Multimedia Workloads versus SPEC CPU2000

Christopher Martinez, Mythri Pinnamaneni and Eugene B. John
Department of Electrical and Computer Engineering
The University of Texas
San Antonio, TX 78249-0669

Abstract - **With the use of peer-to-peer media sharing, the typical users now have a huge collection of media at their fingertips. Digitized audio and video is becoming the norm to store music, pictures and motion. State of the art computers, especially personal computers are spending a large part of their cycles on workloads performing audio and video encoding and decoding.**

The SPEC CPU2000 benchmarks are very popularly used in computer design and benchmarking, however, often many raise questions on the representativeness of SPEC benchmarks. Many often suggest that the SPEC CPU programs are some CPU intensive programs and may not be representative of real-world workloads. In this paper, we present a comparison of the performance characteristics of several real-world multimedia programs with execution characteristics of SPEC CPU 2000 programs. Our study finds that the SPEC CPU 2000 benchmarks are very diverse. The execution characteristics of multimedia programs lie well within the range exhibited by SPEC CPU benchmarks.

Keywords - **Performance Characterization, Measurement, On-chip performance counters, Microprocessor performance, audio processing, video processing, SPEC benchmarks**

1. Introduction

In the past few years, the computer industry has undergone tremendous changes in the application domain. Multimedia applications are becoming an increasingly common computer workload. The computer is no longer a business tool but it also has become a popular form of entertainment. While the home user still uses the typical applications from word processing, email, spreadsheets, and Internet surfing, the amount of multimedia being used has increased significantly.

The Internet has played a huge role in changing the workload of the computer. With the use of peer-to-peer media sharing, the typical users now have a huge collection of media at their fingertips. Digitized audio and video is becoming the norm to store music, pictures and motion. State of the art computers, especially personal computers are spending a large part of their cycles on workloads performing audio and video encoding and decoding. By observing the computer usage trends of different demographic spectrum, including college and school age users, one can observe that the computers are widely used to perform a wide variety of multimedia tasks including, music playback, DVD playback, video playback, and video games.

The SPEC CPU2000 benchmarks [15] are very popularly used in computer design and benchmarking. This benchmark suite contains 26 different programs belonging to integer and floating point categories. While SPEC benchmarks are popular, often many raise questions on the representativeness of SPEC benchmarks. Many often suggest that the SPEC CPU programs are some CPU intensive programs and may not be representative of real-world workloads. We set to investigate whether the performance characteristics of popular multimedia applications (circa 2005) are within the realm of the SPEC CPU benchmarks

Using several popular commercial audio and video applications on a state of the art microprocessor, the Intel Pentium 4, we make a comparison of execution time characteristics of these applications using measurement using the on-chip performance counters of the Pentium 4 processor. We find that the SPEC benchmark suite exhibits a wider range of characteristics than those exhibited by several audio and video applications.

2. Multimedia Workloads

The workloads that this paper examines are the most commonly used multimedia applications used today. The applications we used belong to audio and video encoding and decoding. The various applications and instruction counts are summarized in Table 1.

Audio:

In audio encoding, the application takes the original uncompressed audio file and applies an algorithm to map the file to a model perceivable by the human ear. The mapping into the model will remove all the sounds that the ear cannot perceive. Audio decoding will take the compressed data and extract into the standard pulse code modulated (PCM) format to be sent out to the sound card. Different applications are used to test the encoding of audio because the codec defines only the way the bit stream should look. Therefore each application will have its own way of encoding by choosing different levels of detail and type of hearing models.

Three popular audio applications were used in this study for studying audio decoding, *Winamp*, *RealPlayer* and *iTunes*. *Winamp v5.05* was one of the first applications available for MP3 decoding and it is also one of the most popular audio applications. *RealPlayer 10.5* is a popular multimedia application that is often used for streaming audio but can also be used as an all-purpose media program. *Apple's iTunes 4.6.0.15* is the application to interface and create content for the *iPod*.

Two popular audio codecs, *MP3* and *AAC* are used with each of the three aforementioned applications. (The word **codec** is often used as a short form to represent **co**ding and **dec**oding. MP3 and AAC contain different schemes to perform encoding and decoding and are referred to as codecs in the rest of this paper.) We chose to investigate *MP3* because it is the current standard for audio. Most users who use the computer to listen to music will have some *MP3* files. *AAC* is a very important format, and is the format used on the popular *iPod*. While *MP3* is currently more popular, *AAC* is becoming increasingly common due to the success of the iPod. To the best of our knowledge not much work has been previously done on *AAC*, and our study unveils more performance aspects of *AAC*.

