ISSN 1512-1232

ANALYSIS OF SPEC CPUINT2006 BENCHMARKS: PERFORMANCE AND CLASSIFICATION

Abdul Kareem. P and R. A. Singh

Department of Physics, Dr Hari Singh Gour University, Sagar 470003, India, E-mail: kareemskpa@hotmail.com

Abstract

High performance is a critical requirement to all microprocessors manufacturers. 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 CPU INT 2006 benchmarks workload and their classification. Today we need a processor which can provide a performance boost for many key application areas. 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 four commercial processors. We calculated three most significant PCs, which are retained for 91.6% of the variance. 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, Benchmarks, Moore's Law*

1. Introduction

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 [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 four 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 Xeon 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

ISSN 1512-1232

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 more than one billion transistors.

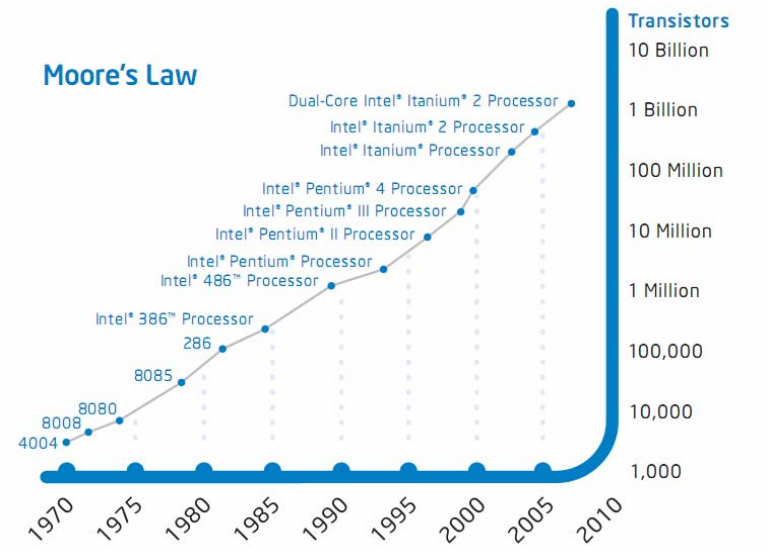

Fig.1. Scaling transistors. The number of transistors is expected to continue to double about every two years, in accordance with Moore's Law. Over time, the number of additional transistors will allow designers to increase the number of cores per chip. [Source from [3]]

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].

ISSN 1512-1232

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 CFP 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

4. Analysis of SPEC CPU2006 Benchmarks

4.1 Methodology

To analyze the benchmarks, we have used recently published SPEC CPUint2006 benchmark scores of a commercial CPUs (Intel Xeon X5260, Intel Xeon X5460, Intel Xeon E5450 and Intel XeonL5320). All scores are collected on same OS Windows Server 2003 Enterprise Edition X64 Edition. The detailed hardware configuration of processors is shown in Table 3. Each benchmark runs on these machines three times. There are 12 performance numbers, one per each benchmark for four most advanced commercial machines.

ISSN 1512-1232

Table 3. The Hardware specification of Intel Xenon machines on SPEC CPU 2006 Benchmarks

CPU Name	Intel Xeon X5260	Intel Xeon X5460	Intel Xeon E5450	Intel Xeon L5320
CPU Characteristics	1333MHz system bus	1333MHz system bus	1333MHz system bus	1066MHz system bus
CPU MHz	3333	3160	3000	1860
FPU	Integrated	Integrated	Integrated	Integrated
CPU(s) enabled	4 cores, 2 chips, 2 cores/chip	8 cores, 2 chips, 4 cores/chip	8 cores, 2 chips, 4 cores/chip	8 cores, 2 chips, 4 cores/chip
CPU(s) orderable	1, 2 chips	1, 2 chips	1, 2 chips	1, 2 chips
Primary Cache	32 KB I + 32 KB D on chip per core	32 KB I + 32 KB D on chip per core	32 KB I + 32 KB D on chip per core	32 KB I + 32 KB D on chip per core
Secondary Cache	6 MB I+D on chip per chip	12 MB I+D on chip per chip, 6 MB shared / 2 cores	12 MB I+D on chip per chip, 6 MB shared / 2 cores	8 MB I+D on chip per chip, 4 MB shared / 2 cores
Memory	16 GB (4 x 4 GB PC2-5300F CAS 5-5-5)	16 GB (8 x 2 GB PC2-5300F CAS 5-5-5)	16 GB (4 x 4 GB PC2-5300F CAS 5-5-5)	16 GB (4 x 4 GB PC2-5300F CAS 5-5-5)

