Standard Performance Evaluation Corporation (SPEC®)

The SERT® 2 Metric and the Impact of Server Configuration

7001 Heritage Village Plaza, Suite 225
Gainesville, VA 20155,
USA

SPEC RG Power Working Group

Table of Contents

This document version: https://www.spec.org/sert2/SERT-metric-20210331.pdf
Latest SERT 2.x.x version: https://www.spec.org/sert2/SERT-metric.pdf
Previous version: https://www.spec.org/sert2/SERT-metric-20191108.pdf

Abstract

The SERT Suite is an industry standard for measuring and analyzing the energy efficiency of servers. It measures server efficiency using multiple workloads, which in turn consist of small scale mini-workloads called worklets. Using multiple worklets enables the SERT suite to holistically explore the behavior of many different systems and enables thorough analysis. However, multiple workloads also result in multiple energy efficiency scores. This document introduces the single SERT 2 metric (SERT 2 Efficiency Score), which can be used to easily compare systems using a single number. This document explains how the SERT 2 metric is calculated. It also illustrates how a system under test (SUT) configuration and changes to this configuration can impact the SERT 2 Efficiency Score and demonstrates this using a running example.

1. Introduction

The SERT Suite is an established industry standard for measuring and analyzing the energy efficiency of servers. To this end it is used by industry, regulators, and researchers. The SERT suite measures server efficiency using multiple workloads, which in turn are executed by running multiple small scale mini-workloads, called worklets, in series. Using multiple worklets enables the SERT suite to holistically explore the behavior of many different systems and enables thorough analysis. However, multiple workloads also result in multiple energy efficiency scores. In past versions, SERT reports separate scores for each of its workloads. While this enables thorough analysis, it makes comparing different SERT results difficult. Specifically, formal criteria for server labeling and classification by regulators can be greatly aided by providing a single score metric.

This document introduces the SERT 2 metric, also called the SERT 2 Efficiency Score, which is a single-number metric that characterizes a complete SERT result. This metric was introduced in SERT 2.0 suite and is designed to showcase a system's overall energy efficiency during operation and is used by regulators to define formal server labeling criteria. This document describes the metric in detail, explaining each step of the metric's calculation.

Considering the metric's use by regulators, achieving reproducible and representative scores is of great importance to many testers. As scores can change greatly depending on system configuration, this document provides guidelines and examples on system configuration and test execution with the goal of enabling testers to consistently achieve reliable and representative scores for their specific systems.

This document is structured as follows: Section 2 outlines the design of the SERT suite, its workloads and worklets. Section 3 then introduces the SERT 2 metric, explains how to calculate it, and provides examples for metric calculation. It also provides factors that may influence SERT results. Finally, this document concludes in Section 4.

1.1. Authors

Written by the SPEC RG Power Working Group:

Jóakim von Kistowski	joakim.kistowski @ uni-wuerzburg.de	University of Würzburg
Klaus-Dieter Lange	klaus.lange @ hpe.com	Hewlett Packard Enterprise
Jeremy A. Arnold	arnoldje @ us.ibm.com	IBM Corporation
Hansfried Block	hansfried.block @ t-online.de	Freelance
Greg Darnell	greg.darnel @ intel.com	Intel Corporation
John Beckett	john_beckett @ dell.com	Dell Inc.
Mike Tricker	mike.tricker @ microsoft.com	Microsoft Corporation

1.2. Trademark and Copyright Notice

2. SERT Workloads and Worklets

The SERT suite features three separate workloads for CPU, Memory, and Storage. Each of these workloads consist of a number of separate mini-workloads, called worklets. Each of the worklets is executed at several load levels. The SERT load levels are defined using the transaction rate at which the worklet is run. Each worklet features transactions, which are the basic work units performed by the worklet (e.g., one array being compressed is one transaction for the Compress worklet). The throughput (amount of transactions performed per second) is the main performance metric for each SERT worklet. The maximum (100%) load level is then defined as the load level at which as many transactions as possible are executed per second. Lower load levels are achieved by deliberately waiting in between separate transactions in order to achieve lower performance and lower system utilization. E.g., at the 75% load level only 75% of the maximum amount of transactions possible is executed each second. Please note, that load levels do not equal CPU utilization (this is a common misconception). A load level of 50% only means that a worklet only runs 50% as many transactions as compared to full load, it does not mean that the CPU is utilized at 50%. Also consider that some worklets, such as the Storage worklets, are not designed to fully utilize the CPU, yet still have a 100% load level.

Each worklet's performance and power consumption are measured separately for each load level and the energy efficiency and normalized energy efficiency are calculated from the measurement results. The workload score is an aggregate of all the worklet scores and provides a workload efficiency score on how well the tested system performed for all the worklets in the specified category.

The following subsections define and describe each of the workloads in detail. Each of the workloads was designed so that it primarily stresses the server subsystem after which it was named (CPU workload stresses CPU, Memory workload stresses memory, etc.). However, it is important to keep in mind that workloads do not exclusively stress that subsystem. Workloads also measure and characterize the energy efficiency of interactions between multiple subsystems. To that end the CPU workload also utilizes some memory, the memory workload utilizes some CPU, and the storage workload utilizes some CPU and memory.

2.1. CPU

The SERT design document [1] defines CPU worklets through the following properties:

- A worklet requires consistent processor characteristics per simulated ''user'' regardless of the number of processors, cores, enabled threads, etc.
- At the 100% load level, the performance bottleneck is the processor subsystem.
- A worklet's performance should increase with increasing amount of processing resources, such as the number of physical CPUs, the number of cores, possibly the number of logical processors, higher clock rate, larger available cache, lower latency, and faster interconnect between CPU sockets.

The SERT suite features a total of seven different CPU worklets, which we describe in short in this section. The performance metric employed for each of these worklets is throughput measured in transactions per second. Each CPU worklet, with the exception of SSJ, is executed at target loads of 100%, 75%, 50%, and 25% by default. The worklets are:

1. **Compress:** Implements a transaction that compresses and decompresses data using a modified Lempel-Ziv-Welch (LZW) method following an algorithm introduced in [2]. It finds common substrings and replaces them with a variable size code. This is deterministic and it is done on-the-fly. Thus, the decompression procedure needs no input table, but tracks the way the initial table was built.
2. **CryptoAES:** Implements a transaction that encrypts and decrypts data using the AES block cipher algorithms using the Java Cryptographic Extension (JCE) framework. Consequently, CryptoAES may use the specialized AES instructions of modern CPUs.
3. **LU:** Implements a transaction that computes the LU factorization of a dense matrix using partial pivoting. It exercises linear algebra kernels (BLAS) and dense matrix operations.
4. **SHA256:** Utilizes standard Java functions to perform SHA-256 hashing transformations on a byte array. This byte array is perturbed by one byte for each transaction so that each transaction operates on different input data and generates a different hash value.

