers, scientists and developers to better understand the performance scaling in modern generation processors. In this paper we apply statistical analysis techniques such as Linear Regression, Principal Component Analysis (PCA) and Cluster Analysis to analyze the workload characterization of SPEC CPU2006 benchmarks.

The rest of the paper is organized as follows. In section 2 we describe the growth of device density in modern generation processors. We describe SPEC CPU2006 benchmarks in section 3 and the analysis of SPEC CPU2006 benchmarks in section 4. Section 5 presents results of our analysis done using Principal Component analysis and Cluster Analysis. Finally section 6 contain summary of the results.

2. GROWTH OF DEVICE DENSITY IN PROCESSORS

The performance of modern processors is rapidly increasing as both clock frequency and the number of transistors required for a given implementation grow. Moore's Law says that the device density of the processor double in every 18 months. Figure 1 shows the transistor count per die of processors introduced by Intel over the past 35 years [2] [3] [4]. Today's processor contains approximately one billion transistors.

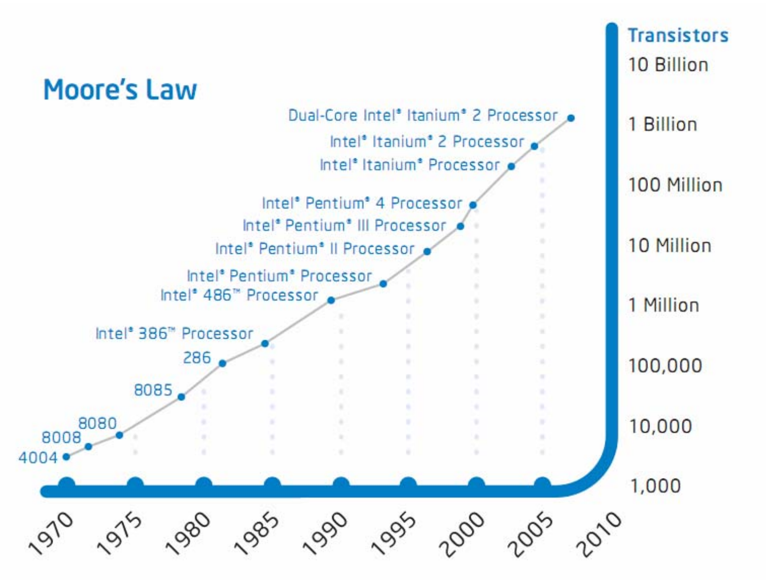


Figure.1: Scaling transistors. The number of transistors is expected to continue to double about every two years, in accordance with Moore's Law. Over time, the number of additional transistors will allow designers to increase the number of cores per chip. [Source from [3]]

3. SPEC CPU BENCHMARKS

Benchmarks are used for the performance evolution of the processors. There are different types of benchmarks available. Among all SPEC, HINT, and TPC are most important and popular benchmarks for performance evolution. SPEC is a nonprofit corporation formed to establish, maintain, and endorse a standardized set of 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 [5] [6] [7].

Table 1: The CINT 2006 Suite Benchmarks

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 2: The CFP2006 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

4. ANALYSIS OF SPEC CPU2006 BENCHMARKS

4.1 METHODOLOGY

To analyze the benchmarks, we have used recently published SPEC CPUint2006 benchmark scores of thirty commercial Intel Xeon, Intel Xeon Quad Core, and Intel Xeon Dual Core processors. The performance numbers of these processors collected on the 64-Bit SUSE LINUX Enterprise Server 10 SP1 RC1 Operating System. Each benchmark runs on these machines three times. There are 12 performance numbers, one per each benchmark for thirty most advanced commercial machines.

We reported the scaling of processor performance in these processors using linear regression analysis [8] to study the performance scaling in Intel processors. The results are discussed in section 5. We use statistical data analysis techniques called Principal Component Analysis (PCA) and Cluster Analysis (CA) to analyze the benchmark workload. These results are also discussed in section 5. For this analysis we used a commercial software package STATISTICA [9] for statistical computation.

4. 2 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 [10] [11] [12].

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 $X_1, X_2, \dots, X_p$ into p principal components $Z_1, Z_2, \dots, Z_p$ with $Z_i = \sum_{i}^{p} a_{ij} X_j$, This transformation has the properties

- ❖ $Var[Z_1] > Var[Z_2] > \dots > Var[Z_p]$, 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

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

4.3 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 [13] [14].

5. RESULTS AND DISCUSSIONS

5.1 LINEAR REGRESSION ANALYSIS

Table 3 explains the summary of computation done in this statistical analysis, by using 12×30 benchmark performance matrix [1]. We have calculated Memory wait time @1GHz, @2GHz and @3GHz processor frequency [15].

Table 3: Memory Wait Time @1GHz, 2GHz and 3GHz processor frequency.

