Performance Characterization of SPEC CPU Benchmarks on Intel's Core Microarchitecture based processor

Sarah Bird[φ], Aashish Phansalkar[φ], Lizy K. John[φ], Alex Mericas[α] and Rajeev Indukuru[α]
[φ]University of Texas at Austin, [α]IBM Austin

Abstract - **The newly released CPU2006 benchmarks are long and have large data access footprint. In this paper we study the behavior of CPU2006 benchmarks on the newly released Intel's Woodcrest processor based on the Core microarchitecture. CPU2000 benchmarks, the predecessors of CPU2006 benchmarks, are also characterized to see if they both stress the system in the same way. Specifically, we compare the differences between the ability of SPEC CPU2000 and CPU2006 to stress areas traditionally shown to impact CPI such as branch prediction, first and second level caches and new unique features of the Woodcrest processor.**

The recently released Core microarchitecture based processors have many new features that help to increase the performance per watt rating. However, the impact of these features on various workloads has not been thoroughly studied. We use our results to analyze the impact of new feature called "Macro-fusion" on the SPEC Benchmarks. Macro-fusion reduces the run time and hence improves absolute performance. We found that although floating point benchmarks do not gain a lot from macro-fusion, it has a significant impact on a majority of the integer benchmarks.

1. Introduction

The fifth generation of SPEC CPU benchmark suites (CPU2006) was recently released. It has 29 benchmarks with 12 integer and 17 floating point programs. When a new benchmark suite is released it is always interesting to see the basic characteristics of programs and see how they stress a state of the art processor and its memory hierarchy. The research community in computer architecture often uses the SPEC CPU benchmarks [4] to evaluate their ideas. For a particular study e.g. memory hierarchy, researchers typically want to know the range of cache miss-rates in the suite. Also, it is important to compare them with the CPU2000 benchmarks. This paper characterizes the benchmarks from the CPU2000 and CPU2006 suites on the state of the art Intel's Woodcrest system. Performance counters are used to measure the characteristics.

Intel recently released "Woodcrest" [9] which is a processor based on the Core microarchiture [6] design. It has multiple new features which improve the performance. One of these features is "Macro-fusion". We evaluate this feature for CPU2006 benchmarks and see if it helps to reduce the cycles of the benchmarks and hence improve performance. The idea behind Macro-fusion [5][7] is to fuse pairs of compare and jump instructions so that instead of decoding 4 instructions per cycle, it can decode 5. This results in effectively increasing the width of the processor. In this paper we measure the macro-fused operations and correlate this measurement with the resulting reduction in time i.e. improvement in performance. We find that the improvement in performance is well correlated to percentage of fused operations for integer programs but not in case of floating point programs. We also study which performance events show a good correlation to the percentage of fused operations.

2. Performance Characterization of SPEC CPU benchmarks

Table 1 (a) shows the instruction mix for the integer benchmarks and (b) shows the same for the floating point benchmarks of CPU2006 benchmarks. Benchmarks *456.hmmer,464.h264ref,410.bwaves,436.cactusADM,433.leslie3D , and 459.GemsFDTD* have a very high percentage of loads. Benchmarks *400.perlbench, 403.gcc, 429.mcf, 458.sjeng, and 483.xalancbmk* show a high percentage of branches. Instruction mix may not always give an idea about the bottleneck for a benchmark but it can give an idea about which parts are stressed by each of the benchmarks.

In this section we see the instructions mix of CPU2006 benchmarks and study how the integer benchmarks behave on the Core microarchitecture based processor. We also show a similar characterization for CPU2000 integer benchmarks.