The goal of our work is address in two concerns. First, we propose the linear regression analysis [8] to study the performance scaling in Intel Xeon 5000+ series. The results are discussed in section 5. As a second step we use statistical data analysis techniques called Principal Component Analysis (PCA) and Cluster Analysis (CA). 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 $X_1, X_2, \ldots, X_p$ into p principal components $Z_1, Z_2, \ldots, Z_p$ 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 Zp 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 ClusterAnalysis

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].

ISSN 1512-1232

5. Results and Discussion

5.1 Linear Regression Analysis

Figure 2 shows the scatter plot of Intel Xeon 5000+ series processors and a fitting line with 95 % confidence. In all four machines Intel Xeon X5260 shows least execution time, i.e. high performance. We used STATISTIC v.7 [9] for this analysis.

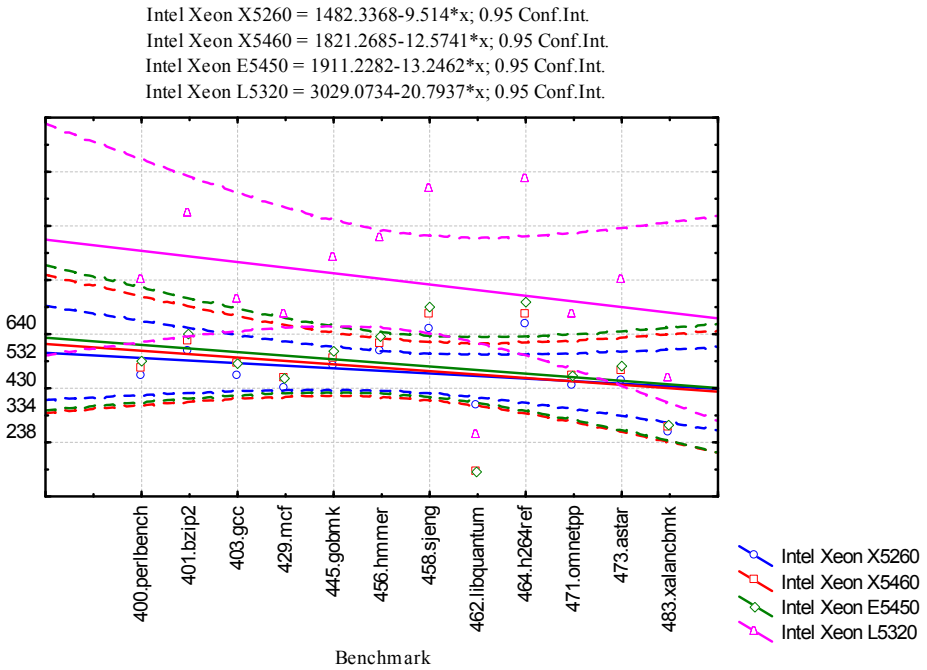

Fig 2. Scaling and scatter plot of performance of Intel Xeon 5000+ series processor.

Table 4 explains the summary of computation done in this statistical analysis, by using 12×4 benchmark performance matrix [1]. We have calculated Memory wait time @1GHz, @2GHz and @3GHz processor frequency.

Table 4. Memory Wait Time @1GHz, 2GHz and 3GHz processor frequency.

Core Frequency	3333	3160	3000	1860	RESULTS			Memory wait time, % of TCT		
Base score	22.7	23.8	22.9	13.8	A	B	R^2			
Core Clock, ns	0.30	0.31	0.33	0.53	Slope	Intercept	RSQ	@3GHz	@2GHz	@1GHz
400.perlbench	447	470	498	804	1504.003	-4.5	100.0%	-0.9%	-0.6%	-0.3%
401.bzip2	536	569	596	1042	2143.854	-111.4	100.0%	-18.5%	-11.6%	-5.5%
403.gcc	449	488	494	732	1156.577	110.6	99.6%	22.3%	16.1%	8.7%
429.mcf	401	437	435	673	1123.415	68.7	99.5%	15.5%	10.9%	5.8%
445.gobmk	485	511	538	880	1666.823	-16.3	100.0%	-3.0%	-2.0%	-1.0%
456.hmmer	532	562	593	958	1790.739	-4.6	100.0%	-0.8%	-0.5%	-0.3%
458.sjeng	616	674	696	1133	2134.92	-14.1	99.8%	-2.0%	-1.3%	-0.7%
462.libquantum	334	86.6	94.2	224	126.6995	137.5	1.4%	76.5%	68.5%	52.1%
464.h264ref	640	677	715	1172	2238.235	-31.3	100.0%	-4.4%	-2.9%	-1.4%
471.omnetpp	407	442	442	672	1092.671	84.4	99.5%	18.8%	13.4%	7.2%
473.astar	430	460	480	796	1534.723	-29.2	100.0%	-6.1%	-4.0%	-1.9%
483.xalancbmk	238	258	266	439	837.0288	-11.0	99.9%	-4.1%	-2.7%	-1.3%