5. **SOR (Jacobi Successive Over-Relaxation):** Implements a transaction that exercises typical access patterns in finite difference applications, for example, solving Laplace's equation in 2D with Drichlet boundary conditions. The algorithm exercises basic "grid averaging" memory patterns, where each A(i,j) is assigned an average weighting of its four nearest neighbors.
6. **SORT:** Implements a sorting algorithm, which sorts a randomized 64-bit integer array during each transaction.
7. **Hybrid / SSJ:** The hybrid SSJ worklet stresses both CPU and memory, with either serving as the primary bottleneck, depending on system configuration. It is measured at eight load levels; 12.5, 25, 37.5, 50, 62.5, 75, 87.5, and 100%. SSJ performs multiple different transactions at a time that simulate an enterprise application. Its performance metric is transactions per second (throughput).

2.2. Memory

According to the SERT design document [1], the Memory workload consists of worklets that have been designed to scale with installed memory. Specifically, this means that the worklets are designed to measure a higher (better) performance score with improved memory characteristics (e.g. higher bandwidth, lower latency, total memory size). The primary memory characteristics being tested are bandwidth and capacity. The memory workload features two worklets:

1. **Flood:** A sequential memory bandwidth test that exercises memory using arithmetic operations and copy instructions. Flood is multi-threaded to reward servers that can utilize more memory concurrently with multiple CPUs and DIMMs. It automatically adjusts to use all available system RAM. It runs at two load levels called "Full" and "Half", utilizing all and half of the system memory, respectively. Flood's performance score is a function of the bandwidth measured during testing.
2. **Capacity:** A memory capacity test that performs XML operations on a minimum and maximum data set. Capacity is the only worklet to scale with memory capacity rather than transaction rate or user count. The Capacity worklet caches validation results in memory; validation of input data that is not already in the cache results in a performance penalty as the input data is validated. Systems with larger amounts of memory will be able to cache more results and therefore support a larger data set. The SERT 2 suite calculates the largest data set that the system can contain in memory with minimal performance penalties. The final metric is a function of the data set size and the transaction rate, which includes the performance penalties.

2.3. Storage

The SERT suite includes a workload for testing storage in order to enable a well-rounded system test. These storage worklets only test the server's internal storage devices and not external storage. Storage tests are run on two load levels: 100% and 50%, with transactions per second (throughput) serving as the performance metric. Their main characteristics are [1]:

- Worklets contain consistent IO characteristics per simulated "user" regardless of size and number of installed disks or memory.
- At the 100% load level, the performance bottleneck is the storage subsystem.
- Worklet performance score higher (better) performance score for higher bandwidth and lower latency.
- Storage worklets are primarily designed to test a server's disk I/O, not the storage subsystem as a whole. Consequently, disk caches must be disabled during storage testing.

The Storage workload includes two worklets, each with read and write transactions:

1. **Random:** Reads and writes data to/from random file locations.
2. **Sequential:** Reads and writes data to/from file locations that are picked sequentially.

2.4. Idle

Idle is the only workload that does not have a performance metric as it does not perform any actions on the tested system, but keeps it in an idle state in order to measure the idle power consumption. As such, it targets no particular subsystem and does not measure any efficiency (only consumption). It is not included in any scores, but its measurement results are included for additional information.

3. SERT 2 Metric

The SERT 2 metric, also called SERT 2 Efficiency Score, is a final aggregate of all the power and performance values measured during a SERT run. It is designed to enable regulators to make a decision on whether or not to apply an energy label to a tested system. The SERT suite calculates the SERT 2 metric from the separate workload scores, which in turn are aggregates of all efficiency scores of the worklets within the given workload. The SERT 2 Efficiency Score is a single number that indicates the overall energy efficiency of the tested system.

Number of Sockets	2
CPU name	Intel Xeon E5-2699 v3
Cores per CPU	18
Threads per Core	2
CPU Frequency	2.3 GHz (3.6 GHz Turbo)
Memory Type	8GB 2Rx8 PC4-2133P-R
Number of DIMMs	8
Operating System	Windows Server 2012 R2
Java Virtual Machine (JVM)	Oracle HotSpot 1.8.0

Table 1: Basic Server Configuration Example

The SERT 2 metric is also used in regulators' decision making on whether or not a SERT result passes for regulatory standards. To this end, regulators have proposed SERT 2 metric thresholds that concrete results must pass in order to be accepted for certification or labeling. These thresholds may change over time. However, the SERT 2 metric and its calculation method remain the same.

This section explains the SERT 2 metric in detail. It defines it using mathematical formulae and explains its calculation. It also explains how configuration of a system affects the scores and the SERT 2 metric. To illustrate these effects, we use a running example throughout this document. We use the basic system specified in Table 1 and alter its configuration to explain the impact that configuration differences have on the metric result.

3.1. Calculating the SERT 2 Metric

The SERT 2 metric is computed in three steps:

1. Calculate worklet efficiency scores from measurement data.
2. Calculate workload efficiency from worklet efficiency scores.
3. Calculate the SERT 2 metric from workload efficiency scores.

Each of those steps is an aggregation. The worklet efficiency scores aggregate the power consumption and performance of the worklet at its various load levels (1). The workload efficiency score then combines the scores of all worklets in this workload (2). Finally, the SERT 2 metric aggregates all the workload scores (3). Figure 1 sketches the overall approach of metric calculation.

