SPEC CPU2006 Sensitivity to Memory Page Sizes

Wendy Korn
IBM
wildkorn@excite.com

Moon S. Chang
IBM
cmoon@us.ibm.com

ABSTRACT

SPEC CPU2006 is a compute-intensive industry standard benchmark suite published in August 2006. This paper characterizes the memory access behavior of SPEC CPU2006 running on IBM POWER5+ microprocessors. We measure the maximum and average memory usage of the benchmarks to validate SPEC's memory requirement criteria. This paper also analyzes how different page sizes affect the performance of the benchmarks. The experiment reveals that 64 KB and 16 MB pages improve the performance up to 46.9% and 50.9%, respectively.

Keywords

SPEC CPU2006 benchmarks, Workload characterization, CPI analysis, Memory usage, Large page size, Performance optimization

1 INTRODUCTION

1.1 SPEC CPU2006 Description and Goals

SPEC has developed a new CPU benchmark suite for the purpose of comparing "compute-intensive workloads on different computer systems" [2]. The resulting suite, CPU2006, is made up of a total of 29 new benchmarks. As part of SPEC's rules, vendors or others wishing to publish results may not make changes to the benchmarks' source code. The remaining parts of the system that can be tuned for a SPEC benchmark run are the compiler, hardware, and operating system. This paper focuses on the suite's memory accesses using the AIX 5L V5.3 operating system on an IBM System p5 with POWER5+ processor.

Two of SPEC's criteria for allowing a benchmark into this suite included: 1) that it spend over 95% of the execution time in the submitted code [10]; and 2) that it use less than 900 MB of memory in 32-bit mode. Although the benchmarks do not use many OS resources, they require effective memory page management techniques to handle the working set size presented by the second criterion. We show how AIX's memory management affects the performance of the benchmarks.

1.2 Related Work

Characterization of memory access behavior has been documented for previous versions of SPEC CPU including CPU2000 [1]. There have been numerous research papers published on large pages and multiple page sizes using SPEC CPU2000 [5] [8]. However, there are many differences between CPU2000 and CPU2006, some of which include new benchmarks, new input sets, and new versions of older benchmarks. Since the release of the suite in August of 2006, there have been few published studies on CPU2006.

2 HARDWARE AND SOFTWARE ENVIRONMENT

2.1 IBM POWER5+ Processor

IBM POWER5+ is a speculative superscalar processor with out-of-order execution capabilities [4] [9]. The processor core contains one instruction fetch unit, one decode unit, two load/store pipelines, two fixed-point execution pipelines, two floating-point execution pipelines, and two branch execution pipelines. The processor core can fetch up to eight instructions per cycle and can dispatch and complete five instructions per cycle.

The POWER5+ multi-core chip contains two processor cores. Each core has a 64 KB L1 instruction cache and a 32 KB L1 data cache. The L1 data cache implements FIFO replacement and store-through policy (all data stored to cache lines that exist in the L1 data cache are also sent to the L2 cache). Two cores on a chip share 1.9 MB of unified L2 cache through a core interface unit. The L3 cache controller and directory is on the chip while the actual 36 MB L3 cache data array is located on a separate chip. The fabric controller is responsible for the data communication between the L2 cache, L3 cache, and other POWER5+ chips.

For the memory management unit, the POWER5+ microprocessor has three types of cache to speed address translation: a translation look-aside buffer (TLB), a segment look-aside buffer (SLB), and an effective-to-real address table (ERAT). The SLB and TLB are only searched when translation cannot be accomplished using the ERAT.

The POWER5+ processor provides for simultaneous multi-threading (SMT), which is a processor design that allows multiple hardware threads to execute multiple instruction streams per cycle [9] [6]. Because the CPU2006 benchmarks are single-threaded (when not using compiler-

Table 1. POWER5+ microprocessor memory hierarchy

Processor component	Size	Organization
IERAT	128 entries	2-way
DERAT	128 entries	fully associative
SLB	64 entries	fully associative
TLB	2048 entries	4-way
L1 Icache	64 KB	2-way, FIFO, 128-byte line
L1 Dcache	32 KB	4-way, LRU, 128-byte line, store-through
L2 cache	1920 KB	10-way unified, 128-byte line, store-back, on-chip
L3 cache	36 MB	12-way unified, 512-byte line, store-back, off-chip

based automatic parallelization), the experiments in this paper were run in single-threaded mode.