Each individual time scales are in accord with the general observation, or Time = Ax + B,

ISSN 1512-1232

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.

Figure 3 shows the variation of task completion time with core clock frequency. All data points are linearly good fitted in trendline with R^2=0.997, Extrapolation of the runtime trendlines down to zero core clock period gives basis for useful interpretation of system behavior. The extended trendline touches task completion time axis, at 197.97 sec, which gives significant parameters memory wait time and core utilization time. Figure 4 shows the average memory wait time and core utilization time in Intel Xeon 5000+ series processors. These two parameters are useful to find the accurate performance of the processors.

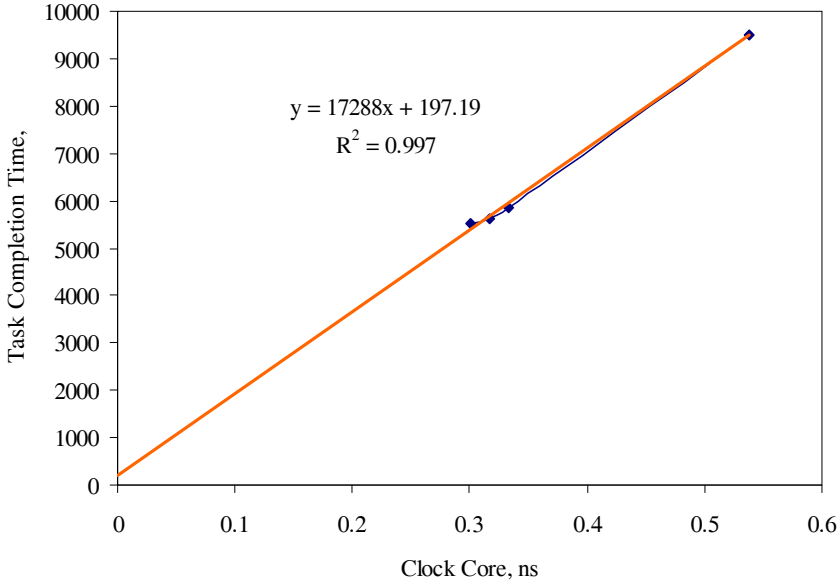

Fig 3. The benchmark runtime vs. core clock period shows scaling of performance of Intel Xeon X5260, X5460, E5450 & L5320 series processors, Extrapolation of the runtime trendlines down to zero core clock period gives basis for useful interpretation of system behavior

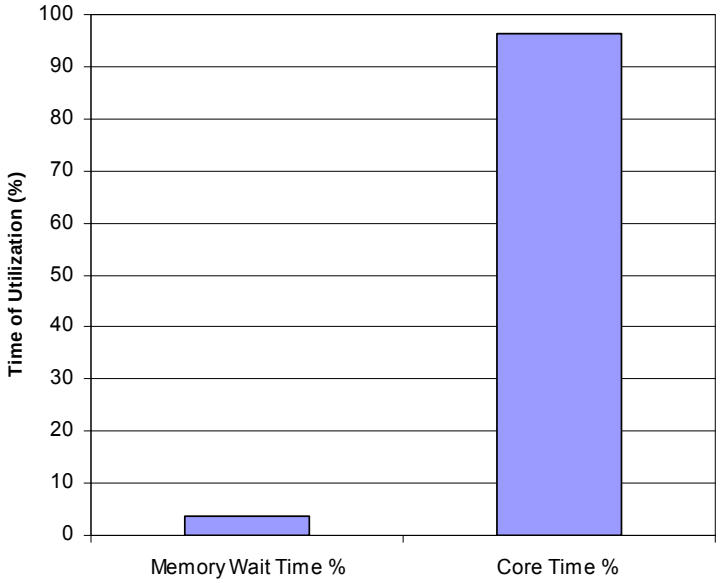

Fig 4. The comparison of memory wait time and core utilization time in Intel Xeon X5260, X5460, E5450 & L5320 series processors.