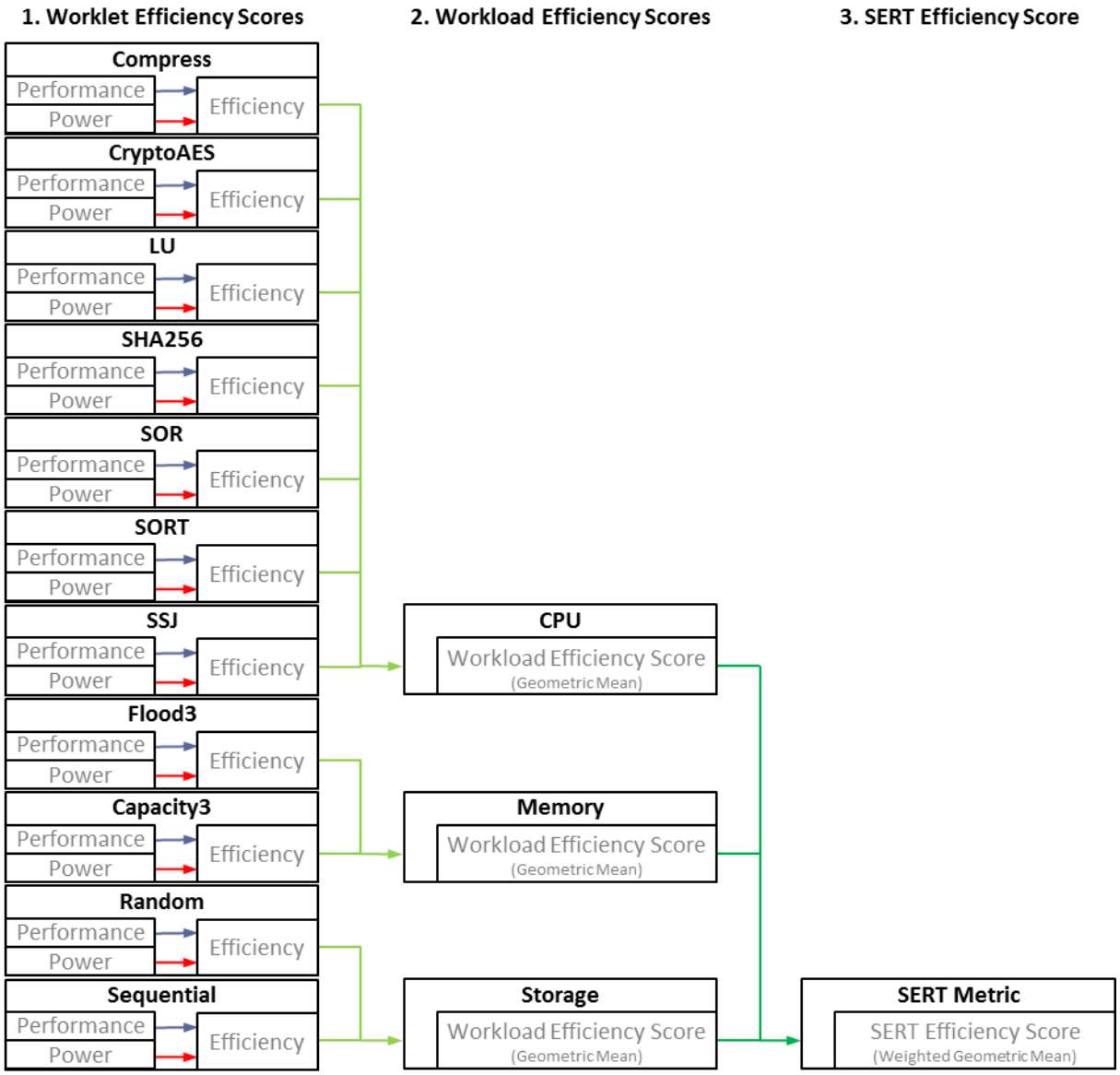

Figure 1: SERT 2 metric calculation

3.1.1. Worklet Efficiency Calculation

All SERT worklets, except Idle run at multiple load levels. For each of those load levels, energy efficiency is calculated separately. We define per load level energy efficiency Eff_{load} as follows (Eq 1):

$$Eff_{load} = \frac{\text{Normalized Performance}}{\text{Power Consumption}} \quad (1)$$

Normalized performance is normalized throughput for most worklets, with memory worklets being a notable exception. Power consumption in this context is the average measured power consumption for each load level. As such, the per load level energy efficiency metric represents the amount of (normalized) transactions that were executed per unit of energy (Joule).

The worklet efficiency score $Eff_{worklet}$ is calculated using the geometric mean of the separate load level scores (Eq. 2). The geometric mean has been chosen in favor of the arithmetic mean or sum. The major difference between the geometric mean and arithmetic mean is that the arithmetic mean favors load levels with higher efficiency scores.

Traditionally, higher load levels also feature higher efficiency scores on most systems. As a result, the arithmetic mean usually favors the results of high load levels. A change in energy efficiency at a higher load level has a greater impact on the SERT 2 metric than it would at a lower load level. The geometric mean, on the other hand, treats relative changes in efficiency equally at each load level. Finally, the resulting geometric mean is multiplied with a factor of 1000. This is a cosmetic factor that has been chosen in order to move the resulting score into a number range that is easier to read for a human reader.

$$Eff_{worklet} = \exp\left(\frac{1}{n} \times \sum_{i=1}^{n} ln(Eff_{load_i})\right) \times 1000 \quad (2)$$

n represents the number of load levels per worklet and Eff_{load_i} the energy efficiency for load level i.

Example: The CryptoAES worklet is measured at four load levels (25%, 50%, 75%, and 100% load, see Table 2). The power and normalized performance are measured directly from the system. Energy efficiency is calculated according to Eq. 1:

Load Level	Normalized Performance	Power	Efficiency
25 %	49.616	176.583 W	0.281
50 %	99.221	294.086 W	0.337
75 %	148.823	407.581 W	0.365
100 %	199.476	469.005 W	0.425

Table 2: Example measurement results and efficiency for CryptoAES worklet.

Using the energy efficiency scores for each load level, the worklet efficiency score is computed by using the geometric mean, as defined in Eq. 2:

$$\text{Eff}_{\text{worklet}} = \exp(1/4 * (\ln(0.281) + \ln(0.337) + \ln(0.365) + \ln(0.425))) * 1000 = 348.33$$

3.1.2. Workload Efficiency Calculation

Workload efficiency $Eff_{workload}$ is calculated by aggregating the efficiency scores of all worklets within the workload using the geometric mean (Eq. 3). The worklet efficiency scores being aggregated are the results of the score calculation in Eq. 2 in Section 3.1.1.

$$Eff_{workload} = \exp\left(\frac{1}{n} \times \sum_{i=1}^{n} ln(Eff_{worklet_i})\right) \quad (3)$$

n represents the number of worklets per workload and $Eff_{worklet\_i}$ is the energy efficiency for each specific worklet.

Example: As an example, we calculate the workload energy efficiency for the storage workload. Storage consists of two worklets: Random and Sequential. For our example system, they feature the following Worklet Efficiency Scores:

Worklet	Worklet Efficiency Score
Random	14.36
Sequential	33.86

Table 3: Example Storage Worklet Efficiency Scores.

