CPU 2006 Sensitivity to Memory Page Sizes

Wendy Korn, Moon Chang (IBM)

ACM SIGARCH Computer Architecture News
Vol. 35, No. 1, March 2007

Memory usage

1. Minimum and Maximum memory used
2. Sensitivity to page sizes
3. 4K, 64K, 16M
4. 16GB also supported by AIX but not studied
5. AIX 5L V5.3
6. IBM System p5 with POWER5+ processor
7. Memory criterion for SPEC CPU 2005 selection
8. 95% mem consumed in the code submitted
9. Less than 900MB in 32-bit mode

IBM POWER 5+

Speculative superscalar processor

OOO (Out of order) capabilities

1 fetch unit, 1 decode unit

2 load/store pipes, 2 fixed-point pipes

2 floating point pipes

2 branch execution pipes

Fetch-width

8 instrns per cycle

Dispatch/Complete

5 instrsn per cycle

IBM POWER 5

It’s a multicore chip

2 processor cores per chip

Cache size for local per core L1 caches

64KB – I; 32 KB- D

FIFO replacement

Store-through write policy to L2

Unified, shared 1.9MB L2 cache

36 MB L3 cache

Communication between L2, L3 & other POWER5s

Done by Fabric controller

IBM POWER5 MMU

TLB, SLB, ERAT

TLB - Translation Look-aside Buffer

SLB – Segment Look-aside Bufer

ERAT – Effective to Real Address Table

SLB and TLB are searched only when ERAT cannot accomplish the translation

SMT processor

Simulataneous Multi-threading – multiple hardware threads can run simultaneously

But CPU2006 is single-threaded

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

Table 2 System Configuration

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

POWER5 PMU

PMU – Performance Monitor Unit

2 dedicated registers that count

a.Instructions completed b. cycles

4 programmable registers that can count

4 out of 300+ hardware events from CPU or memory

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.

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_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

AIX Support for Multiple Page Sizes

4 different page sizes supported by AIX 5L V5.3

AIX allocates a boot-time determined number of 4KB and 64KB pages for various segments

3 regions of address space

Text, data, stack

Kernel uses 64KB pages for shared library segments

4KB and 64KB are supported for all 3 regions

16MB supported for text and data regions only

AIX has a command called vmo to enable large pages

Multiple Page Size Support

3 ways to bind a page size to an executable

Linker options to tag the executable

Linker tool to tag the executable

Environment variables

Superpages very common these days (even 1TB)

It is important to understand page behavior in presence of superpages

Counter support exists in most archs

Data Collection

OS commands like svmon and perf-counters used

Elapsed run time (fixed counter)

Speed-run

As opposed to rate–run

Speed-run – means single threaded run

Rate-run means multiple copies of the typically single-threaded SPEC cpu programs

Snapshot of text, data and library regions every second using svmon

Svmon results for maximum and average memory usage (MB)

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

Integer	4KB pagesize		64KB pagesize		16MB pagesize	
	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

