

# Analysis of the Influences on Server Power Consumption and Energy Efficiency for CPU-Intensive Workloads

Jóakim v. Kistowski  
University of Würzburg  
joakim.kistowski@  
uni-wuerzburg.de

Hansfried Block  
Fujitsu Technology Solutions  
GmbH  
hansfried.block@  
ts.fujitsu.com

John Beckett  
Dell Inc.  
john\_beckett@dell.com

Klaus-Dieter Lange  
Hewlett-Packard Company  
klaus.lange@hp.com

Jeremy A. Arnold  
IBM Corporation  
arnoldje@us.ibm.com

Samuel Kounev  
University of Würzburg  
samuel.kounev@  
uni-wuerzburg.de

## ABSTRACT

Energy efficiency of servers has become a significant research topic over the last years, as server energy consumption varies depending on multiple factors, such as server utilization and workload type. Server energy analysis and estimation must take all relevant factors into account to ensure reliable estimates and conclusions. Thorough system analysis requires benchmarks capable of testing different system resources at different load levels using multiple workload types. Server energy estimation approaches, on the other hand, require knowledge about the interactions of these factors for the creation of accurate power models. Common approaches to energy-aware workload classification categorize workloads depending on the resource types used by the different workloads. However, they rarely take into account differences in workloads targeting the same resources. Industrial energy-efficiency benchmarks typically do not evaluate the system's energy consumption at different resource load levels, and they only provide data for system analysis at maximum system load.

In this paper, we benchmark multiple server configurations using the CPU worklets included in SPEC's Server Efficiency Rating Tool (SERT). We evaluate the impact of load levels and different CPU workloads on power consumption and energy efficiency. We analyze how functions approximating the measured power consumption differ over multiple server configurations and architectures.

We show that workloads targeting the same resource can differ significantly in their power draw and energy efficiency. The power consumption of a given workload type varies depending on utilization, hardware and software configuration. The power consumption of CPU-intensive workloads does not scale uniformly with increased load, nor do hardware or software configuration changes affect it in a uniform manner.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*ICPE'15*, Jan. 31–Feb. 4, 2015, Austin, Texas, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3248-4/15/01 ...\$15.00.

<http://dx.doi.org/10.1145/2668930.2688057>.

## Categories and Subject Descriptors

C.4 [Computer Systems Organization]: Performance of Systems—*Performance attributes*

## General Terms

Benchmarking, Workload, Energy Efficiency

## Keywords

SPEC, SERT, Power, Workload Characterization, Energy Efficiency, Metrics, Utilization

## 1. INTRODUCTION

The energy efficiency of servers has become a significant issue as data center energy consumption has risen dramatically over the past decade. In 2010, the U.S. Environmental Protection Agency (U.S. EPA) estimated 3% of all electricity consumed in the US to be used in running data centers [8]. According to a New York Times study from 2012, data centers worldwide consume about 30 billion watts per hour. This is equivalent to the approximate energy output of 30 nuclear power plants [1].

This leads to an increasing pressure on hardware vendors to design systems with a high energy efficiency. Equally, software developers are tasked with the design and development of energy efficient applications.

Conventional end-user hardware makes use of device power-saving states to reduce energy consumption. Such states enable significant power savings during system idle times. Servers, however, are rarely fully idle. Instead they often serve requests that arrive at low frequencies leading to a steady load at a low-resource utilization level [2]. Furthermore, servers in data centers nowadays have to deal with highly variable application load intensities [17], translating into varying resource utilization levels. Therefore, considering the energy efficiency exhibited by a server at different load levels is equally as important as considering the server's efficiency at peak load.

To enable the design of energy efficient data centers and software systems, methods for the reliable measurement and estimation of server energy efficiency are needed. These methods must be able to accurately measure and estimate energy efficiency for a variety of workloads and load levels. The authors of [12] state that fine granular energy estima-

tion models in particular are highly dependent on the type of workload for which energy is estimated.

As a result, many models for energy estimation classify workloads by the type of resources used. Models, such as the models proposed in [10, 3] decompose workloads to derive their respective CPU, memory, and storage I/O usage in order to obtain accurate power estimates based on device utilization and load levels. Utilization-based models, however, rarely take into account the effect of different types of workloads that use the same resource type. CPU-heavy workloads, for example, can lead to different power usage characteristics depending on their use of different instructions, caches, and so on.

Current approaches to model such effects make use of performance counters for the estimation of power usage based on the type of instructions [4]. Performance counters are, however, difficult to measure and classify, especially if the approach is to be extended over different machines with different architectures. As a result, we see the need to classify workloads that use the same resource type, based on their power and energy-efficiency characteristics at multiple load levels.

In this paper, we show the impact of different CPU-heavy workloads on the power consumption and energy efficiency of servers. For this evaluation, we employ the SERT provided by the Standard Performance Evaluation Corporation (SPEC). SERT is a tool suite for the measurement and analysis of server energy efficiency using different worklets exercising different aspects of the system under test (SUT) under multiple load levels. We employ SERT's seven CPU worklets to measure and characterize the power consumption and energy efficiency of these workloads over a range of at least 10 load levels. We present the measurement results of 20 selected machines, featuring three different CPU architectures.

The goal of this paper is to gain insight into the different classes of CPU workloads and how they vary in power consumption and energy efficiency at different load levels.

The major contributions of the paper are:

1. We explore the differences in power consumption and energy efficiency of different CPU worklets over a range of target load levels.
2. We demonstrate that different workloads utilizing the same resource (CPU) can have a significantly different energy draw. We analyze and characterize the changes in the energy efficiency of these workloads over multiple target load levels.
3. We explore the impact of different hardware and software configurations, including different architectures on the energy curves over target load levels.

Our measurements show that different CPU-heavy workloads exhibit significant differences in power consumption. These differences also translate directly to differences in the workload energy efficiency. Comparison of different hardware architectures and the introduction of hardware bottlenecks also shows a non-trivial relationship between power consumption and efficiency, as it introduces complexities in system performance without necessarily affecting the power draw in the same manner. We also debunk common assumptions such as constant operating system power overhead and maximum energy efficiency at full utilization.

The remainder of this paper is structured as follows: Sec-

tion 2 introduces SERT, its architecture, and measurement approach. Section 3 then introduces the CPU workloads contained in SERT. Section 4 details the considered SUTs, and our measurement approach. Following that, Section 5 details the measurement results and describes the workloads power and energy-efficiency behavior over different load levels, architectures, and hardware and software configurations. We conclude the paper in Section 6.

## 2. SERT

SERT has been developed by the SPEC OSG Power Committee as a tool for the analysis and evaluation of the energy efficiency of server systems. Its design and development goals and process have been introduced in [8].

In contrast to energy-efficiency benchmarks, such as Joule-Sort [13], SPECpower\_ssj2008 [6], and the TPC-Energy benchmarks [11], SERT is not intended to be used as a benchmark for a single system energy efficiency score. It does not aim to generate specific end user workloads, but instead provides a set of focused synthetic micro-workloads called worklets that exercise selected aspects of the Server (or System) Under Test (SUT). Specifically, the worklets have been developed to exercise the processor, memory and storage I/O subsystems, and may be combined into various configurations running serially or in parallel to provide "system" tests as part of a larger workload.

For each of the server components to be stressed, SERT offers a range of worklets designed to exercise the targeted component in a different manner. This allows for thorough analysis of system energy behavior under different workload types designed to target the same component. As an example, the CryptoAES worklet profits from both specialized instruction sets, as well as better CPU to memory connectivity, whereas the SOR worklet primarily scales with processor frequency.

SERT and its worklets are designed for the measurement of system energy efficiency at multiple load levels. This sets it further apart from conventional performance benchmarks, such as SPEC CPU [5], which targets maximum load and performance. A detailed description of SERT memory worklets and their applicability can be found in [7]. A detailed description of the storage I/O worklets and their properties can be found in [9].

