Data Center Performance Model for Evaluating Load Dependent Energy Efficiency

Daniel Schlitt
OFFIS
Institute for Information Technology
Oldenburg, Germany
Email: schlitt@offis.de

Wolfgang Nebel
C.v.O. University of Oldenburg
Oldenburg, Germany
Email: wolfgang.nebel@uni-oldenburg.de

***Abstract*—Energy efficiency metrics are important tools for data center operators to optimize their facilities and thereby decreasing operational expenses while strengthen competitiveness. However, commonly used metrics like Power Usage Effectiveness do not consider productivity or suitable proxy indicators, thus lacking the ability for correctly comparing energy efficiency between data centers. Also, other known metrics which consider productivity, do this in a subjective way, i.e. results are only comparable for the same definitions. In order to address these shortcomings we proposed the Load Dependent Energy Efficiency (LDEE) metric, which uses a combination of utilization, performance, and power models to provide detailed efficiency data. By using load dependent models the concrete workloads are abstracted realizing comparability. Furthermore, models are trained with public information such as hardware specifications and benchmark results to avoid disruption of operation and thereby increasing applicability. This paper focuses on the utilization and performance models of LDEE.**

***Index Terms*—Data center, performance modeling, load modeling, energy efficiency metric, benchmarks**

I. Introduction

Data centers (DCs) with their computing, storage, and network capacities are the backbone of today's information and communication technology (ICT). With rising usage of ICT-supported services also the demand for DCs and their capacities increases. While IT components are getting more and more energy efficient, the total energy demand and thereby operational expenses are still rising. Thus, improving energy efficiency (EE) of DCs is an important topic for their operators to stay competitive in a highly contested industry. The first step, however, is the EE measurement in order to evaluate the as-is state and finding suitable starting points for improvements.

Currently, the most commonly used EE metric is Power Usage Effectiveness (PUE) by The Green Grid [1], which relates total DC energy to IT energy usage. While it is easy to measure as well as to understand, PUE is not significant for EE in its typical definition – useful output related to the invested energy – but is an indicator for energy overheads in DC infrastructure, i.e. inefficient climatisation or power supply. However, there are other known metrics which try to implement the EE definition by evaluating the accomplished useful work, such as DC Energy Productivity (DCeP) or DC Performance per Watt (DCPpW). These metrics try to abstract productivity by measuring the number of relevant successful tasks, which have to be defined and prioritized in relation to each other prior to application. These definitions have to be done manually with the consequence that comparisons between DCs with different task loads are not feasible, i.e. such metrics can only be used for single DCs.

In contrast to the mentioned metrics the Load Dependent Energy Efficiency (LDEE) takes an other approach on abstracting productivity. By using component models for objectively measurable utilization, performance, and power, comparability between DCs can be ensured. Furthermore, the models can be trained with publicly available data like results from performance and energy benchmarks in order to enhance the applicability of LDEE, which is another drawback of current productivity metrics. In addition to this, the usage of component models allows the evaluation of component EE next to the high level view of a complete DC, i.e. while LDEE provides a general overview over the as-is state of EE it also enables identification of suitable optimization points. The main concept of LDEE has already been introduced in prior published work [2]. In this paper, we focus on the utilization and performance evaluation part of the LDEE, which shall abstract the DC productivity. Therefore, respective models are proposed which utilize IT hardware specifications as well as public performance benchmarks to realize preferably non-invasive surveys.

The remainder of this paper is as follows. In Section II related work in the topics metrics, benchmarks, and performance modeling is presented. Afterwards, the proposed models for IT load (Section IV) and performance (Section V) are described in detail. In Section VI the models will be evaluated and compared to other approaches. The paper closes with a conclusion.

II. Related Work

The most commonly used EE metric is the aforementioned PUE [1], which has the drawback of not considering the output in the sense of work done but as energy needed by the IT. As actual accomplished work may differ enormously for same power draws, the PUE is not meaningful as EE metric for whole DCs but only for infrastructure energy overhead. Metrics with focus on productivity are DCeP [3], DCPpW [4], DPPE [5], and DC-EEP [6]. However, all these metrics need subjective manual definitions of accomplished work (e.g. which tasks to count) which prevents comparisons between DCs. Further information about EE metrics in DC context is given in [2] and [7].

Interest groups like Standard Performance Evaluation Corporation (SPEC), Transaction Processing Performance Council (TPC), and Storage Performance Council (SPC) cover a broad range of performance and also power benchmarks for typical DC equipment which can be utilized for EE evaluations. They differ mainly in the used workloads and the target system under test (SUT). In contrast, high performance computing (HPC) benchmarks such as LINPACK for the Top500 and Green500 lists evaluate huge, unique HW systems which are not representative for typical DCs. There are also benchmarks published in research like JouleSort [8] or SWEEP [9]. JouleSort uses the same sorting algorithm for different IT systems and measures energy demand in addition to speed. SWEEP offers customizable synthetic workloads. However, as these benchmarks are largely unknown, they are not providing sufficient public results for performance modeling of most IT systems. The specifically used benchmarks and the reasoning behind will be explained in Section V-A1.

Solutions for performance evaluation of complete DCs are rare. An exemplary approach applied by Google was an evaluation of SW platforms like MapReduce by directly applying a sort algorithm on billions of data sets in one of their DC facilities [10]. However, this method needs specific infrastructures and is adapted to Google's demands, i.e. a general application is not realistic. An evaluation method for cloud performance of DCs is proposed by Bruneo [11]. Requests and workloads are distributed on the HW capacities and metrics such as utilization, availability, queue times, service times, and response times – in sum representing user QoS demands – are measured and used for evaluation. However, as concrete HW specifications are not considered in the model, detailed performance results are not quantifiable.

III. Load Dependent Energy Efficiency Metric

The LDEE is a holistic data center energy efficiency metric with focus on load dependency of DC components' EE. The metric consists of two main parts with (1) the performance of IT hardware and (2) the power demand of any equipment located inside DC. The LDEE is thus represented by following Equation 1,

$$LDEE(\vec{l}, T) = \frac{perf_{DC}(\vec{l})}{power_{DC}(\vec{l}, T)} \qquad (1)$$

with DC IT utilization $\vec{l} \in \mathbb{R}^2$ consisting of l_1 for server and l_2 for storage load with $l_1, l_2 \in [0, 1]$ and outside temperature T [°C] at the free cooling system as inputs. The LDEE is then defined by the relation of performance $perf_{DC}$ to the corresponding power demand $power_{DC}$ for the given input parameters.