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

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

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**

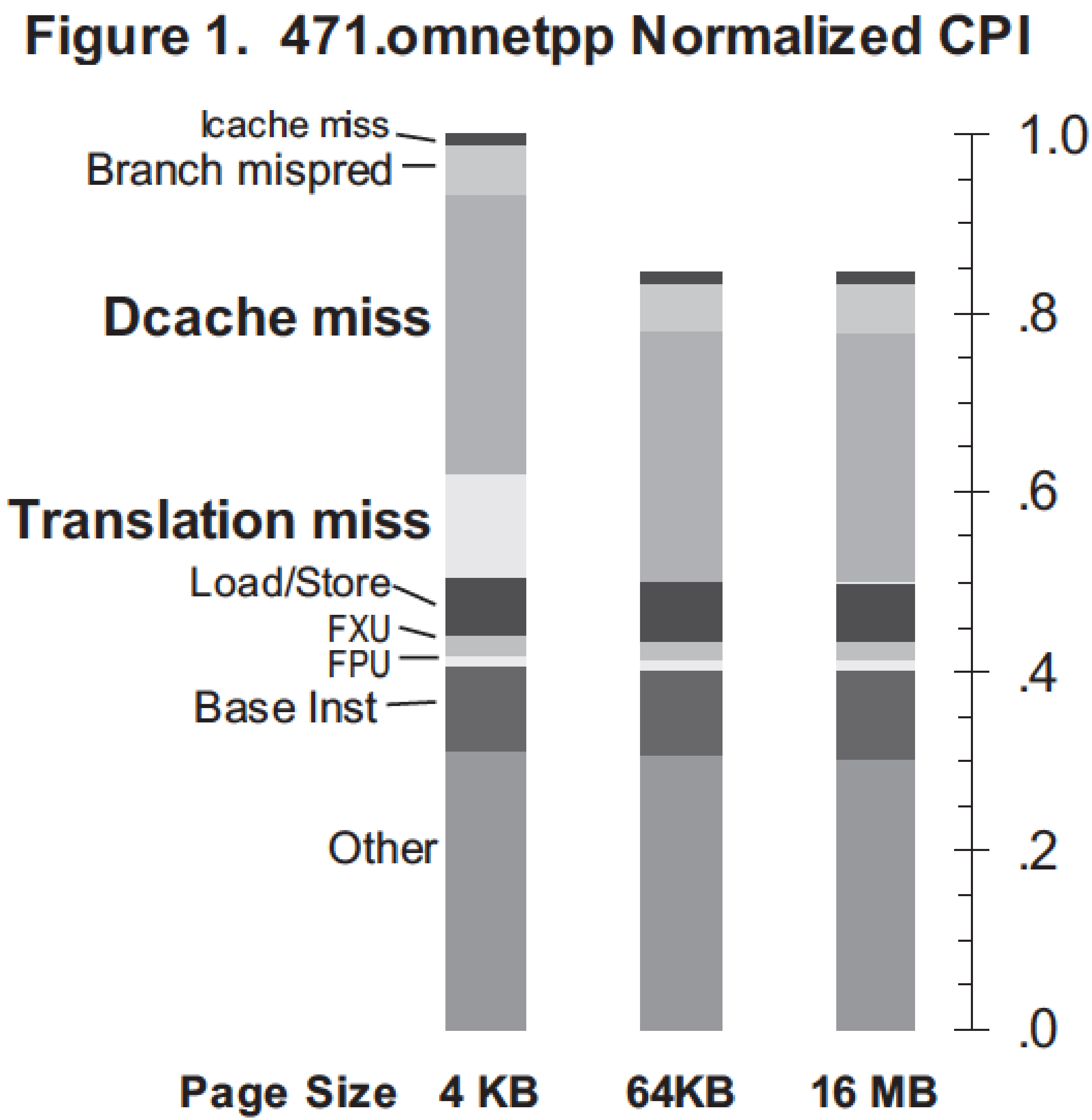
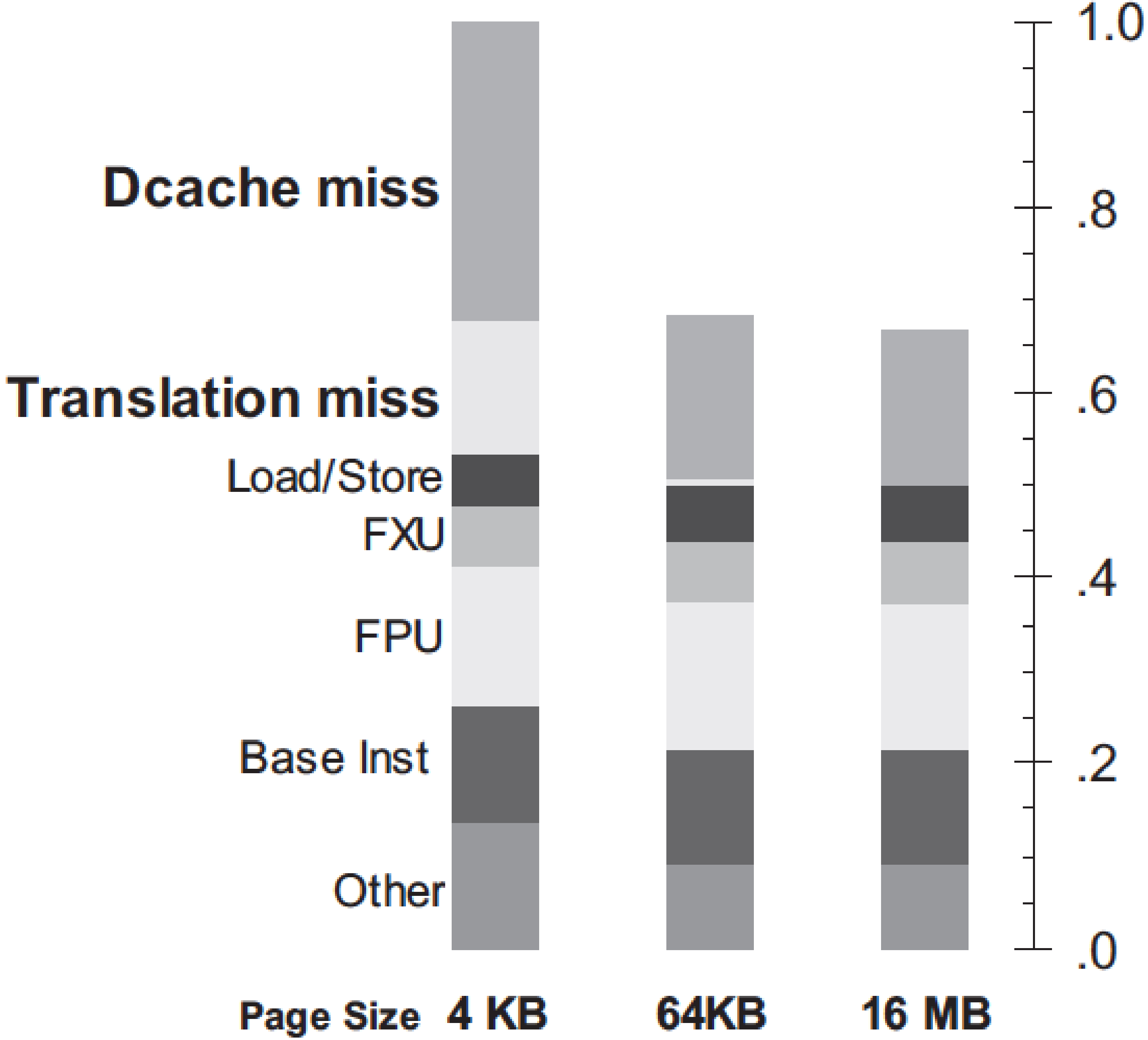