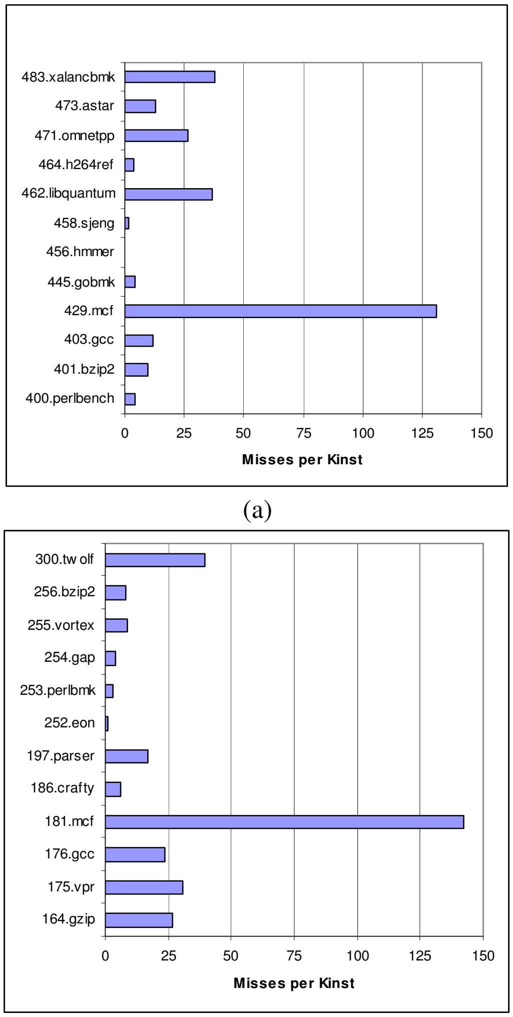
Table 1: Instruction mix for CPU2006 (a) integer benchmarks and (b) floating point benchmarks

(a)

Integer Benchmarks	% Branches	% Loads	% Stores
400.perlbench	23.3%	23.9%	11.5%
401.bzip2	15.3%	26.4%	8.9%
403.gcc	21.9%	25.6%	13.1%
429.mcf	19.2%	30.6%	8.6%
445.gobmk	20.7%	27.9%	14.2%
456.hmmer	8.4%	40.8%	16.2%
458.sjeng	21.4%	21.1%	8.0%
462.libquantum	27.3%	14.4%	5.0%
464.h264ref	7.5%	35.0%	12.1%
471.omnetpp	20.7%	34.2%	17.7%
473.astar	17.1%	26.9%	4.6%
483.xalancbmk	25.7%	32.1%	9.0%

(b)

FP Benchmarks	% Branches	% Loads	% Stores
410.bwaves	0.7%	46.5%	8.5%
416.gamess	7.9%	34.6%	9.2%
433.milc	1.5%	37.3%	10.7%
434.zeusmp	4.0%	28.7%	8.1%
435.gromacs	3.4%	29.4%	14.5%
436.cactusADM	0.2%	46.5%	13.2%
437.leslie3d	3.2%	45.4%	10.6%
444.namd	4.9%	23.3%	6.0%
447.dealII	17.2%	34.6%	7.3%
450.soplex	16.4%	38.9%	7.5%
453.povray	14.3%	30.0%	8.8%
454.calculix	4.6%	31.9%	3.1%
459.GemsFDTD	1.5%	45.1%	10.0%
465.tonto	5.9%	34.8%	10.8%
470.lbm	0.9%	26.3%	8.5%
481.wrf	5.7%	30.7%	7.5%
482.sphinx3	10.2%	30.4%	3.0%



(a)

(b)

Figure 1: (a) Shows L1 data cache misses per 1000 instructions for CPU2006 benchmarks and (b) shows the same for CPU2000 benchmarks