### 2.1 Workload Calibration

According to [2], servers nowadays spend most of their time in a CPU utilization range between 10% and 50%. As a result, server energy-efficiency evaluation tools should support testing at different load levels. SERT offers functionality to do this.

SERT contains a test harness named *Chauffeur* which, among other tasks, handles the task of calibrating the workload to run at target load levels. To this end it runs each worklet in a calibration mode to determine the maximum transaction rate that the worklet can achieve on the SUT. For each Target Load Level (e.g., 100%, 75%, 50%, 25%), *Chauffeur* calculates the target transaction rate and derives the corresponding mean time from the start of one transaction to the start of the next transaction. During the measurement interval, these delays are randomized using an exponential distribution that statistically converges to the desired transaction rate. As a result, lower target loads consist of short bursts of activity separated by periods of inactivity.

## 2.2 Tool Architecture

SERT's measurements are controlled by a **controller** system. This system runs the Chauffeur harness, the reporter, the optional graphical user interface, and instances of the SPEC PTDaemon.

Chauffeur is the framework on which SERT is built. It handles both the coordination with the SUT director triggering the execution of worklets, as well as the communication with other controller-internal components, such as the PTDaemon and the reporter.

The Reporter generates the final report on measured and derived results including performance and energy measurements, as well as energy-efficiency scores.

The SPEC PTDaemon is a tool that allows network-based communication with a host connected to power and temperature measurement devices. PTDaemon supports a range of SPEC-approved devices, all featuring a maximum measurement uncertainty of 1% or better.

SERT requires at least one power analyzer and one temperature sensor. The power analyzer measures the wall power of the entire SUT, while the temperature sensor verifies the validity of measurements by assuring that all experiments are conducted under similar environmental conditions.

SERT also provides a graphical user interface (GUI) for easy test-run execution.

All SERT hardware and software components and their relationships to one another are illustrated in Fig. 1.

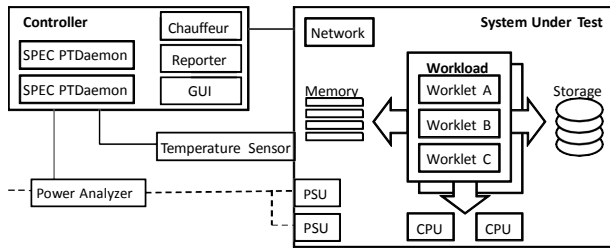


Figure 1: SERT Architecture [15]

## 3. CPU WORKLETS

This paper focuses on the characterization of worklets and their effect on system energy efficiency at different load levels. We focus on CPU-heavy worklets as CPUs offer a clear definition of different utilization levels and the calibration of a worklet's execution frequency offers a highly accurate method of reaching a given target load. Note that all load levels in this paper are workload specific loads, as they signify the number of executing transactions in relation to the maximum possible transaction rate without performance loss. This load definition offers the advantage of being platform independent and thus allowing better comparability of utilization based power and energy efficiency over multiple architectures, workloads, and machine types.

The SERT design document [15] 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.

SERT 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 is executed at a target load 25%, 50%, 75%, and 100% per default. For more detailed analysis, we have reconfigured our SERT runs to use 10% load level interval steps.

1. **Compress**: Implements a transaction that compresses and decompresses data using a modified Lempel-Ziv-Welch (LZW) method following an algorithm introduced in [18]. 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 or DES block cipher algorithms using the Java Cryptographic Extension (JCE) framework.
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 and encryption/decryption transformations on a byte array. This byte array is perturbed by one byte for each transaction.
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 Dirichlet 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. **XMLValidate**: Implements a transaction that exercises Java's XML validation package *javax.xml.validation*. Using both SAX and DOM APIs, an XML file (.xml) is validated against an XML schemata file (.xsd). To randomize input data, an algorithm is applied that swaps the position of commented regions within the XML input data.

## 4. MEASUREMENT METHODOLOGY

We measure all results according to the SPEC Power and Performance Benchmark Methodology [16]. The devices are setup and configured as required by SERT (see Section 2.2). The controller with the Chauffeur harness runs on an external machine, while the worklets are executed on the JVM within the SUT. We employ a Hioki 3334 Power Analyzer for most power measurements. A Digi Watchpart/H temperature sensor monitors the environmental temperature to

ensure that it ranges between 22 and 23 °C for the duration of all test runs.

For the purpose of this paper, we have modified SERT’s default configuration to execute measurements at intervals of 10% load levels, instead of the usual 25%. We confirm the expressiveness of results obtained using this configuration by comparing one set of measurements taken with 10% load intervals to a measurement series with 50 data-points for each worklet, differing by 2% in load. Detailed results of this measurement series are discussed in Section 5.1.4.

## 4.1 Systems under Test

We use a number of similar systems to explore the different bottlenecks that influence CPU-heavy worklets and cause differing power behavior. These systems are based on a Fujitsu PRIMERGY RX300S7 2 socket system using Intel Sandy Bridge processors. Our baseline configuration is shown in Table 1. Almost all servers feature two 8-core processors, with two hardware threads each. As a result, the workload features 32 threads with affinity to each of the virtual cores.

	<b>RX300S7_RHEL6.4 _E5-2690_8x8GB</b>
PSU Output Power	450 W
Number of Sockets	2
CPU name	Intel Xeon E5-2690 (Sandy Bridge)
Cores per CPU	8
Threads per Core	2
CPU frequency	2.9 GHz (3.3 GHz Turbo)
Memory Type	8GB 2Rx4 PC3L-12800R ECC
Number of DIMMs	8
Operating System	Red Hat Enterprise Linux Server 6.4 (Santiago)
JVM	Oracle HotSpot 1.7.0_51-b13

**Table 1: Baseline Server Configuration**

For our basic configuration, we introduce multiple variation points. We vary either the processor, the number of DIMMs, the DIMM size, or the operating system. The alternate operating system to the Linux Server is the Windows Server 2008 R2 Enterprise Edition OS.

When varying the processor, the following processors are used: Intel Xeon E5-2620, E5-2643, and E5-2650L. The Xeon E5-2643 and E5-2620 processors vary the number of available cores, having 4 and 6 cores per socket, respectively. The Intel Xeon E5-2650L processor features an identical number of cores as the baseline E5-2690 (8 cores per socket), having a reduced frequency of 1.8 GHz (up to 2.3 GHz with Turbo).

The number of DIMMs varies between 2, 8, and 24, whereas DIMM capacity is either 8 or 16 GB.

For comparison with other architectures, we also employ Fujitsu PRIMERGY RX600S6, Fujitsu PRIMERGY RX200S8, Dell PowerEdge R720 servers, and an AMD Opteron based HP ProLiant DL385p Gen8 machine. The Fujitsu PRIMERGY RX200S8 and Dell PowerEdge R720 servers are introduced in detail in Sections 5.4 and 5.5. The PRIMERGY RX600S6 Servers feature 4 processor sockets, each

carrying an Intel Xeon E7-4870 Westmere-EX CPU at 2.4 GHz (up to 2.8 GHz with Turbo) processing frequency with 10 cores each and 2 hardware threads per core.

## 4.2 Normalized Power/Efficiency Values

For each measurement interval, we measure the amount of dispatched and completed work units, the interval’s length, the average power use, load level, and environmental temperature. In this paper, we focus on the performance, power, and efficiency metrics in relation to the load level at which they were measured.

As the goal of this paper is the characterization of distinctive power and energy efficiency behavior measured at different load levels for different CPU heavy workloads under different platforms and platform configurations, we display the measured power-load and energy efficiency-load curves in a manner that allows for easy comparison. To achieve this, we normalize all power and energy efficiency measurements to the minimum value for each load level based curve. As a result, all measurements feature a normalized power or efficiency value of one for the smallest CPU load interval. This approach allows for easy comparison of the measurements corresponding to different target load levels, as each measurement now displays the ratio between the measurement at the respective load level and the measurement at the minimum load level.