Figure 1. 471.omnetpp Normalized CPI
Icache miss
Branch mispred
Dcache miss
Translation miss
Load/Store
FXU
FPU
Base Inst
Other
1.0
.8
.6
.4
.2
.0
Page Size
4 KB
64KB
16 MB

Figure 2. 410.bwaves Normalized CPI



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 Multi-threaded 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,

Finding a Single Number to indicate Performance of a Benchmark Suite

Lizy Kurian John

AM, GM, HM

$$\overline{x_A} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\overline{x_G} = \sqrt[n]{x_1 x_2 \cdots x_i \cdots x_n}$$

$$= \left(\prod_{i=1}^{n} x_i\right)^{1/n}$$

$$\overline{x_H} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Example illustrating Arithmetic Mean doesn't correctly summarize Speedup

	M1 time	M2 time	Speedup M1 over M2	Speedup M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM			5.05	5.05
GM			1	1

What's wrong with AM?

Normalizing wrto M1 says M2 is 5x faster
normalizing over M2 says M1 is 5x faster.

GM is consistent irrespective of which machine was used as reference

	M1 time	M2 time	Speedup M1 over M2	Speedup M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM			5.05	5.05
GM			1	1

But is GM correct?

GM is consistent but consistently wrong

	M1 time	M2 time	Speedup M1 over M2	Speedup M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM			5.05	5.05
GM			1	1

Why?

Compare execution times

GM is consistent but consistently wrong

	M1 time	M2 time	Speedup M1 over M2	Speedup M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM	500.5	55	5.05	5.05
GM	31.6227766	31.6227766	1	1

Based on execution times, which machine is faster?

