Towards an Enhanced Benchmark Advocating Energy-Efficient Systems

Daniel Schall[1], Volker Hoefner[1], Manuel Kern[2]

[1] Database and Information Systems Group
University of Kaiserslautern, Germany
schall,hoefner@cs.uni-kl.de

[2] SPH AG
Stuttgart, Germany
m.kern@sph-ag.com

Abstract. The growing energy consumption of data centers has become an area of research interest lately. For this reason, the research focus has broadened from a solely performance-oriented system evaluation to an exploration where energy efficiency is considered as well. The Transaction Processing Performance Council (TPC) has also reflected this shift by introducing the TPC-Energy benchmark. In this paper, we recommend extensions, refinements, and variations for such benchmarks. For this purpose, we present performance measurements of real-world DB servers and show that their mean utilization is far from peak and, thus, benchmarking results, even in conjunction with TPC-Energy, lead to inadequate assessment decisions, e.g., when a database server has to be purchased. Therefore, we propose a new kind of benchmarking paradigm that includes more realistic power measures. Our proposal will enable appraisals of database servers based on broader requirement profiles instead of focusing on sole performance. Furthermore, our energy-centric benchmarks will encourage the design and development of energy-proportional hardware and the evolution of energy-aware DBMSs.

1 Introduction

The TPC-* benchmarking suites are widely used to assess the performance of database servers. To consider a sufficiently wide spectrum of practical demands, various benchmarks were developed for simulating different kinds of applications scenarios. To illustrate the application and hardware dimensions of these benchmarks, a short overview is given.

TPC-C is an on-line transaction processing (OLTP) benchmark that measures transaction throughput of order processing on a single database instance. Typical systems under test (SUT) consist of multiple database nodes, each having several multi-core CPUs with plenty of DRAM attached (up to 512 GB per node). Additionally, a huge storage array of several thousand disks (or, recently, SSDs) is used.[3]

[3] For detailed TPC-C results, see http://www.tpc.org/tpcc/results/tpcc_perf_results.asp

TPC-E is another OLTP benchmark – simulating the workload of a brokerage firm – with similar hardware requirements as TCP-C. At the time this paper was written, the fastest SUT consisted of a single server with 8 processors (80 cores) and 2 TB DRAM. For storage, 16 SAS controllers and more than 800 disk drives were used.

TPC-H is an ad-hoc decision support benchmark processing a set of OLAP queries. Although the query types are different compared to OLTP, the hardware requirements are equally demanding. For example, the fastest server running this benchmark (for the 30 TB benchmark configuration) consists of a single node, equipped with 64 processors, 1 TB of DRAM, and more than 3,000 hard disk drives.

Obviously, the huge amount of hardware is consuming a lot of power – for all three benchmarks. As an example, the top TPC-E system mentioned earlier is consuming up to 4,500 watts at peak performance. Despite such substantial energy needs, power consumption was ignored by TPC in the past. However, the newly created benchmark *TPC-Energy* is approaching the emerging demand for energy-related measures. While TPC-[C,E and H] are reporting performance only, TPC-Energy is introducing measures for the energy consumption during query processing. TPC-Energy additionally regulates how power measurements must be performed, e.g., what measurement devices should be used and what measurement precision must be maintained. The metrics defined by TPC-Energy is *Energy Consumption* over *Work done* expressed in *Joule* per *transactions*, which translates to *Power Consumption* over *Work delivered* expressed in *Watts* per *tps*.

Although TPC-Energy made a first step towards energy-related measures compared to the former performance-centric TCP benchmarks, we advocate appropriate measures which are still missing to get meaningful insights into the servers' energy consumption behavior, e.g., power usage over system utilization. For this reason, we propose a new benchmarking paradigm which extends the already existing TPC-Energy.

This paper is structured as follows: In Section 2, we introduce some definitions regarding power consumption and energy and pinpoint the influence of the server load to the power consumption. In Section 3, we will briefly review related energy-centric benchmarks, whereas we will revisit TPC-Energy and point out its limitations by showing measurements of real-world servers in Section 4. We will argue that servers are usually not running at peak load and discuss the implications on the power consumption. Based on our findings in the preceding sections, we propose a new kind of benchmarking paradigm in Section 5. Our proposal will overcome the limitations we identified earlier. Finally, in Section 6, we conclude our contribution and give an outlook, how a new benchmarking paradigm can change both the way systems are built and the customers' view on new systems.