2.2 POWER5+ Performance Monitor Unit

The POWER5+ microprocessor provides on-chip logic to monitor processor-related performance events. The POWER5+ Performance Monitor Unit (PMU) contains two dedicated registers that count instructions completed and total cycles as well as four programmable registers which can count more than 300 hardware events occurring in the processor or memory system. Some of the hardware events that can be measured by the dedicated registers include L1/L2/L3 cache misses, TLB misses, branch mispredictions, stall cycles, etc. To acquire the data for this paper, we collected 19 groups including instructions completed, ERAT misses, TLB misses, L1/L2/L3 misses, 4K/64K/16M page references, and other resource stall cycles.

2.3 AIX Support for Multiple Page Sizes

The POWER5+ processor combined with AIX 5L V5.3 recommend Technology Level (TL) 04 supports up to four different page sizes: 4 KB, 64 KB, 16 MB, and 16 GB [3]. The AIX 5L V5.3 TL05 kernel allocates a boot-time-determined number of 4 KB and 64 KB pages for its different segments. For example, the kernel uses 64KB pages for the shared library segments.

A user can specify a different page size to use for each of the three regions of a process's address space: text, data, and stack. The 4 KB and 64 KB pages are supported for all three memory regions, and 16 MB page size is supported for the process text and data regions.

Table 2 System Configuration

Hardware	IBM System p5 520
	2.1 GHz POWER5+
	2 processor chips
	16GB Memory
OS	AIX 5L V5.3 TL05
Compilers	XL Fortran Enterprise Edition 10.01 for AIX
	XL C/C++ Enterprise Edition 8.0 for AIX

3 METHODS

3.1 System Configuration

Table 2 summarizes the system configuration used for the measurement in this paper.

3.2 Enabling Multiple Page Sizes

During CPU2006 development, SPEC's criteria for memory usage required IBM to closely monitor the benchmarks on our systems to measure maximum memory usage.

Originally we saw a performance improvement in AIX 5L V5.3 ML03 when enabling large pages with the `vmo` AIX command [3]. AIX 5L V5.3 ML03 only supported 4KB small pages, 16MB and 16GB large page sizes. AIX 5L 5.3 TL04 and above also support 64KB medium pages.

We studied the suite using different page sizes for each benchmark's data segment to see how AIX's Virtual Memory Manager (VMM) handled the suite's memory accesses.

There are multiple ways to bind a page size to a particular executable.

1. linker options to tag the executable
2. linker tool to tag the executable
3. environment variables

We used the linker tool to modify each executable. We then ran our experiments using three different page sizes:

- 4KB - small pages
- 64KB - medium pages
- 16MB - large pages

3.3 Data Collection

For each page size we collected several types of data:

Runtimes of a "speed" run: We extracted the elapsed runtime for each benchmark from the output file generated by the `runspec` command. A 'speed run' consists of running one instance of the benchmark on a single core.

`svmon` data: For each configuration, we collected statistics with the `svmon` tool. `Svmon` shows the number of pages for the text, data, and stack segments of the kernel, shared libraries, and benchmarks. We took a snapshot of `svmon` data once per second while running the benchmark with the reference input set. We extracted the maximum amount of pages the benchmark used at anytime during the run and noted them, as well as the average across the entire run. We used these metrics for memory usage in megabytes. Refer to "Appendix A: Average and Maximum Memory Usage of CPU2006 Benchmarks" at the end of this paper.

Counter data: Performance monitoring counter data was collected on each benchmark, for each reference input set. For example, we started the event monitors, ran the benchmark using the `specinvoke` tool [11], then when the benchmark finished, we stopped the monitors. The resulting output file shows us the total number of events that occurred during the run. Table 3 lists the 35 counter events that were collected to estimate the CPI (Cycles Per Instruction) breakdown for Power5+ [7].