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

ISSN 1512-1232

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.

Table 5. 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 5, benchmark 462.libquantum shows maximum task completion time of all benchmark programs over @1GHz, @2GHz and @3GHz.

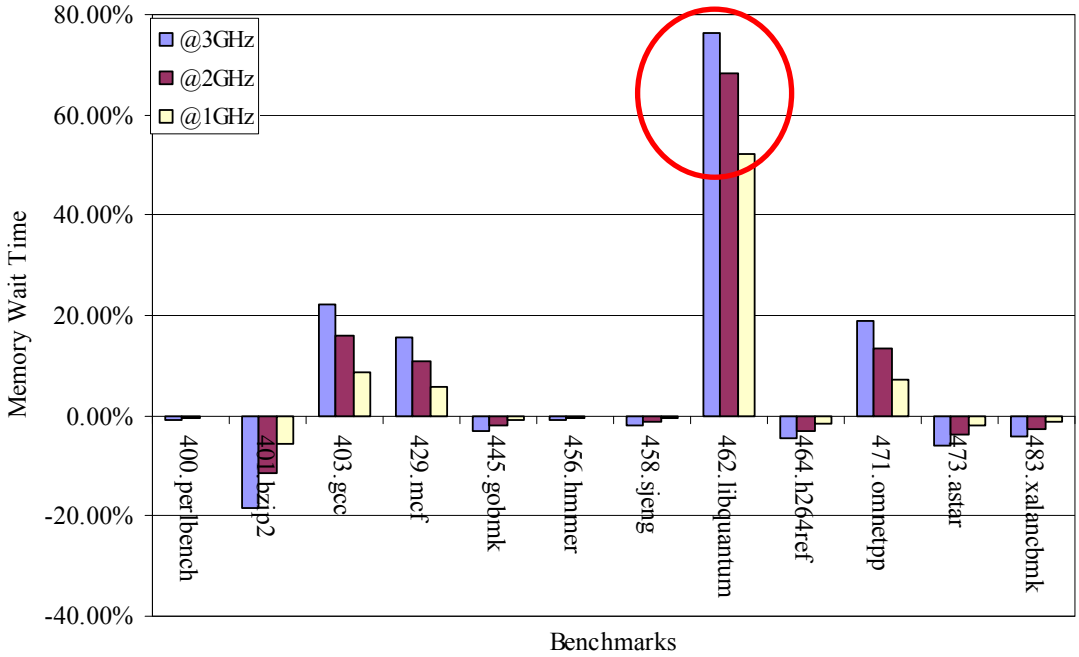

Fig 5. The comparison of normalized task completion time @1GHz, @2GHz and @3GHz processor frequency on Intel Xeon series processors.

5.2 Principal Component Analysis

The analyses of principal components results are discussed in this section. We generated three significant principal components PC1, PC2 and PC3 using benchmark workload and commercial statistical simulation software STATISTICA v.7 [14]. Three principal components are retained for 91.6% of the variance. Figure 6 shows the summary of variance estimated in the benchmark workload, PC2, PC3 holds 8.2% and 0.05% variance respectively.