For audio encoding, *WindowsMediaPlayer*, *RealPlayer* and *iTunes* are run with *MP3* and the latter two are run with *AAC*. *Windows Media Player 10* is the default all-purpose multimedia application that is available in all Microsoft Windows versions.

When conducting the performance evaluation of the different codecs and applications, to have a controlled experiment, the same procedure was used. The encoding of audio was performed on part of the *Beethoven Symphonie Pastoraie*. All performance tests were conducted on the same portion of the data. The audio decoding experiments were done using the song *"Boulevard Of Broken Dreams"* by Greenday. The same song was used for both *MP3* and *AAC* decoding.

Video:

Video codecs are more diverse than their audio counterparts. Most users pick their audio based on what their personal audio devices use or based on overall file size. When picking video codecs one must consider file size, video resolution, and quality. Different codecs are desirable from different perspectives. Our study focuses on a wide range of codecs such as low file size codec (MPEG4)[17], commercially common codec (MPEG2 or DVD), and high quality Windows Media Video (WMV) High Definition (HD). The different video applications are listed in Table 1. All video playback was done using Windows Media Player. The video used for video decoding and video encoding experiments was captured from a regular TV broadcast. The captured video was of a football game involving a complex scene with many moving objects in the scene. We strongly believe that this video presents a reasonable workload to the encoding and decoding routines rather than a simple scene involving a monologue of a close up person.

We investigated the performance of *Windows Media Video 9 (WMV), MPEG-2, and MPEG-4 (DIVX).* There has been very little prior work on the performance of *WMV*, which is becoming increasingly popular. Many multimedia experts believe that it will become one of the major video codecs in the near future. The importance of *WMV* is enhanced by the fact that part of the new Blu-ray DVD standard will be using a codec developed by Microsoft that is very similar to *MWV*. Many researchers believe the popularity of *WMV* will increase since it is one of the best in quality in the *HD* realm. To the best of our knowledge previous papers in this area has not investigated the *WMV HD* codec.

We chose to investigate encoding in *WMV HD* because none of the previous papers have studied this codec. To encode the video we captured a one-minute video in AVI format and encoded the video using the *Windows Media Encoder* program. *WMV HD* encoding is made up of taking two passes through the video. Encoding pass 1 is an analysis phase of encoding while pass 2 is doing the encoding of the video stream. Pass 1 takes a short amount of time to finish while pass 2 takes longer time.

Table 1: Multimedia Applications used in this study

Application Category	Codec Name	Instruction Count
Audio Decoding	*Winamp with MP3*	744 mil
	RealPlayer with MP3	246 mil
	iTunes with MP3	675 mil
	Winamp with AAC	825 mil
	RealPlayer with AAC	309 mil
	iTunes with AAC	576 mil
Audio Encoding	*Windows Media Player with MP3*	9756 mil
	RealPlayer with MP3	6438 mil
	iTunes with MP3	24759 mil
	RealPlayer with AAC	17190 mil
	iTunes with AAC	20574 mil
Video Decoding	*MPEG2 (DVD)*	2937 mil
	MPEG4	5757 mil
	WMV DVD	6657 mil
	WMV HD	7863 mil
Video Encoding	*WMV HD -Pass1*	26814 mil
	WMV HD - Pass2	30648 mil

The same audio file was used to encode into MP3 and AAC and the same video was used for the video applications. The instruction counts of *Realplayer* and *Winamp* are different due to differences in the algorithms, formats, buffer lengths, etc.

3. Description of the Pentium 4 microarchitecture:

The measurements in this paper were conducted on an Intel Pentium 4 fabricated at 90nm technology [1]. In x86 architectures since mid 90s, the native x86 instructions are first broken down into microoperations or ***uops*** of RISC nature. The Pentium 4 processor is capable of executing up to six *uops* per cycle. An overview of the Pentium 4 microarchitecture is shown in Figure 1.