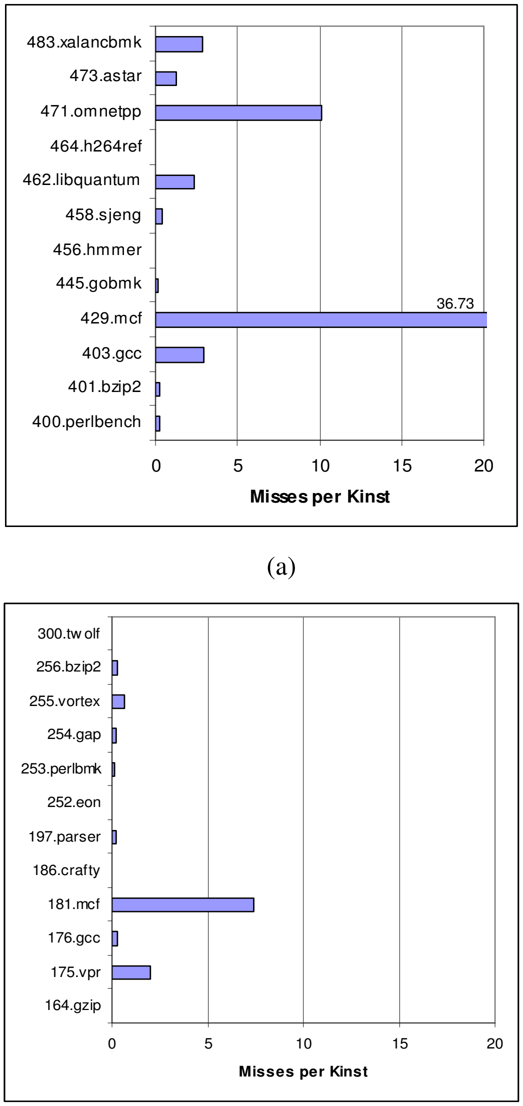
Figure 1(a) shows the L1 data cache misses per 1000 instructions for CPU2006 benchmarks and Figure 1(b) shows similar characterization for CPU2000 benchmarks. It is interesting to see that the *mcf* benchmark in CPU2000 suite shows a higher L1 data cache miss-rate than the *mcf* benchmark in CPU2006. There are total 5 benchmarks in CPU2000 which are close to or greater than 25 MPKI (misses per 1000 instructions) but only 4 benchmarks in CPU2006 which are equal to or greater than 25 MPKI. From the results it appears that the CPU2006 benchmarks do not stress the L1 data cache significantly more than CPU2000. However, it is also important to see how they stress the L2 data cache. In CPU2006 there are benchmarks e.g. *456.hmmr* and *458.sjeng* which show MPKI which is smaller than 5. From Table 1 the program which has the highest percentage of load instructions shows the smallest cache miss rate. This shows that *456.hmmer* exhibits good locality and hence shows a very low cache miss-rate. The same could be said about *464.h264ref* for which the percentage of loads is 42% but the percentage of L1 cache miss-rate is only close to 5 MPKI

Figure 2(a) and 2(b) show the MPKI (L2 cache misses per 1000 instructions for CPU2006 and CPU2000 benchmarks. Even though the L1 cache miss-rates of both suites vary in the same range, there is a drastic increase in the L2 cache miss-rates for CPU2006 benchmarks as compared to CPU2000 benchmarks. This validates the fact that the CPU2006 benchmarks have much bigger data footprint and stress the L2 cache more with the data accesses. Figure 3(a) and 3(b) show the branch mispredictions per 1000 instructions for CPU2006 and CPU2000 benchmarks respectively. It is reasonable to say that there are a few benchmarks in CPU2006 which show worse branch behavior than the CPU2000 benchmarks.

Table 2 shows the correlation coefficients of each of the above measured characteristics with CPI using the characteristics measured for CPU2006 benchmarks. If the value of the correlation coefficient is closer to 1 it has more impact on the performance (CPI). It is evident from Table 2 that the L2 cache misses per 1000 instructions has the greatest impact on CPI of the metrics studied, followed by the L1 misses. The branch mispredictions do not affect the CPI as much which suggests that the benchmarks which show poor branch behavior will not necessarily show worse performance.

Table 2: Correlation coefficients of performance characteristics with CPI

Characteristics	Correlation coefficient
Brach mispredictions per KI and CPI	0.150
L1-D cache misses per KI and CPI	0.918
L2 misses per KI and CPI	0.964



