More on finding a Single Number to indicate Overall Performance of a Benchmark Suite

Lizy Kurian John
Electrical and Computer Engineering Department
The University of Texas at Austin
ljohn@ece.utexas.edu

The topic of finding a single number to summarize overall performance over a benchmark suite is continuing to be a difficult issue 14 years after Smith's paper [1]. While significant insight into the problem has been provided by Smith [1], Hennessey and Patterson [2], Cragon [3], etc, the research community still seems to be unclear on the correct mean to use for different performance metrics. How should metrics obtained from individual benchmarks be aggregated to present a summary of the performance over the entire suite? What are valid central tendency measures over the whole benchmark suite for speedup, CPI, IPC, MIPS, MFLOPS, cache miss rates, cache hit rates, branch misprediction rates, etc?

Arithmetic mean has been touted to be appropriate for time-based metrics, while harmonic mean is touted to be appropriate for rate-based metrics. Is cache miss rate a rate-based metric and hence is harmonic mean appropriate? Geometric mean is a valid measure of central tendency for ratios or dimensionless quantities [3], however, it is also advised that geometric mean should not be used for summarizing any performance measure [1,4]. Speedup, which is a popular metric in most architecture papers to indicate performance enhancement by the proposed architecture is dimensionless and is a ratio-based measure. What will be an appropriate measure to summarize speedups from individual benchmarks?

It is known that weighted means should be used if the benchmarks are not equally weighted. What does equally weighted mean? Does equal weight mean- each benchmark is run once, each benchmark is equally likely to be in a workload of the user, all benchmarks have an equal number of instructions or that all benchmarks run for equal numbers of cycles? Whenever two machines are compared, there is always the question whether the benchmarks are equally weighted in the baseline machine or the enhanced machine. And note that both cannot be true unless each benchmark is enhanced equally.

This paper provides some answers to the above questions – in the context of aggregating metrics from individual benchmarks in a benchmark suite. We show that **weighted arithmetic or harmonic mean can be used interchangeably and correctly provided the appropriate weights are applied.** We give mathematical proofs to establish this.

MIPS as an example

Let us start with MIPS as an example metric. Let's assume that the benchmark suite is composed of n benchmarks and their individual MIPS are known.

We know that the overall MIPS of the entire suite is the total instruction count in millions divided by the total time taken for execution. Hence,

$$\text{Overall MIPS} = \frac{\sum_{i=1}^{n} I_i}{\sum_{i=1}^{n} t_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

where I_i is the instruction count of each component benchmark (in millions) and t_i is the execution time of each benchmark.

Assume $MIPS_i$ is the MIPS rating of each individual benchmark. The overall MIPS is essentially the MIPS when the n benchmarks are considered as parts of a big application. We find that the overall MIPS of the suite can be obtained by computing a Weighted Harmonic Mean (W.H.M) of the MIPS of the individual benchmarks weighted according to the instruction counts or by computing a Weighted Arithmetic Mean (W.A.M) of the individual MIPS with weights corresponding to the execution times spent in each benchmark in the suite. Let us establish this mathematically.

The weights of the individual benchmarks according to instruction counts (ω_i) are $\frac{I_1}{\sum I_i}$, $\frac{I_2}{\sum I_i}$, etc. All summations in this paper are for the n benchmarks as in eq.1, and hence, for compactness we are going to just use the summation sign from now on. The weights of the

individual benchmarks according to execution times (ωt_i) are $\frac{t_1}{\sum t_i}$, $\frac{t_2}{\sum t_i}$, etc. Now,

W.H.M. with weights corresponding to instruction count

$= \frac{1}{\sum \frac{\omega_i}{MIPS_i}}$, where ω_i is the weight of benchmark i according to instruction count……..(2)

$$= \frac{1}{\frac{I_1}{\sum I_i} \cdot \frac{1}{MIPS_1} + \frac{I_2}{\sum I_i} \cdot \frac{1}{MIPS_2} + ...}$$

$$= \frac{1}{\frac{1}{\sum I_i} \sum \frac{I_i}{MIPS_i}}$$

$$= \frac{\sum I_i}{\sum \frac{I_i}{MIPS_i}}$$

$$= \frac{\sum I_i}{\sum \frac{I_i t_i}{I_i}}$$

$$= \frac{\sum I_i}{\sum t_i},$$

which, we know is overall MIPS according to equation 1.