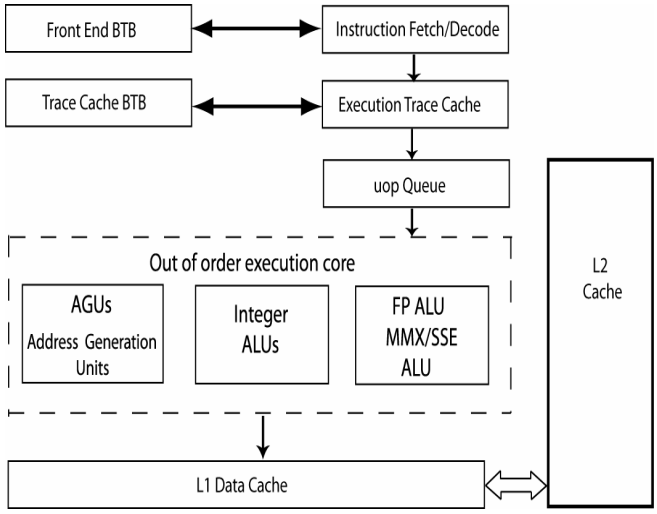

Figure 1: Block Diagram of the Pentium 4 Processor

The Pentium 4 microarchitecture uses an advanced instruction cache called the Execution Trace Cache. The trace cache is able to deliver up to three *uops* per clock cycle to the out of order execution core [1]. The out-of-order execution core contains several integer ALUs, Address Generation Units, Floating point and Media processing units (MMX, SSE, SSE2, SSE3) and memory queues. The out of ordering logic considers a large window of *uops*, in order to find several independent *uops* that are ready to execute. The out of order core can contain up to 126 *uops*, of which, 48 can be *load* operations and 32 can be *store* operations. The ALUs are "double-pumped" i.e. they can take an instruction every half clock cycle. The *uops* are reordered to execute them as fast as possible, results are held in buffers and the instructions retire (commit) later in program order guaranteeing the correctness of the program. The architecture includes a 16K L1 data cache and a1MB L2 cache.

The trace cache provides several advantages to the processor. It stores decoded instructions in the form of microoperations (or *uops*) rather than in the form of native x86 instructions as in a conventional instruction cache. Storing *uops* instead of original instructions allows the complicated instruction decoding logic to be removed from the main execution loop [1]. Instructions are fetched and decoded and the decoded instructions are stored into a Trace Cache, which can then be accessed repeatedly just like a conventional instruction cache.

During instruction fetch, conventional instruction caches typically provide instructions up to the next branch, but no instructions from the path after a taken branch. If the first instruction in a cache line is a branch, only a single useful instruction is obtained by the fetch. Conventional instruction caches also often add a clock cycle delay to get to the target of the taken branch due to the latency of the branch predictor and then accessing the new location in the instruction cache. The trace cache avoids both these instruction delivery delays [1]. The trace cache takes *uops* from the instruction decoder and assembles them into traces, which represent the sequences of instructions encountered during execution. These traces go beyond branches and can include a taken branch and its target instruction even if the two are thousands of bytes apart in the static code. As Figure 1 indicates the trace cache utilizes a Trace Cache Branch Target Buffer in its operation, in addition to the BTB used by the instruction fetcher. The trace cache on the Pentium 4 holds up to 12K *uops*. When an instruction fetch misses in the trace cache, the instruction is fetched from the unified second level (L2) cache.

3.1 Performance Metrics

The Intel Pentium 4 processor, as is evident from the description of the architecture, has many advanced microarchitecture features to allow issue and execution of several instructions every cycle. This paper examines the effectiveness of these micro-architectural features, in the context of the considered media applications. The following features are examined in this study: MMX/SSE instruction usage, cache performance, branch prediction performance, and trace cache performance.

The processor supports several instruction set extensions to support multimedia applications: MMX, SSE, SSE2, and SSE3 SIMD extensions. One of our objectives in this research is to determine whether modern audio and video applications utilize these instructions. Our experimental set up measured the usage of MMX and SSE instructions in the studied applications.

The architecture includes a 16K L1 data cache, a trace cache that can hold 12K *uops*, and a 1MB unified L2 cache. It is important to know how effective the various caches are. As mentioned in the architecture description, two BTBs are used by the architecture: one as in any conventional processor and the other one for the trace cache. Branch mispredictions are expensive from the perspective of performance, and it is important to know the effectiveness of the branch prediction and the trace cache's ability to deliver traces in response to fetch requests.