## 5. EXPERIMENTAL ANALYSIS

In this section, we show and analyze the measurement results. First, we discuss the differences between CPU worklets when run on the same machine. Next, we compare the performance and power of each worklet under different machine configurations. We then go into detail on configuration changes that have a significant impact on power consumption. Finally, we categorize the different types of power and energy efficiency per load curves.

### 5.1 Power Consumption Differences between CPU Workloads

We first compare the power consumption of the different CPU worklets on our baseline system before comparing their overall energy efficiency. Following that, we provide an outlook on how these inter-worklet differences can be affected using other system configurations.

#### 5.1.1 Power Consumption

Figure 2 shows the power consumption of SERT’s CPU worklets on the baseline system (see Table 1). The power consumption for the worklets differs the least at 10% load level with average power draw ranging between 118.3 W for SOR and 126 W for the XMLValidate worklet. The 100% target load level features a larger total difference in power consumption of 87.6 W. The worklet using the most power here is again XMLValidate with a power draw of 431.4 W, and the least power is again drawn by SOR at 343.8 W. XMLValidate is not always the biggest power consumer though, nor does SOR always consume the least power. Between 30% and 90% load, XMLValidate is replaced by CryptoAES as the biggest power consumer, and SHA256 consumes less power than SOR in the range between 30% and 70% load. Even though worklets’ power consumption differences are most pronounced at full utilization, it is notable that significant differences of power draws do exist at lower utilization

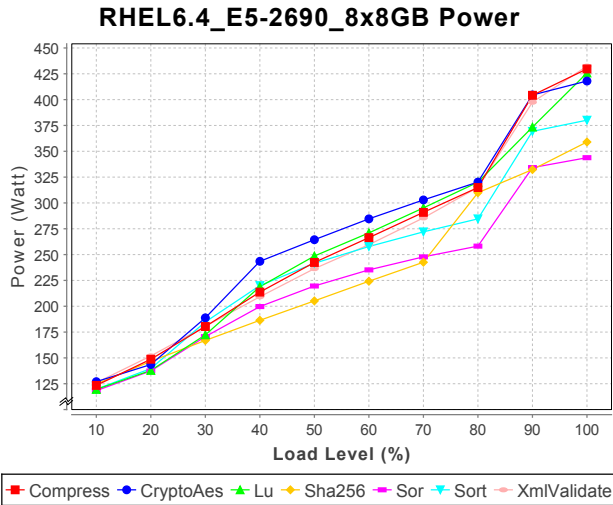


Figure 2: Power Consumption on Baseline Server

ranges. For instance, at 40% load, the biggest power consumer (CryptoAES, 243.5 W) draws 57.1 W more than the smallest power consumer (SHA256 186.4 W).

The power / load curves display a similar pattern for all seven worklets. At lower load levels, power draw increases in an either linear or slightly concave manner. After this gradual power increase, we can observe a drastic rise in power draw between the 80% and 90% load intervals. Exceptions are the SHA256 and LU worklets. SHA256’s sudden power increase is located between the 70% and 80% intervals, whereas LU features a longer linear increase in power consumption between 80% and 100%. In most cases, after the sudden increase in power consumption, the power increase reverts back to a similar shape as for the lower load levels.

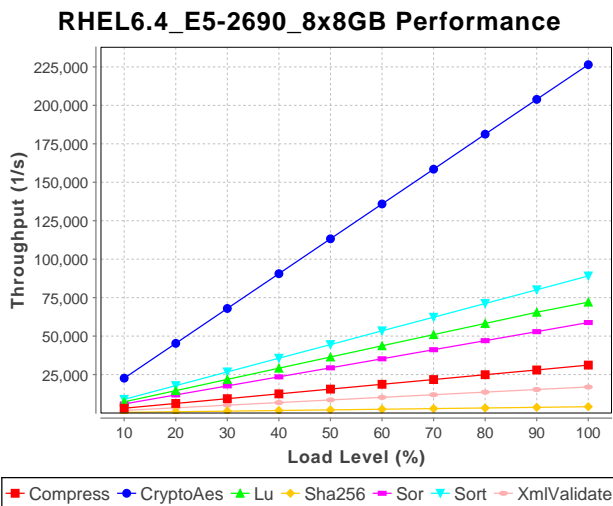


Figure 3: Throughput on Baseline Server

### 5.1.2 Energy Efficiency

Energy efficiency for CPU worklets is measured in throughput/power [15]. SERT achieves the different target load levels by scaling the number of executed transactions per unit of time. As a result, the throughput of the different worklets

risers linearly with CPU utilization (see Fig. 3). Throughput per load level differs greatly depending on the actual worklet, as each worklet type induces different loads on the target CPU. To enable comparability, the energy efficiency displayed in Fig. 4 applies the normalization approach introduced in Section 4.2.

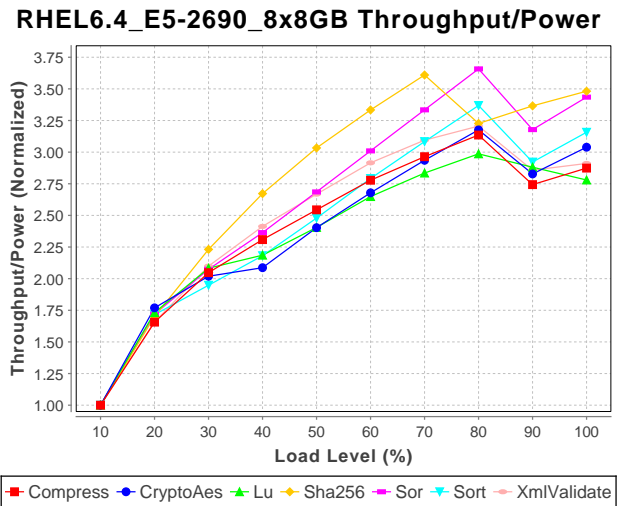


Figure 4: Energy Efficiency on Baseline Server.

Due to the linear nature of the throughput increase over the load intervals, energy efficiency mirrors the observations made for the pure power measurements.

Energy efficiency increases steadily, as long as the throughput increases faster than the power consumption. The sudden rise in power consumption between the 80% and 90% load mark, however, features a power demand that exceeds the linear increase in throughput, thus leading to a slight drop in energy efficiency. This observation contradicts the common assumption of maximum energy efficiency at full machine load. Observations of a "U"-shaped energy efficiency curve [14], though more accurate, also do not picture the entirety of the energy-efficiency curve, as energy efficiency increases again at the 90% load level.

The LU and CryptoAES worklets feature a relatively steep rise in power consumption between 20% and 40% load. As a result, the increase in energy efficiency is slightly diminished at these intervals.

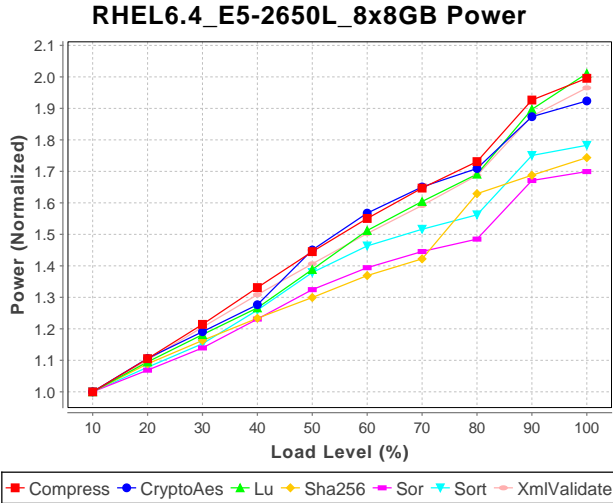
### 5.1.3 Worklet Comparison for Lower CPU Frequency

Lower CPU frequencies have some effect on the worklets’ normalized power consumption. Fig. 5 shows the normalized power consumption for SERT’s CPU worklets on a server in which the CPUs have been exchanged with Intel Xeon E5-2650L processors. These processors feature the same number of cores and hardware threads as the Xeon E5-2690, which was installed in the baseline system. They do, however, work at a reduced base frequency of 1.8GHz compared to the 2.9GHz of the Xeon E5-2690.