Is AM correct or GM correct for exec times? AM

Weighted means

$$\sum_{i=1}^{n} w_i = 1$$

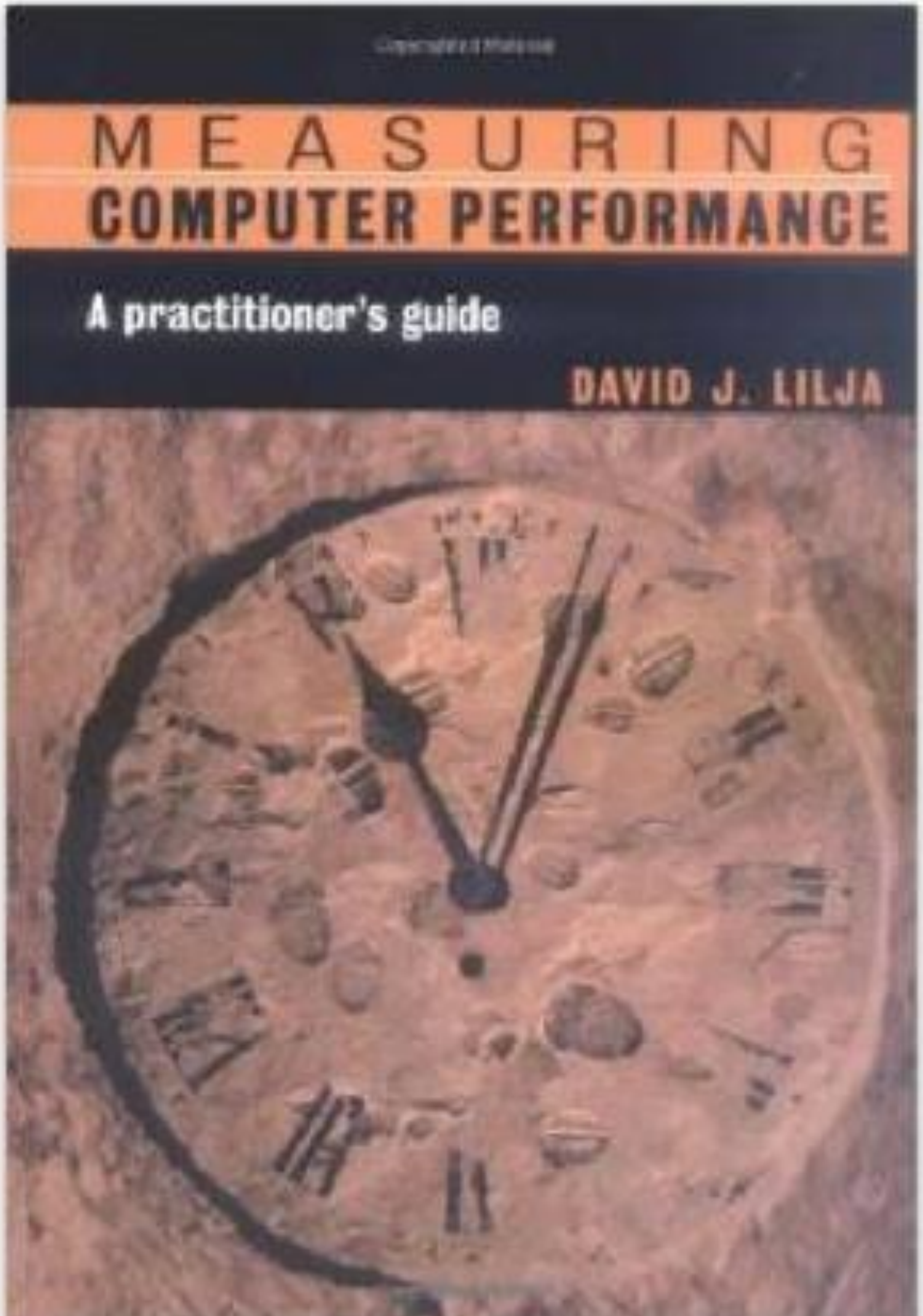
$$\bar{x}_A = \sum_{i=1}^{n} w_i x_i$$

$$\bar{x}_H = \frac{1}{\sum_{i=1}^{n} \frac{w_i}{x_i}}$$

- Standard definition of mean assumes all measurements are equally important
- Instead, choose weights to represent relative importance of measurement *i*

What makes a good mean?