3.2 Performance Monitoring Tools

All modern high performance microprocessors contain on-chip performance monitoring counters that can be measured to learn about the execution characteristics of real world

applications on actual platforms. The Pentium 4 also provides a variety of performance counters. We used Intel's *Vtune* program to access the hardware performance counters in the Pentium 4. The counters we measured are listed in Table 2.

Table 2. Performance Counters measured in our experiment

Performance Metric	VTune Capture Event
Execution Time (cycles)	Clock Ticks
Non Halt Clock Ticks	Non Halt Clock Ticks
MMX Instructions	128bit MMX Instruction Retire & 64bit MMX Instructions Retire
SSE Instructions	Scalar-Single Precision Streaming SIMD Instructions Retired & Scalar-Double Precision Streaming SIMD Instructions Retired
Mispredictions	Mispredict Branch Instructions Retired
Memory Order Pipeline Stalls	Memory Order Machine Clear
Self-modifying code stalls	Self Modifying Code Clear
Trace Cache Delivery Rate	Trace Cache Deliver Mode
Data Cache Misses	1st Level Cache Load Miss Retire & 2nd Level Cache Load Miss Retire

We collected a total of 11 runs of the *Vtune* sampling collection to get all the counters to determine the performance of the application. Each sampling run was carried out for 10 seconds. We used the time frame of 10 seconds due to the speed of the audio encoding. The primary metrics that were computed from the counter measurements are Clock ticks per instruction (CPI), branch prediction rate, cache hit rate, trace cache delivery rate, percentage of MMX and SSE instructions, and the percentage of floating point instructions.

4. Results

4.1 Clockticks Per Instruction (CPI)

Clockticks per Instruction (CPI) is an indicator of overall performance and use of resources on the superscalar processor. Table 3 shows the breakdown of the CPI for all test cases that we investigated in both overall CPI and non-halted CPI. The non-halted CPI is the measurement of the number of instruction with respect to the number of clockticks where the CPU is actively executing (i.e. without counting the halted cycles). In Table 3 we also present the micro-operations *(uops)* per instruction, and the cycles taken per micro-operation. It should be remembered that the Pentium 4 is capable of executing up to six *uops* in one clock cycle.

Our experimental results indicate that commercial audio and video applications take between 1.4 and 3.5 cycles (per instruction) to execute. The applications have 1.29 to 1.7 *uops* per instruction. Many of the SPEC programs have CPIs, which were similar to that of the media programs, however the two floating point benchmarks (*art* and *equake*) had higher CPI than the media benchmarks. The SPEC programs were seen to be diverse in that their CPI, ranged from 1.16 to 8.54, whereas the CPIs of the media applications only ranged from 1.4 to 3.55. The benchmark program *mcf* had very high cache miss rates, and the *equake* program had poor trace cache delivery rate. Overall the SPEC2000 benchmarks and the multimedia applications have similar *uops* per instruction; this observation helps us to conclude that similar instructions were being used in both types of applications.

4.2 Branch Prediction rate:

One of the main differences among the applications used in this study is their branch prediction rate. Table 4 shows the prediction rates for the different codecs and SPEC programs. Fig 2 presents the correlation between branch mispredictions per instruction and CPI of multimedia and SPEC programs. The 31 programs are listed on the x-axis in the order they are in the various tables. The first 18 programs are the media programs that are used in this study and the rest are the SPEC programs that we used for comparison purposes. The CPI is in cycles and the branch misprediction is expressed as mispredictions per 1000 instructions.

When compared to the SPEC2000 applications, the variability in branch misprediction rates is higher in SPEC programs than the media applications. The misprediction rates and CPI appear to be much correlated for the media programs, while there is no such serious correlation in the SPEC benchmark programs. As we analyze other performance metrics in the upcoming sections of this paper, it will become clear that other factors such as cache miss rates impact CPI more strongly than branch misprediction in the SPEC programs. We also note that the video and audio encoding have lower percentage of branches than the SPEC application. It is interesting to see that the media applications with fewer branches exhibit such strong correlation between CPI and mispredictions.

4.3 Cache Performance

Our study also investigated the cache hit rate for both 1st level and 2nd level caches. Table 5 gives the hit rates for the caches. All of the multimedia applications that we investigated had good cache hit rates (93% or more), while some of the SPEC2000 benchmarks suffered from poor cache performance (only 57% hits). The sequential and structured streaming data in media applications does result in cache hits due to spatial locality. Some media applications also involve repeated use of the same constants/coefficients, which leads to temporal locality. The high temporal and spatial locality results in good cache hit rates. Some of the SPEC programs incur unusually high cache miss rates, eg: mcf and art. These are the programs that had very high CPIs in Table 3.