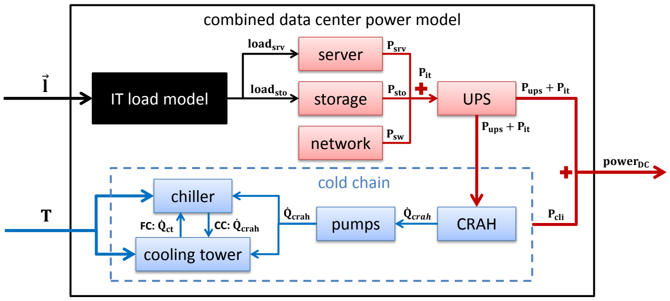


Fig. 1. DC power model composed of IT, power supply, and cooling power models with load as input and aggregated power demand as output.

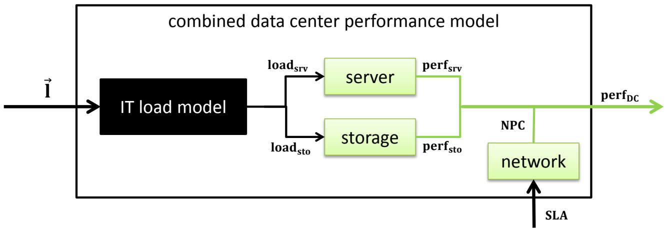


Fig. 2. DC performance model composed of IT component performance models with load as input and aggregated used performance as output.

The power demand $power_{DC}$ is estimated by a combined DC power model shown in Figure 1. A single component's power demand is modeled regarding its corresponding load dependency. For IT equipment the load is defined by the utilization of subsystems (CPU, RAM, I/O, ...), which is estimated by the IT load model based on DC load and applied distribution algorithms. The resulting IT power P_{it} will then be aggregated and used as input for the uninterruptible power supply (UPS) model delivering the according power dissipation P_{ups}. The combined power usage is then again input parameter for the cold chain models, as the power demand equals dissipated heat that has to be removed from the server rooms. The chiller and free cooling models also depend on outside temperature T besides the heat flow $\dot{Q}$, which results from the heat absorbed by the computer room air handler (CRAH). The aggregated power usage of IT equipment, power supply, and climate infrastructure then represents the total DC power demand.

The combined DC performance is estimated in a similar way – as depicted in Figure 2. The resulting DC performance $perf_{DC}$ is composed of the 'used performance' (maximum performance scaled with usage) of every IT component. The particular usage is again estimated by the same IT load model and allocated to the server and storage systems. Only the network equipment will not be scaled with the load but will be incorporated as network performance factor in the overall DC performance. The concrete performance modeling process is explained in detail in Section V.

In contrast to common EE metrics, cf. II, the LDEE uses a combination of IT utilization and performance as substitute for useful work or productivity, which would be the natural pa-

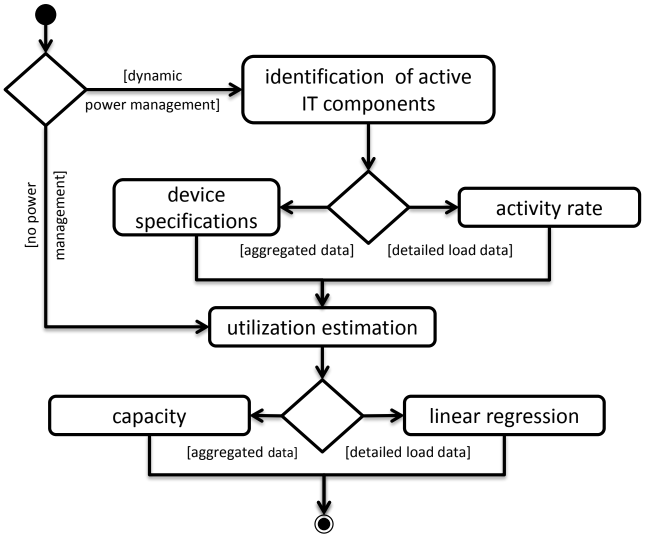


Fig. 3. Activity diagram describing the sequencing of IT load modeling.

rameter based on typical efficiency definitions [12]. Although the resulting 'used performance' is only an approximate value for productivity, as all tasks is considered and not only the useful ones, this approach has the advantage of being objective. Especially metrics with focus on useful workloads or productivity have the issue that tasks running in the DC and their priority have to be defined manually, such that these metrics lack the possibility for correct comparisons between DCs due to differences in task definitions.

Another advantage for using the LDEE approach is the possibility to be independent from actual measurements by using publicly available data to pre-configure the needed power and performance models of each DC component. This has the effect that the LDEE can be ascertained for arbitrary input data and correct comparisons between different DC sites and types can be achieved, as both DCs can be examined for the same workload levels.

IV. IT Load Model

Power and performance models of IT equipment need utilization data as input in order to deliver the load dependent outputs. As the LDEE metric shall be applicable without a need for measured data, an IT load model is used to estimate the workload distribution and concrete utilization levels among IT systems. The IT load model abstracts the behavior of power management (PM) algorithms in different submodels, which are selected based on the available training data. The summarized process of IT load modeling is depicted in Figure 3.

The IT load model is separated in two main parts: (1) identification of active IT components in the case PM is applied and (2) utilization estimation for each IT component. Each part is also sensitive to the input or training data available. If historical utilization traces of IT components have been measured, the IT load model makes use of them with more accurate methods. If there are no detailed activity and utilization data available, the IT load model makes assumptions based on the IT equipment configurations and PM techniques.

A. Utilization Conversion

The LDEE has the input vector $\vec{l}$ with utilization data for server l_1 and storage l_2, cf. Section III. However, the IT load model uses the aggregated, absolute load of all IT systems, so utilization has to be converted to these values beforehand.

The storage systems' utilization is defined as the ratio of absolute allocated storage to the maximum storage capacity. As only the actual available storage is considered, utilization values are comparable and transferable independent from applied redundancy mechanisms such as RAID. So, the absolute storage load can be computed by aggregating the total available storage capacity and multiplying it with the utilization l_2.

In contrast to storage systems, server utilization is not directly transferable between server types with differing performance characteristics. A server with higher performance will finish the same workloads in a shorter time than a server with low performance, i.e. the high-performing server has a lower average utilization for the same time window. Thus, server utilization values have to be normalized to a reference machine before they can be aggregated to an absolute value. Furthermore, in order to transfer utilization between servers, it has to be scaled corresponding to the performance relations of these servers.

