SERT Metric Analysis

Jeremy Arnold
arnoldje@us.ibm.com

August 1, 2016

1 Introduction

The Server Efficiency Rating Tool (SERT)[1] consists of a suite of worklets grouped into multiple workloads. Each worklet runs multiple intervals, or load levels, with performance and power consumption measured during each interval. These are generally designed to exercise some component of the system at different levels of utilization.

The challenge of developing a metric for SERT is to combine all of these individual data points into a single number which appropriately expresses the efficiency of a large number of servers with a vast range of capabilities.

2 Glossary

One of the challenges in describing different metric proposals is a matter of nomenclature – there are a large number of different terms, most of which start with the letters 'w' or 'p' (making meaningful but short variable names difficult). Some of the important terms used by the SERT include:

Worklet one of the specific applications tested as part of a SERT run.

Workload a group of worklets which are intended to exercise a particular component of the system. SERT 1.x consists of 4 workloads (plus the Idle measurement): CPU, Memory, Storage, and Hybrid. Most of the proposed SERT metrics treat the Hybrid/SSJ worklet as part of the CPU workload instead of a separate Hybrid workload.

Suite the complete group of worklets that are part of SERT. SERT 1.x uses a suite of 12 worklets (plus the Idle worklet).

Interval a specific period of measurement while running a worklet. The SERT measures performance and power consumption during each interval.

[1] see `http://www.spec.org/sert`

Calibration a series of measurements used to identify the maximum level of sustained throughput for a particular worklet

The equations used in this document refer to many different values obtained during a SERT run. I've attempted to use consistent variable names across these equations to reduce confusion. These include:

a an index to refer to different worklets (mnemonic: application)

c an index to refer to different workloads (mnemonic: component)

i an index to refer to different measurement intervals

p_{ai} the normalized performance score for worklet a in interval i

w_{ai} the average watts for worklet a in interval i (mnemonic: Watts)

n_a the number of measurement intervals for worklet a

C_a the calibrated normalized performance score for worklet a

E_a an efficiency score for worklet a. Note that there are multiple proposed definitions for E_a within this document. Specific proposed definitions are denoted $E_{\alpha a}$, $E_{\beta a}$, etc

P_a an aggregate performance score for worklet a. Note that there are multiple proposed methods of aggregating performance described within this document, denoted $P_{\alpha a}$, $P_{\beta a}$, etc

W_a an aggregate power measurement for worklet a. Note that there are multiple proposed methods of aggregating power described within this document, denoted $W_{\alpha a}$, $W_{\beta a}$, etc

m_c the number of worklets in workload c

z the number of workloads

F_c an efficiency score for workload c. (mnemonic: F is after E for Efficiency) Note that there are multiple proposed definitions for F_c within this document.

Q an aggregate performance score for all workloads/worklets. (mnemonic: Q is after P for Performance) Note that there are multiple proposed methods of aggregating performance described within this document, denoted Q_α, Q_β, etc

X an aggregate power measurement for all workloads/worklets. (mnemonic: X is after W for Watts) Note that there are multiple proposed methods of aggregating power described within this document, denoted X_α, X_β, etc

y_c is the weight of workload c in the overall efficiency score. By convention, the weights of all of the workloads add up to 1, though with minor changes to the equations this isn't necessary.

S an overall SERT efficiency score (i.e. "the metric"). Note that there are multiple proposed definitions for S within this document, denoted S_α, S_β, etc

3 Worklet performance aggregation

Most SERT worklets (all but the Memory worklets) run three phases of measurements: warmup, calibration, and measurement. The warmup intervals help the system to reach steady state before measurements begin. The calibration phases is used to establish the maximum performance the system can sustain for the worklet. And finally the measurement phase is used for the actual measurements that will contribute to the worklet's score.

Each of these worklets uses a *GraduatedMeasurementSeries* to first run at 100% of the calibrated throughput, and then successively lower percentages of the calibrated throughput, such as 75%, 50%, and 25% (for a worklet with 4 measurement intervals).