While the power / load curves bear some overall similarity to the curves on the baseline server, we can make some additional observations from this measurement.

First, we can see that the sudden rise in power consumption between the 80% and 90% measurement intervals, while still notable, is not as steep as it was in the previous case.

A new observation is that the concave segment of the



**Figure 5: Normalized Power Consumption on Server with Xeon E5-2650L Processor**

power / load curve does not start directly at the 10% load level. Instead, a linear segment leading up to the concave curve is clearly visible for all worklets. CryptoAES demonstrates this best with a linear segment leading up to the 40% load mark, followed by a concave curve segment up to the 80% load level. Knowing that this effect exists, we can also find it when revisiting the original measurement on the baseline system. Here, the concave function begins at an earlier point. In most cases, the linear segment terminates at the 20% load level.

### 5.1.4 Accuracy of Measurements at 10% Utilization Intervals

The majority of measurements in this paper show the power consumption and energy efficiency measured at 10% CPU load intervals. From this, we extrapolate functions and curves approximating the energy behavior of the considered servers when subjected to CPU heavy workloads.

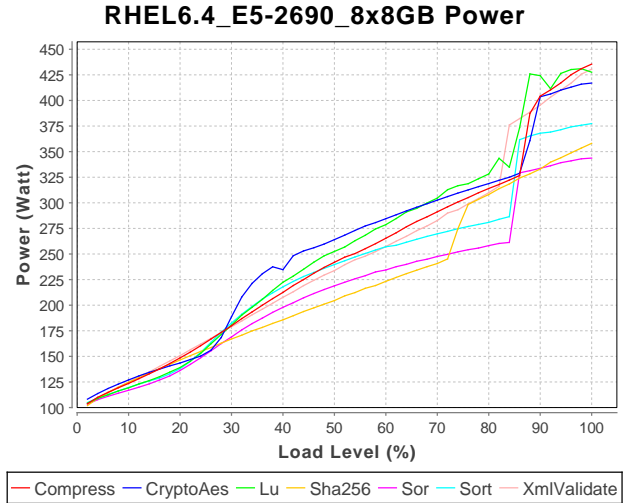
To verify that 10% load intervals can lead to representative insights on the overall behavior of the machine’s power consumption at different load levels, we perform one series of measurements at the fine granularity of 2% load intervals. Fig. 6 shows the results of those measurements.

Compared to the measurements at 10% load intervals in Fig. 2, we can confirm all of our observations in Section 5.1. A slow rise in power consumption is followed by a concave curve segment starting around the 20% load level in most cases. This segment then ends with a steep linear increase in power consumption usually occurring between the 80% and 90% load levels, which again is followed by a continuation of the increase of the previous segment.

While confirming the previous observations, the finer granularity of load intervals also reveals some additional insights. Most notably, the LU and CryptoAES worklets exhibit instances of short decreases in power use for some load level increases. These effects are repeatable over several measurements and warrant some further research.

## 5.2 Per Workload Results

We analyze the impact of varying the system architecture and configuration on the power consumption and energy ef-

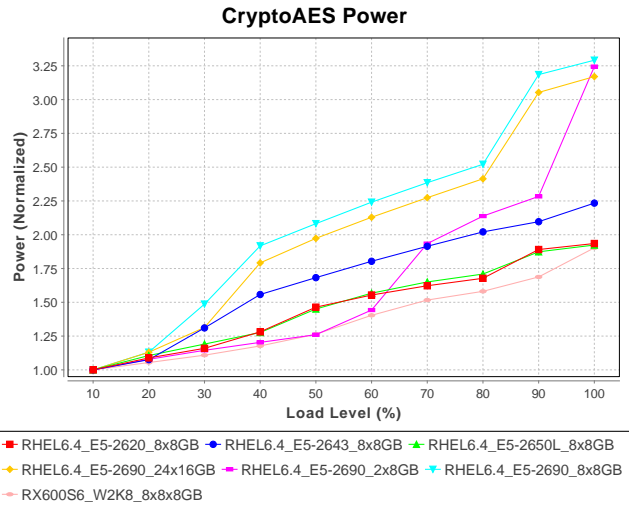


**Figure 6: Power Consumption of the Baseline Server Measured at 2% Load Intervals**

iciency of selected worklets. More specifically, we evaluate the behavior of CryptoAES, SHA256, and SORT as they deviate the most from the observed power pattern on our baseline machine. They are also representative of most deviations, which occur on our selected SUTs.

### 5.2.1 CryptoAES

For CryptoAES, the amount of available memory channels has a significant impact on the power draw caused by the worklet. When increasing the baseline machine’s amount of RAM to 24 DIMMs, each with a capacity of 16 GB, the function characterizing power consumption over load changes little. The only visible difference is a slight reduction in the relative difference between power consumption at 10% and 100% load, resulting in a slightly smaller normalized power consumption in Fig. 7. This effect can be attributed to the increase in constant system power draw caused by the additional DIMMs.



**Figure 7: Normalized Power Consumption of Different Servers Running CryptoAES**

Decreasing the amount of available RAM to two DIMMs,

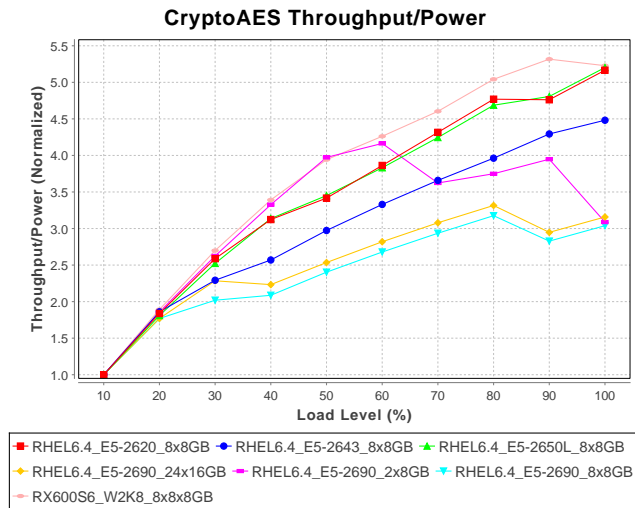


however, has a far more drastic effect. The normalized power draw does not increase as heavily at early load stages, as all DIMMs are already under high load due to the smaller constant power overhead. This effect is mitigated at 100% load, as the CPU power consumption outweighs the memory effects.

Exchanging the processor also impacts the power draw of CryptoAES. The use of processors with lower clock rates or fewer cores lowers the difference between the power draw at low vs. full load. Both the Intel Xeon E5-2650L with its 8 cores at 1.8 GHz and the Xeon E5-2620L with its 6 cores at 2.0 GHz demonstrate this effect well. Both show a decreased difference in power consumption between load levels, yet both still follow the same curve shape as the baseline system.

Running CryptoAES on a Xeon E5-2643 quad-core processor, however, results in a different power / load function, as the steep rise in power consumption between the 80% and 90% load level seems to be missing. Instead, we can only observe a minor increase at the 90% to 100% level.

Running CryptoAES on a four socket system using Windows results in the smallest normalized difference between low vs. full load power draw. This can easily be attributed to the higher constant energy costs of keeping a four socket system running. Considering the system’s energy efficiency in Fig. 8, we can see a dramatic difference in energy efficiency, which is due to the high combined performance potential of 40 cores. Generally speaking, larger systems tend to feature higher constant power draw. This leads to poor energy efficiency at lower load levels. In return, these systems are capable of far greater performance at high load levels, increasing their energy efficiency at those levels.