Benchmark	RESULTS			Memory wait time, % of TCT		
	A	B	R^2			
	Slope	Intercept	RSQ	@3GHz	@2GHz	@1GHz
400.perlbench	1430.91	23.2	97.3%	4.7%	3.1%	1.6%
401.bzip2	1944.96	-31.1	95.3%	-5.1%	-3.3%	-1.6%
403.gcc	993.89	199.1	49.0%	37.5%	28.6%	16.7%
429.mcf	687.91	227.7	74.8%	49.8%	39.8%	24.9%
445.gobmk	1593.94	6.0	99.2%	1.1%	0.7%	0.4%
456.hmmer	1745.58	26.6	93.2%	4.4%	3.0%	1.5%
458.sjeng	1910.34	53.0	99.1%	7.7%	5.3%	2.7%
462.libquantum	494.76	223.3	1.2%	57.5%	47.5%	31.1%
464.h264ref	2194.51	-16.1	99.2%	-2.3%	-1.5%	-0.7%
471.omnetpp	686.73	219.6	74.3%	49.0%	39.0%	24.2%
473.astar	1368.30	33.6	96.1%	6.9%	4.7%	2.4%
483.xalancbmk	732.52	37.0	94.0%	13.2%	9.2%	4.8%

Each individual time scales are in accord with the general observation, or execution time = Ax + B, where x is the Core Clock Cycle in ns, A is the slope, and B is the intercept, then the geometrical mean of all 12 times will be a rather complex transcedental function.

Table 3 shows the variation of Memory Wait Time with benchmark. All individual trends were broken into two categories. First category contains individual tasks where the "memory wait time" (MWT) is very small of the total individual run time. Eight individual tasks fall into the first category. The second group contains four individual tasks where the MWT is grater than zero and above. The second group contains four bench mark programs 471.omnetpp, 462.libquantum 429.mcf, and 403.gcc.

The classification of the benchmarks in to sub groups is shown in Table 4. Benchmark 471.omnetpp and 462.libquantum shows maximum memory wait time with R^2=1.2%, which is the worst fitting value.

Table 4: Classification of SPEC CINT2006 Benchmark programs in to subgroups.

Classification	Benchmarks
Subset of Eight programs	400.perlbench,464.h264ref, 401.bzip2, 445.gobmk, 473.astar, 458.sjeng, 456.hmmer, and 483.xalancbmk
Subset of four programs	471.omnetpp, 403.gcc, 429.mcf, and 462.libquantum

The scaling of task completion time @1GHz, @2GHz and @3GHz frequency of processor for 12 benchmark programs is shown in Figure 2, benchmark 462.libquantum shows maximum task completion time of all benchmark programs over @1GHz, @2GHz and @3GHz.

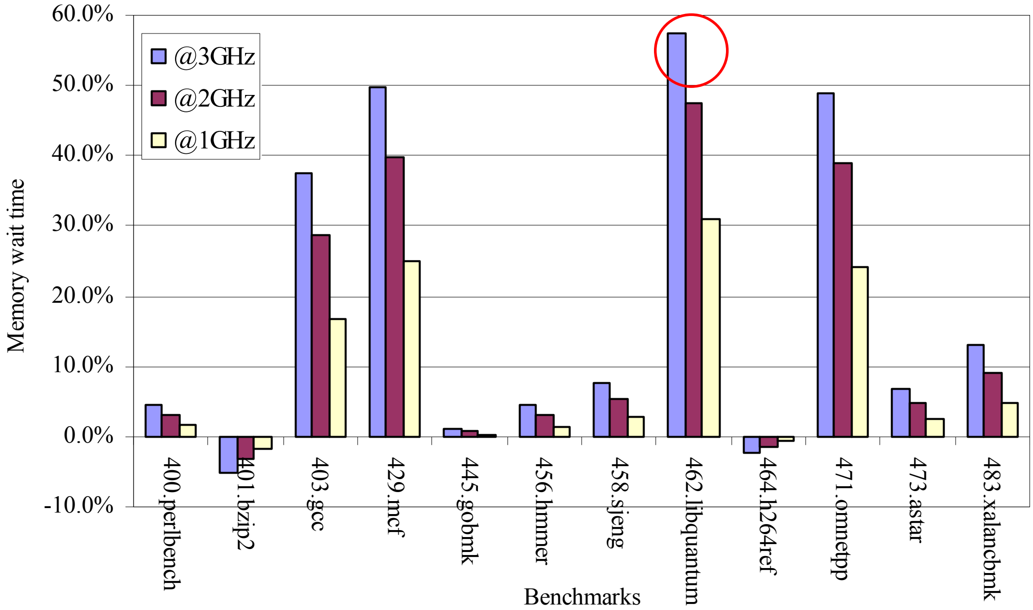


Figure 2: The comparison of normalized task completion time @1GHz, @2GHz and @3GHz processor frequency.

5.2 PRINCIPAL COMPONENT ANALYSIS

The analyses of principal components results are discussed in this section. We generated five significant principal components-PCs using benchmark workload and commercial statistical simulation software STATISTICA v.7 [14]. Five principal components are retained for 85% of the variance. Figure 3 shows the summary of variance estimated in the benchmark workload, PC2, PC3, PC4 and PC5 holds 11.1% , 2.9%, 0.6% and 0.1% variance respectively.