Now, it can be seen that the same result can be obtained by taking a weighted arithmetic mean of the individual MIPS with weights corresponding to the execution times spent in each benchmark in the suite.

W.A.M. weighted with 'time'

$= \sum \omega t_i \cdot MIPS_i$,

where ωt_i is the weights according to execution time

$$= \frac{t_1}{\sum t_i} \cdot MIPS_1 + \frac{t_2}{\sum t_i} \cdot MIPS_2 + \ldots$$

$$= \frac{1}{\sum t_i} \left[t_1 \cdot \frac{I_1}{t_1} + t_2 \cdot \frac{I_2}{t_2} + \ldots \right]$$

$$= \frac{1}{\sum t_i} [\ \sum I_i\ ]$$

$$= \frac{\sum I_i}{\sum t_i}$$

= Overall MIPS

Thus, if the individual MIPS and the relative weights of instruction counts or execution times are known, the overall can be computed. Table 1 illustrates an example benchmark suite with 5 benchmarks, their individual instruction counts, individual execution times and the individual MIPS. Let us calculate the overall MIPS of the suite directly from the overall instruction count and the overall execution time.

Overall instruction count=2000 million
Overall execution time=10 sec
Overall MIPS= 2000/10=200

Table 1: An example benchmark suite with 5 benchmarks, their individual instruction counts, individual execution times and individual MIPS

Benchmarks	Instruction Count (in million)	Time (sec)	Individual MIPS
1	500	2	250
2	50	1	50
3	200	1	200
4	1000	5	200
5	250	1	250

We can also calculate the overall MIPS from the individual MIPS and the weights of the individual benchmarks.

Weights of the benchmarks with respect to I-count
=500/2000, 50/2000, 200/2000, 1000/2000, 250/2000
=0.25: 0.025 : 0.1 : 0.5 : 0.125

Weights of benchmarks with respect to time
= 0.2: 0.1 : 0.1 : 0.5 : 0.1

WHM of individual MIPS (weighted with I-counts)
=1/(0.25/250+0.025/50+0.1/200+ 0.5/200 + 0.125/250)
=200

WAM of individual MIPS (weighted with time)
= 250*0.2 +50*0.1+200*0.1+200*0.5+250*0.1
= 200

Thus, either weighted arithmetic mean or weighted harmonic mean can be used to find overall means, if the appropriate weights can be properly applied. It can also be seen that the simple (unweighted) arithmetic mean or simple (unweighted) harmonic mean are not correct, if the target workload is the sum of the five component benchmarks.

Unweighted AM of individual MIPS =190
Unweighted HM of individual MIPS=131.58

Neither of these are indicative of the overall MIPS. Of course, the benchmarks are not equally weighted in the suite, and hence the unweighted means are not correct.

In general, if a metric is obtained by dividing A by B, if A is weighed equally between the benchmarks, harmonic mean is correct and if B is weighed equally among the component benchmarks in a suite, arithmetic mean is correct while calculating the central tendency of the metric obtained by A/B. In other words, harmonic mean with weights corresponding to the measure in the numerator or arithmetic mean with weights corresponding to the measure in the denominator is valid, when trying to find the aggregate measure from the values of the measures in the individual benchmarks. We use this principle to find the correct means for a variety of performance metrics. This is shown in Table 2.

Somehow there seems to be an impression that arithmetic mean is naïve and useless. Arithmetic mean is meaningless for MIPS or MFLOPS when each benchmark contains equal number of instructions or equal number of floating point operations, however, it is meaningful in many situations. Consider the following situation: A computer runs digital logic simulation for half the time (in a day) and it runs chemistry codes for the other half of the day. A benchmark suite is created consisting of 2 benchmarks, one of each kind. It achieves MIPS1 on the digital logic simulation benchmark and achieves MIPS2 on the chemistry benchmark. The overall MIPS of the target system is the arithmetic mean of the MIPS from the two individual benchmarks and not the harmonic mean.

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

Speedup:

Speedup is a very commonly used metric in the architecture community; perhaps, it is the single most frequently used metric. Let us consider the example in Table 3.