ISSN 1512-1232

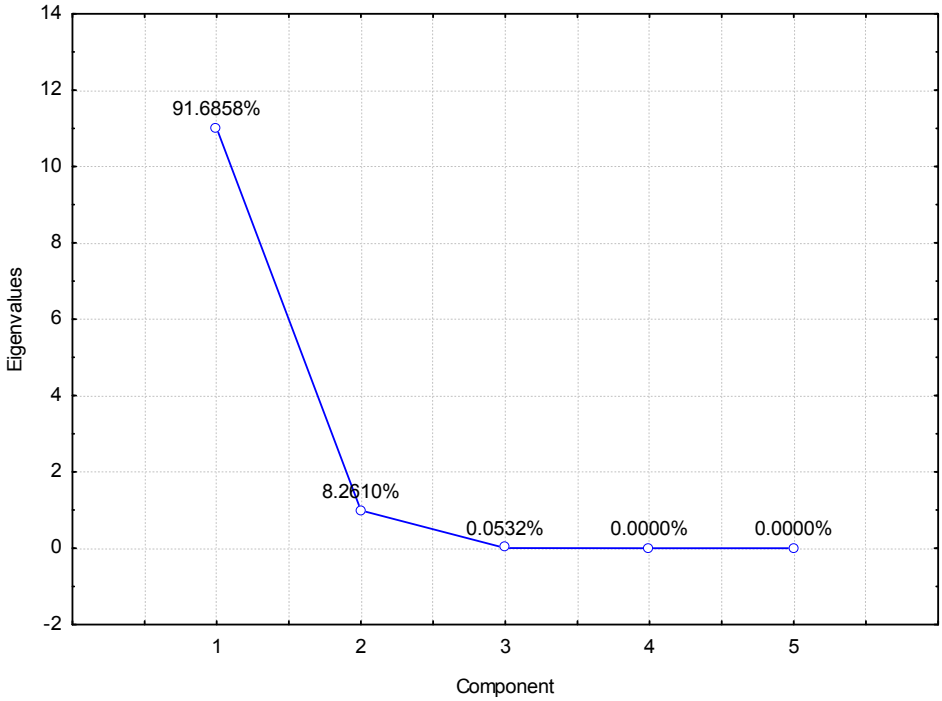

Fig 6. Eigenvalues scree plot of all principal components, which explain the variance in the workload (PC1, PC2 and PC3)

The summary of principal component analysis over the present benchmark workload on four Intel Xeon machines is summarized in figure 7, which represents R^2X and Q^2(Blue, Red).

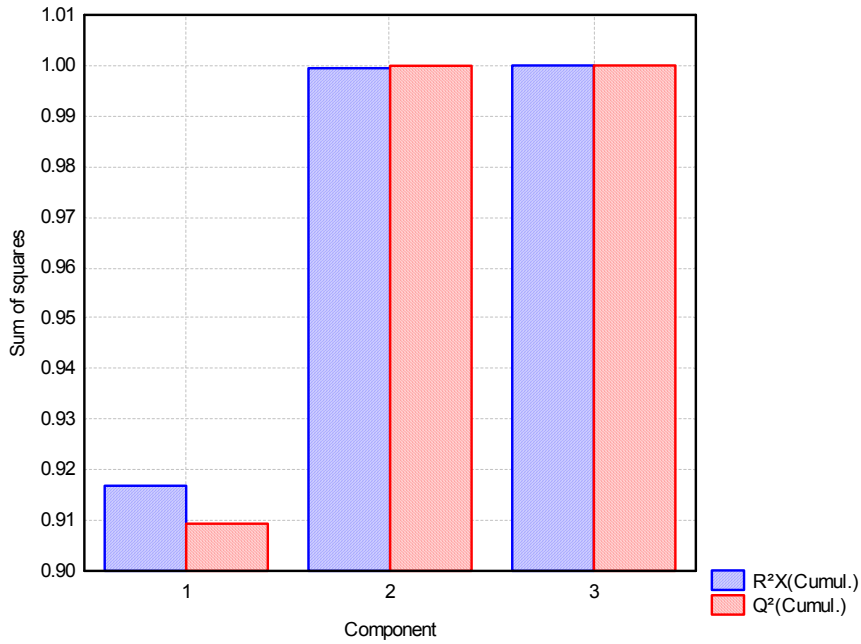

Fig 7: Principal Component Analysis (PCA) Summary Overview

Figure 8 shows the scatter plot of first two PCs, i.e. PC1 vs. PC2. Figure 9 and figure 10 shows the scatter plot of PC1 vs. PC3 and PC3 vs. PC2 respectively. In all PCs space the benchmark 462.libquantum is more scattered as compared to other benchmark.