Table 3 POWER5 Performance Monitor Events

Event Name
PM_GRP_CMPL
PM_RUN_INST_CMPL
PM_RUN_CYC
PM_GCT_NOSLOT_CYC
PM_GCT_NOSLOT_IC_MISS
PM_GCT_NOSLOT_SRQ_FULL
PM_GCT_NOSLOT_BR_MPRED
PM_CMPLU_STALL_LSU
PM_IOPS_CMPL
PM_CMPLU_STALL_REJECT
PM_CMPLU_STALL_DCACHE_MISS
PM_CMPLU_STALL_ERAT_MISS
PM_CMPLU_STALL_FXU
PM_CMPLU_STALL_DIV
PM_CMPLU_STALL_FDIV
PM_CMPLU_STALL_FPU
PM_CMPLU_STALL_FDIV
PM_CMPLU_STALL_FPU
PM_LSU_LMQ_S0_ALLOC
PM_LSU_LMQ_S0_VALID
PM_LSU_SRQ_SYNC_CYC
PM_LWSYNC_HELD
PM_DATA_TABLEWALK_CYC
PM_DATA_FROM_L2
PM_DATA_FROM_L3
PM_DATA_FROM_LMEM
PM_DATA_FROM_RMEM
PM_DATA_FROM_L25_SHR
PM_DATA_FROM_L25_MOD
PM_DATA_FROM_L275_SHR
PM_DATA_FROM_L275_MOD
PM_DATA_FROM_L35_SHR
PM_DATA_FROM_L35_MOD
PM_DATA_FROM_L375_SHR
PM_DATA_FROM_L375_MOD

4 RESULTS

4.1 Memory Usage

Appendix A shows the results of data collection using the `svmon` *AIX* command, with both max and average memory usages. We can see that there is very little difference between the memory requirements for 4K and 64K. The benchmark `436.cactusADM` shows the greatest difference, with 16MB pages. With 64K pages, its increased memory requirement is <1%.

4.2 Runtime Effects of Different Page Sizes

Table 4 shows the runtime improvement gained by using 64KB and 16MB page sizes. We normalized the runtimes by using 4 KB as our baseline to show the improvement when switching to 64 KB and 16MB pages. Thus, the table is read as "The speed for `410.bwaves` improved 51% when using large pages over 4 KB pages".

Almost all of the workloads show a benefit from the larger page sizes. The geometric mean of the speedup is higher with the FP benchmarks, at 11%, whereas INT improves 8%. We expected this result since the FP benchmarks have a larger memory requirement (see Appendix A: Maximum and Average Memory Usage of CPU2006 Benchmarks).

The benchmarks that gained the most speedup were `410.bwaves`, `433.milc`, `450.soplex`, `459.GemsFDTD`, `471.omnetpp`, and `473.astar`. We discuss `471.omnetpp` and `410.bwaves` in more detail in Section 4.3.

Table 4. Normalized Speedup Over 4KB Pages Using 64KB and 16MB

INT	64KB	16MB
400.perlbench	1.008	1.008
401.bzip2	1.047	1.050
403.gcc	1.056	1.056
429.mcf	1.119	1.175
445.gobmk	1.004	1.004
456.hmmer	1.015	1.023
458.sjeng	1.031	1.031
462.libquantum	1.168	1.179
464.h264ref	1.008	1.008
471.omnetpp	1.185	1.190
473.astar	1.179	1.187
483.xalancbmk	1.057	1.072
Geomean	**1.071**	**1.079**

FP	64KB	16MB
410.bwaves	1.469	1.509
416.gamess	1.000	1.000
433.milc	1.289	1.314
434.zeusmp	1.046	1.052
435.gromacs	1.003	1.003
436.cactusADM	0.998	1.018
437.leslie3d	1.163	1.172
444.namd	0.999	0.997
447.dealII	1.055	1.056
450.soplex	1.204	1.219
453.povray	1.000	1.003
454.calculix	1.006	1.006
459.GemsFDTD	1.380	1.407
465.tonto	1.007	1.003
470.lbm	1.142	1.157
481.wrf	1.048	1.051
482.sphinx3	1.080	1.080
Geomean	**1.103**	**1.111**

4.3 CPI Breakdown for Two Benchmarks

In Table 4, we observed that the large page size improved the performance of nearly all the workloads of SPEC CPU2006. To analyze the impact of the large page size in detail, we performed a CPI (cycles per instruction) breakdown analysis on selected workloads: `471.omnetpp` from the integer suite and `410.bwaves` from the floating point suite. These two workloads were selected because they showed the largest performance improvement from INT and FP with large page sizes.

Figure 1 and Figure 2 illustrate the breakdown of CPI components and their contributions to the total CPI. We use the same CPI breakdown model for POWER5+ used in [7]. Note that the legend was put in the same order as the bars. Figures 1 and 2 show that CPI decreases significantly as the page size increases from 4 KB to 64 KB to 16 MB. The primary contributors to the CPI improvement are the reduction in the stall cycles from Dcache and translation misses. Note that stalls from translation misses are mainly caused by ERAT or TLB misses.