The utilization conversion makes use of performance benchmarks, which evaluate the server performance by applying one or more concrete workloads on them. By measuring the actual workload dependent performance, different hardware architectures are considered in contrast to simply scaling with CPU frequency [13]. However, a disadvantage of using performance benchmarks is the requirement of actual measurements or the need for publicly available extensive data sets with benchmark results. In this work, the SPECpower_ssj2008[1] will be used to normalize and convert server utilization.

The utilization u_1 of server s_1 will be multiplied with the corresponding maximum measured throughput p_1^{max} of the SPECpower_ssj2008 benchmark (ssj_ops @ 100%) to get the throughput corresponding to the load: $p_1(u_1) = u_1 \cdot p_1^{max}$. By dividing the load dependent throughput with the maximum throughput p_2^{max} of another server s_2, the corresponding utilization u_2 on this server may be computed: $u_2 = \frac{p_1(u_1)}{p_2^{max}}$. In order to aggregate all server loads, these will be converted to the utilization of a defined reference machine s_r. Therefore, the same method can be used with $s_2 = s_r$. The resulting aggregated load then always has to be considered with regard to the reference machine and represents the number of necessary reference machines to handle the workload.

B. IT Activity Modeling

In the case a DC runs PM for EE reasons, some of the IT equipment will be shut down or transferred to an energy-saving

[1] https://www.spec.org/power_ssj2008/

state to increase the utilization of active systems. Thus, the first task of IT load modeling is to abstract the switching behavior of the used PM in order to estimate the active systems. As PM is currently mainly applied to server systems, if at all, the description of the activity model will concentrate on the application on servers. Also, the concrete modeling depends on the concrete PM running in an IT environment. While the general approach is transferable to other PM algorithms, the method will be shown for the PM developed in the AC4DC project [14]. This management algorithm is based on the work of Hoyer et al. [15] and implements proactive migrations of virtual machines (VMs) in fully virtualized environments using utilization forecasts and EE data of servers.

In a first step, the minimum needed capacities will be computed with Equation 2, as the most important requirement of PM is to guarantee the provision of sufficient computing capacities at all times.

$$l_{max_i} = max(l_j),\ \forall j \in \mathbb{N} :\ i \leq j \leq i + t_{pl} \tag{2}$$

$$t_{pl} = \frac{f_{sch} \cdot |V| \cdot t_m}{res} \tag{3}$$

For every time point i of the aggregated DC load l the maximum load l_{max_i} of a time window t_{pl} starting from point i will be ascertained. This time window represents the planning time of the PM algorithms, for which a certain capacity has to be reserved. It is affected by several operational parameters, which depend on the type of management algorithm and the software environment. For the algorithm developed in AC4DC project the planning time is derived from the following parameters: (1) a scheduling factor f_{sch}, which is the factor of migrated to total VMs, i.e. the distance to a pre-defined starting allocation, (2) the number $|V|$ of managed VMs, (3) the migration time t_m per VM in minutes, and (4) the time resolution of aggregated load l, i.e. the number of minutes per time point in l. So, in the AC4DC case the time window t_{pl} is the estimated time to return to the starting allocation. In Figure 4, the application on an aggregated CPU trace is shown as an example. The red curve is the minimum CPU capacitiy that has to be reserved.

On basis of the minimum required resources, the number of active servers can be ascertained. Thus, for each time point i the load l on system resource r (CPU, RAM, I/O, ...) will be allocated theoretically to the available servers s by using the primitive recursive function $f(l, n)$ in Equation 4. The result is the number of active servers so that the ordering of the sorted server set $(S, \leq)$ determines the active servers, as the ordering represents the usage priority.

$$f(l,n) = \begin{cases} n & \text{if } l \leq 0 \\ f(l - (c^r_{s_{n+1}} \cdot f_{tl}), n+1) & \text{if } l > 0 \end{cases} \tag{4}$$

The function will be called initially by $f(l^r_{max_i}, 0)$. With every function call the capacity c^r_s of resource r and server s with index $n+1$ of the ordered set $(S, \leq)$ will be subtracted from the load l, while the target load is also considered by applying the factor f_{tl} with $0 \leq t_l \leq 1$. Basically, with each function call a server will be added to the list of active servers until the resource capacities are sufficient while keeping the actual load below the defined target load. The required capacity is reached, when the remaining load $l \leq 0$. The number of server systems needed at a specific time point is the maximum of all considered resources: $max(f(l^r_{max_i}, 0))\, \forall r \in R$ with R as set of considered system resources.

The active servers are determined by the ordered set of server systems. As shown in Figure 3, there are two approaches with different precision to sort the servers: (1) a method using the historical activity rate and (2) a method using device specifications. The activity rate can be extracted from detailed server utilization data and represents the fraction of time a server was active. The assumption is now that servers with high past activity rates will also most probably be selected by the PM algorithms in future. The alternative approach, if no detailed utilization data is accessible, regards server system capacities for CPU and RAM as well as EE characteristics (e.g. SPECpower results). The two approaches will be compared in Section VI.

C. IT Utilization Modeling

After determining the potential active systems, the next step of the IT load model is to estimate utilization of each system. As depicted in Figure 3, there are again two different approaches depending on the available data: (1) linear regression on historical data or (2) distribution based on capacities.

The first approach is applicable, if detailed historical utilization data for each system are available. The relation of these data to aggregated DC load will be abstracted in a model, which then can be used to get system utilization for arbitrary aggregated load levels. In a first step, DC load is grouped into utilization classes. Then the probability distribution of time points with specific load corresponding to these classes is computed, as shown in Figure 5 (left) for an example of 20 classes with 5% utilization each. In a next step, the system utilization regarding the utilization classes is modeled using linear regression, such that there exist linear regression parameters $\binom{a}{b} = LR_{x,y}$ for each system s_x and utilization class cl_y. These are saved in a look-up table as demonstrated in Figure 5 (right). In order to get a system utilization abstraction l_x, the regression parameters $LR_{x,y}$ corresponding to system x and aggregated load l in class y will be inserted into $l_x = a + b \cdot l + e$.