2 Energy Efficiency and Other Measures

Energy consumption of data centers is steadily growing due to the ascending number of server installations and due to the increasing power consumption of each server. At the same time, energy costs are rising continuously. For an average utilization period (∼5 years), energy costs have now drawn level with the server's acquisition cost [3]. The total cost of ownership is therefore heavily influenced by the energy footprint of the devices. Further costs, over and above the plain energy cost of powering the servers, come to play, i.e., cooling cost, additional batteries, and amply dimensioned power switches.

In recent years, a lot of efforts have been made to limit the spendings on energy. Some of these efforts include the building of data centers in regions, where power can be cheaply acquired, or the augmented utilization of servers by virtualization techniques [7]. Nevertheless, the best approach is reducing the power consumption of the hardware in the first place. As the power consumption of a server does not scale linearly with its utilization, overdimensioned hardware has a huge impact on its overall energy consumption. Figure 1 charts the power consumption at various compute loads. As the load level drops, the power consumption does not scale well. Even at idle, the system uses about 50 % of its peak power.

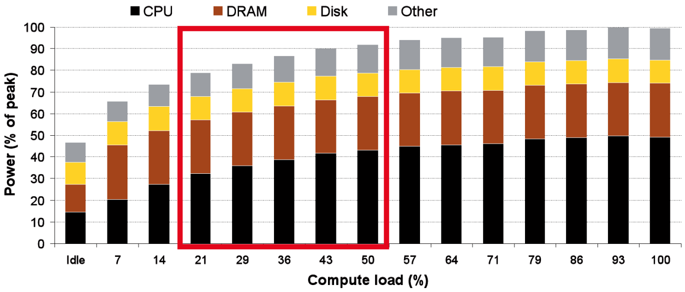


Fig. 1. Power by component at different activity levels, from [6]

Due to the growing influence of the energy consumption on buyers' decisions, it is crucial to provide sound energy measurement results besides the performance data for servers. It is especially necessary to provide energy measurements that cover the whole operating area of a database server. Hence, not only the peak performance and respective power consumption are important to buyers. The average energy consumption expected for a certain machine has much more influence on the anticipated spendings during the next years and are therefore equally relevant. Because the power consumption of servers is not linear to the delivered performance, it is important to know the load curve and the resulting energy consumption of a system. The relation between consumed power and delivered work relative to their peak is called *Energy Proportionality* and will be explained in the following definitions.

2.1 Definitions

In this paper, we will use the term *power* (or power consumption) to denote the current consumption of electricity; hence, the unit of power is *Watt*. The power consumption of a server/component over time is called *energy* (or energy consumption). Energy consumption is expressed in *Joule*:

$$energy = \int power\ dt$$
$$1\ Joule = 1\ Watt \cdot 1\ second$$

To quantify the amount of computational work done on a server, different hardware-related measures can be employed, e.g. $MIPS$, $FLOPS$, $IOPS$, or more complex, often application-related measures. In the database community, and for the TPC-* benchmarks as well, the number of transactions – defined as specific units of work in the application environments addressed by the TPC-* workloads – has prevailed as an application-related measure for the quantity of computations. We are using the same (generic) measure referring to *transactions per time unit*, in particular, *transactions per second (tps)* here:

$$1\ tps = \frac{1\ transaction}{1\ second}$$

Because of the transactions' dependency on the specific application scenario, only results from the same benchmark, hence, with the same dataset and the same set of queries, are comparable. Hence, such performance figures must always be qualified by the respective benchmark (such as *tpmC* or *tpsE*). In this paper, performance measures are expressed in *tps*, but other qualifiers can be used exchangeablely.

To express how efficiently a certain set of queries can be processed using a given amount of energy, we use the term *energy efficiency*:

$$energy\ efficiency = \frac{\#\ of\ transactions}{energy\ consumption}$$

which can be transformed to the amount of work done per time unit when a certain amount of power is given:

$$energy\ efficiency = \frac{tps}{Watt}$$

The higher the energy efficiency, the better a given system transforms electricity into "work". Note, this is the inverse of the formula used in TPC-Energy which applies $\frac{Watt}{tps}$ as its metrics. The rationale of the TPC for choosing the inverse was the desire to be similar to the traditional TPC metrics *price over throughput* and, furthermore, to allow a secondary metrics for each of the subsystems. To conform with already established practices, we will use $\frac{Watt}{tps}$ as well.

In addition to that absolute measure, we are using the term *energy proportionality*, coined by [1], when we want to reference the power consumption of a server (at a given level of system utilization) *relative* to its peak consumption. Ideally, the power consumption of a system should be determined by its utilization [2]. Hence, energy proportionality describes the ability of a system to scale its power consumption linearly with the utilization.