Since the expected performance of each interval is a certain percentage of the calibrated throughput (and the SERT validation ensures that this is true within some level of tolerance for expected validation), the performance for each interval can be approximated in terms of the calibrated throughput C_a, as in (1). This observation is used to simplify many of the equations in the sections that follow.

$$p_{ai} \approx C_a \frac{n_a - i + 1}{n_a} \tag{1}$$

There are several possible ways to aggregate the performance results from the multiple intervals into a single aggregate worklet performance score P_a. The sections below describe various methods that may be appropriate for the non-memory worklets. The memory worklets have different characteristics, which are discussed in section 6.

3.1 Peak worklet performance

The simplest of these variations is to use the peak (or calibrated) throughput for the worklet, rather than aggregating all of the interval scores. Because (for the non-memory worklets) each interval runs at some specific percentage of the calibrated throughput, there is no real loss of information by using the calibrated throughput to represent the performance of the worklet. Note that since the 100% measurement interval has performance approximately the same as the calibrated throughput, this 100% performance value could also be used in place of C_a in the calculations that follow.

The aggregate worklet performance in this case is simply $P_{\alpha a} = C_a$.

3.2 Geometric mean of worklet interval performance

A second proposal is to calculate the aggregate worklet performance using the geometric mean of all of the worklet's measurement intervals, as in (2). These two equations are both equivalent calculations of the geometric mean; the product notation (2a) is easier to work with in some cases, while the exponential notation (2b) is easier in others. Throughout the rest of this document, either notation will be used where it is most convenient.

$$P_{\beta a} = \left(\prod_{i=1}^{n_a} p_{ai} \right)^{\frac{1}{n_a}} \tag{2a}$$

$$P_{\beta a} = \exp \left(\frac{1}{n_a} \sum_{i=1}^{n_a} \ln p_{ai} \right) \tag{2b}$$

As in (1), the worklet aggregate performance can be approximated using the calibrated throughput instead of the individual performance values (3).

$$\begin{aligned} P_{\beta a} &\approx \left(\prod_{i=1}^{n_a} C_a \frac{n_a - i + 1}{n_a} \right)^{\frac{1}{n_a}} \\ &\approx \left(\prod_{i=1}^{n_a} C_a \frac{i}{n_a} \right)^{\frac{1}{n_a}} \\ &\approx \left({C_a}^{n_a} \prod_{i=1}^{n_a} \frac{i}{n_a} \right)^{\frac{1}{n_a}} \\ &\approx C_a \left(\prod_{i=1}^{n_a} \frac{i}{n_a} \right)^{\frac{1}{n_a}} \end{aligned} \tag{3}$$

For any worklet with n intervals, the last term in this formula is a constant. For example, for worklets with $n_a = 4$ intervals, $(\prod_{i=1}^{n_a} \frac{i}{n_a})^{\frac{1}{n_a}} = (\frac{1}{4} \cdot \frac{2}{4} \cdot \frac{3}{4} \cdot \frac{4}{4})^{\frac{1}{4}} \approx 0.553$. As a result, for any particular worklet, the aggregated performance score using this method is directly proportional to the calibrated throughput, where the constant of proportionality is dependent on the number of intervals in that worklet. The specific constant values for different numbers of intervals are listed in table 1.

Therefore, for the non-memory worklets, the aggregate worklet performance value will have the same characteristics whether it uses the calibrated throughput or the geometric mean of the performance score in each interval, i.e. $P_{\beta a} \propto P_{\alpha a}$. However, since the CPU workload consists of worklets with different numbers of intervals (4 intervals for each of the 7 CPU worklets but 8 intervals for Hybrid SSJ), the aggregate workload performance value will implicitly weight these worklet scores differently depending on which aggregation method is chosen.

n_a	Equation	Exact Weight	Approx Weight
2	$(\frac{1}{2} \cdot \frac{2}{2})^{(\frac{1}{2})}$	$\frac{\sqrt{2}}{2}$	0.707107
4	$(\frac{1}{4} \cdot \frac{2}{4} \cdot \frac{3}{4} \cdot \frac{4}{4})^{(\frac{1}{4})}$	$(\frac{3}{32})^{\frac{1}{4}}$	0.553341
8	$(\frac{1}{8} \cdot \frac{2}{8} \cdot \frac{3}{8} \cdot \frac{4}{8} \cdot \frac{5}{8} \cdot \frac{6}{8} \cdot \frac{7}{8} \cdot \frac{8}{8})^{(\frac{1}{8})}$	$(\frac{315}{131072})^{\frac{1}{8}}$	0.470544