In the case only aggregated data center load data is available, the distribution of workloads on the formerly estimated active systems is determined by their resource capacities. Thus, the load l_x of system x is computed by Equation 5 with aggregated DC load l and maximum resource capacity $l_{x_{max}}$.

$$l_x = \frac{l}{\sum_x l_{x_{max}}} \cdot l_{x_{max}} \tag{5}$$

The utilization model will be applied for every resource type that is considered in the performance and power models of IT equipment. Furthermore, the model has to be applied

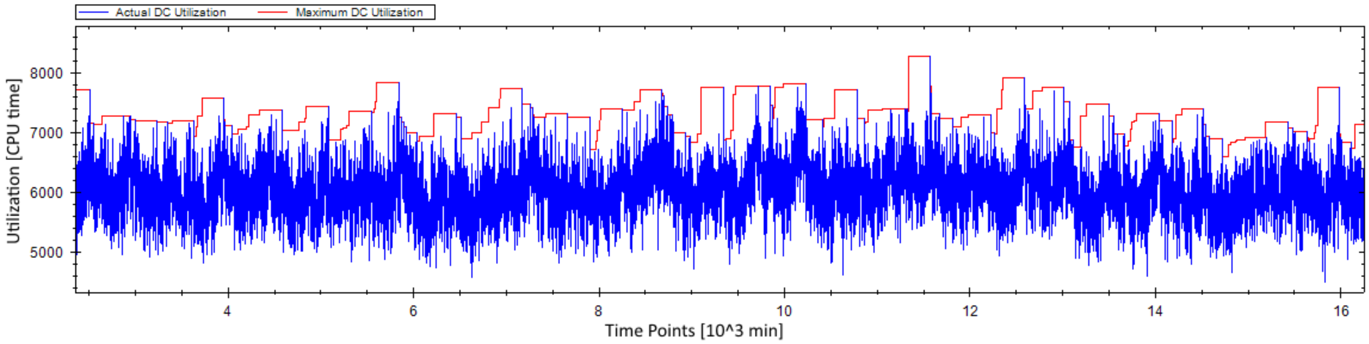
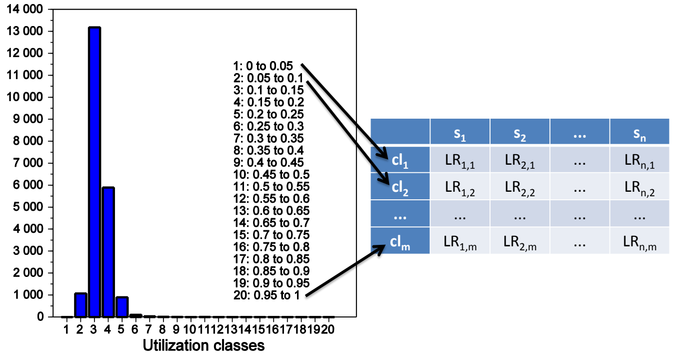


Fig. 4. Maxima l_{max_i} (red) of the aggregated resource demand (blue) represents the minimum resource capacity to provide.



	s_1	s_2	...	s_n
cl_1	$LR_{1,1}$	$LR_{2,1}$	...	$LR_{n,1}$
cl_2	$LR_{1,2}$	$LR_{2,2}$	...	$LR_{n,2}$
...	...	...	...	...
cl_m	$LR_{1,m}$	$LR_{2,m}$	...	$LR_{n,m}$

Fig. 5. Utilization abstraction for an IT system s by using a linear model LR for each utilization class cl.

for each time point of aggregated DC load time series, as the aggregated load changes over time as well as the number of active IT systems, if PM is used.

V. Data Center Performance Model

As the DC performance model is part of the LDEE, it has to fulfill the following requirements:

1) Comparability: Performance values for different DCs as well as points in time should be directly comparable.
2) Completeness: Results should represent a cross section of different typical DC workloads.
3) Applicability: The model should be applicable for productive DCs without disturbing operation.
4) Load Dependency: As the LDEE considers actual usage, the influence of DC load has to be respected by the performance model.
5) Comprehensibility: Performance results should be in a format, which is easy to understand with brief explanations.

A typical approach for performance estimation would be to analyze the hardware specifications and summarize the quantitatively identifiable key characteristics, e.g. CPU and RAM by number of cores, frequency, and memory size, respectively. However, due to different hardware architectures and technologies, these numbers are often not comparable between different workloads, which restricts the applicability of this approach for the LDEE.

Another approach is the direct measurement of productive systems in order to capture the amount of work done. However, each application has its own performance indicators such as operations per second or response times in different magnitudes. Summarizing these indicators requires a manual definition in relation to each other. Due to this subjective definition, the comparability between DCs is not given and thus, direct measurements are not suitable for the LDEE.

In order to realize comparability of performance values, IT systems have to be stressed with the same workload in an objective manner. Thus, performance benchmarks are predestined to fulfill this requirement. However, applying a single benchmark is not sufficient for achieving completeness, as benchmarks usually represent one single use case. So, several different benchmarks should be used to characterize the performance of an IT system to stress the diverse subsystems. A big disadvantage of directly applying performance benchmarks on IT systems is the high effort that is needed to isolate such systems from production and reintegrate them again afterwards. As another obstacle some systems may not even be allowed to be isolated.

An alternative approach which profits from the advantages of performance benchmarks without the need for real measurements is the indirect usage of benchmarks by analyzing already available performance results. This way the applicability is ensured, however, the number of potential benchmarks is limited as there have to be enough performance results publicly available to cover the majority of IT systems. Thus, the completeness is worse than with direct application of benchmarks. However, as the applicability is the most important requirement for DC metrics, this compromise will be accepted.

By using performance benchmarks, the first three requirements can be covered. Regarding load dependency, benchmarks usually provide the maximum performance of an SUT. Thus, the resulting maximum performance will be scaled with the actual or estimated utilization. This ’used performance’ then stands for the totally processed work in the DC and is an approximation to useful work, which is easier to estimate. And the fifth requirement – comprehensibility – will be achieved by normalizing the single performance values by defined reference systems.

A. Modeling Approach

The basic approach for performance modeling starts with the one-time determination of used performance benchmarks for server and storage systems. On basis of system performance data, a reference system will be defined which then will be used to normalize the single performance values to comparable performance factors. After that, the performance factors corresponding to each IT system are combined to a single performance value. Subsequently, the single steps will be explained more in detail.

1) Benchmark Selection: The selection of suitable performance benchmarks bases on the requirements for performance modeling mentioned above. Especially the completeness and the applicability affect the benchmark choices. However, both requirements represent a trade-off, for which the best proportion has to be found. On the one hand, the completeness requires the usage of diverse benchmarks to consider several kinds of workloads which stress the subsystems from different angles. On the other hand, the applicability demands for publicly available performance results for a high variety of IT systems.