- *Time*–based mean (e.g. seconds)
  - Should be *directly proportional* to total weighted time
  - If time doubles, mean value should double
- *Rate*–based mean (e.g. operations/sec)
  - Should be *inversely proportional* to total weighted time
  - If time doubles, mean value should reduce by half
- Which means satisfy these criteria?


MEASURING
COMPUTER PERFORMANCE
A practitioner's guide
DAVID J. LILJA

Arithmetic mean for times

- Produces a mean value that is *directly proportional to total time*

→ Correct mean to summarize *execution time*

$$\overline{T_A} = \frac{1}{n} \sum_{i=1}^{n} T_i$$

Arithmetic mean for rates

- Produces a mean value that is proportional to *sum of inverse of times*
- But we want *inversely proportional to sum of times*

$$\overline{M_A} = \frac{1}{n}\sum_{i=1}^{n} M_i$$

$$= \sum_{i=1}^{n} \frac{F / T_i}{n}$$

$$= \frac{F}{n}\sum_{i=1}^{n} \frac{1}{T_i}$$

Arithmetic mean for rates

- Produces a mean value that is proportional to *sum of inverse of times*
- But we want *inversely proportional to sum of times*

→ Arithmetic mean is ***not*** appropriate for summarizing rates

$$\overline{M_A} = \frac{1}{n}\sum_{i=1}^{n} M_i$$

$$= \sum_{i=1}^{n} \frac{F / T_i}{n}$$

$$= \frac{F}{n}\sum_{i=1}^{n} \frac{1}{T_i}$$

Harmonic mean for times

- Not directly proportional to *sum of times*

$$\overline{T_H} = \frac{n}{\sum_{i=1}^{n} \frac{1}{T_i}}$$

Harmonic mean for times

- Not directly proportional to *sum of times*

→ Harmonic mean is ***not*** appropriate for summarizing times

$$\overline{T}_H = \frac{n}{\sum_{i=1}^{n} \frac{1}{T_i}}$$

Harmonic mean for rates

- Produces
  (total number of ops)
  ÷ (sum execution times)
- Inversely proportional to total execution time

→ Harmonic mean is appropriate to summarize rates

$$\overline{M_H} = \frac{n}{\sum_{i=1}^{n} \frac{1}{M_i}}$$

$$= \frac{n}{\sum_{i=1}^{n} \frac{T_i}{F}}$$

$$= \frac{Fn}{\sum_{i=1}^{n} T_i}$$

Geometric mean

- Correct mean for averaging normalized values or ratios, right?
- Used to compute SPECmark
- Good when averaging measurements with wide range of values, right?
- Maintains consistent relationships when comparing normalized values
  - Independent of basis used to normalize

But we saw it is consistently wrong

Geometric mean for times

- Not directly proportional to *sum of times*

$$\overline{T_G} = \left(\prod_{i=1}^{n} T_i \right)^{1/n}$$

Geometric mean for times

- Not directly proportional to *sum of times*

→ Geometric mean is ***not*** appropriate for summarizing times

$$\overline{T_G} = \left(\prod_{i=1}^{n} T_i \right)^{1/n}$$

Geometric mean for rates

- Not inversely proportional to *sum of times*

$$\overline{T_G} = \left(\prod_{i=1}^{n} M_i \right)^{1/n}$$

$$= \left(\prod_{i=1}^{n} \frac{F}{T_i} \right)^{1/n}$$

Geometric mean for rates

- Not inversely proportional to *sum of times*

→ Geometric mean is ***not*** appropriate for summarizing rates

$$\overline{T_G} = \left(\prod_{i=1}^{n} M_i \right)^{1/n}$$

$$= \left(\prod_{i=1}^{n} \frac{F}{T_i} \right)^{1/n}$$

Geometric mean for ratios

- Does provide consistent rankings
  - Independent of basis for normalization
- But can be consistently wrong!
- Value can be computed
  - But has no physical meaning