**Figure 8: Normalized Energy Efficiency of Different Servers Running CryptoAES**

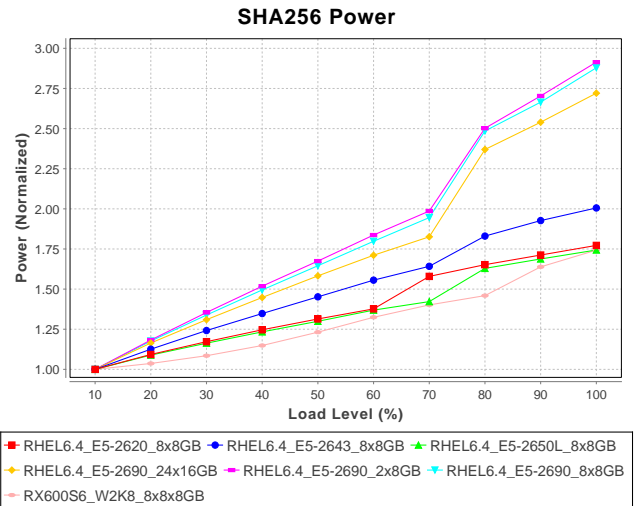
Evaluation of energy efficiency also confirms that CryptoAES requires a minimal number of available memory access channels. Using 2 DIMMs only results in greater energy efficiency for CPU load of 60% and less. At higher load levels, however, the memory bottleneck leads to a decrease in performance and causes efficiency to drop.

Even though the 16 x 24GB RAM configuration of our baseline system shows smaller differences in power consumption, it does show a slight increase in normalized energy efficiency. This is to be expected as performance increases with

the additionally provided memory channels. The addition of more DIMMs does however also lead to a decrease in effective memory frequency from 1600 MHz to 1066 MHz. As a result, the increase in normalized efficiency is only minimal.

### 5.2.2 SHA256

Experiments with the SHA256 worklet confirm most of the observations made for CryptoAES, with some key differences. The overall shape of the power / load curves still matches the previously observed shapes, with the four socket Windows system showing the smallest differences between the load levels. The Windows system also displays a different power scaling behavior over load. We discuss the differences in power scaling of SERT’s worklets for different operating systems in Section 5.3.



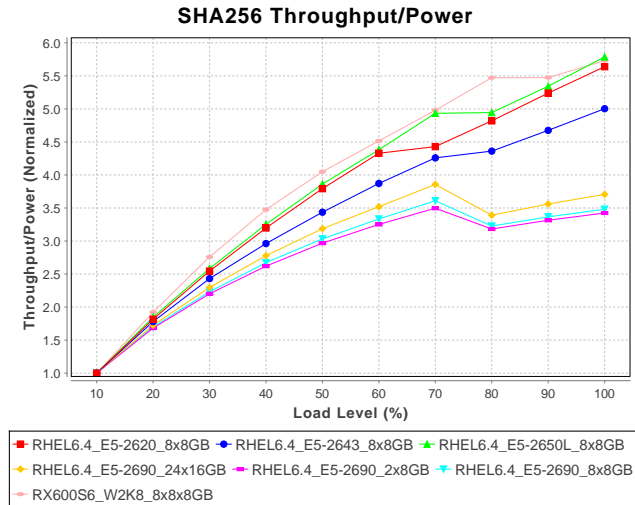
**Figure 9: Normalized Power Consumption of Different Servers Running SHA256**

A key difference in SHA256 and CryptoAES is found in the way in which power consumption over load scales depending on the number of installed DIMMs. For CryptoAES, a minimum number of memory access channels is necessary for high performance and energy efficiency. An insufficient number of memory channels also leads to significant deviations in power behavior. SHA256 shows little dependency on RAM connectivity. It has little impact on power, with smaller numbers of DIMMs only causing a slightly lower constant power overhead. It also has no significant impact on energy efficiency, as seen in Fig. 10.

Another new observation is the deviation between the normalized power consumption of SHA256 using the Xeon E5-2620 and 2650L processors. Both processors still behave similarly, yet show difference in normalized power at 70% load, with the E5-2620 system having its steep increase in power consumption between the 60% and 70% intervals, and the E5-2650L system increasing its energy draw at the 70% to 80% interval. This deviation is not surprising, as the processors feature different numbers of cores and processor clocks.

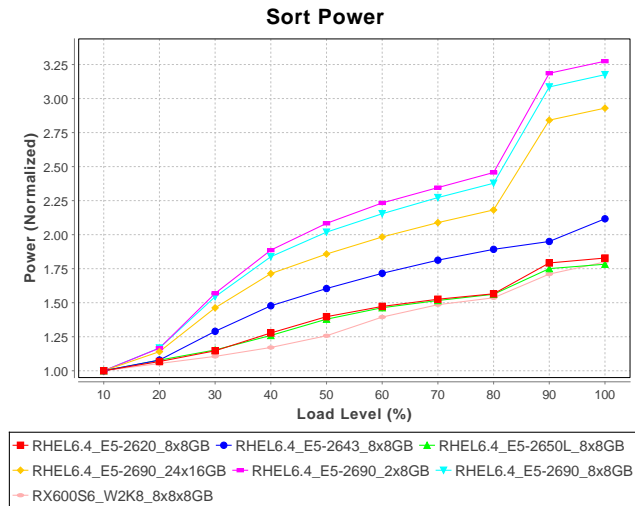
Running on the Xeon E5-2643 processor, SHA256 also features a smaller sudden increase between the 70% and 80% load levels. Other than CryptoAES running on the same processor, it does show some increase at this level. This increase is, however, not as pronounced as it is on the other

system configurations.



**Figure 10: Normalized Energy Efficiency of Different Servers Running SHA256**

The normalized worklet energy efficiency in Fig. 10 provides another observation for the Xeon E5-2620 and E5-2650L processors. While systems with these processors exhibit smaller differences in power consumption than the baseline system, they show greater differences in energy efficiency, matching the energy efficiency range of the 4 socket system.



**Figure 11: Normalized Power Consumption of different Servers running Sort**

### 5.2.3 Sort

The Sort worklet presents a mix of the observations from CryptoAES and SHA256. Sort’s normalized power consumption (see Fig. 11) mirrors CryptoAES for the system running Xeon E5-2643 processors, as the power over load function features only a slight additional increase in power consumption at the 90% to 100% load interval.

Similarly to SHA256, however, Sort shows little RAM dependency featuring a similar normalized power consumption for the different RAM configurations of the baseline system.

The biggest difference between Sort and the other worklets is the observation that the concave segment of the power function is far more pronounced. It is also clearly visible for all system configurations with the exception of the 4 socket Windows system.

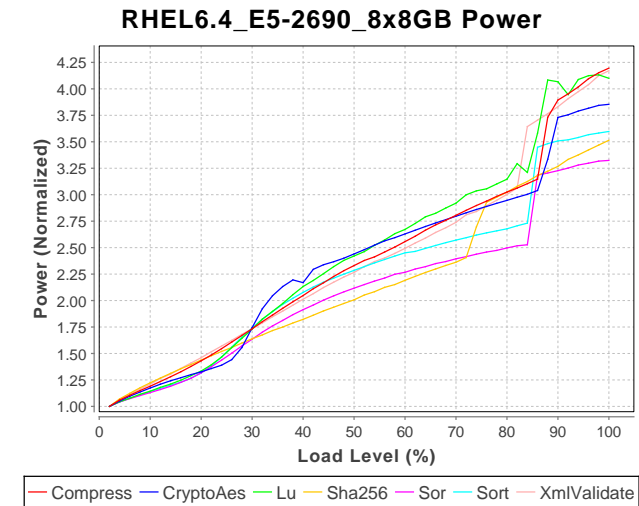
### 5.2.4 Conclusions

Our analysis of the impact of hardware configurations on worklet power consumption and energy efficiency has revealed several insights. Our experiments confirmed that a greater number of hardware components decreases the relative difference between maximum and minimum power consumption, as these components introduce an additional constant power draw. These systems have a lower energy efficiency at low load levels, in return they feature greater efficiency at high load.

The relative differences between power draw at low vs. high load are also smaller for processors with fewer cores and lower clock rates. We have also demonstrated that bottlenecks, in our case a memory access bottleneck, have significant impact on both power consumption and energy efficiency.