Server Performance Benchmarks: In case of server systems there are currently three suitable choices for performance benchmarks: (1) SPEC CPU2006[2], (2) SPECpower_ssj2008, and (3) SAP Sales and Distribution (SD) Standard Application Benchmark[3]. To the best of our knowledge, no other performance benchmark can comply with the requirements. Many benchmarks are designed to assess an SUT consisting of several servers, storage systems, and connecting network equipment, which is not ideal for performance modeling of single components. Examples are SPECvirt_sc2010, VMmark 2.X, or TPC-C/E/H. In some other cases the problem is the low count of benchmark results, i.e. only a small number of server systems could be modeled.

The three selected server benchmarks have slightly different focal points. SPEC CPU2006 is a CPU-intensive benchmark which stresses the processor and memory subsystems by applying workloads based on real user applications in different application areas like artificial intelligence, optimization and search problems, compression, fluid/molecular/... dynamics, etc. – divided into integer and floating point [16]. For performance modeling, the four base metrics of elapsed execution time and throughput each for integer and floating point will be used. SPECpower is a performance/power benchmark, which stresses the SUT on eleven load levels from active idle to full load in 10% steps with a Server Side Java (ssj) workload exercising the CPU, caches, memory hierarchy, and the scalability of shared memory processors [17]. Results are given in throughput per power (ssj_ops/W) for each load level as well as maximum performance which will be used by the proposed model. SAP SD Standard Application Benchmark covers a sell-from-stock scenario simulating user orders including processing and delivery [18]. The number of simulated users is increased until the response time of the SUT arrives at one second (two seconds before 2009). The workload is mainly CPU-driven but also stresses the memory subsystem. The achieved throughput (order line items per hour) at maximum load is then converted into SAPS (SAP Application Performance Standard), which will be used for performance modeling.

Storage Performance Benchmarks: In case of storage performance modeling there are currently two suitable public benchmarks: (1) SPECsfs2008[4] and (2) SPC Benchmark 1[5]. Other benchmarks either have no data base with publicly available results like FileBench or have too few entries such as SPC Benchmark 2. Apart from these problems, the two exemplary benchmarks would be suitable for storage performance modeling and may be integrated in the future.

SPECsfs2008 is a benchmark for network file systems measuring a file server's throughput and response time. There are two different workloads: A NFS workload and one for CIFS which both are based primarily on data collected from tens of thousands of file servers from SPEC member companies [19]. Load generating clients stress the file server on at least ten different load levels, reporting throughput (in operations per second) and average response time per operation values. For the storage performance model results for the NFS workload will be used, as the CIFS workload has too few results so far. SPC-1 has a single workload performing typical functions of business critical applications on storage systems consisting predominately of random I/O, query, and update operations like OLTP, mail server, or database applications [20]. SPC-1 evaluates the maximum I/O request throughput in SPC-1 IOPS for systems with a high amount of parallel tasks and the least I/O request response time in SPC-1 LRT for business critical applications. For both benchmarks the maximum throughput as well as the response times will be used for storage performance modeling.

2) Normalization Process: As the performance model is supposed to estimate an aggregated performance value over all systems, the single performance values from the selected benchmarks have to be comparable and combinable for the respective type of IT system. However, the benchmarks evaluate performance with different indicators in different ranges, which have to be scaled in relation to each other. E.g. SPEC CPU measures execution times in relation to a predefined reference machine with factors from 1 to 2000+ while SPECpower returns maximum performance by the number of ssj_ops with up to 10+ millions.

In order to achieve the required comparability between different benchmarks, the single results have to be normalized regarding a reference machine, i.e. dividing all results by the result of the reference machine. Selecting a suitable reference

[2] http://www.spec.org/cpu2006/

[3] http://www.sap.com/campaigns/benchmark/appbm_sd.epx

[4] http://www.spec.org/sfs2008/

[5] http://www.storageperformance.org/results/#spc1_overview

machine, however, is not trivial. Choosing a real existing IT system with results for each selected benchmark may be the common approach, but if this system is not balanced throughout all benchmarks, this imbalance will transfer to the normalized results with the consequence that some benchmarks will be emphasized, which is not desired. E.g., if reference machine rm belongs to the slowest 20% of SUTs in benchmark B and is in the middle of the field (50%) for every other benchmark, normalized values for B would be much higher compared to the other benchmarks. Thus, benchmark B would have a higher weight which restrains comparability.

Instead of using an existing IT system as reference machine, the normalization process will make use of a virtual reference machine representing the average IT system. This machine will be computed from the set R of all IT systems s, which have a result PB_s for each performance benchmark PB of the selected set A, as shown in Equation 6. With set R the concrete reference performance value PB_R^i for every benchmark i can be computed by averaging the performance results PB_s^i of every IT system $s \in R$, cf. Equation 7. This reference performance PB_R^i can then be used in Equation 8 to normalize all performance results PB_s^i of benchmark i to a performance factor PF_s^i, which represents the relation of system s to the virtual reference machine regarding performance.

$$R = \{s | \exists PB_s, \forall PB \in A\} \tag{6}$$

$$PB_R^i = \frac{\sum\limits_{s \in R} PB_s^i}{|R|} \tag{7}$$

$$PF_s^i = \frac{PB_s^i}{PB_R^i} \tag{8}$$

The set R of IT systems representing the reference system is determined exactly one time and will be constant for each version of LDEE. Otherwise, comparability of performance and consequently of EE results cannot be guaranteed.

3) Combination of Performance Factors: After normalization of performance results, every IT system s has a performance factor PF_s^i for each benchmark i, which then have to be combined to a single result pf_s. Therefore, the weighted geometrical mean of all performance factors of system s with weights w_i is computed, cf. Equation 9.

$$pf_s = \sqrt[w]{\prod_i^{|A|} (PF_s^i)^{w_i}} \quad with \quad w = \sum_i^{|A|} w_i \tag{9}$$

For a general performance evaluation all benchmarks have the same share, i.e. $\forall i \, w_i = 1$. However, to focus on specific workloads or DC types, the benchmarks may be weighted differently to emphasize the ones representing respective workloads.

In case of general server performance, the benchmarks SPECpower and SAP SD have a weighting of 1, while SPEC CPU subdivides in four individual results and thus, each single performance value is weighted with 0.25 in order to achieve a balance between all three benchmarks. The two selected storage performance benchmarks consist both of two performance values, such that each single value is weighted with 0.5. However, before combining the storage performance factors, the factors for the response times have to be inverted to their reciprocals: $1/PF_s^i$. While all other performance indicators represent higher performance with higher values, the response times behave exactly converse, i.e. a smaller response time indicates higher performance.