Using these two worklet efficiency scores in Table 3, we calculate the workload efficiency score according to Eq. 3:

$$\text{Eff}_{\text{workload}} = \exp(1/2 * (\ln(14.36) + \ln(33.86)) = 22.05$$

3.1.3. SERT 2 Metric Calculation

The SERT 2 metric (or SERT 2 Efficiency Score) is the final aggregate of the workload scores. In contrast to the other mean aggregates, the SERT 2 metric does not consider all workloads equally. Instead, it uses a weighted geometric mean, putting a different focus on each of the workload scores. The particular workload weights are as follows:

- CPU weight: 65% **(High)**
- Memory weight: 30% **(Medium)**
- Storage weight: 5% **(Low)**

The weights have been decided by expert groups and represent a realistic weighting of the importance of the different workloads when comparing the SERT suite to real world operation workloads. Note, again, that the workloads do not purely exercise the hardware component after which they are named. The CPU workloads also exercise some memory, the Memory workloads perform some work on the CPU, etc. This means that the weighted CPU workloads (with their weight of 65%) already include some memory power and memory energy efficiency components. The weights have been created with this in mind. Consequently, the SERT 2 metric is calculated as follows (Eq. 4):

$$\text{SERT 2 Efficiency Score} = \exp(0.65 * \ln(\text{Eff}_{\text{CPU}}) + 0.3 * \ln(\text{Eff}_{\text{Memory}}) + 0.05 * \ln(\text{Eff}_{\text{Storage}})) \quad (4)$$

Example: We calculate the SERT 2 metric of our example system based on the efficiency scores of the separate workloads, which in turn have been calculated using the geometric mean of their worklet scores (see Section 3.1.2). They are as follows:

Workload	Workload Efficiency Score
CPU	85.75
Memory	36.50
Storage	22.05

Table 4: Example Workload Efficiency Scores.

Using these workload Efficiency Scores, the SERT 2 Efficiency Score is calculated using the weighted geometric mean, as defined in Eq. 4:

$$\text{SERT 2 Efficiency Score} = \exp(0.65 * \ln(85.75) + 0.3 * \ln(106.76) + 0.05 * \ln(22.05)) = 62.01$$

3.2. Primary influences on metrics

The SERT 2 metric is primarily influenced by the performance and efficiency of the worklet (and subsequently, workload) scores. This means that changes in single workload scores directly affect the SERT 2 Efficiency Score of a system under test. As workloads are not equally weighted for the SERT 2 metric, changing certain workloads has a greater impact on the SERT 2 metric. Specifically, the CPU workload has the greatest impact, followed by the Memory workload.

Changes to the system configuration will typically result in changes to the workload scores. System hardware components may be exchanged for other components that offer better performance or lower power consumption. This may affect different workload scores in a different manner, depending on the hardware component in question. E.g., a better performing CPU may increase the energy efficiency for the CPU workload, which utilizes this CPU, but may reduce energy efficiency for the storage workload, for which this CPU remains at low load or idle. The CPU workload's efficiency score is primarily influenced by the choice of CPU in the tested system. An energy efficient CPU increases the energy efficiency scores for the separate load level measurements, which leads to better worklet scores, which, in turn, leads to a better CPU workload score.

Understanding which CPUs will result in the best (most energy efficient) SERT 2 Efficiency Scores requires a basic understanding of the design decision behind the SERT CPU worklets: Specifically, the design decisions regarding scalability and parallelism. According to the SERT design document [1], CPU worklets are designed to be bottlenecked by the CPU hardware. All CPU worklets are designed to utilize the CPU to its fullest extent. This includes high

parallelism, as much as the CPU in the tested system permits. By fully utilizing the entire CPU, the SERT suite is able to investigate its energy efficiency for most possible states. These design decisions have as a consequence that high performance CPU with great parallelism can achieve excellent scores regarding energy efficiency, as these CPUs often feature good energy efficiency at full load, which is achieved for many SERT measurements.

We demonstrate the effect of the CPU on the workload scores using our example system. The example system features two Intel Xeon E5-2699 v3 processors with 18 cores each, running at a base frequency of 2.3 GHz (3.6 GHz with Turbo). We exchange these CPUs for smaller ones: Namely, two Intel Xeon E5-2660 v3 CPUs with 10 cores each, running at 2.6 GHz (3.3 GHz with Turbo) and two Intel Xeon E5-2603 v3 CPUs with 6 cores, running at 1.6 GHz.

Processor	CPU	Memory	Storage	SERT 2 Efficiency Score
E5-2699 v3 – 2.3 GHz – 18 cores	85.75	36.50	22.05	62.01
E5-2660 v3 – 2.6 GHz – 10 cores	75.75	41.92	19.69	59.30
E5-2603 v3 – 1.6 GHz – 6 cores	41.69	39.80	23.94	39.99

Table 5: Score change for different CPU models.

In this example (Table 5), the biggest CPU features the best efficiency, considering that the SERT suite utilizes the CPU resources efficiently during its high load measurements. The low load measurements account for the CPU score not dropping off as steeply as it would if measurements were taken at full load only on the lower performance CPUs. Of course, the low performance CPUs may cause score increases for the Memory and Storage workloads, considering that these workloads do not utilize the CPU at its maximum capacity. However, these increases must be very significant in order to make up for or exceed the effect of the 65% CPU weight.

Similarly, the memory configuration can have a significant impact on the Memory workload score. For this, configuring memory basically comes down to two primary choices: Number of DIMMS and DIMM capacity. For this decision, it is important to consider the effect that the amount of DIMMs has on bandwidth. Always install sufficient DIMMs to utilize all memory channels. We illustrate this using measurements from earlier SERT analyses [3]. In these analyses, we observed that both CPU and memory scores drop dramatically if insufficient memory DIMMs are used. Figure 2 illustrates this for the worst case worklet: CryptoAES. In this example CryptoAES is executed once with only two DIMMs (not utilizing all channels), and once with 8 DIMMs. The figure shows that energy efficiency using all 8 DIMMs is significantly better. The memory worklets rely heavily on good bandwidth (as do most CPU worklets, such as CryptoAES). Ensuring good bandwidth is paramount for achieving high performance and good efficiency.