Figure 1 shows the CPIs of `471.omnetpp` normalized to that of 4 KB page size. For `471.omnetpp`, the normalized CPI decreases to 0.847 when using 64KB page size. The total CPI further decreases to 0.845 when 16 MB pages are used.

Figure 1 also shows that the `471.omnetpp` workload spends 31.3% and 11.4% of cycles due to Dcache misses and translation misses when using 4 KB page size. When the page size is set to 64 KB, the Dcache stall cycles reduce to 27.7% and the translation stall cycles decrease to as low as 0.3%, which reduced the total CPI by 15.3%. The large page also improves the data prefetch by eliminating the need to restart prefetch operations on 4 KB boundaries [3] [8].

A similar improvement can be observed in the `410.bwaves` workload in Figure 2. With a 4 KB page size for its data segment, the 410.bwaves workload spends 32.4% and 14.1% of cycles due to Dcache misses and translation misses, respectively. When the page size is set to 64 KB, the Dcache stall cycles decreases to 17.5% and the translation stall cycles fall to 0.08%, which reduces the total CPI by 31.7%.

Other than Dcache misses and Translation misses, there were very few improvements when using large pages, which is what we expect. The other stall cycles remain constant as the page size increases. Time spent in Branch predictions and Load/Store execution also change very little.

Figure 1. 471.omnetpp Normalized CPI

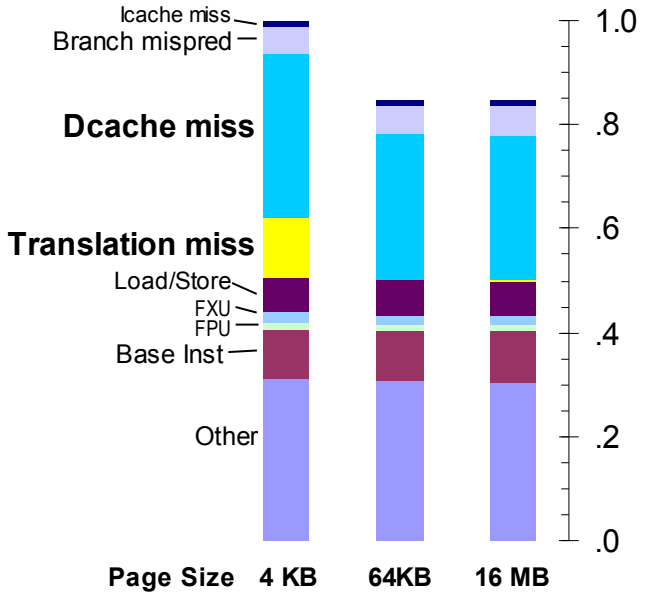
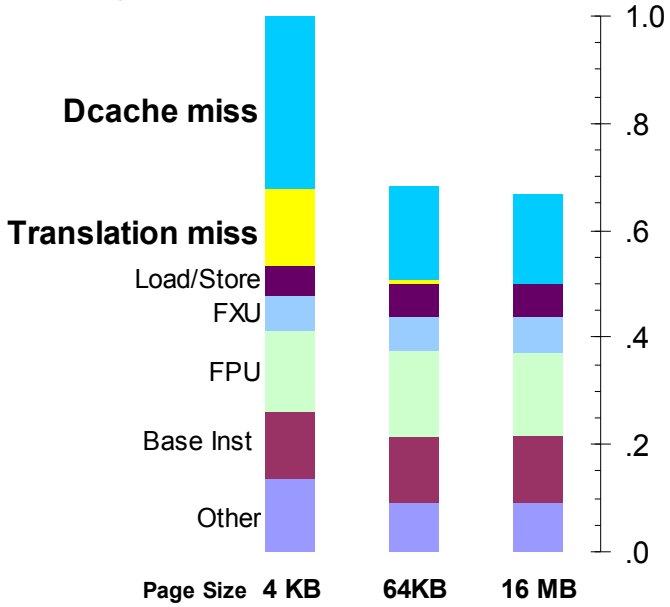


Figure 2. 410.bwaves Normalized CPI



5 CONCLUSIONS

The operating system can supply memory efficiently regardless of 4K or 64K or 64 MB page sizes. We saw that there is little difference with each benchmark's working set size when comparing small and medium page memory usage for 4KB vs. 64KB pages. With 16MB pages, a few benchmarks show increased size.