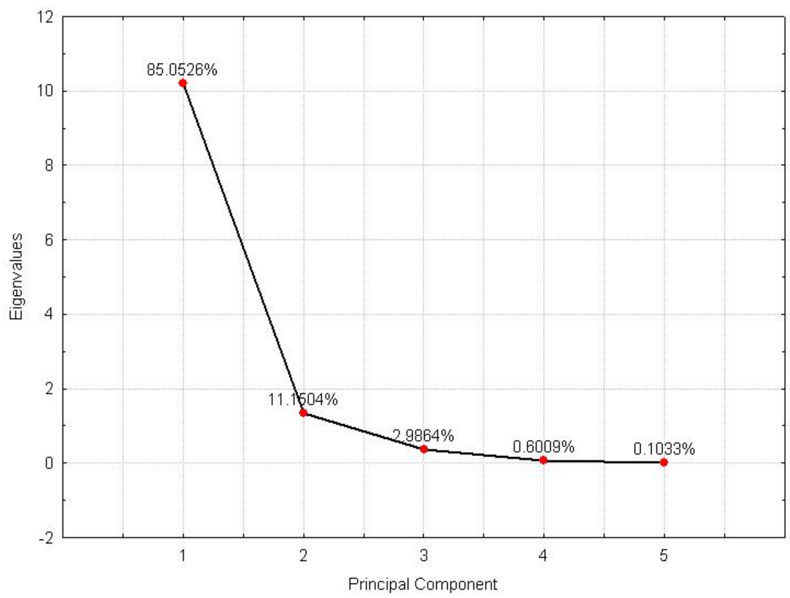


Figure 3: Eigenvalues scree plot of all principal components, which explain the variance in the workload (PC1 to PC5).

Figure 4 shows the scatter plot of first two PCs, i.e. PC1 vs. PC2. Figure 5 to Figure 7 shows the scatter plot of PC2 vs. PC3, PC3 vs. PC4, and PC4 vs. PC5 respectively. In all PCs space the benchmark 462.libquantum is more deviated as compared to other benchmark. Figure 8 to Figure 10 explains the variation of individual principal components for each benchmark. They show the dissimilar behavior of the benchmarks, 429.mcf, 471.omnetpp, 403.gcc and 462.libquantum. This information is useful in selecting input data set for the implementation of processor architecture.

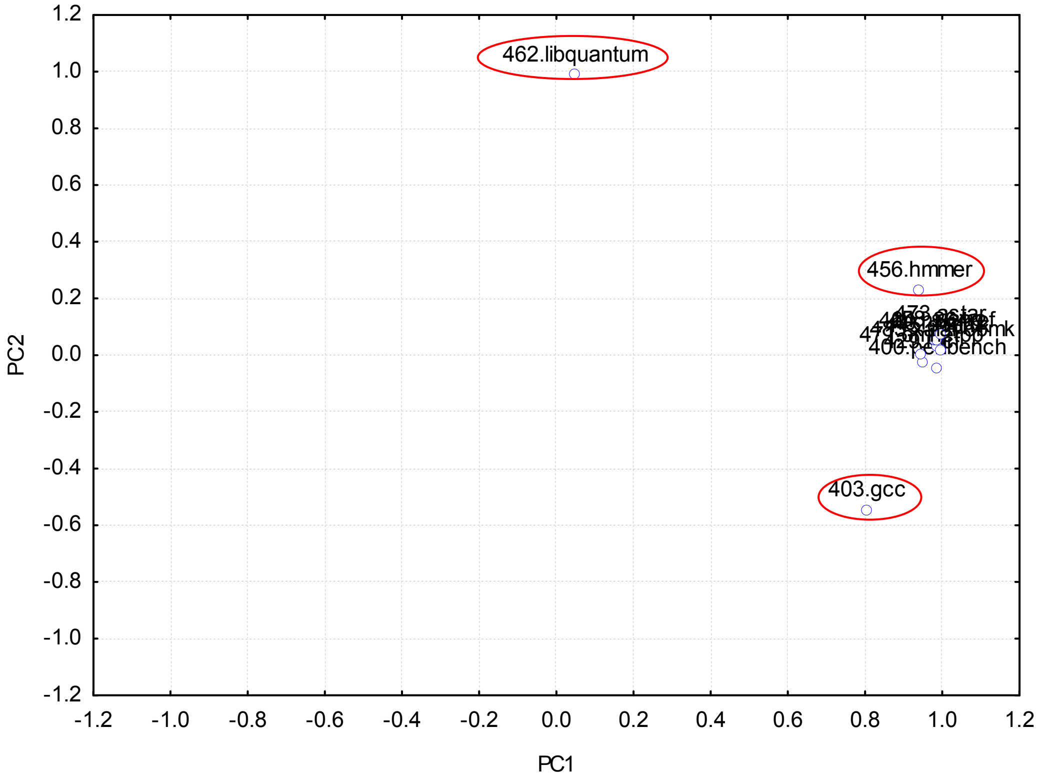


Figure 4: SPEC CINT 2006 programs plotted in the PC space using memory access characteristics (PC1 vs. PC2).