## 5.3 Effect of Operating System

Common power models such as [3] often assume a constant power overhead created by the respective operating system. As a result, operating systems can be compared with respect to the amount of constant power overhead they induce.



**Figure 12: Normalized Power Consumption on Baseline Server Running Linux**

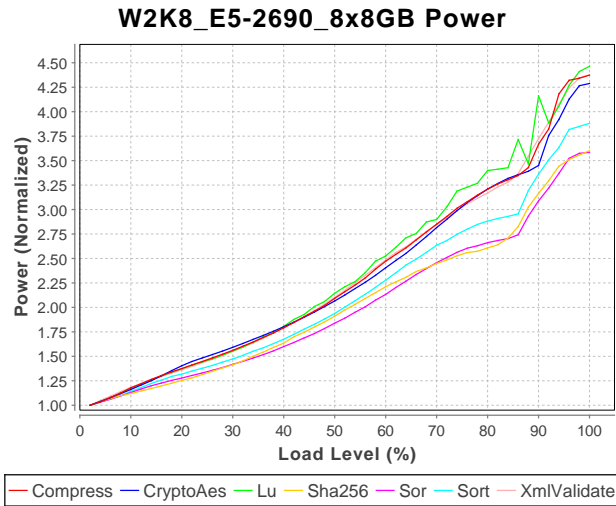
We test this assumption by executing our CPU worklets on both Linux and Windows Server systems. Fig. 12 shows the normalized power consumption of our baseline system running Linux using load level intervals of 2%, whereas Fig. 13 shows the normalized power consumption of the same server running Windows Server 2008 R2 as its operating system.

Both figures show that a constant power overhead per operating system is a gross oversimplification of the diverse power behaviors.

On Linux, the different worklets exhibit a range of different power / load curves. Windows, on the other hand, creates a more uniform power draw. Workloads still differ in their power consumption, specifically in the rate of power



increase with increasing utilization. The shape of the power / load curve, however, is very similar for each worklet. This relatively uniform manner of power over load increase results in smaller normalized power differences between the worklets at lower load levels. These differences in normalized power then increase with higher load levels.



**Figure 13: Normalized Power Consumption on Baseline Server Running Windows Server 2008 R2**

The only worklets deviating from the common scheme are Compress and CryptoAES, which show a lesser increase than the other worklets at the 90% load level.

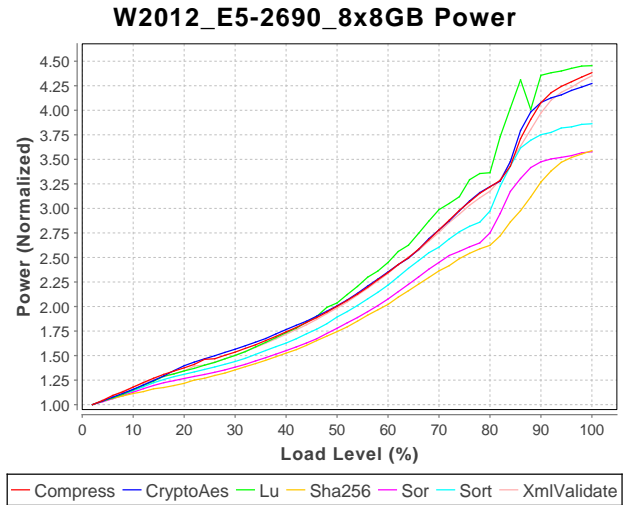
Overall, the power / loadation curve on Windows can be approximated using an exponential increase in power draw over load. Deviations from this exponential curve are less pronounced and less diverse than on the Linux system. We can still observe a slightly concave part overlapping the exponential curve between the 50% and 80% load levels. This deviation is clearly visible for the SHA256 and SOR worklets. The visibility of this deviation differs between architectures. The four socket Windows system, running Westmere-EX processors (see Figures in Section 5.2), for example, demonstrates a greater deviation from the exponential curve than the baseline system. Compared to the non-Windows system, however, the base function of power over load follows an exponential pattern, which it does not on the Linux system.

An additional measurement series of SERT’s CPU worklets on the baseline system running Windows Server 2012 also confirmed the results of our comparison and are shown in Fig. 14. Windows Server 2012 features smaller changes to the exponential base power over load function compared to Windows Server 2008 R2. As a result, the concave deviations of worklet power as described above are not as pronounced.

Concluding, we find that differences in power draw over load cannot be approximated using a constant power overhead per operating system. Operating systems are not simply “better” or “worse” in their impact on power. Power draw differs depending on worklet and load level, with power over load being approximated by completely different functions.

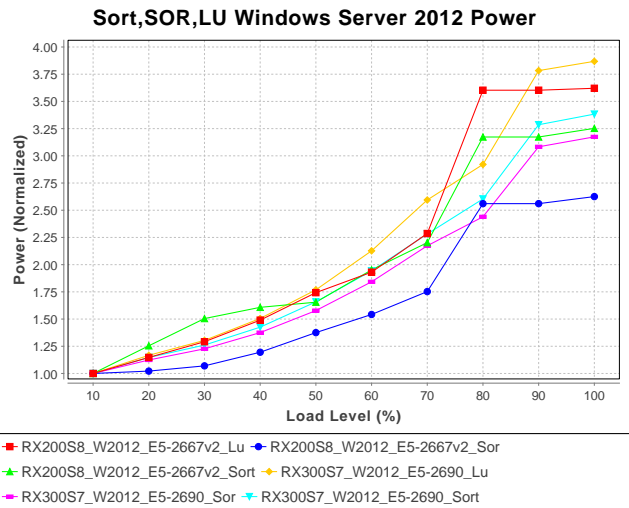
#### 5.4 Effect of Processor Architecture

We now evaluate the effect of the processor architecture



**Figure 14: Normalized Power Consumption of the Baseline System Running Windows Server 2012**

on the character of the power over load function on both Linux and Windows. We run the Windows comparison on our baseline Fujitsu system with its Intel Sandy Bridge type processor and on a Fujitsu PRIMERGY RX200S8 with 8 core Intel Xeon E5-2667v2 processors at 3.3 GHz (3.6 GHz with Turbo). Both systems feature 8x8 GB RAM and two processor sockets. The Xeon E5-2667v2 processor was selected as it is also an 8 core processor with an only slightly higher clock frequency.



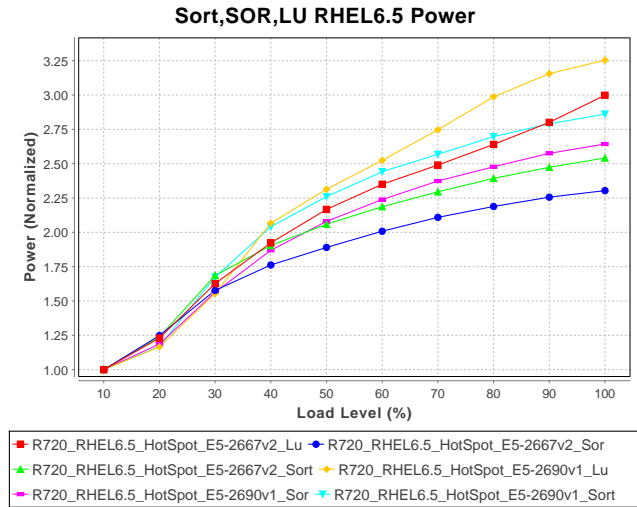
**Figure 15: Normalized Power Consumption of the Baseline Sandy Bridge System and an Ivy Bridge System, both Running Windows Server 2012**

Linux CPU architecture comparisons are performed on a Dell PowerEdge R720 system, also using the same two processors and 8x8 GB RAM. These systems run Red Hat Enterprise Linux 6.5. Dell BIOS power management has been disabled on those systems to enable an accurate accounting of the effects of operating system power management functionality.

For better visibility we present the results for the Sort, SOR, and LU worklets. The differences between these worklets

are representative when comparing SERT’s CPU worklet power consumption over the two architectures.