B. Network Performance Model

Performance analysis of computer networks is much more complex than evaluating server or storage systems, as not only one or few hardware and software systems are involved but all network nodes as well as every application that uses network functions. I.e., benchmarking all network components like switches and routers separately is not sufficient due to the influence of the network architecture and the shape of traffic on the overall DC network performance. Also, as the network architecture is specific for each DC, performance is hardly comparable and therefore public data bases with performance results are not available.

For the usage in the LDEE metric, the network performance will not be considered in absolute, comparable numbers, but as a percentage of sufficient network capacities. Therefore, network parameters regarding quality of service (QoS) have to be continuously measured (e.g. via DC infrastructure management) in order to check the adherence to the network's service level agreements (SLA). By using a DC's specific SLA, the assumption is met that adhering to SLA implies the provisioning of sufficient network capacities. Thus, the network performance coefficient (NPC) is introduced, representing the times QoS is fulfilled. In Equation 10, an example for some typical QoS parameters is shown.

$$\begin{aligned} NPC = \frac{1}{n} \sum_{i=0}^{n} min \left(min \left(1, \left\lfloor \frac{thr_{qos}}{thr_i} \right\rfloor \right), \right. \\ \left. min \left(1, \left\lfloor \frac{lat_{qos}}{lat_i} \right\rfloor \right), min \left(1, \left\lfloor \frac{pl_{qos}}{pl_i} \right\rfloor \right) \right) \end{aligned} \tag{10}$$

The parameters data throughput thr (MBit/s), latency lat (ms), and relative packet loss pl (%) will be compared with their QoS requirements for each time point i of the considered historical time frame. If a single QoS target cannot be met at one time point, i.e. parameter $pm_i > pm_{qos}$, the whole term will be 0 representing network shortcomings at time point i. Otherwise, if targets are met, the whole term is 1 which stands for a smooth operation. The NPC then is the percentage of time points without shortcomings in network operation, i.e. if NPC is 1, network capacities were always sufficient regarding historical application traffic.

The main advantage in using SLA and their corresponding QoS parameters for modeling network performance is the abstraction from the actual network architecture. SLA are specifically adapted to a certain DC with its individual network topologies and architectures as well as to the installed software and services. Otherwise, the network has to be modeled specifically for each DC. By using SLA in the NPC, network shortcomings indirectly affect the DC performance. The other case of inefficiently used (wasted) network performance cannot be considered by this approach. However, the objective of LDEE is to survey EE and therefore performance regarding load or usage, i.e. wasted performance is not relevant for LDEE. Then again, inefficient networks will be considered on the energy side of LDEE as higher network performance usually means higher energy demands.

C. Combined DC Performance Rating

In order to increase the comprehensibility of LDEE, which is required for a broad adoption in DC monitoring and controlling, both server and storage performance will be converged to a single performance rating representing the number of used converged reference units (abbreviated as ru).

The server (i) and storage (j) performance models provide maximum performance pf_i and pf_j for single IT systems, which have to be adapted with the actual usage l_i and l_j. Used performance for server and storage systems will then be aggregated separately and subsequently converged to a single value, whereby different weights w_{srv} and w_{sto} are used. This combined result will then be scaled with the NPC to the final used DC performance pf_{DC}, cf. Equation 11.

$$pf_{DC} = NPC\left(w_{srv}\sum_{i\in srv} l_i pf_i + w_{sto}\sum_{j\in sto} l_j pf_j\right) \qquad (11)$$

The standard case for weighting between server and storage is simply adding the single results, i.e. $w_{srv} = w_{sto} = 1$. Differing weights are used, when specific data center types are relevant and the data center will be compared to others of the same kind. Although weights may be adapted individually (Free Choice), it is not recommended doing so as the comparability is only ensured for identical weights. Thus, there are some recommendations for DC classes:

- Standard (S): $w^S_{srv} = 1$; $w^S_{sto} = 1$
- Compute-Intensive (C): $w^C_{srv} = 0.8$; $w^C_{sto} = 0.2$
- Data-Intensive (D): $w^D_{srv} = 0.2$; $w^D_{sto} = 0.8$
- Free Choice (F): $w^F_{srv} = x$; $w^F_{sto} = 1 - x$ $\{x \in \mathbb{R} | 0 \leq x \leq 1\}$

The stated ratios between server and storage systems are currently not based on hardware numbers, as such data is not available. However, the strong difference of 0.8 to 0.2 in compute- and data-intensive DCs is recommended to emphasize clearly the respective IT systems in contrast to a balanced weighting.

VI. Evaluation

This section describes evaluations for the proposed IT load model (Section IV) and DC performance model (Section V).

A. Evaluating the IT Load Model

For evaluating the IT load model, data based on the DC power management simulation from the AC4DC project [14] were used. The simulation has been executed for several given numbers of VMs with ten different arbitrarily chosen DC configurations for each number of VMs. The resulting time series with server activity data and utilizations as well as aggregated DC CPU and RAM utilization have been separated into training and test data, with training data used for model characterization and test data to analyze the model error.

In the evaluation of the IT activity model the two presented approaches – activity rate AR and server specifications Sp – were compared to each other. These two approaches were also evaluated in a slightly adapted version ($w/o\ max$), where no maximum utilization for time windows was computed prior to estimating the active servers as it would be the case for reactive PM without forecasts. Furthermore, Av estimated server activity based on historic average utilization and Rdm conducted a random selection of servers. While AR and Av used detailed utilization information, Sp and Rdm based on aggregated DC load only. The relative mean square errors are shown in Figure 6.

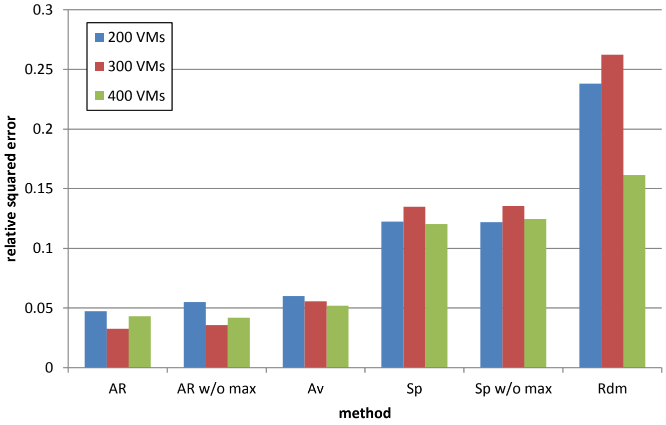


Fig. 6. Mean errors of estimated number of active servers to actual one.