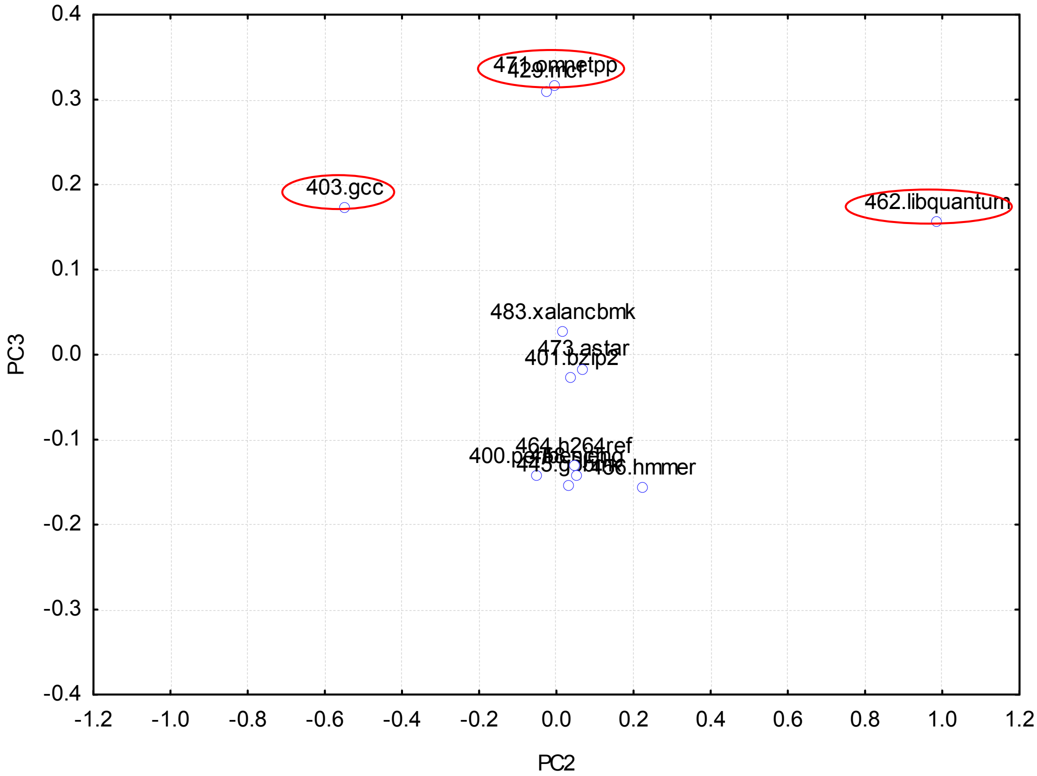


Figure 5: SPEC CINT programs plotted in the PC space using memory access characteristics (PC2 vs. PC3).

