2009

Taming Energy Costs of Large Enterprise Systems Through Adaptive Provisioning

Markus Hedwig
Albert-Ludwig-University of Freiburg, markus.hedwig@is.uni-freiburg.de

Simon Malkowski
Georgia Institute of Technology - Main Campus, simon.malkowski@cc.gatech.edu

Dirk Neumann
Albert-Ludwigs-University of Freiburg, dirk.neumann@is.uni-freiburg.de

TAMING ENERGY COSTS OF LARGE ENTERPRISE SYSTEMS THROUGH ADAPTIVE PROVISIONING

Completed Research Paper

Markus Hedwig
Chair of Information Systems Research
Albert-Ludwigs-University of Freiburg
Platz der Alten Synagoge
79098 Freiburg, Germany
markus.hedwig@is.uni-freiburg.de

Simon Malkowski
CERCS
Georgia Institute of Technology
266 Ferst Drive
30332-0765 Atlanta, USA
simon.malkowski@cc.gatech.edu

Dirk Neumann
Chair of Information Systems Research
Albert-Ludwigs-University of Freiburg
Platz der Alten Synagoge
79098 Freiburg, Germany
dirk.neumann@is.uni-freiburg.de

Abstract

One of the most pressing concerns in modern datacenter management is the rising cost of operation. Therefore, reducing variable expense, such as energy cost, has become a number one priority. However, reducing energy cost in large distributed enterprise system is an open research topic. These systems are commonly subjected to highly volatile workload processes and characterized by complex performance dependencies. This paper explicitly addresses this challenge and presents a novel approach to Taming Energy Costs of Larger Enterprise Systems (Tecless). Our adaptive provisioning methodology combines a low-level technical perspective on distributed systems with a high-level treatment of workload processes. More concretely, Tecless fuses an empirical bottleneck detection model with a statistical workload prediction model. Our methodology forecasts the system load online, which enables on-demand infrastructure adaption while continuously guaranteeing quality of service. In our analysis we show that the prediction of future workload allows adaptive provisioning with a power saving potential of up 25 percent of the total energy cost.

Keywords: Green IT, Bottleneck Detection, Workload Analysis, Adaptive Provisioning

Introduction

One of the most pressing concerns in modern datacenter management is the ever-growing cost of operation. Especially, the expenses for electrical energy have become a significant cost factor. Steadily rising prices for electricity, increasing power density of IT systems, as well as the growth of the installed base of computing infrastructure, transformed energy cost reduction into a decisive decision criterion in modern datacenter design. Accompanied by increasing demand of electrical energy, the carbon footprint of the IT industry has been increasing rapidly, which resulted in high public attention. Nowadays, the IT industry emits about two percent of the total CO_2 emissions, which approximately equals the output of the global air traffic (Singh et al., 2007).

In the past, IT was considered a clean industry, but recently, a new type of “green” awareness has grown. The umbrella term “Green IT” denotes all activities and efforts incorporating ecologically friendly technologies and processes into the entire lifecycle of information and communication technology. The sustainable operation of datacenters plays a central role in this domain and focuses on the reduction of energy consumption during operation of datacenters. At a first glance, this objective seems economically motivated. However, considering the direct relation between energy consumption and green house gas emissions, economic and ecologic goals are coherent in this case.

There exist several impressive examples that show the extent of the environmental impact of the IT industry. For instance, datacenters worldwide have consumed the energy output of nineteen mid-sized power plants in 2007. The market-value of the energy was approximately $ 7.2B (Koomey, 2007). A more tangible example on the level of single website requests is the fact that a single Google search consumes 8Wh of energy. With 40M search requests per day, the daily energy consumption of Google searches is around 300MWh. This is roughly equivalent to circling around the earth sixty times in a car or 350t of CO_2 based on data of 2005 (Kersten, 2007). Given these numbers, it is not surprising that the IT industry and research departments have devoted high efforts to slow down the increasing energy hunger of computing.

However, the heterogeneity and complexity of IT systems dictates that the development of new system designs and new computing paradigms is a challenging task. In the meantime, several concepts to cut down energy consumption already reached market maturity. A prominent example is virtualization technology that is dominant means of consolidation. Nevertheless, today’s datacenter layouts and operation concepts still have large potential for improvement. Some efforts deal with the development of new technologies to increase energy efficiency while others aim to remove shortcomings resulting from obsolete and inefficient designs. The complexity of modern datacenters makes the development of holistic improvement concepts non-trivial. Accordingly, the first green initiatives concentrated on easily accessible *“low hanging fruit”* with low complexity, and not on sustainable impact on energy efficiency.