Therefore, energy proportionality can not be expressed using a scalar value. Instead, a function or graph is needed to display the characteristics of a system. For each level x, $0 \leq x \leq 1$, of system utilization[4], we can measure the power used and denote this value as the actual power consumption at load level x . To facilitate comparison, we use relative figures and normalize the actual power consumption at peak load ($x = 1$) to 1, i.e., $PC_{act}(x = 1) = 1$. Using this notation, we can characterize a system whose power consumption is constant and independent of the actual load by $PC_{act}(x) = 1$.

Note, we obtain by definition true energy proportionality at peak load, i.e., $PC_{ideal}(x = 1) = 1$. In turn, a truly energy-proportional system would consume no energy when it is idle (zero energy needs), i.e., $PC_{ideal}(x = 0) = 0$. Due to the linear relationship of energy proportionality to the level of system utilization, we can express the ideal power consumption at load level x by $PC_{ideal}(x) = x$.

With these definitions, we can express the energy proportionality $EP(x)$ of a system as a function of the load level x:

$$EP(x) = \frac{PC_{ideal}(x)}{PC_{act}(x)} = \frac{x}{PC_{act}(x)} \tag{1}$$

This formula delivers EP values ranging from 0 to 1. Note, for $x < 1$ in a real system, $PC_{act}(x) > x$. According to our definition, each system is perfectly energy proportional at $x = 1$. If a system reaches $EP(x) = 1$, it is perfectly energy proportional for all load levels x. In turn, the more $EP(x)$ deviates from 1, the more it loses its ideal characteristics.

Using the results of Figure 1 as an example, we yield $EP(x = 0.5) = 0.55$, $EP(x = 0.3) = 0.35$, and $EP(x = 0.01) = 0.02$. Special care must be taken for defining $EP(x = 0)$ to avoid having a zero value in the numerator of the formula. In this paper, we have agreed to define $EP(x = 0) := EP(x = 0.01)$. Therefore, this value should be taken with care. Nevertheless, the worst EP figure is provided by a constant-energy-consuming system in idle mode: $EP(x = 0) = 0$.

Obviously, assessing energy proportionality is a lot more expressive than mere energy consumption. While the latter only captures a single point of the system's energy characteristics, the former reveals the ability of the system to adapt the power consumption to the current load.

[4] By multiplying x by 100%, the percentage of system utilization can be obtained

3 Related Benchmarks

In this section, we will give a short overview of existing benchmarks and their relation to energy measurements. Poess et al. provide a more detailed summary in [4].

As the first observation, the TPC-[C, E, H] benchmarks are not considering energy consumption at all. These benchmarks are purely performance-centric. The TPC-Energy benchmark – an extension to any of the three benchmarks – is defining measurements for energy consumption. It gives advice how to measure the power consumption while the benchmark is running and provides additional guidelines how to measure the power consumption of the idle system.

SPEC (the Standard Performance Evaluation Corporation) has introduced the *SPECpower_ssj2008* benchmark for measuring the performance and energy consumption of a system running Java-based workloads. In contrast to TPC-Energy, the SPEC benchmark does measure power consumption at 11 load levels (from 0% load to 100% load) and aggregates the measurements by the geometric mean to form a single result. Additionally, newer releases of the SPEC benchmark like *SPECweb_2009* and *SPECvirt_sc2010* incorporate the power measurement methodologies from SPECpower.

Apart from benchmarks specified by benchmark-centric organizations, the database community itself moved forward to propose an energy-related benchmark to evaluate the energy efficiency of computer systems. The *JouleSort* [5] benchmark is a sort benchmark, whose idea is to account the energy consumed for sorting a given input size. Instead of benchmarking *sorted records per $*, JouleSort is reporting *sorted records per Joule*. Thus, this benchmark reveals the energy efficiency of a computer system close to 100% load, but the scenario (just focusing on sorting) is rather narrow and differing from real-world database workloads.

In addition, the SPC (Storage Performance Council), whose benchmarks are targeted on evaluating storage components, defined energy-related extensions for their benchmarks. These extensions do not track the power consumed at peak load, but measure it at 80% (denoted as heavy) and 50% (denoted as moderate) of the peak performance as well as in idle mode. Furthermore, they introduce the weighted average power consumption based on three different usage patterns (low, medium, and high).