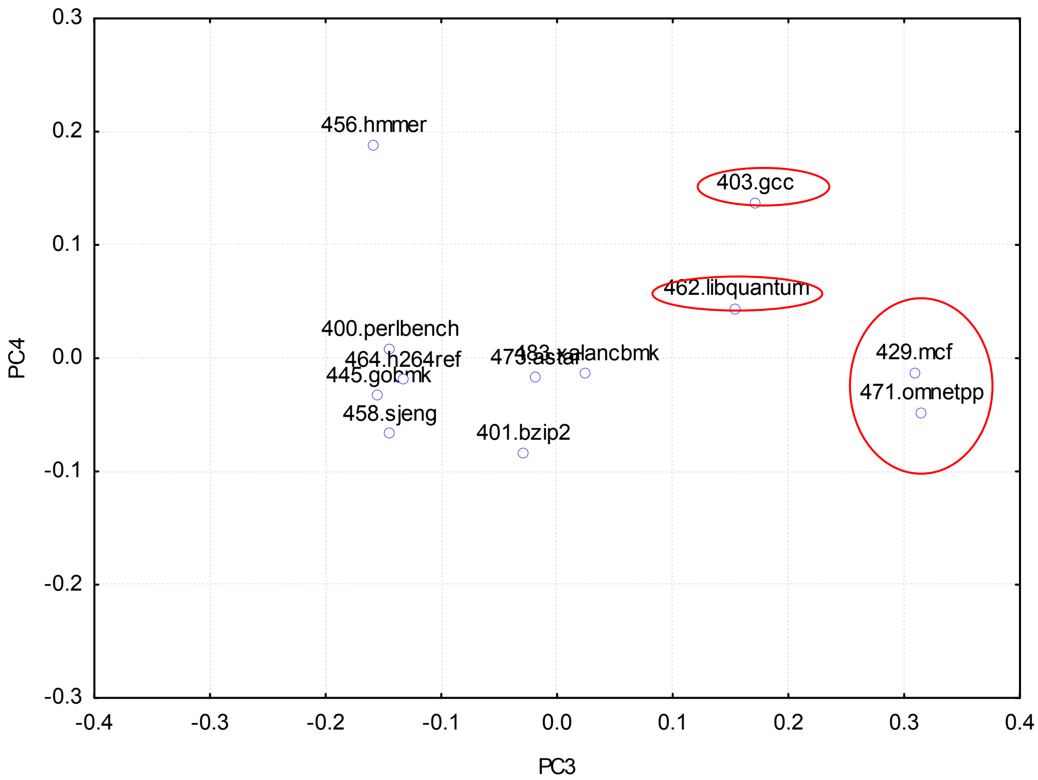
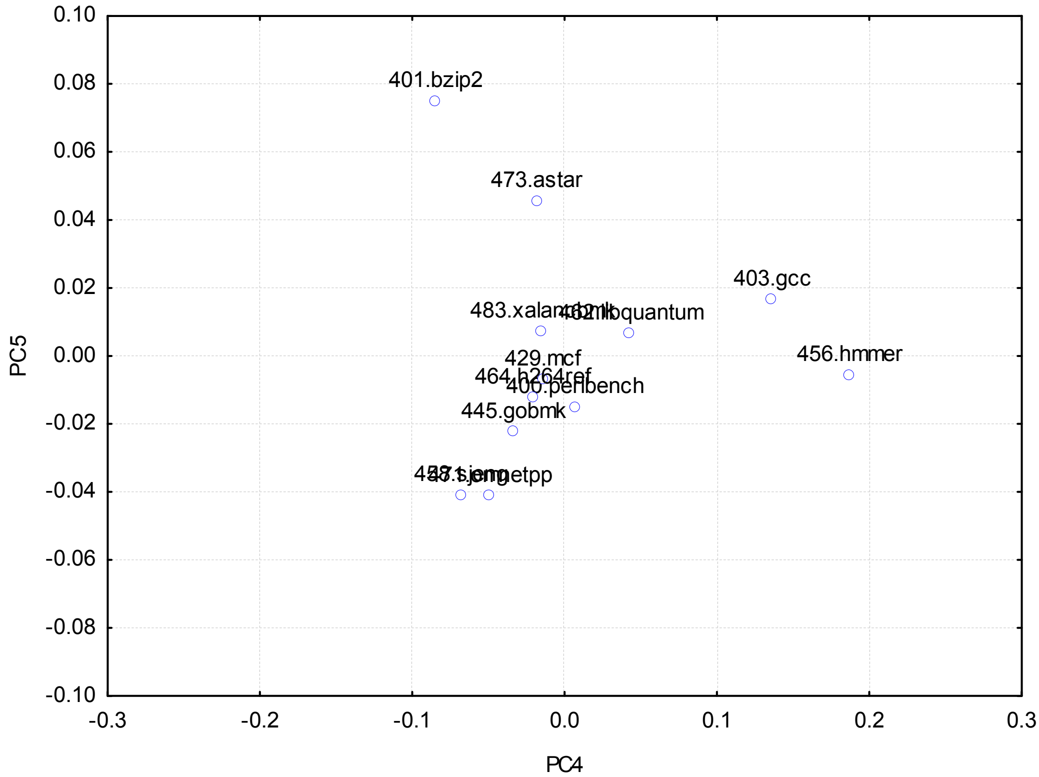


Figure 6: SPEC CINT programs plotted in the PC space using memory access characteristics (PC3 vs. PC4).



Figur
e 7: SPEC CINT programs plotted in the PC space using memory access characteristics (PC4 vs. PC5).