Table 1: Relative weights of geometric mean of worklet performance based on number of intervals

3.3 Arithmetic mean of worklet interval performance

Another approach to aggregating the worklet interval performance is to use the arithmetic mean of the interval performance values (4).

$$P_{\gamma a} = \frac{1}{n_a} \sum_{i=1}^{n_a} p_{ai} \tag{4}$$

If we again approximate p_{ai} using the calibrated performance C_a (5), we see that for each worklet this value is also directly proportional to the calibrated performance, and therefore $P_{\gamma a} \propto P_{\alpha a}$. Once again, the relative weights of the worklets will change based on the number of intervals.

$$\begin{aligned} P_{\gamma a} &\approx \frac{1}{n_a} \sum_{i=1}^{n_a} C_a \frac{n_a - i + 1}{n_a} \\ &\approx \frac{1}{n_a} \sum_{i=1}^{n_a} C_a \frac{i}{n_a} \\ &\approx \frac{1}{n_a} \frac{C_a}{n_a} \sum_{i=1}^{n_a} i \\ &\approx \frac{C_a}{(n_a)^2} \sum_{i=1}^{n_a} i \\ &\approx \frac{C_a}{(n_a)^2} \frac{n_a(n_a+1)}{2} \\ &\approx C_a \frac{n_a + 1}{2n_a} \end{aligned} \tag{5}$$

The values of the constant term $\frac{n_a+1}{2n_a}$ are shown in table 2.

3.4 Sum of worklet interval performance

An aggregate worklet performance score based on the sum of worklet interval performance is very similar to a score based on the arithmetic mean (6).

n_a	Equation	Weight
2	$\frac{2+1}{2\cdot 2}$	0.75
4	$\frac{4+1}{2\cdot 4}$	0.625
8	$\frac{8+1}{2\cdot 8}$	0.5625

Table 2: Relative weights of arithmetic mean of worklet performance based on number of intervals

$$\begin{aligned}
P_{\delta a} &= \sum_{i=1}^{n_a} p_{ai} \\
&\approx \sum_{i=1}^{n_a} C_a \frac{n_a - i + 1}{n_a} \\
&\approx \sum_{i=1}^{n_a} C_a \frac{i}{n_a} \\
&\approx C_a \sum_{i=1}^{n_a} \frac{i}{n_a} \\
&\approx C_a \frac{1}{n_a} \sum_{i=1}^{n_a} i \\
&\approx C_a \frac{1}{n_a} \frac{n_a(n_a+1)}{2} \\
&\approx C_a \frac{n_a+1}{2}
\end{aligned} \tag{6}$$

As in the previous sections, $P_{\delta a} \propto P_{\alpha a}$, this time with the constants shown in table 3.

n_a	Equation	Weight
2	$\frac{2+1}{2}$	1.5
4	$\frac{4+1}{2}$	2.5
8	$\frac{8+1}{2}$	4.5

Table 3: Relative weights of sum of worklet performance based on number of intervals

3.5 Harmonic mean of worklet interval performance

The harmonic mean is often the appropriate method of aggregation for rates and ratios. Since the performance score reported by each worklet interval is normalized to a reference value, the performance is really a ratio to this reference value. The aggregate worklet performance using the harmonic mean is once again proportional to the calibrated throughput (7), with constant weights shown in table 4.