ISSN 1512-1232

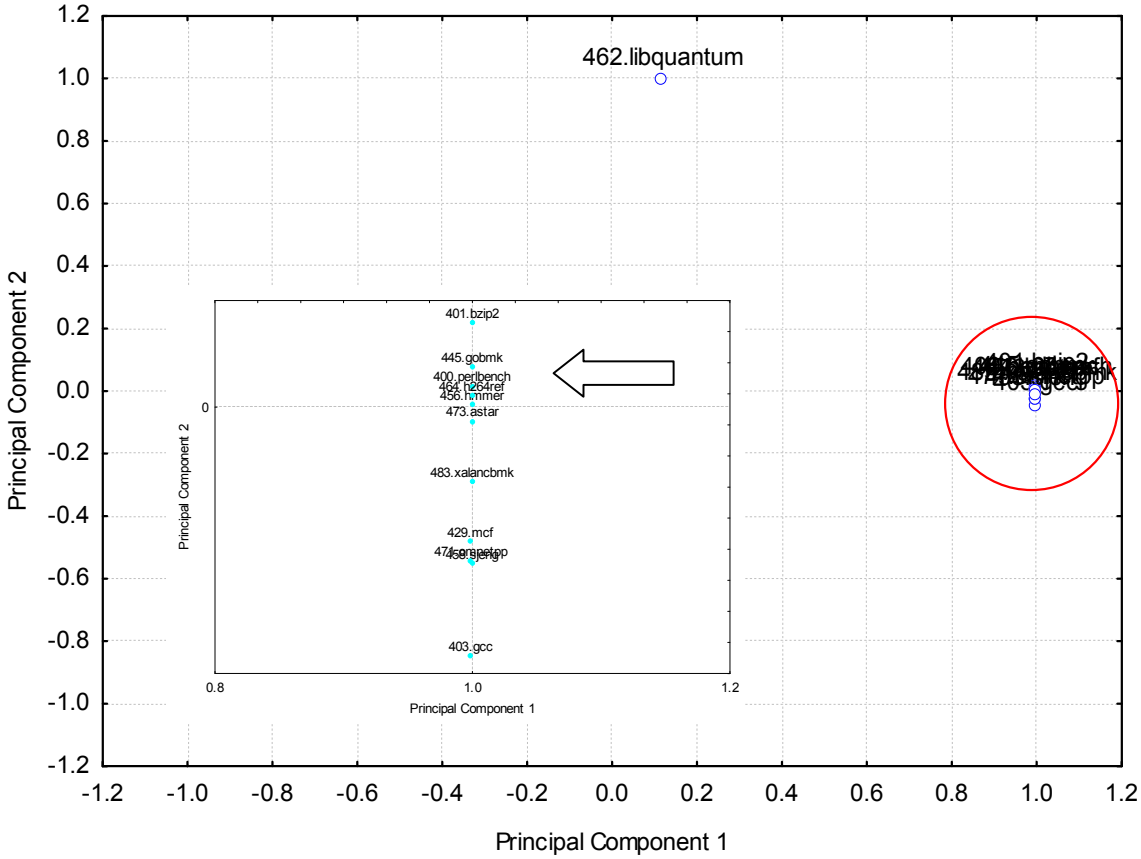

Fig 8. SPEC CINT 2006 programs plotted in the PC space using memory access characteristics (PC1 vs. PC2)