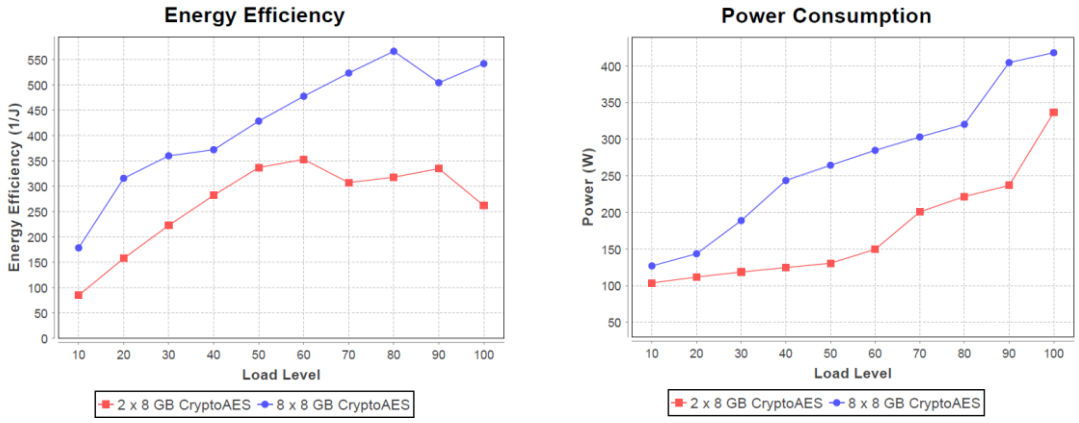

Figure 2: Memory channel impact on CryptoAES.

Increasing memory capacity also increases memory scores, due to the way the Capacity worklet operates. The Capacity worklet validates XML data, which is mostly kept in-memory. If the memory is not sufficiently large, data will be recalculated, which results in a hit to performance, as simply fetching it would be faster. This way, Capacity inherently rewards systems for large capacity memory. In addition, it features a memory scaling factor that adjusts performance scores based on the capacity of the system. Table 6 shows this effect. Increasing memory capacity

significantly increases memory scores. This can often outweigh the small negative impact that capacity increase may have on CPU scores, considering that the CPU worklets do not utilize that much memory.

The storage workloads utilize all storage devices (HDDs or SSDs) on the SUT. All devices are used in parallel, meaning that additional devices can be used to increase bandwidth and, consequently, system throughput. Of course, additional storage devices also increase overall system power consumption. As shown in Table 7, the performance increase can outweigh the negative impact of the additional power consumption, as storage efficiency is increased by a significant margin.

Memory	CPU	Memory	Storage	SERT 2 Efficiency Score
8 x 8 GB	85.75	36.50	22.05	62.01
8 x 16 GB	81.32	41.44	22.69	62.32

Table 6: Score change for different memory capacities.

HDD	CPU	Memory	Storage	SERT 2 Efficiency Score
1 HDD	85.75	36.50	22.05	62.01
8 HDDs	79.15	34.28	124.64	63.00

Table 7: Score change for different number of HDDs.

3.3. Secondary influences on scores

Section 3.2 demonstrates that the selection of tested hardware components has a direct influence on the workload scores and thus the SERT 2 metric. As expected each workload's score is primarily influenced by the hardware component after which it is named. A more energy efficient CPU increases the CPU score, RAM with a greater bandwidth and/or capacity increases the memory score, and additional storage devices increase the Storage score.

Apart from these primary influences, hardware component selection may also change the score of the other workloads. E.g., the CPU may impact the memory score. Some of these influences were already mentioned and observed in Section 3.2. However, these secondary influences are often not as straight forward as the direct primary hardware component selection influences. This section demonstrates some of the potential secondary influences based on our running example.

3.3.1. CPU workload score

The CPU workload's score is obviously primarily affected through choice of the CPU. However, RAM and HDD/SSD selection also plays a role and influences final efficiency. RAM choice can have a very large influence on CPU scores, as memory bandwidth is important for several CPU worklets. Table 8 shows memory bandwidth measurements for SERT worklets at full load on an example system using Intel Xeon E5-2699 v3 processors. Note that CryptoAES, Compress, and SSJ all require very significant bandwidth of 55-85% of that consumed by the Flood worklet.

Worklet	Bandwidth	% of Flood3 bandwidth
CryptoAES	92.57 GB/s	85.4%
Compress	60.03 GB/s	55.4%
SSJ	72.07 GB/s	66.5%
Sort	7.80 GB/s	7.2%
LU	0.15 GB/s	0.1%
SOR	0.11 GB/s	0.1%
SHA256	0.09 GB/s	0.1%
Flood3 (memory)	108.39 GB/s	100.0%

Table 8: Example CPU and memory worklet system memory bandwidth.

Bandwidth can be influenced by many factors, number of memory channels and memory clock being the most common. Again, the rule of thumb is to never under-provision memory channels. Always provide sufficient DIMMs to utilize all the memory channels available to the SUT. Table 9 illustrates this for two systems with Intel Xeon E5-2603 v3 processors with one and two DIMMs of RAM. As expected, the Memory score increases both with bandwidth and capacity. In this it differs from the CPU score, which only increases with bandwidth but not with capacity. In this example, the second DIMM increases the number of memory channels, which has a significant impact on bandwidth, causing the CPU score to increase. Storage performance is not affected. The additional power consumption of the second DIMM only serves to reduce storage energy efficiency.

DIMMs	CPU	Memory	Storage	SERT 2 Efficiency Score
1 x 8 GB	35.93	13.04	32.71	26.39
2 x 8 GB	44.71	19.61	25.54	33.96
Delta	24%	50%	-22%	29%

Table 9: Score change for different memory capacities.

To further stress the importance the number of available memory channels have on the CPU score, we demonstrate the impact the number of memory channels can have on CPU worklet performance. Figure 3 shows the effect that memory channels can have on the CPU performance score.