4 Server Load Profiles

In contrast to the assumptions made for existing benchmarks, i.e., testing a system at peak performance, real-world applications do not utilize servers that way. Typically, the hardware of database servers is designed to handle peak load; hence, it is overprovisioned for the average daily work. In the rare events of peak load, the servers reach their processing limits, but most of the time, their hardware is heavily underutilized. In the following, we will outline two studies that analyzed the energy consumption and load of servers.

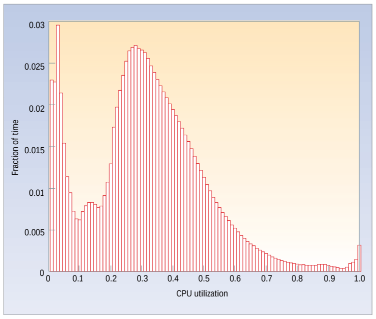


Fig. 2. Average CPU utilization of Google's servers, from [1]

4.1 Google Study

In 2007, Louiz Barroso and Urs Hölzle published a study that revealed performance data of Google's MapReduce server cluster [1]. According to this study, the servers are typically operating at 10% to 50% of their maximum performance. That way, servers are barely idle but, as well, barely fully utilized. Figure 2 charts the aggregate histogram for the CPU utilization of 5,000 servers hosted at Google.

4.2 SQL- and BI-Server Data

SPH AG monitored the performance of some of the database and analysis servers of its customers – data we use to visualize typical utilization and workload behavior of SQL and Business Intelligence (BI) applications. SPH AG is a mid-sized ERP-developing company that specializes in the branches mail order and direct marketing. Its ERP products are based on IBM System i5 or (in the considered case) on Microsoft Dynamics AX. For some of its customers, SPH AG is hosting the ERP servers in-house, including SQL servers and BI servers. The SQL servers are used to store the ERP data, such as customer, sales order, and invoice information. For the ERP system, 24/7 availability is also needed, because on-line shops are connected to the ERP systems. The BI servers are used to process data of the SQL servers for the preparation of reports for the company management. This data is updated by a nightly job. On all servers, a thorough performance and load monitoring is installed.

Every customer gets its own SQL and BI server to isolate the user data of different customers at hardware level. Figure 3 shows a sketch of the systems' layout. The OLTP server on the left-hand side is processing transactional workloads issued by an upstream ERP system. The BI server at the right-hand side is pulling all tables of interest in a nightly job from the OLTP server. After the new data has arrived, the BI server starts building indexes and running OLAP queries. The results are stored on the BI server and can be accessed by the management. Both servers consist of two Intel Xeon E5620 2.4 GHz with 8 cores per CPU and 32 GB DRAM in total. They have two hard disk drives attached, one for the database files which has 800 GB, and the second one for the log files

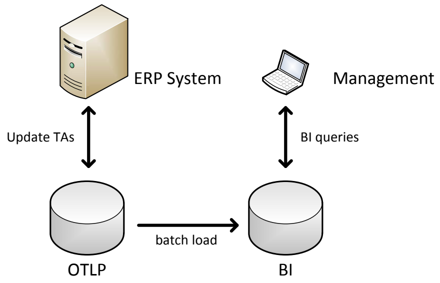
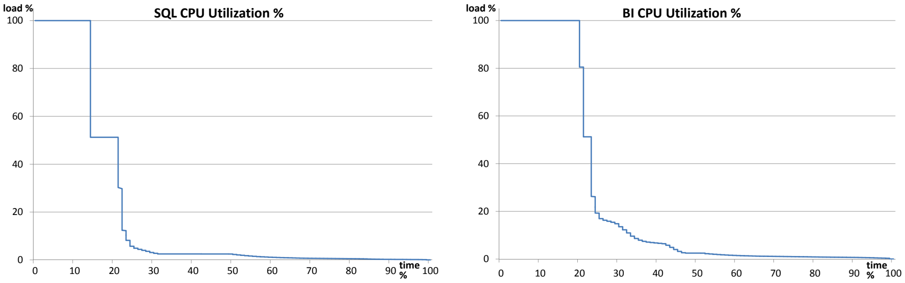


Fig. 3. SQL and BI server

(300 GB). The servers are interconnected via Ethernet. One may argue that this hardware configuration is not very powerful for servers typically benchmarked with TPC-*, but it delivers sufficient power for these tasks. As we will see, even this configuration is overprovisioned for the average daily load.