Table 3: CPI of Applications used in this study

Application Category	Codec Name	CPI	Non-halted CPI	*uops* per Ins.	Cycle per uop
Audio Decoding	*Winamp with MP3*	3.11	3.03	1.71	1.82
	RealPlayer with MP3	3.55	3.04	1.54	2.30
	iTunes with MP3	1.85	1.81	1.54	1.20
	Winamp with AAC	2.43	2.40	1.38	1.76
	RealPlayer with AAC	2.82	2.61	1.57	1.80
	iTunes with AAC	1.98	1.91	1.61	1.23
Audio Encoding	*Windows Media Player with MP3*	1.66	1.65	1.49	1.12
	RealPlayer with MP3	2.02	2.00	1.38	1.47
	iTunes with MP3	2.07	2.07	1.41	1.47
	RealPlayer with AAC	1.71	1.71	1.38	1.24
	iTunes with AAC	1.40	1.40	1.30	1.08
Video Decoding	*MPEG2 (DVD)*	2.38	2.41	1.43	1.67
	MPEG4	2.59	2.58	1.37	1.89
	WMV DVD	1.96	1.96	1.28	1.53
	WMV HD	2.14	2.14	1.31	1.64
Video Encoding	*WMV HD -Pass1*	2.08	2.08	1.31	1.59
	WMV HD – Pass2	1.82	1.83	1.29	1.42
SPEC2000	*gcc*	1.81	1.80	1.72	1.05
	gzip	1.52	1.52	1.35	1.13
	mcf	8.54	8.50	1.29	6.60
	vortex	1.32	1.31	1.60	0.82
	vpr	3.17	3.15	1.46	2.17
	art	4.73	4.70	1.32	3.57
	equake	8.31	8.26	2.48	3.35
	parser	1.86	1.85	1.52	1.22
	crafty	1.81	1.80	1.31	1.38
	eon	2.53	2.53	2.11	1.20
	gap	1.40	1.40	1.53	0.92
	perlbmk	1.16	1.16	1.48	0.79
	bzip2	2.06	2.05	1.42	1.44
	twolf	3.36	3.35	1.56	2.15

4.4 Trace Cache Performance

Trace cache performance is indicative of instruction fetch performance. In order to measure the trace cache delivery rate, we used the Intel's VTune Trace Cache Delivery Rate ratio. The ratio is *TraceCacheDeliverMode * 100/Clockticks*, where a lower number means the trace cache is able to deliver micro-ops to the execution units only in fewer clock ticks. Table 6 gives the trace cache delivery rate.

The trace cache delivery rates of many SPEC programs are as high as 98%, whereas the multimedia programs did not exhibit trace cache delivery rates higher than 91%. *Equake* and *crafty* among the SPEC programs exhibit very low trace cache delivery rates. As in other metrics, the SPEC programs look more versatile than the media programs in the large range of the metrics.

5. Conclusion

We have moved into an era where multimedia is a dominant computer workload. In this case study, we investigated the performance of audio and video applications, while decoding and encoding with the most common commercial encoding/decoding software. We used several real commercial applications from audio and video fields and compared them with the popular SPEC CPU benchmarks.

Our comparison of media programs with the SPEC benchmarks illustrated that the SPEC programs are very diverse. The range of metrics observed by the media applications was smaller than the range for the SPEC programs. The SPEC suite does have some programs with very high and very low CPIs.

References:

[1]. D. Boggs, A. Baktha, J. Hawkins, J. Miller, P. Roussel, R. Singhal, B. S. Venkatraman, "The Microarchitecture of the Intel Pentium 4 Processor on 90nm Technology," Intel Technology Journal, Vol 8, Issue 1, 2004.