Comparison of the worklet’s power draw on Windows (see Fig. 15) shows, that power consumption tops out far earlier for all worklets on the E5-2667v2 processor. On Sandy Bridge power consumption stops its exponential increase at 90% load. Ivy Bridge, on the other hand, features an only minimal increase starting at 80% load. Another visible deviation is found for the Sort worklet, which features a bigger deviation from the exponential base pattern on the E5-2667v2 CPU than on the older E5-2690. Deviations of this kind are also visible on Sandy Bridge for other worklets (such as LU), yet occur at higher load levels than the Sort deviation on the E5-2667v2.

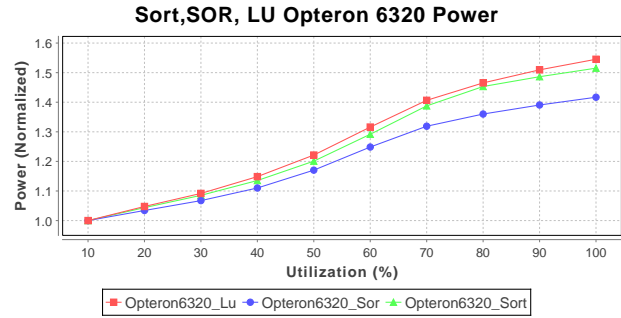


**Figure 16: Normalized Power Consumption of a Sandy Bridge System and Ivy Bridge System, both Running Linux**

The power consumption of the worklets on Linux (see Fig. 16) shows only small differences between the processor architectures. LU features a more linear power increase over load on Ivy Bridge, with a slight additional increase around 90% load. Differences between power consumption at 10% and 100% load are also not as large overall. The curve shapes are relatively similar, however.

Notably, the Dell R720 system running RHEL 6.5 does not show the steep rise in power consumption around the 80% load mark that was measured on the systems in Section 5.2 running RHEL 6.4. In our previous measurements, we already observed the absence of this sudden increase when using the Intel Xeon E5-2643 processor (see Fig. 7). The RHEL 6.4 measurements using the Xeon E5-2690 processor, however, did feature the increase after the concave function segment.

We also evaluate the shape of the power curve for an AMD Opteron based system on an HP ProLiant DL385p Gen8 machine, running Windows Server 2008R2 (Fig. 17) with two AMD Opteron 6320 processors (8 cores each) at 2.8 GHz (3.3 GHz with Turbo) and 8 x 8 GB of RAM. This curve shows major differences with the power curves of other architectures. It is still clearly a power curve as measured on a Windows system, as it features the characteristic exponential power increase over utilization also measured on the Intel systems. The sudden increase in power around the 80%

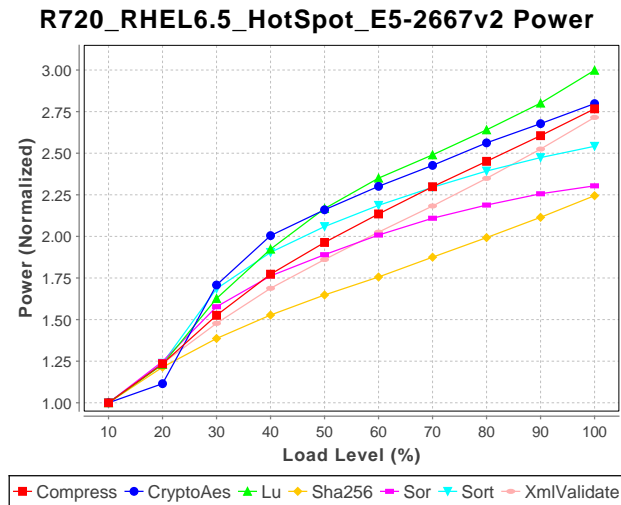


**Figure 17: Normalized Power Consumption of an AMD Opteron based System, running Windows Server 2008R2**

load level, however, is missing and is replaced by a concave function segment beginning at 60% load.

### 5.5 Effect of JVM

Similarly to our exploration of operating system impact on the power / load function, we also measure the impact of the Java Virtual Machine (JVM). For these measurements we employ a Dell PowerEdge R720 machine equipped with two Intel Xeon E5-2667v2 processors and 8 DIMMs of 8 GB DDR3 memory at 1866 MHz.



**Figure 18: Normalized Power Consumption of Dell PowerEdge Running the Oracle HotSpot JVM**

Fig. 18 shows the worklets’ normalized power consumption running on Oracle’s HotSpot JVM and Fig. 19 shows the worklets’ power consumption on IBM’s J9 JVM. The choice of JVM does not have as significant of an impact on the overall curve shape as the choice of operating system. Some differences are visible though.

Small changes can be observed in the consumption of the LU, Compress, and XMLValidate worklets. For the LU worklet, the concave function segment is flattened out at higher load levels, when using the J9 JVM. As increase in power consumption drops slightly on HotSpot, J9 features a linear rise. Compress and XMLValidate feature the return of the segment with a steep increase in power consumption following the concave segment on J9. This effect was visible for our HotSpot measurements on the Sandy Bridge

architecture on RHEL 6.4, but has almost completely disappeared on RHEL 6.5. Running the worklets on J9 results in this increase returning in some cases between the 90% and 100% load level. The reverse can be observed for the LU worklet. LU features a minimal increase in the rise of power consumption between the 90% and 100% load level on HotSpot. This effect is not visible on J9.

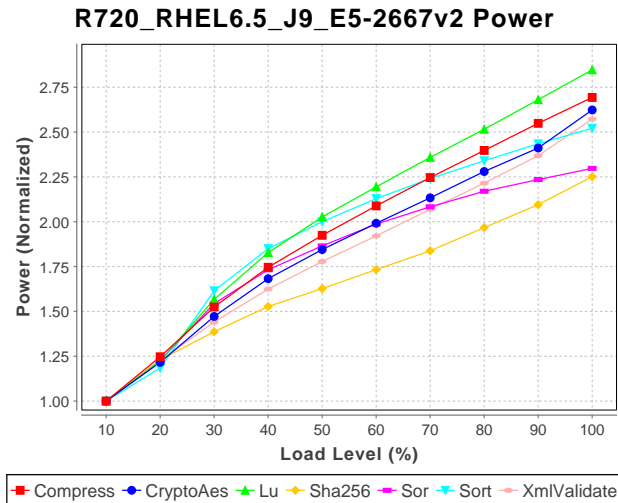


Figure 19: Normalized Power Consumption of Dell PowerEdge Running IBM’s J9 JVM

The biggest differences, however, can be observed for the CryptoAES worklet. Newer versions of HotSpot feature integrated AES support with a significant performance boost. This has an impact on both energy efficiency and power consumption. At low load, normalized power consumption is higher on HotSpot, indicating a steeper increase. The consumption levels out towards higher utilization levels, as the observations for the LU worklet repeat. On HotSpot, the concave segment of the worklet’s power over load function is more pronounced, whereas it seems almost linear on J9. For CryptoAES this effect is even stronger and more visible than for LU. Additionally, CryptoAES features the steep increase in power draw at the 90% to 100% level, whereas it does not on HotSpot.

We confirmed our observations of worklet differences depending on JVM with a second measurement series using an Intel Xeon E5-2697v2 12 core processor. This second measurement series confirmed all of our observations except for one. The slight additional increase in power consumption of LU at the 80% to 100% load interval on HotSpot is missing on the E5-2697v2 system. Instead, it continues the concave power curve in the same manner as CryptoAES and Compress.

Our measurements for the power consumption of the two JVMs confirm that JVMs influence power consumption indirectly, mostly through their impact on worklet performance and support of hardware and operating system features as they do not provide additional power management mechanisms.