Table 3: An example benchmark suite with 5 benchmarks, their individual execution times on 2 systems under comparison and the individual speedups of the benchmarks

Benchmarks	Time on baseline system	Time on enhanced system	Individual Speedup
1	500	250	2
2	50	50	1
3	200	50	4
4	1000	1250	0.8
5	250	200	1.25

Total time on baseline system=2000sec
Total time on enhanced system=1800 sec

If the entire benchmark suite is run on the baseline system and enhanced system, we know that the

Overall speedup=2000/1800=1.111

Now, given the individual speedups, which mean should be used to find the overall speedup? We contend that the overall speedup can be found either by arithmetic or harmonic mean with appropriate weights. One needs to know the relative weights (with respect to execution time) of the different benchmarks on the baseline and/or enhanced system.

Weights of the benchmarks on the baseline system
=500/2000, 50/2000, 200/2000, 1000/2000, 250/2000

Weights of the benchmarks on the enhanced system
= 250/1800, 50/1800, 50/1800, 1250/1800, 200/1800

WHM of individual speedups (weighted with time on the baseline machine)
=1/(500/(2000*2) + 50/(2000*1) + 200/(2000*4) + 1000/(2000*0.8) + 250/(2000*1.25))
=1/(250/2000+50/2000+50/2000+1250/2000+200/2000)
=1/(1800/2000)
=2000/1800
=1.111

WAM of individual speedups (weighted with time on the enhanced machine) =
=2*250/1800+1*50/1800+4*50/1800+0.8*1250/1800+1.25*200/1800=(500/1800+50/1800+200/1800+1000/1800+250/1800)= 2000/1800
=1.111

Thus, if speedup of a system with respect to a baseline system is available for several programs of a benchmark suite, the W.H.M of the speedups for the individual benchmarks with weights corresponding to the execution times in the baseline system or the W. A. M of the speedups for the individual benchmarks with weights corresponding to the execution times in the improved system can yield the overall speedup over the entire suite.

Now, consider a situation as in table 4.

Table 4: An example where the unweighted A. M. of the individual speedups or the weighted H. M. is the correct aggregate speedup

Benchmarks	Time on baseline system	Time on enhanced system	Individual Speedup
1	200	100	2
2	100	100	1
3	400	100	4
4	80	100	0.8
5	125	100	1.25

Based on execution times, we know that the overall speedup is 905/500, which is equal to the unweighted arithmetic mean of the individual speedups. As you can see each program had equal weight on the enhanced machine. This is indicative of a condition where the workload is not fixed, but all types of workloads are equally probable on the target system. Please note that the same correct answer can be obtained if harmonic mean of individual speedups with weights corresponding to execution times on the baseline system is used.

Next, let us consider a situation as in table 5.

Table 5: An example where the unweighted H.M. of the individual speedups or the weighted A.M. is the correct aggregate speedup

Benchmarks	Time on baseline system	Time on enhanced system	Individual Speedup
1	100	50	2
2	100	100	1
3	100	25	4
4	100	125	0.8
5	100	80	1.25

The overall speedup is 500/380, based on the total execution times in the two systems. It can also be derived from the individual speedups as the unweighted harmonic mean of the individual speedups. In this case, the unweighted harmonic mean is correct because the programs are equally weighted on the baseline system. It may be noted that the same correct answer can be obtained if arithmetic mean of the individual speedups with weights corresponding to execution times on the enhanced system is used.

One might notice that the average speedup is heavily swayed by the relative durations of the benchmarks. It is clear that the relative execution times of the benchmarks in a suite are important. However, how much thought has gone into deciding the relative durations of execution of the different benchmarks? In the SPECINT2000, the baseline running times are 1400, 1400, 1100, 1800, 1000, 1800, 1300, 1800, 1100, 1900, 1500 and 3000 time units for gzip, vpr, gcc, mcf, crafty, parser, eon, perlbmk, gap, vortex, bzips2 and twolf respectively [5]. Apparently these running times were derived based on the time these programs took on a reference machine.