As expected, with detailed utilization data per server (AR, $AR\ w/o\ max$, Av) considerably better estimations can be achieved than by methods using aggregated DC load (Sp, $Sp\ w/o\ max$, Rdm) alone. Looking at the hit rate for correct server selection, the difference is even bigger: While chosen severs by AR matched 90% of the actual servers, less than 50% servers matched by using Sp. Preprocessing input data by analyzing maxima is not necessarily needed, as results were marginally better (0.3% for AR and 0.1% for Sp), i.e. the IT load model is more flexibly applicable for other (reactive) PM algorithms.

In order to evaluate the utilization estimation, the proposed approaches based on utilization classes ($Cl + LR$) and capacity information (Cap) were applied to productive utilization data which had been measured in the context of the AC4DC project [14]. Furthermore, other methods for abstracting detailed utilization data were analyzed with harmonic mean (H), arithmetic mean (A), and mode for grouped data ($Mode$) in addition to linear regression (LR). With exception of $Mode$, these methods were also analyzed without prior utilization classification. Besides, for each result an additional interpolation to match the corresponding aggregated DC load was performed and compared to the corresponding unchanged result. For each method, the difference to actual utilization (mean square error) was computed, which is depicted in Figure 7.

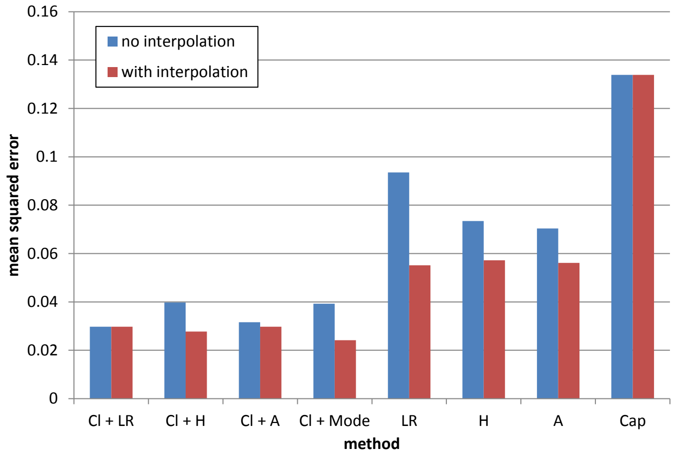


Fig. 7. Mean errors of estimated compared to actual server utilization.

The advantage of classification (Cl) prior to abstracting utilization data is clearly visible. However, the difference between the diverse abstraction methods is marginal. As linear regression does not need a further interpolation to improve the results, it is the chosen method in the IT load model. As expected, the method based on maximum resource capacity (Cap) is the least accurate, as it only makes use of the aggregated DC load and hardware specifications.

B. Evaluating the Performance Model

As the proposed server and storage performance models base on the most common performance benchmarks, an evaluation by comparing the models with these benchmarks has no significance. However, even a comparison with benchmarks, which were not used, or with other means of performance estimation is not meaningful, because performance is not distinct but it depends strongly on the considered characteristics. I.e., performance values may differ exceedingly without meaning that one of these values is wrong.

Thus, only the combined DC performance model was evaluated by analyzing its behavior in comparison to known metrics Data Center Energy Productivity (DCeP) by The Green Grid [3] and Digital Service Efficiency (DSE) by Ebay [21]. For the evaluation, only the numerator of DCeP was used: Useful Work Produced $= \sum_{i=1}^{M} V_i \cdot U_i(t,T) \cdot T_i$, which is the sum of accomplished tasks T_i. Each task is defined by a runtime function $U_i(t,T)$ and a normalization factor V_i, which have to be defined manually. The DSE is similar to DCeP with the restriction that only Ebay key indicators number of generated Buy and Sell URL will be counted without further normalization or consideration of execution times.

By using the AC4DC PM [14], an evaluation scenario consisting of 21 rack servers with four different server models and one storage model has been defined and simulated for 100, 200, and 300 VMs, which had been measured in a productive environment. Moreover, for each number of VMs two different kinds of volatility have been induced: (1) low volatility is created by combining VMs with shifted profiles which form an average utilization with narrow variation and (2) high volatility originates from combining VMs with similar profiles which result in high variations within the time series. For each of the six simulations the 'used DC performance' (as part of the LDEE) was computed by applying the proposed models on the specific IT components with the corresponding load. Also, load and performance of IT systems were converted to a number of accomplished predefined tasks, considered by DCeP and DSE. The results are given in Table I.

TABLE I
PERFORMANCE AND PRODUCTIVITY FOR DIFFERENT WORKLOADS

VM profile	high volatility		
number VMs	**100**	**200**	**300**
used DC performance [ru]	0.419	0.837	1.252
DCeP [M tasks]	13.651	26.222	40.003
DSE [M tasks]	4.838	9.677	14.515
server utilization	0.020	0.040	0.061
storage utilization	0.321	0.642	0.958
VM profile	low volatility		
number VMs	**100**	**200**	**300**
used DC performance [ru]	0.727	1.785	2.483
DCeP [M tasks]	23.198	57.888	80.482
DSE [M tasks]	7.258	19.354	29.030
server utilization	0.040	0.103	0.141
storage utilization	0.321	0.642	0.958

The results show that all three considered metrics are dependent on the load, as with higher loads the used performance as well as productivity increases, and can thus fulfill the fourth requirement in Section V. Regarding comparability, only 'used DC performance' is consistent between different DC configurations and SW environments, which can be seen in Table II. There, a second testing scenario TS_2 with a different SW stack was assumed for an identical DC configuration. While the DC performance model provides the same results by abstracting from the workloads, the numbers of DCeP are quite differing due to different subjective task definitions. In order to achieve comparability for DCeP, task definitions have to be normalized over all tasks in all DCs, which is virtually unaccomplishable.

Regarding the other requirements (cf. Section V), there are also some advantages of the proposed performance model. Applicability has already been discussed and is one of the key benefits compared to common metrics. While comprehensibility is subjective, the performance model has the advantage of delivering a single number in a single format – the number of used reference machines. On the contrary, e.g. DCeP provides an abstract value computed from tasks of undefined types normalized with subjective priorities. Solely regarding com-

TABLE II
Performance and productivity for two testing scenarios TS_i

testing scenario		TS_1	
number VMs	**100**	**200**	**300**
used DC performance [ru]	0.419	0.837	1.252
DCeP [M tasks]	13.651	26.222	40.003
testing scenario		TS_2	
number VMs	**100**	**200**	**300**
used DC performance [ru]	0.419	0.837	1.252
DCeP [M tasks]	4.234	8.467	12.701

pleteness, the DCeP has the advantage of considering every relevant workload, while the performance model makes use of a limited number of predefined workloads from benchmarks.