This paper presents a novel model to increase the efficiency of enterprise systems. Tecless continuously supervises the workload process of an enterprise system and systematically analyses the performance behavior of the system. Based on these observations, Tecless dynamically adapts the infrastructure size of the enterprise system to the demand in order to reduce the energy consumption. Due to the inherent domain complexity, Tecless focuses a single common use case in IT operation. More concretely, Tecless is designed for systems providing services to private and commercial end-users. A detailed analysis of workload processes for such systems revealed that these systems often face highly volatile workload processes. Our empirical investigation has shown that such workload processes, especially for systems with a large user community, are nearly stationary. This allows the forecast of the process for the near future.

Because enterprise systems need to be scaled to handle the maximum expected workload level, the systems are only weakly utilized in off peak times. Consequently, this leads to a low average utilization. According to a recent study, the average utilization of datacenters is only around 20 percent (ITP, 2007). Especially during night times these systems are typically weakly utilized. Motivated by this observation, our paper introduces and evaluates a model that allows cutting down the energy consumption of IT systems by up to 25 percent using state-of-the-art technology. The energy savings are mainly gained by tailoring the IT infrastructure of the system to the actual demand at all times. Unnecessary servers are identified and removed from the system by switching them off. Modern servers require about 50 percent of their peak power consumption in idle mode (SPEC, 2008). This naturally leads to a very high electrical base load. Consequently, only the complete deactivation of these servers allows the conservation of this share.

Hitherto, large IT systems have been developed to maintain large and growing user community with a constant service quality. The performance demand of these systems is often far beyond the capabilities of single machines. Hence, scalability considerations are becoming the main concern. Distributed systems are the state-of-the-art design paradigm, where the application logic is split into several layers being deployed on separate servers. By replicating certain layers of the system, the performance can be scaled. Modern applications, based on state-of-the-art middleware, usually incorporate the capability for dynamic reconfiguration (Alonso et al., 2002). However, the performance characterization of distributed systems remains a highly challenging problem. Determining the maximal possible workload is commonly based the on experience of domain experts (e.g., IT administrators). Non-trivial interactions between the different layers of the systems as well as hidden dependencies make inferences on the performance capabilities an extremely hard problem. Especially, since the hardware configuration is dynamically adapted to the demand, performance analysis becomes incomprehensible and potentially untraceable.

Though the online adaption of the hardware configuration seems feasible, dynamic modifications of the infrastructure are very time consuming. Servers require a certain time to start and the applications might require prior synchronization. Hence, reconfiguration decisions need to be made in advance. Our workload trace analysis revealed that there is a high volatility in workload processes. Elevations of the workload level above 50 percent in less than one hour are common. Apart from the complexity of performance characterization of large systems, this volatility constitutes the second major challenge. An efficient dynamic provisioning model needs to account for the delay between the activation of a server and its availability. Hence, datacenter management must accurately predict the workload process in the near future based on current and post observations. Deviations between the forecast and actual the outcome might lead to performance failures and to the violation of Service Level Agreements (SLA).

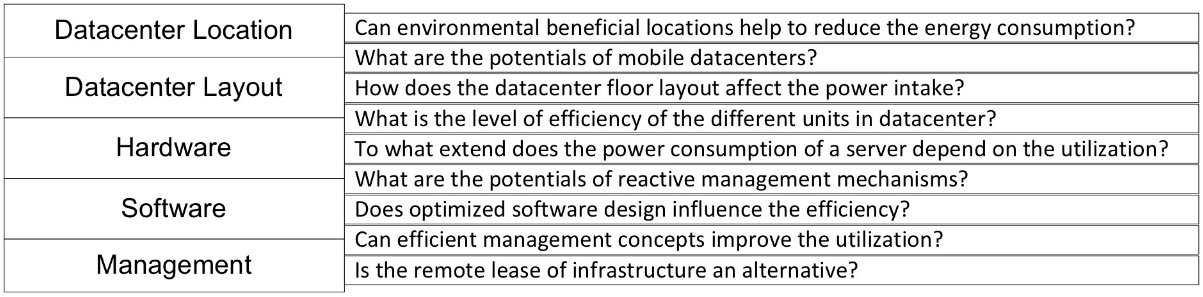
This paper is unique as it advances current norms in provisioning by introducing Tecless, a provisioning model that is designed to accommodate for the complexity of the performance behavior of large systems as well as the volatility of workloads. Guaranteeing the Quality of Service (QoS) is the main goal of the model. Thus, this paper has three main contributions:

- First, the paper designs an iterative model to accurately describe the performance behavior of enterprise systems depending on the workload based on a bottleneck detection model (Malkowski et al., 2007). This advanced performance model bridges the gap between state-of-the-art systems research and green design. The gist of the underlying observation-based method is that it can be applied to any enterprise system as it does not directly rely on software design. This method requires less domain specific knowledge and simplifies the process of provisioning optimal infrastructures.
- Second, the paper introduces a scheme designed to forecast the workload process based on past observations. Compared to related approaches, the proposed scheme identifies and models different factors of influence on the workload process. This results in considerably higher prediction accuracy, which facilitates reliable dynamic reconfiguration decisions in enterprise systems.
- Third, the paper derives a provisioning algorithm, which optimizes hardware configurations according to the performance demand of the system. Our analysis shows that our model has the potential to reduce operational energy consumption up to 25 percent while retaining the same quality-of-service. Given the lower power consumption, the profits of the datacenter operators may substantially increase while the impact on the environment is reduced.

The remainder of this paper is structured as follows. The subsequent section offers a rich overview of green initiatives ordered by their scopes. Several concepts are introduced and the model of this paper is motivated along this scheme. Subsequently, the theoretical details of Tecless are introduced, followed by an evaluation of the model on generic data taken form Wikipedia. The paper concludes with a discussion of the results and an overview on the future work.

Related Work

The term “Green IT” comprises all efforts and activities incorporating ecological friendly technologies and process along the whole lifecycle of IT products. In the field green datacenter operation, this especially targets the reduction of the carbon footprint of the IT industry and therefore increasing energy efficiency is major goal. As the expenses for energy rapidly increase, the goals of green operation are generally coherent with economic goals. The manifoldness of this domain cannot be comprehended with a single solution concept, but requires various efforts on the different layers.



Datacenter Location

Datacenter Layout

Hardware

Software

Management

Can environmental beneficial locations help to reduce the energy consumption?

What are the potentials of mobile datacenters?

How does the datacenter floor layout affect the power intake?

What is the level of efficiency of the different units in datacenter?

To what extend does the power consumption of a server depend on the utilization?

What are the potentials of reactive management mechanisms?

Does optimized software design influence the efficiency?

Can efficient management concepts improve the utilization?

Is the remote lease of infrastructure an alternative?

Figure 1. Fields of Green IT Operation

Figure 1 shows the different aspects of Green IT Operation and introduces the motivating questions of each area. Most Green IT initiatives target more than one layer. The layer "Location" deals with the physical location of a datacenter and examines the economic and ecologic impact of the facilities environment. The second layer is concerned with the layout of a datacenter and aims to remove inefficiency based on obsolete or inefficient designs. The "Hardware" layer is mainly interested in the optimization of single servers in a datacenter, with special regard to their physical properties. The "Software" layer comprises efforts in the field of code optimization and scheduling for resources conserving operation. The last layer deals with economic management aspects. Table 1 offers a broad overview on different Green IT activities and related initiatives

Table 1. Scope of Green IT Initiatives

Object	Scope	Location	Layout	Hardware	Software	Management
Physical Location	Utilizing hydroelectric power or building datacenters in moderate climate zones reduces the cost for energy. For example, several US IT companies have built their datacenters next to the Columbia River, Oregon, USA area to profit from hydroelectric energy (Kurp, 2008).	X				
Mobile Datacenters	Mobile datacenters constitute a new design alternative, allowing to provide computing resources locally to temporary demand spots (e.g. Olympia). The project "Blackbox" of Sun Microsystems integrated a complete datacenter into a standard shipping container (Sun, 2009). Goggle applied for a patent for a "Water-Based Data Center" including concepts for sea-based electricity generators (Clidaras, 2008).	X	X	X		
Physical Configuration	Servers generally require air-conditioned environments. However, conventional layouts do not regard the airflow within the datacenters and disregard the resulting mixture of chilled and hot air. A first step to increase efficiency is setting hot and cold aisle. In state-of-the-art datacenters planning the airflow is simulated and efficiently guided through the infrastructure (Sharma et al., 2005) (Bash et al., 2006) (Sharma et al., 2008).		X	X		
Datacenter Infrastructure	More than 50% of the energy consumed in datacenters is needed for the infrastructure, e.g. cooling, power supplies or light. Especially uninterruptible power supplies (UPS) are often dimensioned to provide the complete datacenter. Reducing UPS to the critical systems, reduces the power consumption significantly (Velte et al., 2008).		X	X		
Energy Efficient Hardware	Server design is often solely focused on performance whereas energy efficiency is often secondary. However, for example power supplies are relatively cheap and increase the efficiency significant. Volunteer certificates have been introduced, e.g. EnergyStar (EPA, 2008), to encourage minimum efficiency standards for hardware. A broad overview on energy efficient hardware is given in (Velte et al., 2008).		X	X		
Chip Level	The processor accounts for about 30% of the energy consumption of a server. Though the main design goal of chip development is performance, energy efficiency gets more and more attention. This includes new chip layouts (Brunschwiler, 2008) (Puttaswamy and Loh, 2006), or new chip-level cooling concepts (Linderman et al., 2007).			X		
Server Design	Innovative server design can significantly help to improve the efficiency. For instance blade systems are one common concept. Instead of using multiple stand alone servers, several systems are installed in one rack, sharing common hardware such as power			X		