Table 4: Branch prediction rates of Applications used in this study

Application Category	Codec Name	% Branches	Prediction rate	Mispred per instrn	CPI
Audio Decoding	*Winamp with MP3*	12.8	94.92	0.0065	3.11
	RealPlayer with MP3	9.41	91.50	0.0080	3.55
	iTunes with MP3	11.76	97.84	0.0025	1.85
	Winamp with AAC	16.85	96.88	0.0053	2.43
	RealPlayer with AAC	13.02	95.36	0.0060	2.82
	iTunes with AAC	12.81	98.16	0.0024	1.98
Audio Encoding	*Windows Media Player with MP3*	9.08	96.96	0.0028	1.66
	RealPlayer with MP3	10.42	95.86	0.0043	2.02
	iTunes with MP3	0.53	94.87	0.0055	2.07
	RealPlayer with AAC	7.74	94.52	0.0043	1.71
	iTunes with AAC	7.68	95.36	0.0035	1.40
Video Decoding	*MPEG2 (DVD)*	8.91	92.93	0.0063	2.38
	MPEG4	8.28	96.76	0.0027	2.59
	WMV DVD	5.12	95.86	0.0021	1.96
	WMV HD	9.89	96.30	0.0018	2.14
Video Encoding	*WMV HD -Pass1*	6.31	94.69	0.0033	2.08
	WMV HD - Pass2	9.28	95.46	0.0042	1.82
SPEC2000	*gcc*	21.84	96.91	0.0067	1.81
	gzip	19.10	94.89	0.0097	1.52
	mcf	24.25	95.78	0.0102	8.54
	vortex	21.22	99.75	0.0005	1.32
	vpr	16.57	92.86	0.0118	3.17
	art	14.21	99.21	0.0011	4.73
	equake	11.00	98.21	0.0020	8.31
	parser	20.80	96.65	0.0074	1.86
	crafty	15.76	94.20	0.0091	1.81
	eon	13.45	97.12	0.0039	2.53
	gap	17.51	98.57	0.0025	1.40
	perlbmk	21.18	98.56	0.0031	1.16
	bzip2	14.83	94.35	0.0084	2.06
	twolf	16.48	88.39	0.0019	3.36

[2].D. Bhandarkar and J. Ding, "Performance Characterization of the Pentium Pro Processor", Proceedings of the 3rd High Performance Computer Architecture Symposium, pp. 288-297, 1997

[3]. R. Bhargava, L. John, B. L. Evans, R. Radhakrishnan, Evaluating MMX Technology Using DSP and Multimedia Applications, In *Proceedings of the IEEE Symposium on Microarchitecture*, pp. 37-46, November, 1998

[4]. Y. Chen, M. Holliman, E. Debes, S. Zheltov, A. Knyazev, S. Bratanov, R. Belenov, and I. Santos, "Media Applications on Hyper-Threading Technology," Intel Technology Journal, Q1, Vol. 6, No. 1, pp 1-11, 2002.

[5]. J. Fritts, W. Wolf and B. Liu,``Understanding multimedia application characteristics for designing programmable media processors", Proceedings of SPIE, vol. 3655, pp. 2-13, Jan 1999

[6]. M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge and R. B. Brown, ``MiBench: A free, commercially representative embedded benchmark suite," In proceedings of the 4th annual IEEE international workshop on workload characterization, pp. 3-14, Dec. 2001

[8]. C. Lee, M. Potkonjak, and W. H. Mangione-Smith, ``MediaBench: A tool for evaluating and synthesizing multimedia and communication systems," in Proceedings of the 30th Annual International Symposium on Microarchitecture, pp. 330-335, Dec. 1997

[9].Y. Luo, J. Rubio, L. John, P. Seshadri and A. Mericas, Benchmarking Internet Servers on Superscalar Machines, *IEEE Computer*, February 2003, pp. 34-40.