What mean should be used for speedups from SPEC benchmarks? If the aggregate number of interest is the speedup, and if the exact same SPEC benchmark suite is run in its entirety on the new system, then W. H. M. with weights of execution times of each of the benchmarks on the baseline system should be used. This represents the condition where the target workload is exactly the same as the SPEC benchmark suite. If one argues that the relative durations of the SPEC benchmarks in the SPEC suite (as dictated by SPEC) mean nothing to him/her, the unweighted harmonic mean of speedups can be used. If one is interested in knowing the speedup if an imaginary workload with each type of SPEC program is run for equal parts of the day on the target system, the A. M. of the individual speedups should be used.

So if someone summarizes individual MIPS using unweighted harmonic mean, what does it indicate? It is a valid indicator of the overall MIPS of the suite, if every benchmark had equal number of instructions. Since either arithmetic or harmonic mean with corresponding weights is appropriate for most metrics, we can summarize the conditions under which unweighted arithmetic and harmonic means are valid for each metric. Table 6 presents this.

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

Smith uses the meaning "equal work" or equal number of floating point operations for equal weights [1]. Under that condition, Table 6 does illustrate that harmonic mean is the right mean for MFLOPS. Weighted Harmonic Mean with weights corresponding to number of floating point operations or W. A. M with weights corresponding to the execution times of the benchmarks correctly yields the overall MFLOPS.

Ideally, the running times of benchmarks should be just enough for performance metrics to stabilize. Then, while aggregating the metrics, each program should be weighed for whatever fraction of time it will run in the user's target workload. For instance, if program 1 is a compiler, program 2 is a digital simulation, and program 3 is compression, for a user whose actual workload is digital simulation for 90% of the day, and 5% compilation and 5% compression, WAM with weights 0.05, 0.9, 0.05 will yield a valid overall MIPS on the target workload. When one does not know the end user's actual application-mix, if the assumption is that each type of benchmark runs for equal period of time, finding a simple (unweighted) arithmetic mean of MIPS is not an invalid approach.

It appears that everything computer architects deal with can be covered by arithmetic or harmonic mean. So what is geometric mean useful for? Cragon [3] provides an example where geometric mean can be used to find the mean gain per stage of a multi-stage amplifier, when the gains of the individual stages are given. He also illustrates that , if improvements in CPI and clock periods are given, the mean improvement for these two design changes can be found by the geometric mean. Since execution time is dependent on the product of the two metrics considered here, the mean improvement per change can be evaluated by the geometric mean. But geometric mean of performance metrics derived from component benchmarks cannot be used to summarize performance over an entire suite. A general rule is that arithmetic or harmonic means make sense when the component quantities are summed to represent the aggregate situation. The geometric mean is meaningful when the component quantities are multiplied to represent the aggregate situation. Since execution times of component benchmarks are added to find the overall execution time, arithmetic or harmonic means should be used.

In summary, it is possible to summarize performance over a benchmark suite by using arithmetic or harmonic means with appropriate weights. If the metric of interest is obtained by dividing A by B, if A is weighed equally between the benchmarks, harmonic mean is correct and if B is weighed equally among the component benchmarks in a suite, arithmetic mean is correct while summarizing the metric over the entire suite. If speedup of a system with respect to a baseline system is available for several programs of a benchmark suite, the W.H.M of the speedups for the individual benchmarks with weights corresponding to the execution times in the baseline system or the W. A. M of the speedups for the individual benchmarks with weights corresponding to the execution times in the improved system can yield the overall speedup over the entire suite. Geometric mean does not represent anything meaningful while aggregating performance metrics over benchmarks in a suite.

Acknowledgement: The feedback from Jim Smith, David Lilja, Doug Burger and my students in the Laboratory of Computer Architecture helped to improve this manuscript. The author's research is supported in part by the National Science Foundation under grant no. 0113105, and by AMD, Intel, IBM and Motorola Corporations.

References

[1] J. E. Smith, "Characterizing Computer Performance with a Single Number", Communications of ACM, 31(10):1202-1206, October 1988

[2] Patterson and Hennessy, Computer Architecture: The Hardware/Software Approach, Morgan Kaufman Publishers

[3] H. Cragon, Computer Architecture and Implementation, Cambridge University Press

[4] David Lilja, Measuring Computer Performance: A Practitioner's Guide, Cambridge University Press, 2000.

[5] The CPU2000 Results published by SPEC at:
http://www.spec.org/cpu2000/results/cpu2000.html#SPECint