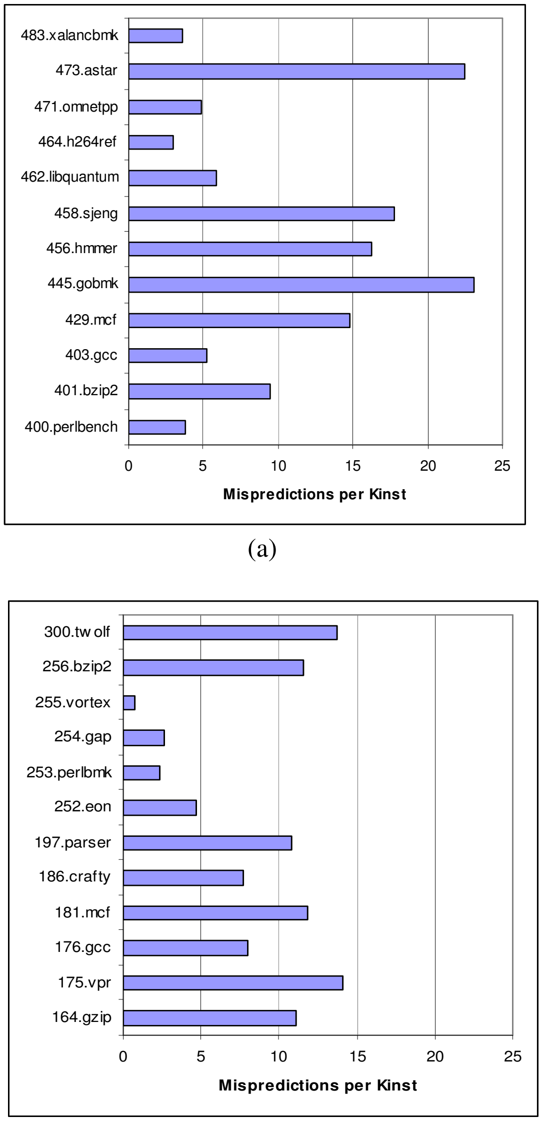
(a)

(b)

Figure 2: (a) Shows the L2 cache misses per 1000 instructions for CPU2006 integer benchmarks and (b) shows the same for CPU2000 integer benchmarks

3. Macro-fusion and Micro-op fusion in the Core Microarchiture based processors

There are several new architectural features that were added in the Core Microarchiture based processors that are different as compared to their predecessors. One such feature is "Macro-fusion" [5][7]. Macro-fusion is a new feature for the Core microarchitecture, which is designed to decrease the number of micro-ops in the instruction stream. The hardware can perform a maximum of one macro-fusion per cycle. Select pairs of compare and branch instruction are fused together during the pre-decode phase and then sent through any one of the four decoders. The decoder then produces a micro-op from the fused pair of instructions. Although it does require some additional hardware, macro-fusion allows a single decoder to process two instructions in one cycle, save entries in the reorder buffer and reservation stations, and allow an ALU to process two instructions at once.



(a)

(b)

Figure 3: (a) Shows the branch mispredictions per 1000 instructions for CPU2006 benchmarks and (b) shows the same for CPU2000 benchmarks

Woodcrest also has another type of fusion called the "Micro-op fusion" which is the enhanced version of the previous fusion that was first introduced on Pentium M [12]. Micro-op fusion [5] [7], which takes place during the decode phase, works by creating a pair of fused micro-ops which are tracked as a single micro-op by the reorder buffer. These fused pairs, which are typically composed of a store/load address micro-op and a data micro-op are issued separately and can be executed in parallel. The advantage of micro-op fusion is that the reorder buffer is able to issue and commit more micro-ops. The remainder of this section describes the methodology of our experiment and then discusses the results.

3.1 Methodology

The cycle count, total fusion (Macro and Micro-op fusion) and macro-fusion for the Core microarchitecture based processor (Woodcrest) were collected using performance counters while running the SPEC CPU2006 benchmarks. The configuration of the Woodcrest processor system is as follows: Tyan S5380 Motherboard with two Xeon 5160 CPUs running at 3.0GHz (although the workload is single threaded) with 4 1GB memory DIMMS at 667MHz. The benchmarks were compiled using Intel C Compiler for 32-bit applications, Version 9.1 and Intel Fortran Compiler for 32-bit applications, Version 9.1. Micro-op fusion was calculated by subtracting the number of macro-fused micro-ops from the total number of fused micro-ops. The time in cycles for the processor code-named Yonah (Intel Core Duo T2500) which is the predecessor of Woodcrest was collected from results published on the SPEC CPU2006 website. The reason why we pick Yonah as our baseline is that it does not have macro-fusion and includes similar architectural features as Woodcrest [5][7]. NetBurst architecture based processor (Intel Pentium Extreme Edition 965) was used as the baseline for the micro-op fusion as well has macro-fusion studies since it does not have both the features and was compared to Woodcrest. The time in cycles for Netburst architecture was also calculated using data from the SPEC website. The number of cycles for each benchmark was calculated based on the runtime for that benchmark and the frequency of the machine. The increase in performance between machines was calculated by subtracting the number of cycles for the benchmark on the Woodcrest machine from the number of cycles on its predecessor and then dividing the result by the original number of cycles. This increase in performance was then used to correlate with the percentage of fused operations. The results of this experiment are discussed in the next subsection.