Summary of Means

- Avoid means if possible
  - Loses information
- Arithmetic
  - When sum of raw values has physical meaning
  - Use for summarizing **times** (not rates)
- Harmonic
  - Use for summarizing **rates** (not times)
- Geometric mean
  - Not useful when *time* is best measure of perf

Geometric mean is correct for things with multiplicative relationships

- Prof. Harvey Cragon's architecture book
- Consider a 3-stage amplifier
- Amplifier 1 has stage gains of 2,3,6
- Some design change makes the gains increase to 3,4,7
- What is the gain improvement per stage?
- G.M. of 3/2, 4/3, and 7/6 = 1.326 or 32.6%

Computer Architecture News

A Publication of the Association for Computing Machinery Special Interest Group on Computer Architecture

Vol. 32, No. 1 - March 2004

CORRESPONDENCE

REGULAR CONTRIBUTIONS

Table 2: The mean to be used to find aggregate measure over a benchmark suite from measures corresponding to individual benchmarks in a suite

Measure	Valid central tendency for summarized measure over the suite	
IPC	W.A.M. weighted with cycles	W.H.M. weighted with I-count
CPI	W.A.M. weighted with I-count	W.H.M. weighted with cycles
Speedup	W.A.M. weighted with execution time ratios in improved system	W.H.M. weighted with execution time ratios in the baseline system
MIPS	W.A.M. weighted with time	W.H.M. weighted with I-count
MFLOPS	W.A.M. weighted with time	W.H.M. weighted with FLOP count
Cache hit rate	W.A.M. weighted with number of references to cache	W.H.M. weighted with number of hits
Cache misses per instruction	W.A.M. weighted with I-count	W.H.M weighted with number of misses
Branch misprediction rate per branch	W.A.M. weighted with branch counts	W.H.M. weighted with number of mispredictions
Normalized execution time	W.A.M. weighted with execution times in system considered as base	W.H.M. weighted with execution times in the system being evaluated
Transactions per minute	W.A.M. weighted with exec times	W.H.M. weighted with proportion of transactions for each benchmark
A/B	W.A.M. weighted with B's	W.H.M. weighted with A's

Table 6: Conditions under which unweighted arithmetic and harmonic means are valid indicators of overall performance

	To summarize measure over the suite	
Measure	**When is AM valid?**	**When is H.M. valid?**
IPC	If equal cycles in each benchmark	If equal work (I-count) in each benchmark
CPI	If equal I-count in each benchmark	If equal cycles in each benchmark
Speedup	If equal execution times in each benchmark in the improved system	If equal execution times in each benchmark in the baseline system
MIPS	If equal times in each benchmark	If equal I-count in each benchmark
MFLOPS	If equal times in each benchmark	If equal FLOPS in each benchmark
Cache hit rate	If equal number of references to cache for each benchmark	If equal number of cache hits in each benchmark
Cache misses per instruction	If equal I-count in each benchmark	If equal number of misses in each benchmark
Branch misprediction rate per branch	If equal number of branches in each benchmark	If equal number of mispredictions in each benchmark
Normalized execution time	If equal execution times in each benchmark in the system considered as base	If equal execution times in each benchmark in the system being evaluated
Transactions per minute	If equal times in each benchmark	If equal number of transactions in each benchmark
A/B	If B's are equal	If A's are equal

Many books write that GM is correct for ratios but that is incorrect

			Speedup	Speedup
	M1 time	M2 time	M1 over M2	M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM	500.5	55	5.05	5.05
GM	31.6227766	31.6227766	1	1

Lot of Bad Press for AM but….

	M1 time	M2 time	Speedup M1 over M2	Speedup M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM	500.5	55	5.05	5.05
GM	31.6227766	31.6227766	1	1

Is AM correct or GM correct for exec times?

AM

GM is consistent but consistently wrong

	M1 time	M2 time	Speedup M1 over M2	Speedup M2 over M1
P1	1	10	10	0.1
P2	1000	100	0.1	10
AM	500.5	55	5.05	5.05
GM	31.6227766	31.6227766	1	1

Can you imagine any situation in which GM is correct?

Computer Architecture News

A Publication of the Association for Computing Machinery Special Interest Group on Computer Architecture

Vol. 32, No. 4 - September 2004

REGULAR CONTRIBUTIONS