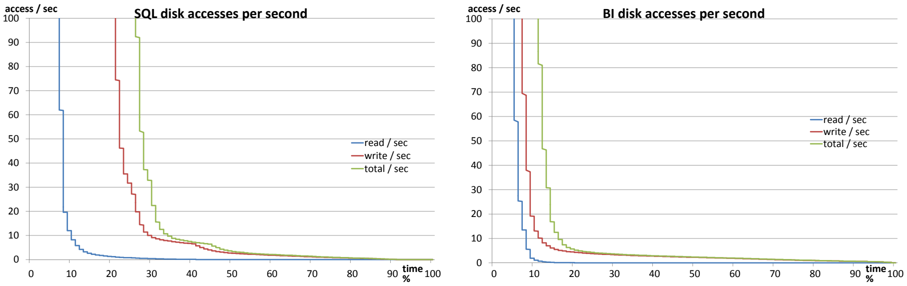
We analyzed the performance-monitoring log files from SPH and charted the servers' CPU and disk utilization for some customer. The overall disk and pro-



(a) SQL server

(b) BI server

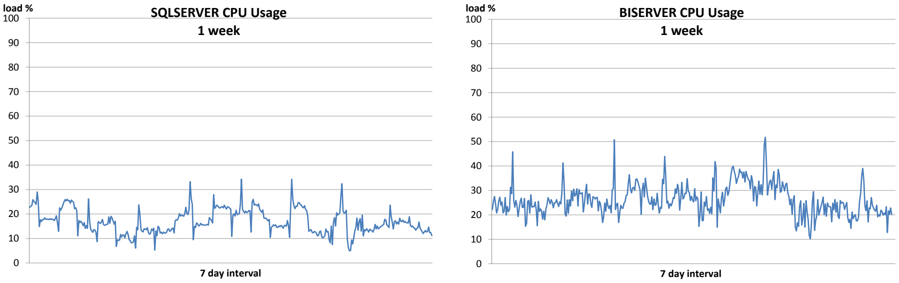
Fig. 4. CPU utilization histograms



(a) SQL server

(b) BI server

Fig. 5. Disk utilization histograms



(a) SQL server

(b) BI server

Fig. 6. Weekly CPU usage

cessor utilization histograms of the servers are depicted in figures 4(a), 4(b), 5(a), and 5(b). For an arbitrary day in the data set, the graphs show the probability distribution of the CPU utilization sorted from 100% to 0% (idle) and the disks' utilization by *accesses/second*, respectively. As the graphs indicate, the servers spend most of their time idle.

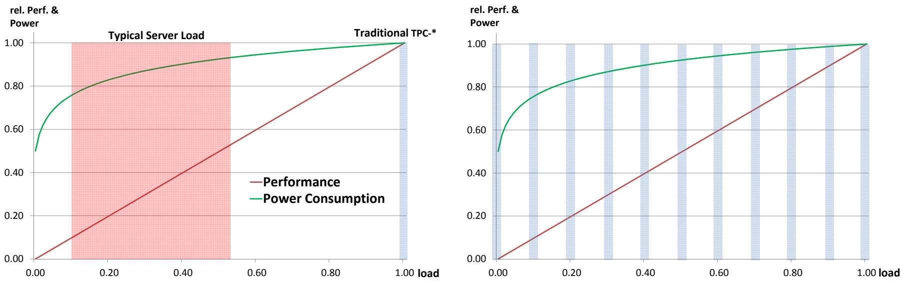
A weekly breakdown of the servers CPU load is depicted in figures 6(a) and 6(b). At night, the BI starts its update cycle and gathers daily transaction data from the SQL server. During some rush hours, the SQL server is processing most of the transactional workload, while customers and employees are accessing the ERP system. During the rest of the day, the servers are heavily underutilized.

Overall, it gets obvious that the claims made by Barroso and Hölzle apply to these servers as well. As the figures 6(a) and 6(b) show, the servers are utilized about 25% of the time and very rarely at peak.

5 Proposal

Based on the observations in the previous section, it is easy to see that current server installations do not behave like the systems measured in traditional benchmarks. While benchmarks usually measure peak performance, typical servers operate far away from that point during most of the time. Nevertheless, benchmark results are comparable and meaningful when it comes to performance only. As long as attention is not turned to energy consumption, the mismatch between benchmarking and real usage of servers does not carry weight. Performance measurements under peak utilization can be easily broken down to lower load situations. Hence, a system, able to process *x tps per second* at peak, can also process *0.5x tps per second*.

In contrast, energy-related measurements obtained at some point of utilization are not transferable to other load situations because of the non-linear scaling of energy consumption of todays computer hardware. Therefore, the whole span of different load situations should be measured separately to obtain meaningful energy data for customers.