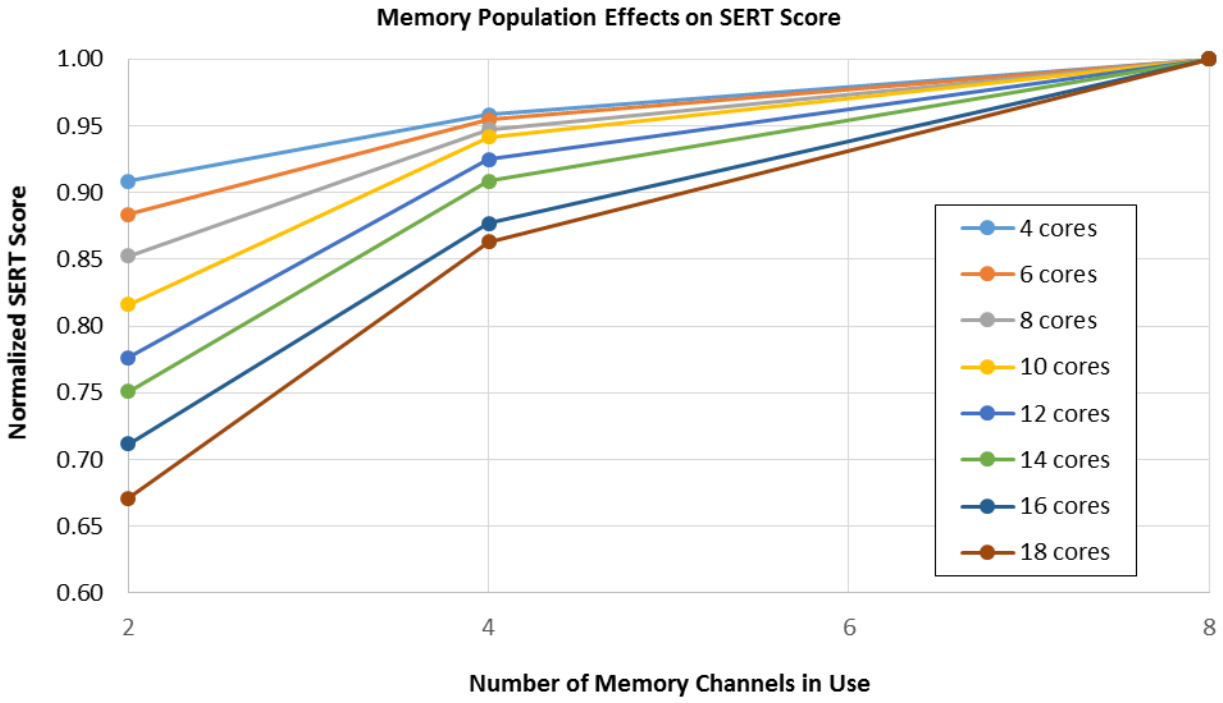

Figure 3: CPU worklet performance for memory channels

The figure is normalized for different core counts, meaning that it shows the relative performance decrease for each system as memory channels are removed (starting at 8 memory channels). The graph shows that systems with many CPU cores can be affected with a performance hit of almost 40% for CPU worklets when the number of memory channels is lowered to two. This hit to performance translates almost directly to the efficiency score, as power is not affected in the same manner as performance.

Adding additional storage devices always negatively affects CPU workload scores, considering that none of the CPU worklets utilizes storage. As a consequence, storage devices with low idle power are useful for SERT measurements, as these devices do not affect CPU (and Memory) scores as much as devices with greater idle consumption.

3.3.2. Memory workload score

As expected, the primary influence on the Memory score are the memory DIMMs themselves. Apart from the memory DIMMs, CPU and storage devices can also impact memory score. The impact of CPU on memory score is the most difficult to assess. Both memory worklets (Flood and Capacity) utilize the CPU quite a bit but, in contrast to CPU worklets, the CPU is not the bottleneck resource. Performance may however be negatively affected if the CPU is not fast enough or if it does not feature sufficient cache. On the other hand, a high performance CPU might be idle for a lot of the time, consuming unnecessary power. We can observe this effect when comparing scores for different CPU models as done in Table 5 in Section 3.2. An excerpt of this table with memory scores only is shown below in Table 10.

Processor	Memory Score
E5-2699 v3 – 2.3 GHz – 18 cores	36.50
E5-2660 v3 – 2.6 GHz – 10 cores	41.92
E5-2603 v3 – 1.6 GHz – 6 cores	39.80

Table 10: Memory score change for different CPU models (excerpt from Table 5).

Table 10 illustrates the mentioned effect. Both the low end and high end CPU achieve worse efficiency scores than the medium sized Intel Xeon E5-2660 v3 CPU. This medium sized CPU offers better performance than the lower end Xeon E5-2603 v3 and consumes less power than the higher end Xeon E5-2699 v3 CPU.

Storage devices do not improve the memory score as they remain unused during the memory tests. Just as observed during the CPU workload, using storage devices with low idle consumption is paramount. Using multiple storage devices has a negative effect on memory efficiency as those remain unused. Consequently, multiple storage devices should only be used if their efficiency increase for the storage workload (which is only weighted with a 5 % weight towards the SERT 2 metric) outweighs the negative impact that those devices have on memory and CPU scores.

3.3.3. Storage workload score

The storage workload score can be influenced by the selection of CPU and memory. Similarly, to the effect of CPU selection on Memory workload results, the impact of CPU and memory choice on the storage score is non-trivial. The storage worklets use some memory and CPU during their operation. This means that using too little or too slow memory and/or a slow CPU can affect storage worklet results negatively. On the other hand, a high-performance power consuming CPU might be utilized too little and consume unnecessary power. Memory can have similar effects on the storage score. In contrast to the secondary influences on the memory worklets, the low power consumption of small CPU and memory configurations can have a positive impact on storage scores. Table 11 demonstrates the effect example CPUs can have on the storage score. It is an excerpt of Table 5 in Section 3.2.

Processor	Storage Score
E5-2699 v3 – 2.3 GHz – 18 cores	22.05
E5-2660 v3 – 2.6 GHz – 10 cores	19.69
E5-2603 v3 – 1.6 GHz – 6 cores	23.94

Table 11: Storage score change for different CPU models.

However, the effects that memory and CPU selection have on storage score are usually negligible in comparison to the change that can be achieved by using multiple drives. In addition, the 5% weight of the storage workload on the SERT 2 metric means that it almost always pays off to optimize CPU and memory for the CPU and memory workloads instead of optimizing them for good storage results.

3.3.4. Untested Hardware Components

SERT utilize all installed CPUs, memory, and hard drives to achieve additional performance that may improve energy efficiency, depending on the performance and power draw of those specific components. Other components, however, are not tested in a SERT run and only consume power, thus impacting the SERT 2 metric negatively.

The following devices are not stressed in a SERT run: Accelerator cards and other specialized co-processors, high-performance network devices, any sort of graphical display devices, peripherals that use the tested system as their primary power source, and many more.