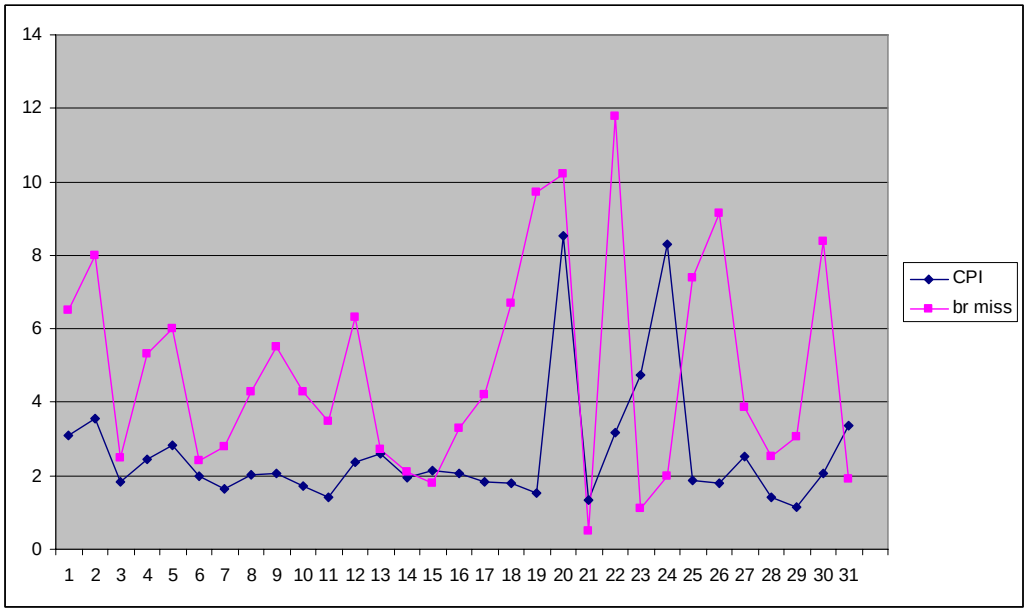

Fig 2. Branch mispredictions and CPI for 32 programs
(Y-axis is CPI in cycles or branch mispredictions for 1000 instructions. X-axis indicates the benchmarks. 1-18 are media, 19-32 are SPEC CPU2000)

Table 5: Data Cache Hit rates of Applications used in this study

Application Category	Codec Name	L1 Hit rate	L2 Hit rate	CPI
Audio Decoding	*Winamp with MP3*	93.79	99.85	3.11
	RealPlayer with MP3	96.45	99.93	3.55
	iTunes with MP3	97.68	99.94	1.85
	Winamp with AAC	95.20	99.82	2.43
	RealPlayer with AAC	94.58	99.88	2.82
	iTunes with AAC	96.95	99.94	1.98
Audio Encoding	*Windows Media Player with MP3*	99.05	99.99	1.66
	RealPlayer with MP3	98.35	99.91	2.02
	iTunes with MP3	98.33	99.95	2.07
	RealPlayer with AAC	97.29	99.97	1.71
	iTunes with AAC	95.15	99.94	1.40
Video Decoding	*MPEG2 (DVD)*	93.07	99.80	2.38
	MPEG4	95.61	99.94	2.59
	WMV DVD	95.63	99.81	1.96
	WMV HD	95.63	99.84	2.14
Video Encoding	*WMV HD -Pass1*	96.97	99.87	2.08
	WMV HD - Pass2	97.97	99.95	1.82
SPEC2000	*gcc*	88.82	99.54	1.81
	gzip	84.13	99.84	1.52
	mcf	57.02	86.16	8.54
	vortex	97.27	99.73	1.32
	vpr	91.42	98.76	3.17
	art	78.36	87.76	4.73
	equake	86.07	98.93	8.31
	parser	93.44	99.77	1.86
	crafty	93.98	99.94	1.81
	eon	98.56	99.99	2.53
	gap	97.17	99.89	1.40
	perlbmk	97.67	99.89	1.16
	bzip2	96.20	99.55	2.06
	twolf	86.94	98.20	3.36

Table 6: Trace Cache Delivery rate

Application Category	Program Name	Delivery rate	CPI
Audio Decoding	*Winamp with MP3*	86.17	3.11
	RealPlayer with MP3	74.13	3.55
	iTunes with MP3	83.14	1.85
	Winamp with AAC	66.65	2.43
	RealPlayer with AAC	70.32	2.82
	iTunes with AAC	80.07	1.98
Audio Encoding	*Windows Media Player with MP3*	87.26	1.66
	RealPlayer with MP3	79.65	2.02
	iTunes with MP3	87.26	2.07
	RealPlayer with AAC	87.61	1.71
	iTunes with AAC	91.88	1.40
Video Decoding	*MPEG2 (DVD)*	87.61	2.38
	MPEG4	80.97	2.59
	WMV DVD	87.34	1.96
	WMV HD	73.91	2.14
Video Encoding	*WMV HD -Pass1*	78.77	2.08
	WMV HD - Pass2	84.19	1.82
	gcc	80.26	1.81
	gzip	98.20	1.52
	mcf	95.24	8.54
	vortex	85.96	1.32
	vpr	98.76	3.17
	art	98.09	4.73
	equake	52.28	8.31