VII. Conclusion and Future Work

The proposed model of used DC performance is an integral part of the LDEE metric, which realizes objective EE analysis with full comparability between DCs while being noninvasively applicable without disrupting productive operation. By using common performance benchmarks with publicly available results, performance models can be trained without real measurements and applied to arbitrary DC loads. The same is true for the IT power models needed for EE analysis, which base on public energy benchmarks. Thus, LDEE handles shortcomings of previous metrics such as incompleteness of PUE or subjectivity and complex applicability of DCeP.

The next step in pushing usage of LDEE in the DC segment is the implementation of an easy to use work flow. Basis would be common DC management tools such as data center infrastructure management (DCIM) or monitoring systems. These tools may be used to inventory the IT and infrastructure components and also readout the concrete system specifications and potential operational data. LDEE models then can be trained either by using available measurement data and/or by using energy and performance data from public data bases. EE results may then be added to the administration dashboards.

Moreover, the LDEE is a significant component of dynamic energy aware workload management. In the EU project M2DC (Modular Microserver Data Center) [22], the LDEE will be used to determine the energy efficiency of IT systems in order to dynamically deploy applications in the most energy efficient way. In addition to this, the extension of LDEE to alternative compute nodes like ARM processors, GPUs or FPGAs will be analyzed in the context of M2DC.

Acknowledgment

This scientific work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688201 (M2DC).

References

[1] D. Azevedo, V. Avelar, and A. French, "Pue: a comprehensive examination of the metric," White Paper #49 – The Green Grid, 2012, http://www.thegreengrid.org/~/media/WhitePapers/WP49-PUE%20A%20Comprehensive%20Examination%20of%20the%20Metric_v6.pdf?lang=en.

[2] D. Schlitt and W. Nebel, "Load dependent data center energy efficiency metric based on component models," in *Energy Aware Computing, 2012 International Conference on*. IEEE, 2012, pp. 1–6.

[3] J. Haas, M. Monroe, J. Pflueger *et al.*, "Proxy proposals for measuring data center productivity," *White Paper #17 – The Green Grid*, 2009, http://www.thegreengrid.org/~/media/WhitePapers/White%20Paper%2017%20-%20Proxies%20Proposals%20for%20Measuring%20Data%20Center%20Efficiencyv2.pdf.

[4] J. Pflueger, "Re-defining the 'green' data center," White Paper – DELL, 2008, http://www.gobiztech.com/docs/Dell-Redefining-Green-Datacenter.pdf.

[5] Green IT Promotion Council, "New data center energy efficiency evaluation index dppe (datacenter performance per energy) measurement guidelines," *Richtlinie – Green IT Promotion Council (GIPC)*, 2012, http://home.jeita.or.jp/greenit-pc/topics/release/pdf/dppe_e_DPPE_Measurement_Guidelines.pdf.

[6] K. Brill, "Data center energy efficiency and productivity," *White Paper – The Uptime Institute*, 2007, https://connect.ufl.edu/cns/DCO/ecdc/ECDC%20Construction%20Project/Whitepapers/Data%20Center%20Energy%20Efficiency%20%28Uptime%20Inst%29.pdf.

[7] B. Schödwell, K. Erek, and R. Zarnekow, "Data center green performance measurement: State of the art and open research challenges," 2013.

[8] 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*. ACM, 2007, pp. 365–376.

[9] K. Du Bois, T. Schaeps, S. Polfliet, F. Ryckbosch, and L. Eeckhout, "Sweep: evaluating computer system energy efficiency using synthetic workloads," in *Proceedings of the 6th International Conference on High Performance and Embedded Architectures and Compilers*. ACM, 2011, pp. 159–166.

[10] D. A. Patterson, "The data center is the computer," *Communications of the ACM*, vol. 51, no. 1, pp. 105–105, 2008.

[11] D. Bruneo, "A stochastic model to investigate data center performance and qos in iaas cloud computing systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 3, pp. 560–569, 2014.

[12] S. Sorrell, "Jevons' Paradox revisited: The evidence for backfire from improved energy efficiency," *Energy policy*, vol. 37, no. 4, pp. 1456–1469, 2009.

[13] C. Hankendi, S. Reda, and A. K. Coskun, "vcap: Adaptive power capping for virtualized servers," in *Proceedings of the International Symposium on Low Power Electronics and Design*. IEEE Press, 2013, pp. 415–420.

[14] D. Schlitt, B. Hanstein, S. Oberthür, R. Hintemann *et al.*, "Abschlussbericht des Verbundprojekts AC4DC – Adaptive Computing for green Data Centers," 2014, https://www.tib.eu/de/suchen/id/TIBKAT%3A833375288/Adaptive-Computing-for-green-Data-Centers-AC4DC/.

[15] M. Hoyer, K. Schröder, D. Schlitt, and W. Nebel, "Proactive dynamic resource management in virtualized data centers," in *Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking*. ACM, 2011, pp. 11–20.

[16] Standard Performance Evaluation Corporation (SPEC), "SPEC CPU2006 Run and Reporting Rules," Specifications, 2011, http://www.spec.org/cpu2006/Docs/runrules.html.

[17] ——, "SPECpower_ssj2008 Design Document SSJ Workload," Specifications, 2012, http://www.spec.org/power/docs/SPECpower_ssj2008-Design_ssj.pdf (letzter Zugriff: 21.01.2016).

[18] U. Marquard and C. Götz, "Sap standard application benchmarks – it benchmarks with a business focus," in *Performance Evaluation: Metrics, Models and Benchmarks*. Springer, 2008, pp. 4–8.

[19] Standard Performance Evaluation Corporation (SPEC), "SPECsfs2008 User's Guide," Specifications, 2008, http://www.spec.org/sfs2008/docs/usersguide.pdf (letzter Zugriff: 21.01.2016).

[20] Storage Performance Council (SPC), "SPC Benchmark 1 (SPC-1) Official Specification," Specifications, 2013, http://www.storageperformance.org/specs/SPC-1_SPC-1E_v1.14.pdf (letzter Zugriff: 21.01.2016).

[21] Ebay Inc., "Digital Service Efficiency," White Paper, 2013, http://tech.ebay.com/sites/default/files/publications/eBay-DSE-130523.pdf.

[22] A. Oleksiak, G. Agosta, M. vor dem Berge, M. Cecowski *et al.*, "The M2DC Project: Modular Microserver DataCentre," in *19th Euromicro Conference on Digital Systems Design (DSD)*. IEEE, 2016.