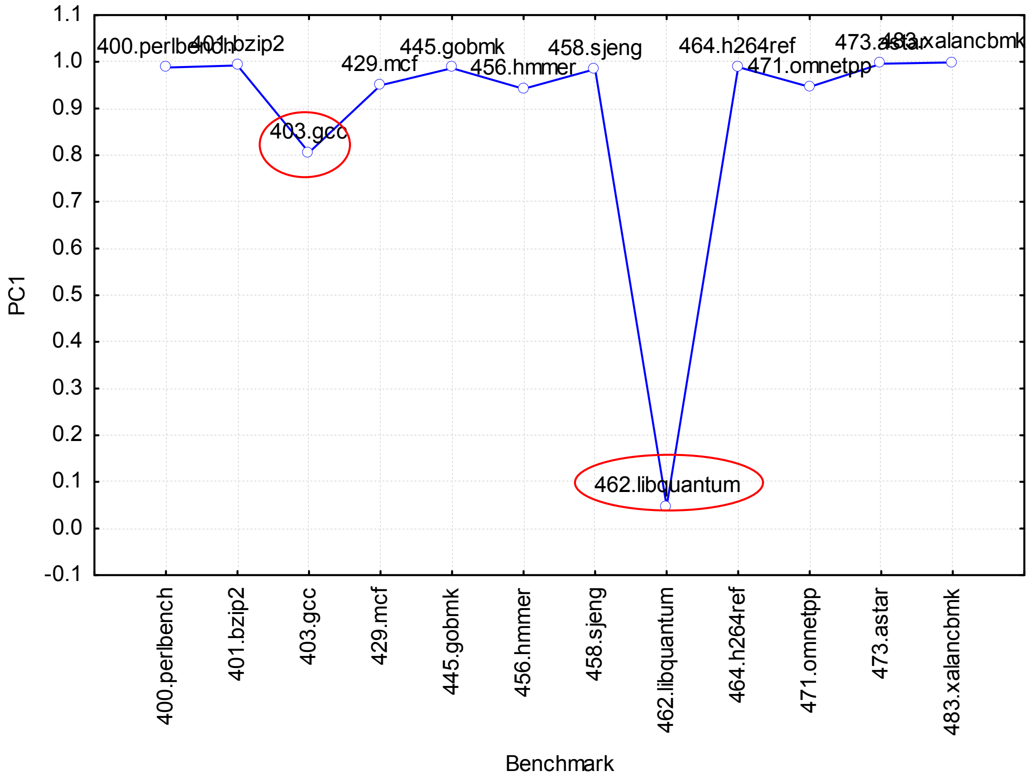


Figure 8: Variation of PC1 for individual SPEC CINT benchmark programs plotted in the PC space.

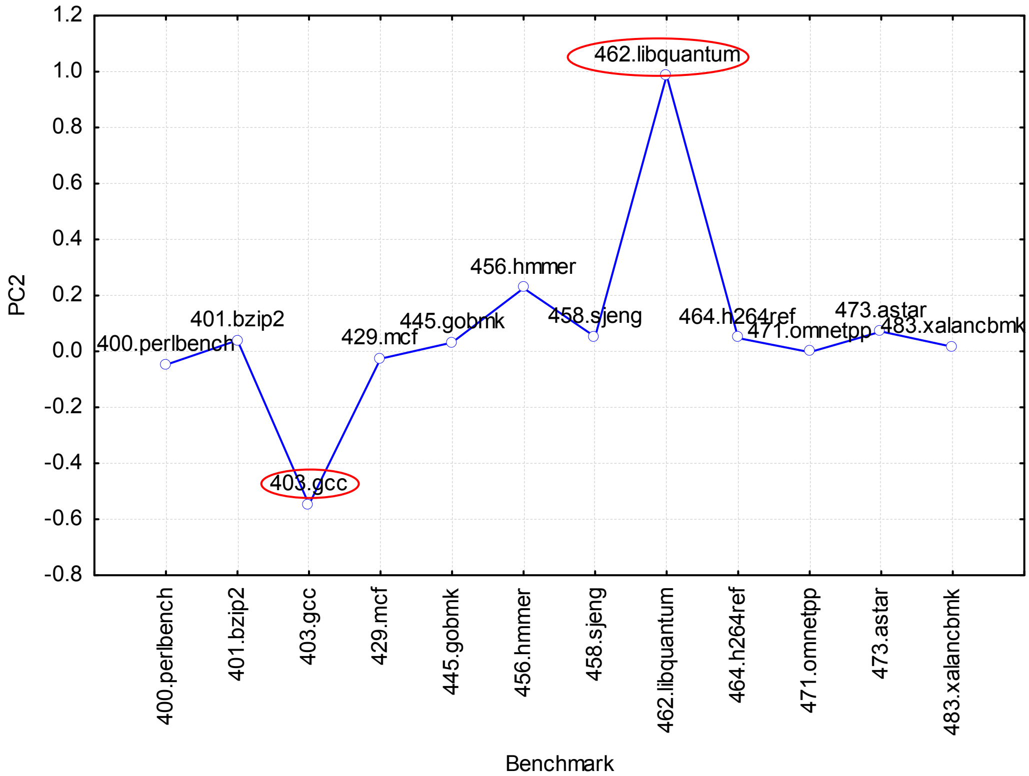


Figure 9: Variation of PC2 for individual SPEC CINT Benchmark programs plotted in the PC space.