Future versions of the SERT suite may test some of these devices. Currently, it is recommended to remove them from the tested system for SERT measurements. Note that hardware availability requirements of the run and reporting rules may apply, meaning that the system under test must be available for consumers in the tested configuration. Also note, that some devices may be required in order to execute the SERT measurement, even though they are not stressed. E.g., the SERT setup requires a network interface card in order to control the SUT during measurement.

3.3.5. Run to Run Variations

Hardware components and background software, such as the operating system, may behave differently during different SERT runs. These differences may be due to environmental factors, such as minor changes in temperature or due to internal hardware behavior that remains unknown to the outside observer. Consequently, different runs of the SERT suite may produce slightly different score results.

The differences in power consumption that can be observed between different measurements are not that significant, however. An existing study [4] analyzes these differences and finds that the power differences range between 0.7% (0.1W on the tested system) for the Idle worklet to 4.0% (1.32W). The impact on the SERT 2 metric is even lower, as these random variations balance out across the multiple worklets. For run-to-run variations a higher power consumption also correlates with a better performance, further diminishing the effect on the SERT 2 metric.

4. Conclusion

This document introduces the SERT 2 metric, a single-number metric to compare SERT results. The SERT 2 metric is used by regulators for server energy efficiency labeling. It is calculated as a weighted geometric mean based on SERT's separate workload scores. Section 3 explains the calculation in detail. In addition to explaining the calculation of the SERT 2 metric, this document also provides hints and guidelines on how to configure servers for testing using the SERT. It describes how hardware component selection may influence workload scores and, finally, the SERT 2 Efficiency Score. The document also describes the potential impact of run to run variations.

SPEC recommends that the full suite of worklets be run and the SERT metric be used for any Government efficiency programs as it offers the best test methodology for reporting server efficiency. SPEC designed the metric to stress a full range of CPU, memory and storage workloads to provide an overall assessment of the server efficiency. The SERT suite was designed as a complete package comprising a balanced suite of worklets and resultant metric. The calculation methods rely upon a balanced set of data points to create a highly effective overall metric to differentiate servers based on their ability to maximize work for each unit of energy consumed. The metric has undergone thousands of hours of testing over a 6 year period and has been validated by SPEC, U.S. EPA, The Green Grid, Digital Europe, JEITA, METI, and others as an effective server energy efficiency metric, and is the required metric for the ISO/IEC 21836 Draft International Standard.

5. References

[1] Standard Performance Evaluation Corporation. The SERT 2 Design Document. https://www.spec.org/sert2/SERT-designdocument.pdf.

[2] T.A Welch. A Technique for High-Performance Data Compression. Computer, 17(6):8-19, June 1984.

[3] Jóakim von Kistowski, Hansfried Block, John Beckett, Klaus-Dieter Lange, Jeremy A. Arnold, and Samuel Kounev. Analysis of the Influences on Server Power Consumption and Energy Efficiency for CPU Intensive Workloads. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering (ICPE 2015)*, ICPE'15, New York, NY, USA, February 2015. ACM.

[4] Jóakim von Kistowski, Hansfried Block, John Beckett, Cloyce Spradling, Klaus-Dieter Lange, and Samuel Kounev. Variations in CPU Power Consumption. In *Proceedings of the 7th ACM/SPEC International Conference on Performance Engineering (ICPE 2016)*, New York, NY, USA, March 2016. ACM.

6. Appendix A – Approximating the SERT 2 Metric from SERT 1.1.1 scores

This approximation is provided to help regulators who wish to use existing SERT 1.1.1 data to define thresholds. It is intended to be used to approximate the SERT 2 metric from the SERT 1.1.1 scores. However, as a conversion applied to existing data it is affected by the original hardware configuration, which will not be apparent from the data being converted. You'll only be able to do conversions for results where all of the interval detail are available. In cases where only the efficiency scores are available, you won't be able to accurately approximate to the SERT 2 suite.

6.1. CPU, Hybrid, and Storage Worklets

All of the CPU, Hybrid, and Storage worklets use the same basic process (but with a different conversion factor for each worklet). For these worklets, there is no difference in the runtime behavior of the SERT suite but the scores are calculated differently. So in order to calculate the worklet efficiency score we simply need to perform the same calculations that SERT suite does based on the raw scores.

To convert these scores, start with the raw performance score and the average active power for each measurement interval. These values can be found in results-details.html in the "Performance Data" table (the "Score" column) and the "Power Data" table (the "Average Active Power (W)" column) for each worklet. In the output of the SERT 2 metrics report (reporter.sh -r results/sert-xxxx/results.xml -c -x results/XSL-File/csv-metrics.xsl), the values are in columns with names such as "Compress 100% Perf" and "Compress 100% Watts". Do not use the "Normalized Perf" values.

For each measurement interval (100%, 75%, 50%, etc), divide the Score by the worklet-specific reference score in Table 12 to calculate the SERT 2 Normalized Performance value. Next, calculate a SERT load level efficiency value by multiplying the SERT 2 Normalized Performance Value by 1000 and then dividing by the Average Active Power for that interval. Then calculate the geometric mean of the load level efficiency scores to calculate the worklet efficiency score.

Worklet	Reference Score
CPU Compress	6924.40
CPU CryptoAES	59008.11
CPU LU	15946.51
CPU SOR	15056.53
CPU XMLvalidate	Not included in the SERT 2 suite
CPU Sort	20887.07
CPU SHA256	976.85
Storage Sequential	338.46
Storage Random	136.59
Hybrid SSJ	354112.34

Table 12: SERT 2 reference scores.

Table 13 shows an example for the Compress worklet from our SERT 2 reference system. The first 3 columns come from the SERT 1.1.1 result and the remaining columns are calculated as described above.

Load Level	SERT 1.1.1 Score (Performance)	SERT 1.1.1 Average Active Power	SERT 2 Reference Score	SERT 2 Normalized Performance	SERT 2 Load Level Efficiency Score
100%	7053.604377	151.407107	6924.40	1.01866	6.727949
75%	5398.947832	139.427934	6924.40	0.779699	5.592129
50%	3591.575764	74.890165	6924.40	0.518684	6.925930
25%	1800.210682	62.280413	6924.40	0.259981	4.174358

Table 13: Example: Compress worklet.

From this we calculate the SERT 2 worklet efficiency score for compress as the geometric mean of the load level efficiency scores: 5.742909. The process is the same for all other non-memory worklets -- the only difference is the reference scores in the table above.