Table 3: Percentage of fused operations for integer programs of CPU2006

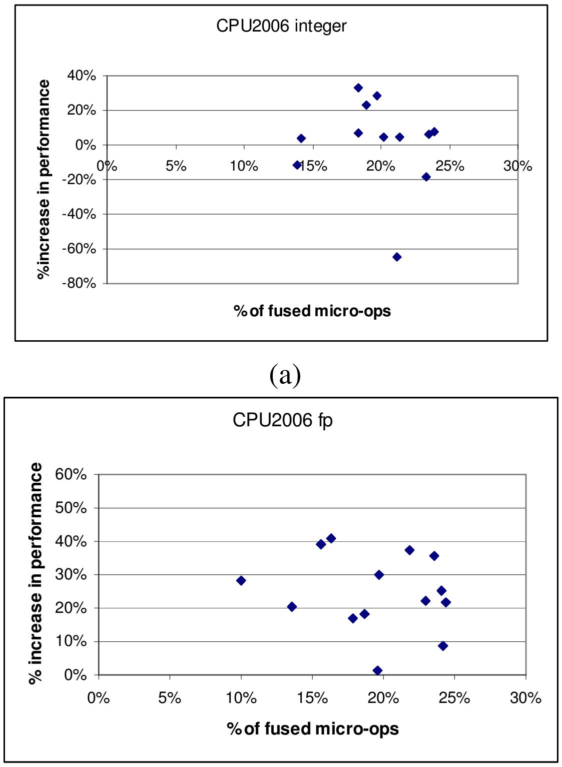
Benchmark	Macro	Micro	Total
400.perlbench	12.18%	19.68%	31.86%
401.bzip2	11.84%	18.95%	30.79%
403.gcc	16.18%	18.39%	34.57%
429.mcf	13.93%	21.40%	35.33%
445.gobmk	11.33%	20.19%	31.52%
456.hmmer	0.13%	23.30%	23.43%
458.sjeng	14.33%	18.32%	32.65%
462.libquantum	1.59%	13.92%	15.51%
464.h264ref	1.51%	23.45%	24.96%
471.omnetpp	8.19%	23.87%	32.06%
473.astar	12.86%	14.20%	27.06%
483.xalancbmk	15.98%	21.12%	37.10%

3.2 Results

Table 3 and Table 4 show the percentage of fused operations for the integer and floating benchmarks in the CPU2006 suite. We can see that the total fusion seen in integer benchmarks is much more than what is seen in the floating point benchmarks. Macro-fusion observed in floating point benchmarks is also drastically less than the integer benchmarks.

Table 4: Percentage of fused operations for floating point programs of CPU2006

Benchmark	Macro	Micro	Total
416.gamess	2.18%	23.58%	25.76%
433.milc	0.35%	17.82%	18.17%
434.zeusmp	0.09%	16.32%	16.41%
435.gromacs	0.71%	15.58%	16.29%
436.cactusADM	0.00%	24.14%	24.14%
437.leslie3d	0.70%	24.12%	24.82%
444.namd	0.36%	10.02%	10.38%
447.dealII	8.03%	22.98%	31.01%
450.soplex	5.10%	13.59%	18.69%
453.povray	4.40%	21.88%	26.28%
454.calculix	0.44%	19.67%	20.11%
459.GemsFDTD	0.38%	18.66%	19.04%
465.tonto	1.69%	24.35%	26.04%
470.ibm	0.22%	19.56%	19.78%