$$\begin{aligned}
P_{\varepsilon a} &= \frac{1}{\sum\limits_{i=1}^{n_a} \frac{1}{p_{ai}}} \\
&\approx \frac{1}{\frac{1}{n_a} \sum\limits_{i=1}^{n_a} \frac{1}{C_a \frac{n_a - i + 1}{n_a}}} \\
&\approx \frac{1}{\frac{1}{n_a} \sum\limits_{i=1}^{n_a} \frac{1}{C_a \frac{i}{n_a}}} \\
&\approx \frac{1}{\frac{1}{n_a} \sum\limits_{i=1}^{n_a} \frac{n_a}{C_a i}} \\
&\approx \frac{1}{\frac{\frac{1}{n_a} \sum_{i=1}^{n_a} \frac{n_a}{i}}{C_a}} \\
&\approx C_a \frac{1}{\sum\limits_{i=1}^{n_a} \frac{1}{i}}
\end{aligned} \tag{7}$$

n_a	Equation	Exact Weight	Approx Weight
2	$\frac{1}{\frac{1}{1}+\frac{1}{2}}$	$\frac{2}{3}$	0.666667
4	$\frac{1}{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}$	$\frac{12}{25}$	0.48
8	$\frac{1}{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}}$	$\frac{280}{761}$	0.367937

Table 4: Relative weights of harmonic mean of worklet performance based on number of intervals

4 Worklet power aggregation

Worklet power consumption can be aggregated in similar ways to the performance, but there are a few key differences which reduce the number of options to be considered.

1. The power consumption in each measurement interval is not proportional to the peak value. In fact, this is one of the key characteristics that an energy efficiency metric should reflect: how much is the system able to change power consumption to reflect changing utilization of the system's resources. So the approximations that simplified the worklet performance aggregation are not applicable for power.

2. The power consumption is an actual measured value, and not normalized to a reference system. So it is not a ratio or rate, and it is not appropriate to use the harmonic mean.

3. The power consumption is in the same range across all worklets. Some worklets may have vastly different performance scores; while these performance scores are relative to a single reference system, a particular server may perform particularly well on specific worklets, so the range of normalized worklet performance scores may vary by at least an order of magnitude. The power measurements, however, will have a tighter range between the idle and max power of the server.

4.1 Geometric mean of worklet interval power

One way to aggregate the power scores for a worklet is by taking the geometric mean of the average power in each measurement interval (8).

$$W_{\beta a} = \exp\left(\frac{1}{n_a}\sum_{i=1}^{n_a} \ln w_{ai}\right) \tag{8}$$

As described above, it is not possible to simplify this formula based on the peak power, since the power consumption in each interval can generally not be estimated based on the peak power consumption.

4.2 Arithmetic mean of worklet interval power

The power can also be aggregated using the arithmetic mean of the worklet interval power (9).

$$W_{\gamma a} = \frac{1}{n_a}\sum_{i=1}^{n_a} w_{ai} \tag{9}$$

4.3 Sum of worklet interval power

Finally, the power can be aggregated as the sum of the power for each worklet interval (10).

$$W_{\delta a} = \sum_{i=1}^{n_a} w_{ai} \tag{10}$$

4.4 Summary

While there are several possible ways to aggregate the power scores for a worklet into an aggregate worklet power value, there are some advantages to using the geometric mean. In particular, the geometric mean naturally rewards results where the system is able to conserve power at low utilizations. This is due to the property of the geometric mean regarding numbers subjected to a mean-preserving spread. If two sets of power values have the same arithmetic mean, but one of the sets has values that are more spread out while the other set has values that are close together, the geometric mean will be lower for the set that is more spread out.

Systems that reduce power aggressively at lower utilizations will have interval power values that are more spread out, and therefore the geometric mean of these values will be lower than it would be for a system that uses the same amount of power regardless of the utilization. Since many servers spend most of their time at less than full utilization, systems whose energy usage is proportional to their utilization will generally be more efficient, so it is appropriate that the aggregated worklet power reflect this with a lower average power consumption.

5 Calculating aggregate workload scores

5.1 SERT 1.1.x Score Calculations

Historically, the SERT will calculate an efficiency score E_a for each worklet using (11).

$$\begin{aligned} E_{\alpha a} &= \frac{\sum_{i=1}^{n_a} p_{ai}}{\sum_{i=1}^{n_a} w_{ai}} \\ &= \frac{P_{\delta a}}{W_{\delta a}} \end{aligned} \tag{11}$$

Actually, the SERT efficiency score is 1000 times this value. This factor is just used to convert the resulting value to a range that is easier for humans to read. Similar scaling factors could be applied to any of the metric proposals in this document, so these scaling factors have been left out to simplify the equations.