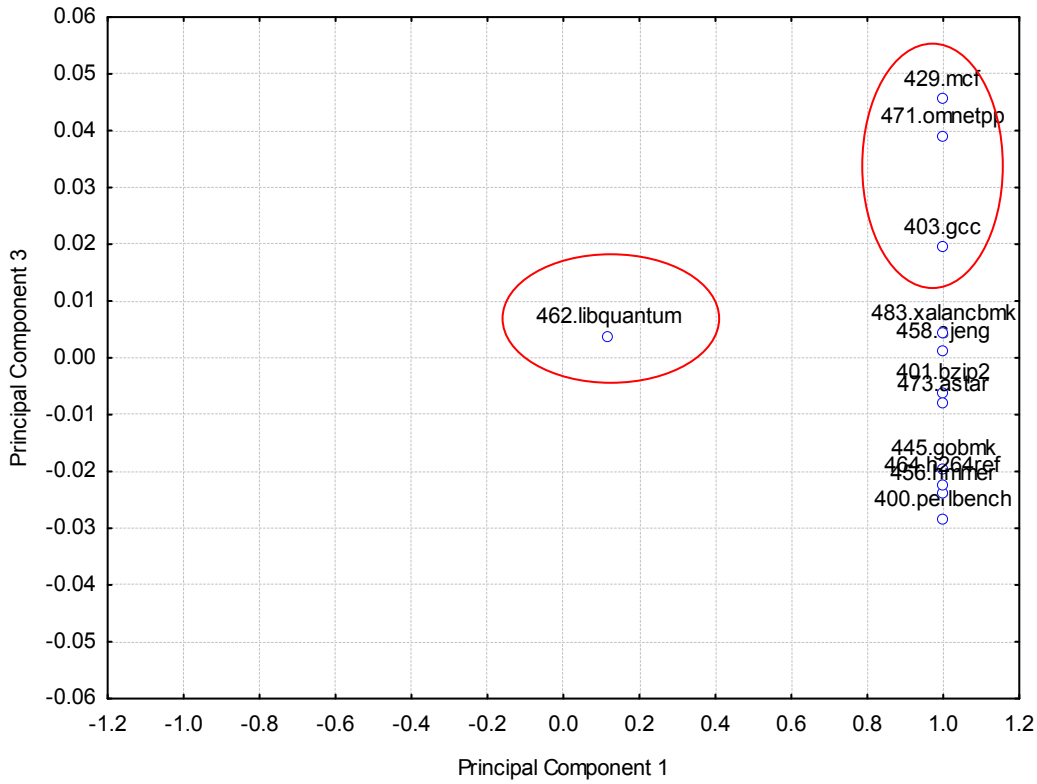

Fig 9. SPEC CINT programs plotted in the PC space using memory access characteristics (PC1 vs. PC3)

ISSN 1512-1232

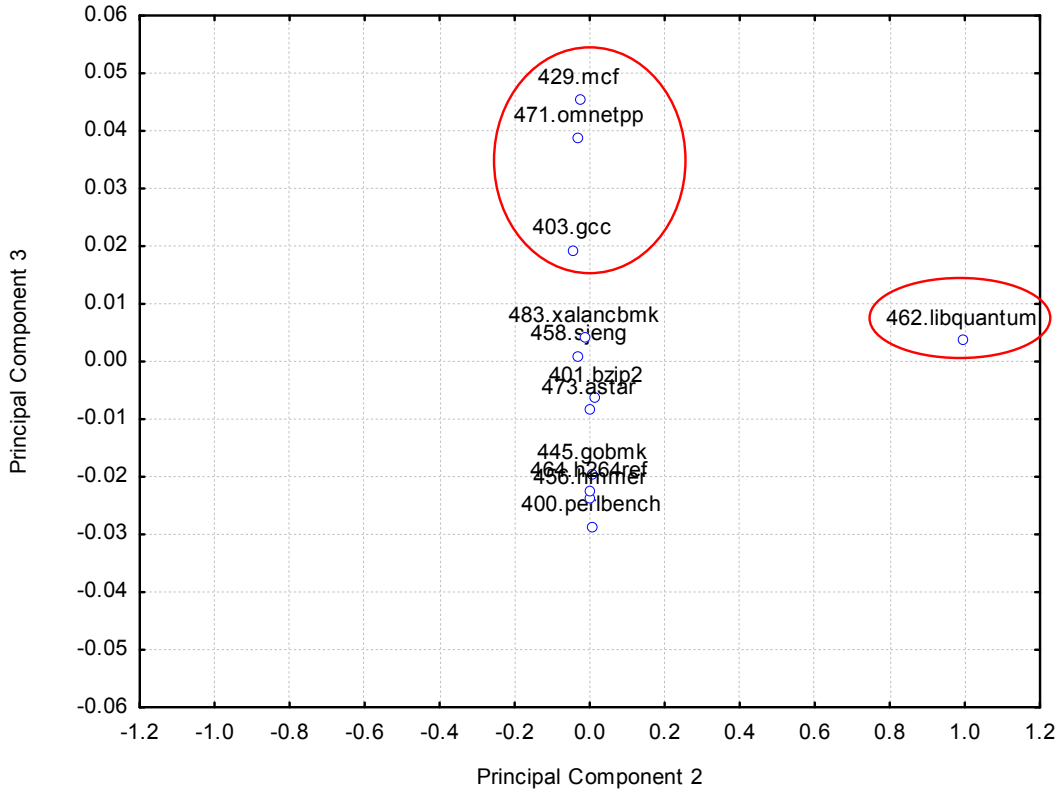

Fig 10. SPEC CINT programs plotted in the PC space using memory access characteristics (PC3 vs. PC2).

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 Xeon machines, since selection of similar benchmark programs will only increases the performance evolution of the processor without providing an 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 mentioned in principal component memory space.

As mentioned in linear regression analysis, we classified the benchmark workload in two main categories (Table 5). From this dendrogram a researcher and scientist working on computer architecture can reduce his benchmark workload by plotting a line at linkage distance to ≈ 140 (K=8) for selecting first subset and draw a line near linkage distance ≈ 260 (K=4) for selecting second subset of benchmarks.