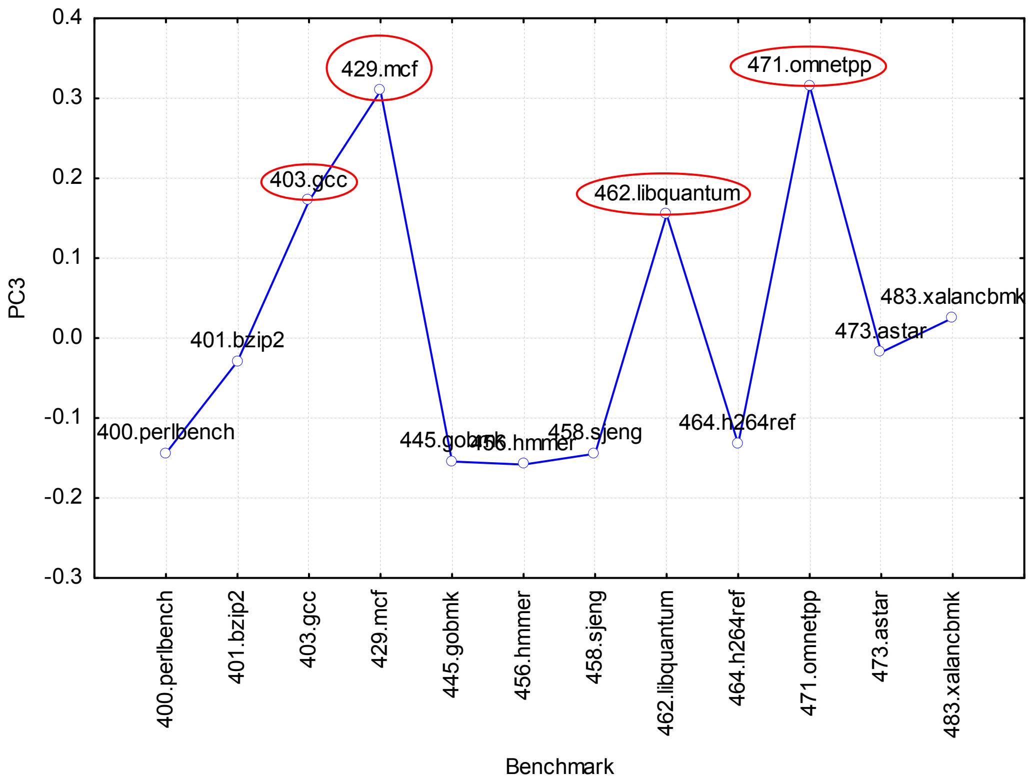


Figure 10: Variation of PC3 for individual SPEC CINT Benchmark programs plotted in the PC space.

5.3 CLUSTER ANALYSIS

Using Cluster Analysis (CA) in two-dimensional space, various groups of similar benchmark programs are identified. The linkage cluster analysis is shown in Figure 11, which explains the similarities and dissimilarities of workload of 12 benchmarks behavior on Intel machines, since selection of similar benchmark programs will only increases the performance evolution time of the processor without providing extra information. Improper selection of benchmark programs may not accurately illustrate the true performance of the processor.

Figure 11 illustrates the similarities and dissimilarities between benchmarks workload from the dendrogram, the behavior of 462.libquantum is significantly differ, which is also identified in principal component memory space.