Obviously, we could use other calculations for E_a that formulate a worklet efficiency score based on other definitions of P_a and W_a, such as using the geometric mean of the interval values rather than the sum.

Then a workload efficiency score is calculated using the geometric mean of the worklet efficiency scores (12).

$$F_{\alpha c} = \left(\prod_{a=1}^{m_c} E_{\alpha a} \right)^{\frac{1}{m_c}} \tag{12}$$

This workload efficiency score could also be applied to other definitions of E_a.

SERT 1.1.x does not compute a single overall score S. The most straightforward calculation would be a weighted geometric mean of the workload efficiency scores, with weights for each workload given by y_c (13).

$$S_\alpha = \exp\left(\sum_{c=1}^{z} y_c \ln F_c\right) \tag{13}$$

5.2 TGG Workload Aggregations

TGG has proposed calculating an overall efficiency score directly from aggregated performance and power (14).

$$S = \frac{Q}{X} \tag{14}$$

This requires defining aggregate performance (Q) and aggregate power (X) values.

5.2.1 Aggregate Performance

There are several advantages to using the geometric mean to aggregate normalized performance data. In particular, the geometric mean preserves the ranking of results regardless of which results are used to determine the normalization factors [1]. SPEC has a long history of using the geometric mean to calculate a combined score for a suite of benchmark applications. The use of the geometric mean also limits the influence of outliers that may have a large effect on the arithmetic mean.

TGG proposes aggregating the performance of the worklets in each workload using the geometric mean of those worklet performance scores (P_a), and then using a weighted geometric mean to combine the workload performance into a single overall aggregate performance value Q, as in (15).

$$Q_\beta = \exp\left(\sum_{c=1}^{z}\left(y_c \ln\left(\exp\left(\frac{1}{m_c}\sum_{a=1}^{m_c} \ln P_a\right)\right)\right)\right) \tag{15}$$

This generic formula can be applied to any of the aggregate worklet performance score calculations described in section 3. Each of those worklet performance score calculations were shown to be directly proportional to the calibrated performance value (for the non-memory worklets), so the general characteristics of the aggregate performance based on any of these scores are quite similar. But the choice of worklet performance aggregation method does influence the relative weighting of the various worklets in the overall score. It is important to note that none of the weightings are necessarily the “correct” weightings.

5.2.2 Aggregate Power

There are two reasonable choices for aggregating the power measurements from multiple worklets: using either the geometric mean (16) or the arithmetic mean

(17). The main argument for the arithmetic mean is that all of the power measurements are measurements of the power consumption of the server, and all are in the same general range. Again, either of these generic calculations could be used with any of the worklet power aggregation formulas described in section 4.

$$X_\beta = \exp\left(\sum_{c=1}^{z}\left(y_c \ln\left(\exp\left(\frac{1}{m_c}\sum_{a=1}^{m_c}\ln W_a\right)\right)\right)\right) \tag{16}$$

$$X_\gamma = \sum_{c}\left(y_c \exp\left(\frac{1}{m_c}\sum_{a=1}^{m_c}\ln W_a\right)\right) \tag{17}$$

However, there are actually important differences in the power consumption by some of the worklets. In particular, the memory and storage worklets do not have much variation in CPU usage or power consumption in different intervals. As a result, the aggregate worklet power (S_a) is generally much higher for the memory worklets than it is for the CPU worklets (where the different measurement intervals typically have significantly different power consumption) or the Storage worklets (which run at near idle power for many system configurations).

Due to the difference in behavior among the different worklets, the geometric mean is the most appropriate choice for aggregating power consumption. The geometric mean reduces the influence of outlier values, providing a value that is meaningful across a range of worklets and load levels.