(a) TPC-* vs. the real world (b) Static energy-efficiency measurement

Fig. 7. Comparing benchmark paradigms

As an analogy from a well-known field, automobiles are benchmarked similarly with additional "energy-related" measures. Hence, the power of a car is estimated by its horse power and its top speed, like database servers are classified by their hardware and their peak *tpmC* / *QphH* / *tpsE*. On the other hand, the gas consumption of a car, estimated at top speed, is meaningless for the average driver, because the measurement does not reveal the gas consumption for average usages. Therefore, a car's mileage is measured by driving the car through a set of standardized usage profiles which reflect the typical use of the vehicle. The same paradigm should be applied to database benchmarks as well, where energy consumption measured at peak utilization is no indicator for the average use case.

Figure 7(a) depicts the point that all TPC-* benchmarks measure compared to the typical working region of a server. Note the mismatch in energy efficiency between both regions. To cope with the limitations we have outlined previously and to keep the TPC benchmarking suite up to date, we propose a new paradigm in benchmarking.

5.1 Static Weighted Energy Proportionality

Nowadays, the measurement paradigm for the TPC benchmarks strictly focuses on performance results, i.e., to get the most (in terms of units of work) out of the SUT. Hence, this methodology collides with the desire to get a meaningful energy-efficiency metrics for the system. Therefore, we propose a sequence of small benchmarks that utilize the SUT at different load levels, instead of a single run at peak load. Figure 7(b) depicts a feasible set of benchmark runs at different utilization ratios. First, a traditional TPC-* run will be performed, i.e., at full utilization. That run is used as a baseline to get the maximum possible performance the SUT can handle (see equation 2). Next, based on the results from the first run, the number of queries per second issued for the other runs is calculated using equation 3, where x denotes the system utilization between 0 and 1.

$$baseline := \frac{transactions}{second} \text{ @100\%} \tag{2}$$

$$\frac{transactions}{second} \text{ @}x := baseline \cdot x \tag{3}$$

Of course, depending on the type of benchmark, the characteristics and knobs for throttling can differ, e.g., for TPC-C increasing the think time seems reasonable while for TPC-H a reduction of concurrent streams is the only possibility. We call this a *static weighted energy-proportionality benchmark*, because the workload does not change in between and, therefore, the system does not have to adapt to new load situations. To allow the system adapting to the current workload, a preparation phase of a certain timespan is preceding each run. During the preparation time, the SUT can identify the workload and adapt its configuration accordingly. It is up to the system whether and how to adapt to the workload, e.g., the system can power down unused CPU cores or consolidate the workload on fewer nodes in order to save energy. After the preparation phase, the overall energy consumption during the run is measured. In other words, instead of measuring the performance of the SUT, we are now measuring the power consumption for a certain system usage.

At each load level, the system's energy proportionality (according to equation 1) is calculated. By multiplying each result with the relative amount of time the system is running at that load level, we can estimate the overall energy proportionality under realistic workloads.

Formula Let EP_i be the energy proportionality at load level i, and let T_i be the relative time, the system operates at that level. Then, the static weighted energy proportionality of the system ($= SWEP$) can be calculated as:

$$SWEP = \int_{i=0.0}^{1.0} EP_i \cdot T_i \; di \tag{4}$$

We can estimate the power consumption ($= PC$) of the SUT during the measured interval by multiplying the (absolute) power consumption of each interval (PC_i) with the relative time, the system operates in that load interval:

$$PC = \int_{i=0.0}^{1.0} PC_i \cdot T_i \; di \qquad [Watts] \tag{5}$$

Furthermore, by adding the system's performance to the formula (denoted as *tps* in the following), we can estimate the overall energy efficiency.

$$EE = \int_{i=0.0}^{1.0} \frac{PC_i}{tps_i} \cdot T_i \; di \qquad \left[\frac{Watts}{tps}\right] \tag{6}$$

Table 1. Example calculation of the SWEP

		real system			energy-proportional system		
load	rel. time	rel. PC	EP	EP · time rel.	rel. PC	EP	EP · time rel.
idle	0.11	0.47	0.00	0.00	0.0	1.0	0.11
0.1	0.08	0.7	0.14	0.01	0.1	1.0	0.08
0.2	0.19	0.78	0.26	0.05	0.2	1.0	0.19
0.3	0.23	0.84	0.36	0.08	0.3	1.0	0.23
0.4	0.18	0.88	0.45	0.08	0.4	1.0	0.18
0.5	0.10	0.91	0.55	0.05	0.5	1.0	0.10
0.6	0.05	0.93	0.65	0.03	0.6	1.0	0.05
0.7	0.02	0.94	0.74	0.01	0.7	1.0	0.02
0.8	0.01	0.98	0.82	0.01	0.8	1.0	0.01
0.9	0.01	0.99	0.91	0.01	0.9	1.0	0.01
1.0	0.02	1.00	1.00	0.02	1.0	1.0	0.02
SWEP ($=\sum$)				0.37			1.00

In a practical application, the integrals in the formulas 4, 5, and 6 are approximated by the sum of load situations measured, e.g., by eleven measurements of loads from 0% to 100% using a 10% increment.