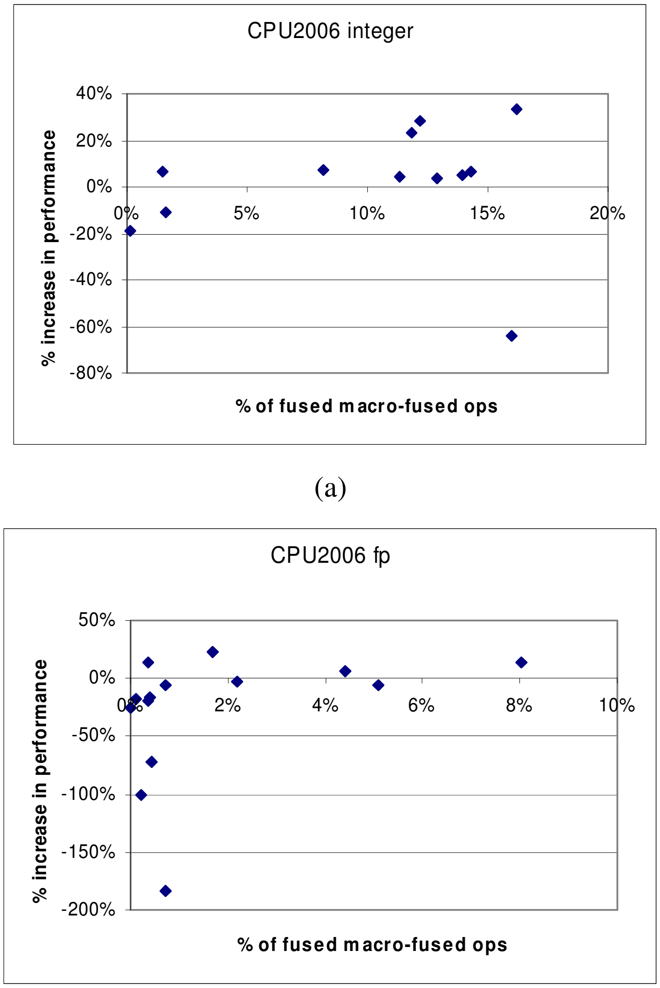
(a)

(b)

Figure 4: Percentage increase in performance and the measured micro-op fusion for (a) CPU2006 integer benchmarks and (b) CPU2006 fp benchmarks

Micro-op Fusion

All of the floating point and integer benchmarks exhibit a high percentage (10-25%) of micro-fused micro-ops. However, in our study we did not find a strong correlation between the percentage increase in performance (calculated as described in Section 3.1) and the measured micro-op fusion. For this reason, the remainder of the fusion results will focus on macro-fusion. Figure 4 (a) and (b) show the plots where we can see that the increase in performance does not correlate well with the percentage of fused micro-ops. For this analysis we used the increase in performance from the NetBurst architecture to the Core microarchitecture based processor.



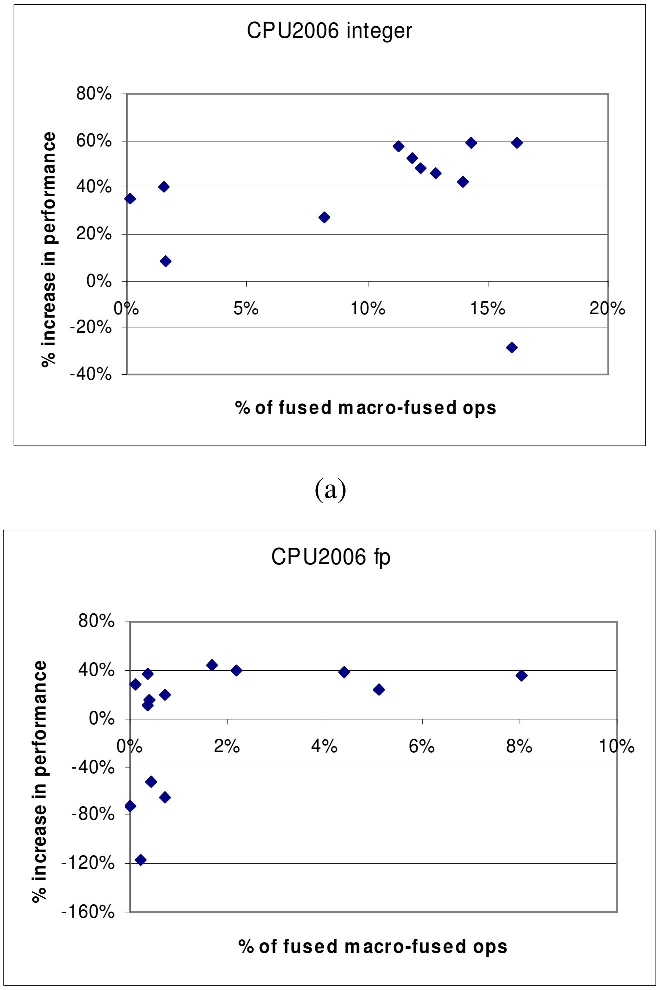
(a)