ISSN 1512-1232

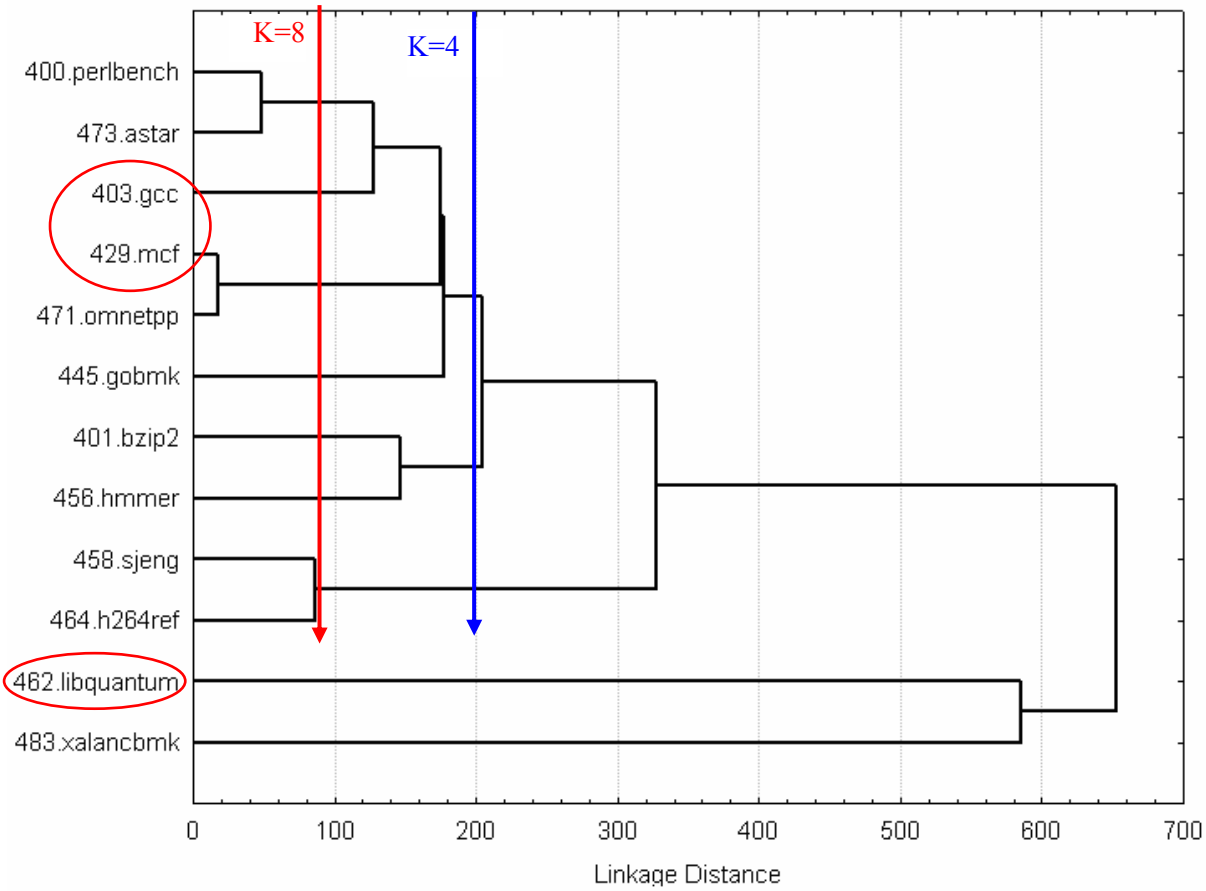

Fig 11. Dendrogram showing similarity between SPEC CINT2006 Benchmark Programs behavior with linkage distance.

6. Conclusion

Using the recently published performance numbers from SPEC CPU INT 2006s benchmark suite of four 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 three most significant PCs, which are retained for 91.6% 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 programs. 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 5. 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, they only increases the execution time. 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 CPU2006int 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 computer architecture simulation research.

ISSN 1512-1232

8. Acknowledgements

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

9. Refrence

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. Alex Predtechenski , AMD "A Method for Benchmarks Analysis" Online Available at http://www.spec.org/events/specworkshop/abs.html,2006.
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.

Article received: 2011-01-15