Since SPEC CPU2006 has larger memory usage than its predecessor, CPU2000, a significant performance improvement was achieved by using large page sizes. The results from the CPI breakdown indicate that the large page sizes reduced Dcache, ERAT, and TLB misses, which in turn contributes to the overall performance improvement of SPEC CPU2006 benchmarks. The experiment showed that 64 KB and 16 MB pages improved the performance up to 46.9% and 50.9%, respectively.

6 ACKNOWLEDGEMENTS

Special thanks to Alan MacKay, our IBM-SPEC CPU representative who helped us get through the development phase of the SPEC suite. Thanks to our AIX gurus Greg Mewhinney and Dirk Michel and to Men-Chow Chiang for the counter script.

7 REFERENCES

[1] Henning, J. SPEC CPU2000: Measuring CPU Performance in the New Millennium. *IEEE Computer*, Vol. 33, No. 7, July 2000.

[2] Henning, J. SPEC CPU2006 Description. *ACM Computer Architecture News*. Vol. 34, No. 4, September, 2006.

[3] Hepkin, D. *Guide to Multiple Page Size Support on AIX 5L Version 5.3*. IBM whitepaper. available at www-03.ibm.com/servers/aix/whitepapers/multiple_page.html

[4] Kalla, R., Sinharoy, B., and Tendler, J.M. IBM POWER5 Chip: A Dual-Core Multithreaded Processor. *IEEE Micro*, Vol. 24, No. 2, 2004.

[5] Kandiraju, G. B., Sivasubramaniam, A. Characterizing the d-TLB Behavior of SPEC CPU 2000 Benchmarks. *Proceedings of the International Conference on Measurement and Modeling of Computer Systems Conference (ACM SIGMETRICS)*, 2002.

[6] Mackerras, P., Matthews, T. S., and Swanberg, R.C. Operating system exploitation of the POWER5 system*, IBM Journal of Research and Development*. Vol. 49, No. 4/5, 2005.

[7] Maron, B., Chen, T., Vianney, D., Olszewski, B., Kunkel S., Mericas, A. Workload Characterization for the Design of Future Servers. *Proceedings of IEEE International Workload Characterization Symposium (IISWC)*, 2005.

[8] Navarro, J., Iyer, S., Druschel, P., and Cox, A. Practical, transparent operating system support for superpages*. Proc. of the 5th Symposium on Operating Systems Design and Implementation (OSDI),* 2002.

[9] Sinharoy, B., Kalla, R.N., Tendler, J.M., Eickemeyer, R.J., and Joyner, J.B. POWER5 system microarchitecture. *IBM Journal of Research and Development*, Vol. 49, No. 4/5, 2005

[10] See the Search program page, archived at http://www.spec.org/cpu2005/search

[11] See the documentation of utility programs for CPU2006, www.spec.org/cpu2006/Docs/utility.html

Appendix A. Average and Maximum Memory Usage (MB) with Various Pagesizes

	4KB pagesize		64KB pagesize		16MB pagesize	
Integer	**AVG**	**MAX**	**AVG**	**MAX**	**AVG**	**MAX**
400.perlbench	288	571	289	571	297	577
401.bzip2	354	847	554	847	569	864
403.gcc	489	924	356	924	366	929
429.mcf	838	838	839	839	848	848
445.gobmk	18	19	19	20	33	33
456.hmmer	19	39	20	39	32	49
458.sjeng	175	175	175	175	177	177
462.libquantum	66	96	67	97	81	112
464.h264ref	36	66	37	67	47	80
471.omnetpp	115	118	116	118	128	129
473.astar	178	304	177	305	187	321
483.xalancbmk	288	323	291	324	294	325
Floating Point						
410.bwaves	873	873	872	874	897	897
416.gamess	5	7	7	9	49	49
433.milc	662	670	662	670	666	673
434.zeusmp	483	484	485	485	495	495
435.gromacs	13	13	15	15	17	17
436.cactusADM	622	623	626	627	1011	1011
437.leslie3d	122	122	123	123	129	129
444.namd	45	45	45	45	49	49
447.dealII	423	634	421	635	429	641
450.soplex	339	604	334	604	349	625
453.povray	2	2	3	3	17	17
454.calculix	159	159	159	159	161	161
459.GemsFDTD	828	829	829	830	835	836
465.tonto	29	33	29	33	32	33
470.lbm	409	409	409	409	416	416
481.wrf	686	692	687	693	697	703
482.sphinx3	52	67	52	67	59	81