Example To clarify the calculation of the weighted average, we will give an example using the load and energy measurements provided by Google (see Figures 1 and 2). Table 1 shows the (relative) average power consumption and time fractions of a hypothetical server for 11 utilization levels. The data is derived from the two studies done by Google. For comparison, the relative energy footprint of a theoretical, perfectly energy-proportional system is shown.

This static approach has certain drawbacks: First, the measurements are rather coarse grained in reality, i.e., “reasonable” static measurements will be employed at 0, 10, 20, . . . , 100% load, but not in greater detail. And second, this calculation does not take transition times from one load/power level to another into account.

5.2 Dynamic Weighted Energy Efficiency

To design an energy-related benchmark that overcomes the drawbacks of the static approach, we are proposing a refinement of the previous benchmark, called *dynamic weighted energy-efficiency benchmark* (DWEE). In order to simulate an even more realistic workload on the SUT, the static measurements at various load levels of the SWEP benchmark are replaced by continuous sequences of different length and different load situations (so called *Scenes*), followed by each other without interruption or preparation times. In contrast to the static approach, all scenes run consecutively, thus transition times are measured as well in this benchmark. That enables us to test the systems ability to dynamically adapt (if possible) while running.

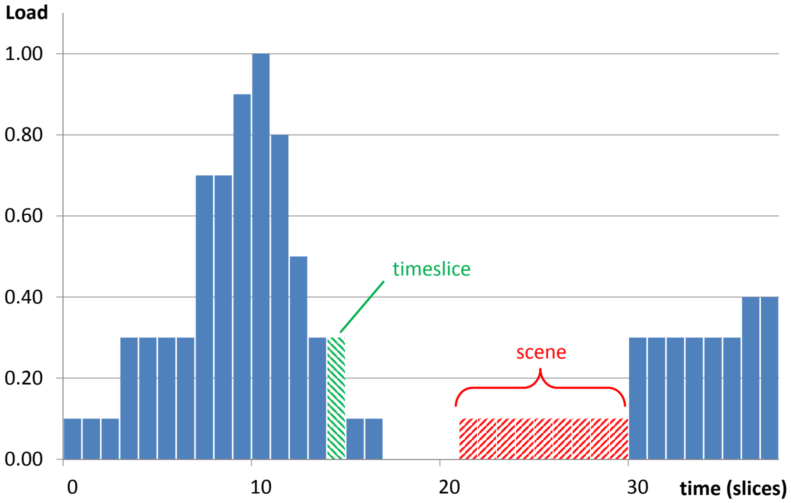


Fig. 8. Dynamic weighted energy-efficiency benchmark – sample load pattern

Every scene will run for a defined timespan T, as sketched in Figure 8. A *timespan* is always a cardinal multiple of a constant *time slice* t, thus, all scenes run for a multiple of that time slice.

The dynamic energy-efficiency benchmark should simulate a typical workload pattern, hence, the sequence of load levels should reflect the intended usage pattern of the SUT.

Influence of the length of the time slice *t* By adjusting the cardinal time slice t to smaller values, all benchmarking scenes will be shorter, hence, the system must react faster to changing loads. Such a course of action enables testing the SUTs ability to quickly react to changes. Of course, when benchmarking different systems, the results for a specific benchmark can only be compared by choosing the same time slice t and the same sequence of scenes.

The minimum length of the time slice should not go below 10 minutes, because real-world utilization usually does not change faster than that.

Formula Calculating the result of the DWEE benchmark is simpler than calculating the SWEP results, because the weighting of the utilization is determined by the selection of scenes. Because we are measuring the overall throughput and energy consumption, we do not have to aggregate several measurements. To obtain comparable results, benchmark runs will be characterized by $Watt/tps$. Hence, the overall result of the dynamic weighted energy-efficiency benchmark, short $DWEE$, is:

$$DWEE = \frac{overall\ Energy\ Consumption}{overall\ \#\ of\ Transactions} \qquad \left[\frac{Joule}{transactions} = \frac{Watt}{tps}\right]$$

Hence, by employing the same sequence of scenes and the same length of t, the energy efficiency of different systems can be compared to each other. Because the benchmark closely simulates a daily workload, the energy consumption to be anticipated by the system under test can be estimated for its prospective use.