(b)

Figure 5: Percentage of increase in performance and percentage fused macro-ops for Woodcrest over Yonah for (a) CPU2006 integer benchmarks and (b) CPU2006 floating point benchmarks.

Macro-fusion

Figure 5 shows the plot of percentage increase in performance and the percentage of macro-fused operations for Woodcrest over Yonah. As Table 2 indicates, the percentage of macro-fused operations is below 3% for 11 of the 14 floating point benchmarks analyzed. When studying correlation between the increase in performance and macro-fusion, we found almost no correlation for the floating point benchmarks, which may be due to the low levels of macro-fusion exhibited by the floating point benchmarks. The percentage of macro-fused operations was considerably greater for the integer benchmarks with 9 of the 12 benchmarks exhibiting levels above 8% and show strong correlation. To strengthen our analysis we did a similar experiment for Woodcrest over the NetBurst architecture based processor and the results are shown in Figure 6 (a) and (b).



(a)

(b)

Figure 6: Percentage of increase in performance and percentage fused macro-ops for Woodcrest over NetBurst architecture based processor for (a) CPU2006 integer benchmarks and (b) CPU2006 floating point benchmarks.

We see good correlation between macro-fusion and the increase in performance for integer benchmarks and very weak correlation for floating point.

We also analyze the data to see why the integer benchmarks have a higher percentage of macro-fused operations than the floating point benchmarks. Since macro-fusion involves fusing a compare instruction with the following jump, we see if the percentage of branch instruction correlates with the percentage of macro-fused ops. Figure 7 shows the plot of this analysis with percentage of macro-fused ops against the percentage of branches within a benchmark for the SPEC CPU2006 integer benchmarks. We see a good correlation in Figure 7 and hence can conclude that the opportunity of macro-fusion offered by integer benchmarks is quite uniform and is directly proportional to the percentage of branch operations. There are however three benchmarks for which the correlation is quite weak, *456.hmmer, 462.libquantum* and *464.h264ref.*

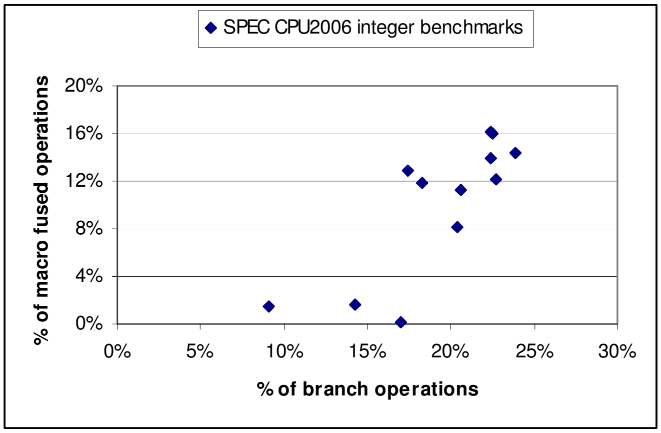


Figure 7: Percentage of macro-fused operations and percentage of branch operations

4. Related Work

New machines and new benchmarks often trigger characterization research analyzing performance characteristics and bottlenecks. Seshadri et.al [2] use program characteristics measured using performance counters to analyze the behavior of java server workloads on two different PowerPC processors. Luo et.al. [1] extend the work in [1] with the data measured on three different systems to characterize internet server benchmarks like SPEC Web99, Volanomark and SPECjbb2000. They also compared the characteristics of the internet servers with the compute intensive SPEC CPU2000 benchmarks to evaluate the impact of front-end and middle-tier servers on modern microprocessor architectures.