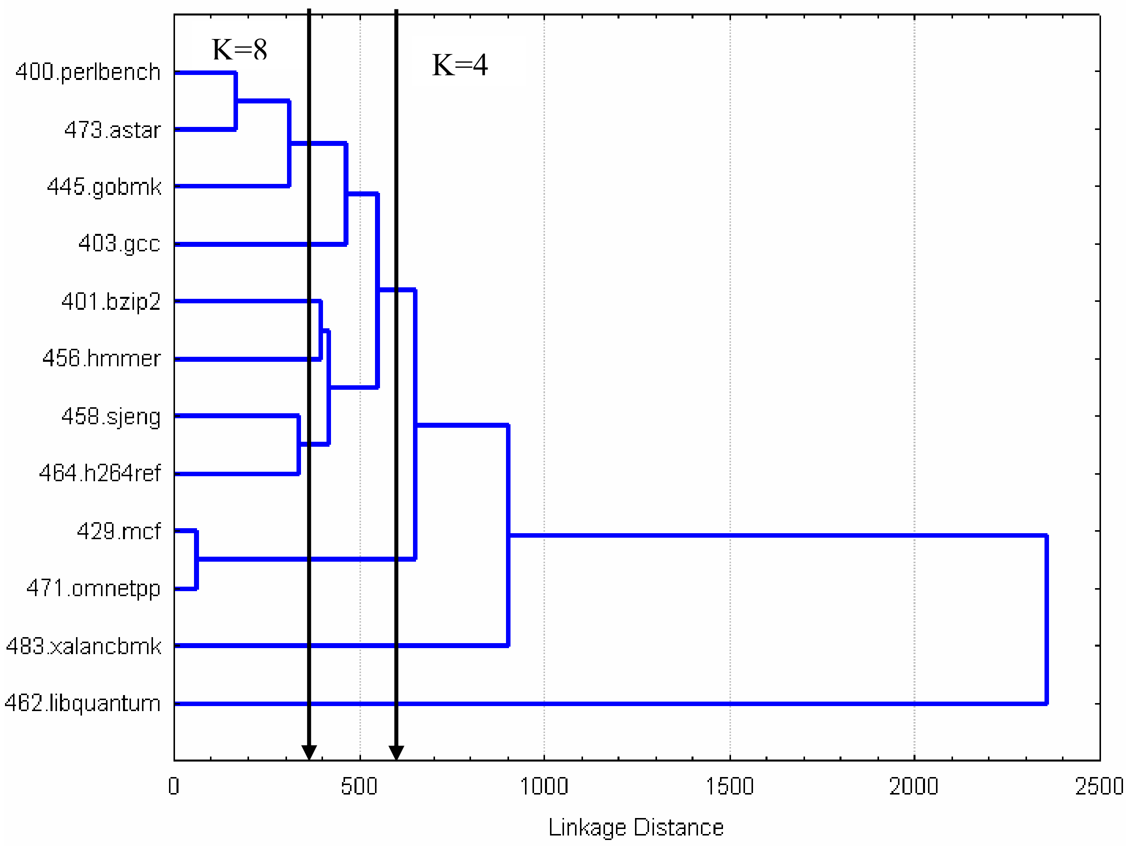


Figure 11: Dendrogram showing similarity between SPEC CINT2006 Benchmark Programs behavior with linkage distance.

As mentioned in linear regression analysis (Table 4), we classified the benchmark workload in two main categories. The classification of benchmarks from Table 4 can be selected from Dendrogram diagram shown in Figure 11. This Dendrogram is useful to a researcher and scientist working on computer architecture, in selecting input data set for their research. They can reduce his benchmark workload by plotting a line at linkage distance to ≈ 350 (K=8) for selecting first subset and draw a line near linkage distance ≈ 600 (K=4) for selecting second subset of benchmarks.

6. CONCLUSIONS

From this analysis we can conclude that benchmark 401.bzip2 which is used for Data Compression and the benchmark 464.h264 for Video Compression are shows least memory wait time. These Intel processors are best for compression work. The benchmark 462.libquantum shows high memory wait time which is used for Physics / Quantum Computing.

Using the recently published performance numbers from SPEC CPU INT 2006s benchmark suite of thirty different state of the art machines and statistical analysis techniques like linear regression analysis, principal component analysis and cluster analysis, we recognize the similarities and dissimilarities of recently released SPEC CPU INT2006 benchmark suite. Dendrogram (Figure 11) shows the behavior 12 integer benchmark programs. From the principal component analysis we identify the five most significant PCs, which are retained for 85% of the variance. It is clear from PCs the benchmarks programs 471.omnetpp, 462.libquantum, 429.mcf, and 403.gcc are more deviated from other benchmark behavior. Depending on memory wait time these benchmarks are classified in two subcategories. The first subset of group consists of 8 benchmarks and second subset consists of four benchmarks as discussed in table 4. We recognize that the one of the benchmark program of second subset group 462.libquantum exhibits higher memory wait time as compared to other benchmark. Different benchmarks have similar linkage distance; the selection of these benchmarks as input data set is only increases the execution time without providing extra information. Our results and analysis can be used by performance engineers, scientists and developers to better understand benchmark programs workload, it is useful to select the benchmark as input data set for better microarchitecture design of the processor.

7. 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.

8. ACKNOWLEDGEMENTS

The authors would like to thank Alex Predtechenski, Advanced Micro Devices, for SPEC Benchmark analysis. One of the authors (A.K.P.) gratefully acknowledges financial support of UGC, New Delhi, INDIA, for a Meritorious Research Fellowship.

9. REFRENCES

[1] Standard performance evaluation corporation (SPEC), Benchmarks, http://www.spec.org.
[2] I. Tuimi, The Lives and Death of Moore's Law, First Monday, Oct 11, 2002.
[3] www.dell.com/powersolutions
[4] Lilja, David J., Measuring Computer Performance: A Practitioner's Guide, Cambridge University Press, New York, NY, 2000.
[5] John L. Henning. SPEC CPU2000: Measuring CPU performance in the new millennium. IEEE Computer, July 2000.
[6] SPEC CPU2000 Press Release FAQ, available at http://www.spec.org/osg/cpu2000/press/faq.html
[7] A. KleinOsowski and D. Lilja, MinneSPEC: A new SPEC benchmark workload for simulation-based computer architecture research. Computer Architecture Letters, Volume 1, June 2002.
[8] 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.
[9] StatSoft, Inc. Commercial statistical software, STATISTICA for Windows. 2006. http://www.statsoft.com.
[10] B.F.J. Manly, Multivariate Statistical Methods: A Primer, 2nd ed., Chapman & Hall, 1994.
[11] I.T. Jolliffe, Principal Component Analysis, 2nd ed., Springer, 2002.
[12] NIST/SEMATECH e-Handbook of Statistical methods, in http://www.itl.nist.gov/div898/handbook/.
[13] B. F. J.Manly. Multivariate Statistical Methods: A primer. Chapman & Hall, second edition, 1994.
[14] StatSoft, Inc. STATISTICA for Windows. Computer program manual. 1999. http://www.statsoft.com.
[15] Alex Predtechenski , AMD "A Method for Benchmarks Analysis" Online Available at http://www.spec.org/events/specworkshop/abs.html,2006.