6 Conclusion and Future Work

In times of high energy cost and rising environmental concerns, it is crucial to shift the focus from a purely performance-centric view to a more comprehensive look. For this reason, we have proposed two additional measures to the widely used TPC-* benchmarks. The results from the benchmark runs can be used to estimate the average power consumption for given usage profiles. By comparing the overall performance and the load-specific energy efficiency, systems can be compared for arbitrary utilization profiles. Of course, customers need to be aware of the specific usage profile of their servers to get meaningful results from the benchmarks.

We have explained, why high throughput as sole optimization criterion and $/*tps* (or $/*tpmC*, $/*tpsE*, etc.) as the solitary driver for purchase decisions are no longer up to date. Therefore, we proposed a paradigm shift for the TPC benchmarks: the same shift that has already moved *SPEC* and *SPC* to energy-related benchmarks. This paper introduced more sophisticated energy measures to allow a more detailed view of the systems' energy efficiency. By comparing the *static weighted energy proprtionality* of two servers, one can easily derive, which server is the more energy proportional one. Additionally, if the usage pattern of the server is known, the servers real energy consumption can be estimated. Finally, for getting more realistic, energy-related results, the *DWEE benchmark* can be run with workloads, that reflect the estimated usage for a customer. These workloads could stem from historical performance data provided by the customer to enable tailor-made comparisons of different systems. Alternatively, a workload specified by the TPC would enable standardized benchmarking as usual, with respect to energy.

Since customers are slowly becoming energy-aware also as far as their computer equipment is concerned, measures revealing energy-related characteristics of servers are gaining increasing attention. Our benchmark proposal will help comparing the energy profile of different systems.

Our approach focuses on the overall power consumption of the SUT. We do not make restrictions regarding the power consumption of individual components. Therefore, it is up to the system designers how to improve the energy footprint of their systems. Some of the possibilities include powering down redundant, but underutilized components, e.g., disks in a storage array, CPU cores, or networking adapters. Other approaches could focus on the hardware level, i.e., choosing more energy-efficient components while building the system. Typical servers can widely vary in power consumption, depending on the

DRAM sizes and modules used. Finally, the software driving the hardware can have a great impact on the behavior of the system. Todays database systems do not consider energy consumption as a first-class optimization goal, e.g., the use of energy-efficient algorithms. Nevertheless, we expect future generations of database servers to show an increasing awareness of energy-related characteristics. Energy-aware operating systems and database management software can leverage energy efficiency of the plain hardware significantly.

For the future, we encourage researchers and benchmarking enthusiasts to focus on energy as well as performance. This proposal exposes a first concept how a comprehensive energy benchmark should look like. As the details have to be worked out yet, we expect our contribution to influence the design of future benchmarks.

Acknowledgements We thank our anonymous reviewers for their insightful comments. We'd also like to thank SPH AG, Stuttgart[5] for providing very useful performance data to us.

References

1. L. A. Barroso and U. Hölzle. The Case for Energy-Proportional Computing. *Computer*, 40:33–37, December 2007.
2. T. Härder, V. Hudlet, Y. Ou, and D. Schall. Energy Efficiency is not Enough, Energy Proportionality is Needed! In *Proceedings of the 16th international conference on Database systems for advanced applications*, DASFAA'11, pages 226–239, Berlin, Heidelberg, 2011. Springer-Verlag.
3. M. Poess and R. O. Nambiar. Energy Cost, The Key Challenge of Today's Data Centers: A Power Consumption Analysis of TPC-C Results. *PVLDB*, 1(2):1229–1240, 2008.
4. 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*, e-Energy '10, pages 131–140, New York, NY, USA, 2010. ACM.
5. S. Rivoire, M. A. Shah, P. Ranganathan, and C. Kozyrakis. JouleSort: A Balanced Energy-Efficiency Benchmark. In *SIGMOD Conference*, pages 365–376, 2007.
6. A. Z. Spector. Distributed Computing at Multi-dimensional Scale (Keynote). In *Proceedings of Int. Middleware Conference*, 2008.
7. A. Weiss. Computing in the Clouds. *netWorker*, 11:16–25, December 2007.

[5] www.sph-ag.com