5.3 Comparison of workload aggregation methods

One of the useful properties of the geometric mean is that the ratio of two geometric means is equivalent to the geometric mean of the ratio (18). (This is not the case for the arithmetic mean or the harmonic mean.)

$$GM\left(\frac{A_i}{B_i}\right) = \frac{GM(A_i)}{GM(B_i)} \tag{18}$$

As a result, when the efficiency score is based on the geometric mean of performance and geometric mean of power (as in (15) and (16)), the SERT and TGG calculations for efficiency score are actually equivalent. This property applies regardless of how the worklet performance P_a and worklet power W_a are calculated.

While the two methods both have the same end result, the SPEC method has the advantage of being more intuitive. The concept of a worklet efficiency score (performance per Watt for that worklet) is reasonably intuitive, and it is a simple extension to calculate the workload efficiency score as the geometric mean of these worklet efficiency scores.

6 Memory Worklet Performance Aggregation

The formulas presented in section 3 make use of the observation in (1) to express the performance in each interval as a percentage of the calibrated throughput. This is used to show that several different methods of aggregation produce results that are directly proportional to the calibrated throughput.

There are some differences, however, for the Memory worklets. Mathematically, the Flood2 interval performance scores behave similarly to the CPU worklets (though the actual runtime behavior is not the same), but the Capacity2 worklet has a distinctively different behavior.

6.1 Flood2

The Flood2 worklet runs two intervals (Flood_Full and Flood_Half). In both cases, the worklet iterates through large in-memory arrays and performs various operations on the data. The difference between the two intervals is that in Flood_Full, nearly all of the physical memory on the system is allocated to these arrays, while in Flood_Half the arrays are only half of the size. So the work being performed is identical in both intervals, but Flood_Half will take approximately half of the time since it has less memory it has to work on. The CPU utilization (and therefore the power consumption) tends to be high, and there is little difference in power consumption between Flood_Full and Flood_Half.

The performance result for Flood_Half is approximately half of the performance of Flood_Full. Therefore, the worklet performance aggregation formulas described in 3 work, and they can be approximated using the peak performance (the performance of Flood_Full) as the CPU worklets can. However, it is important to remember that unlike the CPU worklets, the power consumption for Flood2 will be approximately the same for both Flood_Full and Flood_Half, which will result in different characteristics of the efficiency metric calculated from these two intervals.

6.2 Capacity2

The Capacity2 worklet scales differently than either Flood2 or the other worklets in the SERT. Many enterprise applications benefit from systems with large amounts of memory because it allows them to cache data. They may be able to run with smaller amounts of memory, but performance will suffer because data has to be retrieved or re-processed instead of being obtained from the application's in-memory cache. The Capacity2 worklet mimics this class of applications to measure the increased efficiency of systems with larger amounts of memory.

Each Capacity2 load level (measurement interval) runs using progressively larger data set sizes. For small data sets, most systems will be able to keep most or all of the data set in memory. As a result, most of the computations performed by the worklet are cached, and the system will get a high performance score. When the data set is larger than physical memory, only a portion of the data set can be cached; when a non-cached value is accessed, a CPU-intensive computation will have to be performed. This will require more time, and the performance score will be lower as a result.

So the typical pattern in Capacity2 results is that for small data set sizes (Capacity_4, Capacity_8, etc) the performance result will be approximately the same. At large memory sizes (e.g. Capacity_512, Capacity_1024) where the data set size is greater than available physical memory, the score will be much lower, since most transactions have to perform the extra computation steps. Somewhere in between is a crossover point where the results will be somewhere in between.

Therefore, it is important that the aggregate performance score for Capacity2 incorporates the results from all of the intervals, and not just a "peak" value. As long as the aggregate performance includes all of the intervals, then systems with larger amounts of physical memory will have a higher aggregate performance score than systems with smaller amounts of memory, because these systems will have a greater number of Capacity2 intervals that are able to return most of their data from the application cache.

Because of the nature of the Capacity2 worklet, there is a non-linear relationship in the performance/power ratio at each test interval which complicates the integration of the Capacity2 results into a combined, single efficiency metric. Additional work is needed to determine how these results can be combined appropriately.

References

[1] Philip J. Fleming and John J. Wallace. How not to lie with statistics: The correct way to summarize benchmark results. *Commun. ACM*, 29(3):218–221, March 1986.