## 5.6 Characteristic Power / Utilization Curves

Over the course of our measurements, we have observed two distinct types of power over load curves, depending on the server’s operating system. Within these types the power

over load behavior of worklets can be approximated by piecewise mathematical functions:

- **Linux:** Power consumption over load on Linux follows a similar pattern for all CPU-heavy workloads: Functions approximating this power consumption start with a linear segment followed by a concave segment. The concave segment is optionally followed by a steep increase in power consumption around the 80% load level. After this sudden increase the concave segment resumes.
- **Windows:** Power consumption on Windows resembles an exponential function. This exponential base function can be modified by a concave deviation, usually between the 40% and 80% load levels.

The form in which these basic function templates are realized then depends on the workload type and the hardware configuration. These factors influence both the load levels at which each function segment begins and terminates, as well as the actual power output at these points.

Most notably, the concave function segments on all platforms can be pronounced with differing strengths. They can be clearly visible, as it is for CryptoAES and Sort, but they can also be almost non-existent, resulting in a linear function for the Linux case or the default exponential function on Windows.

### 5.6.1 Characteristic Energy-Efficiency / Utilization Curves

Due to SERT’s linear calibration of target load levels through the number of dispatched transactions, the shape of the function approximating energy efficiency is a direct mirror of the worklet’s power draw.

This direct correlation changes once specific bottlenecks or architectures with different performance attributes are introduced. As both of these factors significantly influence the overall system performance at each load level, the energy efficiency is affected as well.

In our experiments, we observed this change when comparing the 2 socket and 4 socket systems. The 4 socket system features significantly more CPU cores of a different architecture, leading to different performance characteristics. As a result its energy efficiency deviates greatly from the 2 socket system.

The effect of bottlenecks can be easily observed for the CryptoAES worklet using only two DIMMs of RAM. The under-saturation of available memory channels leads to hardware contention for memory access and causes a smaller increase in performance for each amount of additional power. As a result, efficiency drops at higher load levels.

## 6. CONCLUSIONS

In this paper, we demonstrate the need for characterization of server workload energy consumption and efficiency at multiple resource load levels. Specifically, the characterization of power consumption of CPU-intensive workloads is an open field of research.

We demonstrate that different workloads using CPU as their primary resource can feature significant differences in power consumption and energy efficiency. They are also affected differently by changes in the hardware and software configuration.

We characterize the power over CPU load, as well as energy efficiency over CPU load of these workloads and explore the effect of the execution environment and configuration changes on the workloads' power characteristics. Specifically, we derive two characteristic power over load functions and notice that the operating system is the major decider between these two functions.

We also explore the effect of different architectures and hardware configurations on workload power and energy efficiency, showing how they impact the character of power consumption over load. Changes in hardware configuration have a bigger impact when evaluating the overall energy efficiency instead of pure power consumption. This is especially true when considering architecture changes, as they cause shifts in performance behavior, which are reflected in the energy efficiency metric.

Several common assumptions on server energy efficiency are called into question as part of this paper. We show that the assumption of constant operating system power overhead is a gross oversimplification of the complex impact operating systems have on the system's power consumption. We also show that the assumption of maximum energy efficiency at full utilization is equally wrong in many cases, as many workloads exhibit their maximum efficiency at 80% utilization. The "U"-shape of energy efficiency curves, which is found in other literature [14] is also an oversimplification as efficiency does not always decrease monotonically after its maximum peak. We also show that the choice of Java Virtual Machine impacts power consumption mostly through secondary factors, such as performance improvements, and we demonstrate that workload specific hardware bottlenecks have a major, but non-trivial impact on power consumption and energy efficiency.

## 7. ACKNOWLEDGMENTS

The authors also wish to acknowledge current and past members of the SPECpower Committee who have contributed to the design, development, testing, and overall success of SERT: Sanjay Sharma, Nathan Totura, Mike Tricker, Greg Darnell, Karl Huppler, Van Smith, Paul Muehr, David Ott, Cathy Sandifer, Jason Glick, Ashok Emani, and Dianne Rice, as well as the late Alan Adamson and Larry Gray.

SPEC and the names SERT, SPEC PTDaemon, and SPECpower\_ssj are registered trademarks of the Standard Performance Evaluation Corporation. Additional product and service names mentioned herein may be the trademarks of their respective owners.

## 8. REFERENCES

- [1] C. Babcock. NY Times data center indictment misses the big picture. 2012.
- [2] L. Barroso and U. Holzle. The Case for Energy-Proportional Computing. *Computer*, 40(12):33–37, Dec 2007.
- [3] R. Basmadjian, N. Ali, F. Niedermeier, H. de Meer, and G. Giuliani. A Methodology to Predict the Power Consumption of Servers in Data Centres. In *Proceedings of the 2Nd International Conference on Energy-Efficient Computing and Networking, e-Energy '11*, pages 1–10, New York, NY, USA, 2011. ACM.
- [4] F. Bellosa. The Benefits of Event-Driven Energy Accounting in Power-sensitive Systems. In *Proceedings of the 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System, EW 9*, pages 37–42, New York, NY, USA, 2000. ACM.
- [5] J. L. Henning. SPEC CPU2000: measuring CPU performance in the New Millennium. *Computer*, 33(7):28–35, Jul 2000.
- [6] K.-D. Lange. Identifying Shades of Green: The SPECpower Benchmarks. *Computer*, 42(3):95–97, March 2009.
- [7] K.-D. Lange, J. A. Arnold, H. Block, N. Totura, J. Beckett, and M. G. Tricker. Further Implementation Aspects of the Server Efficiency Rating Tool (SERT). In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, ICPE '13*, pages 349–360, New York, NY, USA, 2013. ACM.
- [8] K.-D. Lange and M. G. Tricker. The Design and Development of the Server Efficiency Rating Tool (SERT). In *Proceedings of the 2Nd ACM/SPEC International Conference on Performance Engineering, ICPE '11*, pages 145–150, New York, NY, USA, 2011. ACM.
- [9] K.-D. Lange, M. G. Tricker, J. A. Arnold, H. Block, and C. Koopmann. The Implementation of the Server Efficiency Rating Tool. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, ICPE '12*, pages 133–144, New York, NY, USA, 2012. ACM.
- [10] A. Lewis, S. Ghosh, and N.-F. Tzeng. Run-time Energy Consumption Estimation Based on Workload in Server Systems. In *Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08*, pages 4–4, Berkeley, CA, USA, 2008. USENIX Association.
- [11] M. Poess, R. O. Nambiar, K. Vaid, J. M. Stephens Jr, K. Huppler, and E. Haines. Energy benchmarks: a detailed analysis. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pages 131–140. ACM, 2010.
- [12] S. Rivoire, P. Ranganathan, and C. Kozyrakis. A Comparison of High-level Full-system Power Models. In *Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08*, pages 3–3, Berkeley, CA, USA, 2008. USENIX Association.
- [13] S. Rivoire, M. A. Shah, P. Ranganathan, and C. Kozyrakis. JouleSort: A Balanced Energy-efficiency Benchmark. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, SIGMOD '07*, pages 365–376, New York, NY, USA, 2007. ACM.
- [14] S. Srikantaiah, A. Kansal, and F. Zhao. Energy Aware Consolidation for Cloud Computing. In *Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08*, pages 10–10, Berkeley, CA, USA, 2008. USENIX Association.
- [15] Standard Performance Evaluation Corporation. Server Efficiency Rating Tool (SERT) Design Document. [http://spec.org/sert/docs/SERT-Design\\_Document.pdf](http://spec.org/sert/docs/SERT-Design_Document.pdf).
- [16] Standard Performance Evaluation Corporation. SPEC Power and Performance Benchmark Methodology. [http://spec.org/power/docs/SPEC-Power\\_and\\_Performance\\_Methodology.pdf](http://spec.org/power/docs/SPEC-Power_and_Performance_Methodology.pdf).
- [17] J. G. von Kistowski, N. R. Herbst, and S. Kounev. Modeling Variations in Load Intensity over Time. In *Proceedings of the 3rd International Workshop on Large-Scale Testing (LT 2014), co-located with the 5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*. ACM, March 2014.
- [18] T. Welch. A Technique for High-Performance Data Compression. *Computer*, 17(6):8–19, June 1984.