Bhargava et.al. [3] evaluate the effectiveness of the x86 architecture's multimedia extension set (MMX) on a set of benchmarks. They found that the range of speedup for the MMX version of the benchmarks ranged from less than 1 to 6.1. This work evaluates the new features added to x86 architecture and also gives an insight as to why there was an improvement in performance using the Intel VTune profiling tool.

There are numerous technically strong and resourceful editorials available on the World Wide Web. We referred to [5][6][7][8][9][10] for information about fusion. But as far as we know this paper is the first of its kind analyzing performance improvement of fusion using measurement on actual machines with and without fusion.

5. Conclusion

Based on characterization of SPEC CPU benchmarks on the Woodcrest processor, the new generation of SPEC CPU benchmarks (CPU2006) stresses the L2 cache more than its predecessor CPU2000. This supports the current necessity for benchmarks based on trends seen in the latest processors which have large on die L2 caches. Since SPEC CPU suites contain real life applications, this result also suggests that the current compute intensive engineering and science applications show large data footprints. The increased stress on the L2 cache will benefit researchers who are looking for real-life, easy-to-run benchmarks which stress the memory hierarchy. However, we noticed that the behavior of branch operations has not changed significantly in the new applications.

The paper also analyzes the effect of a new feature called "Macro-fusion" and "Micro-op fusion" of the Woodcrest processor on its performance. We perform correlation analysis of the increase in performance of the SPEC CPU2006 benchmarks on a processor with both types of fused operations. The results of the correlation analysis showed that the increase in performance of Woodcrest over Yonah as well as NetBurst architecture correlated well with the amount of macro-fusion seen in the integer benchmarks of CPU2006. For floating benchmarks the amount of macro-fusion observed was very low and did not correlate well with the effective increase in performance. Although, we saw interesting results for macro-fusion, we did not find significant correlation between increase in performance and micro-op fusion. For future work, it will be interesting to see how the other new architecture features in the Woodcrest processor correlate with the increase in performance and quantify the contribution of each of them.

Acknowledgments:

The University of Texas researchers are supported in part by National Science Foundation under grant number 0429806 and an IBM University Partnership Award.

References

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

[2] P. Seshadri, A. Mericas "Workload Characterization of Multithreaded Java Servers on Two PowerPC Processors" 4th Annual IEEE International Workshop on Workload Characterization. pp 36-44. December 2001

[3] R. Bhargava, R. Radhakrishnan, B. L. Evans, and L. John "Characterization of MMX Enhanced DSP and Multimedia Applications on a General Purpose Processor" Workshop on Performance Analysis and its Impact on Design. pp 16-23. June 1998

[4] SPEC Benchmarks, http://www.spec.org

[5] D. Kanter "Intel's Next Generation Microarchitecture Unveiled" Real World Technologies. March 2006, http://www.realworldtech.com.

[6] "Intel Core Microarchitecture" Intel, http://www.intel.com.

[7] J. Stokes "Into the Core: Intel's Next-Generation Microarchitecture" Arstechnica. April 2006, http://arstechnica.com.

[8] S. Wasson "Intel Reveals details of new CPU Design" The Tech Report. August 2005, http://www.techreport.com.

[9] W. Gruener "Woodcrest launches Intel into a New Era" TG Daily. June 2006, http://www.tgdaily.com.

[10] W. Gruener "AMD Intros new Opertons and promises 68W Quad-Core CPUs" TG Daily. August 2006, http://www.tgdaily.com.

[11] J. De Gelas "Intel Core versus AMD's K8 Architecture" AnandTech. May 2006, http://www.anandtech.com.

[12] "Intel Core Microarchitecture" PCtech guide. September 2006, http://www.pctechguide.com/21Architecture_Intel_Core.htm