6.2. Memory Worklets

6.2.1. Flood2 -> Flood3

The Flood worklet is a measure of memory bandwidth. However, in Flood2 (SERT 1.1.1 suite), the score included a capacity multiplier to scale with the square root of physical memory. In addition, the performance score for Flood_Half was multiplied by 0.5 so that the score was roughly 1/2 of the Flood_Full score, which made it look more like the CPU worklets.

In Flood3 (SERT 2 suite), the performance score is the measured bandwidth with no capacity multiplier or load level adjustment. Therefore these adjustments need to be removed from the raw performance scores before applying the new reference score in order to convert from Flood2 to Flood3.

We will start with the raw (not normalized) Flood2 scores for both Flood_Full and Flood_Half. Divide this value by the square root of physical memory on the system (in GB). Please note that the physical memory is available in column J of the SERT csv-metrics output, but it must be converted from MB to GB (divide by 1024) (Eq 5) and multiplied by 2 for Flood_Half (Eq 6).

SERT 2 Flood_Full raw perf score = SERT 1.1.1 Flood_Full raw perf score / sqrt(physicalMemoryGB) (5)

SERT 2 Flood_Half raw perf score = 2 * SERT 1.1.1 Flood_Half raw perf score / sqrt(physicalMemoryGB) (6)

Then calculate the normalized performance score the same way as you did for the CPU worklets. Divide the raw performance score by the reference score (11.52). Then calculate the load level efficiency scores (1000 * normalizedPerfScore/averageWatts), and take the geometric mean of the Flood_Full and Flood_Half efficiency scores to compute the Flood3 worklet efficiency score.

For example, on the reference system with 8192 MB (8 GB) of physical memory, the conversions from SERT 1.1.1 Flood2 scores to SERT 2 Flood3 are shown below. The first 4 columns come from the SERT 1.1.1 result and the remaining columns are calculated as described in Table 14. Finally, the Flood3 worklet efficiency score is the geometric mean of 6.935 and 7.084, which is 7.009.

Load Level	SERT 1.1.1 Score (Performance)	SERT 1.1.1 Average Active Power	Physical Memory (GB)	SERT 2 Reference Score	SERT 2 Raw Performance	SERT 2 Normalized Performance	SERT 2 Load Level Efficiency Score
Flood_Full	**32.591109**	**144.233333**	**8.0**	**11.52**	**11.522697**	**1.000234**	**6.935**
Flood_Half	**16.285057**	**141.113810**	**8.0**	**11.52**	**11.515274**	**0.999590**	**7.084**

Table 14: Approximating example: Flood2 -> Flood3.

6.2.2. Capacity2 -> Capacity3

Capacity2 (SERT 1.1.1) ran 9 different load levels, instead of only 2 load levels (Base and Max) for Capacity3 (SERT 2). Only the Capacity3_Max load level is included in the score, Capacity3_Base is reported for reference only. So, the first step is to identify which Capacity2 load level we should use to approximate Capacity3_Max.

To do this, we must look at the PhysicalMemoryGB / 2, and find the first Capacity_x value that is approximately equal to this value. For example, for a system with 512 GB memory you would use Capacity_256. Most systems should have a Capacity load level that is very close to half the physical memory, but a few will not. It should be reasonably accurate to use the first load level that is <= to this value. For example, with a system that has 1.5 TB (1536 GB) memory, you could use Capacity_512. In a pinch, you could use Capacity_4 for any system and get reasonably close,

but this will introduce a larger inaccuracy. Please note that one of the advantages of Capacity3 is its improved handling of systems with less typical memory capacities.

Once the correct Capacity_x load level is identified, we'll use that as the starting point for calculating the raw performance score for Capacity_Max. In Capacity2, the score was scaled by the square root of physical memory (in GB). We need to remove this scaling factor, so divide the raw performance by sqrt(PhysicalMemoryGB).

In Capacity3, the score scales with the square root of the data store size (Eq 9). The data store size in each VM is 60% of the max heap size (Eq 8). And the max heap size is equal to the total physical memory minus 1 GB, times 0.85 and divided by the number of client VMs on the host (Eq 7):

maxHeap = (PhysicalMemoryGB - 1) * 0.85 / numJvms (7)

dataStoreSize = 0.6 * maxHeap (8)

totalDataStoreSize = dataStoreSize * numJvms (9)

and therefore:

totalDataStoreSize = (PhysicalMemoryGB - 1) * 0.85 * 0.6 = (PhysicalMemoryGB - 1) * 0.51 (10)

The approximated raw performance score for Capacity3, Capacity_Max is show in (Eq 11).

SERT 2 Capacity_Max raw performance score = SERT 1.1.1 Capacity_x raw performance score *
sqrt((PhysicalMemoryGB - 1) * 0.51) / sqrt(PhysicalMemoryGB) (11)

Finally, the Capacity_Max Normalized Performance is calculated by dividing the raw performance score by the reference score (148095.20). Then, similar to the other worklets, the load level efficiency score is 1000 * the raw performance score divided by the average active power during that load level. Since we use only the score for one load level (Capacity_Max), the Capacity3 worklet efficiency score is equal to the Capacity_Max load level efficiency score.

Since the reference system has 8 GB memory, we will only use the Capacity_4 load level to calculate Capacity_Max. According to the formula above, the estimated raw performance score for Capacity_Max is 146452.188, and the normalized performance score is 0.988906. The average active power was 154.3, so the Capacity_Max load level efficiency score (and therefore the Capacity3 worklet efficiency score) is 6.408981.

For the reference system, the SERT 1.1.1 scores are shown in Table 15.

Load Level	SERT 1.1.1 Score (Performance)	SERT 1.1.1 Average Active Power
Capacity_4	219233.415	154.3
Capacity_8	169565.334	154.2
Capacity_16	152083.244	153.8
Capacity_32	144904.023	154.1
Capacity_64	141253.592	154.1
Capacity_128	139406.868	153.9
Capacity_256	138480.982	154.1
Capacity_512	137897.310	154.2
Capacity_1024	137360.984	154.2

Table 15: SERT 1.1.1 capacity scores – reference system.

6.1. Memory Worklets

The **workload efficiency score** for each workload is calculated as the geometric mean of the SERT 2 worklet efficiency scores for the component worklets (Eq. 3).

Remember that SSJ is considered to be part of the CPU workload in the SERT 2 suite.

6.2. SERT 2 Efficiency Metric

The **overall SERT 2 Efficiency Score** is a weighted geometric mean (Eq. 4) of the workload efficiency scores as described in section 3.1.3.