	supplies, network gear or fans. This helps to reduce the energy demand and allows efficient cooling concepts (Velte et al., 2008).					
Storage Systems	Enhancements of storage systems are an important part of green-IT. (Won et al., 2008) developed an energy-aware scheduling algorithm for bundling disk accesses and setting the disk into idle mode in the mean times. Virtualization and Consolidation together with the help of Network Attached Storage Devices, enables the automated storage of data on appropriate media (hard disks, tape) according to reliability, availability and access frequency concerns (Schulz, 2009).			X	X	
Thermal Aware Scheduling	The heat signature of servers depends on their utilization. To achieve a uniform thermal distribution within the datacenter, the workload is distributed according to the physical layout of center (Moore et al., 2005) (Ghanbari et al., 2007) (Ranganathan et al., 2006).			X	X	
Virtualization and Consolidation	Oftentimes enterprise systems are required to run on dedicated servers for security or compatibility reasons. However these systems are often only weakly utilized. With the help of consolidation, the system can be migrated to a single machine, while virtualization guarantees the independence of the systems (Schulz, 2009).			X	X	X
Cloud Computing	Cloud Computing is a new computing paradigm, providing virtualized resources as a service over the internet. Integration of Cloud Computing into an enterprise system helps to improve the utilization as peak demands are provided with remote resources. However the derivation of the optimal size is a non-trivial (Hedwig et al., 2010).				X	X
Fast adaption of Business Process	Changes in business processes are often accompanied with new software systems. However, legacy systems often remain online for years and are hardly used. Fast adoptions and regularly reviews of the IT infrastructure can lead to significant cost reductions (Velte et al., 2008).					X

Adaptive Provisioning Models

Adaptive Provisioning Models constitute a new Green IT concept and belong to the categories hardware, software and management. The model proposed in this paper is specifically designed for the requirements of distributed enterprise and information systems with a large user community, whereby the varying usage intensity of the clients over time is explicitly modeled. Virtualization and consolidation has a similar scope. Various standard software products already reached market maturity, e.g. Xen or VMWare. However, this concept specifically targets small- and midsized systems (Velte et al., 2008), which do not have the non-trivial complexity of distributed and replicated architectures. A closer examination of the problem setting reveals it's bilaterally. Adaptive provisioning does not only demand a profound understanding of the performance behavior of a large-scaled system, but eventually requires triggering reconfigurations of the hardware infrastructure in advance. This necessity demands to model the workload process in order to forecast the near future of the system load and hence to guarantee a continuous QoS. The approach of designing complex models to improve energy efficiency has also been examined in detail (Viswanathan and Monie, 2006) (Dhiman and Rosing, 2006). Though the scope is similar to these concepts, Tecless specifically models the workload process as well as it utilizes a complex system performance model.

The field of performance characterization of large-scale, distributed systems is already alone a hot topic in research and practice; several concepts have been proposed to solve this problem. Most advanced system performance models abstract from the technical properties of the infrastructure as well as the software design characteristics and utilize empirical data coupled with statistical methods, such as machine learning, or operations research concepts. The main objectives of these approaches are (i) to evaluate the performance characteristics of a system and (ii) in case of performance failures to identify the root cause. One renown approach makes use of Tree-Augmented Bayesian Networks (TANs) to identify combinations of system level metrics with their corresponding threshold values and system performance measures (Cohen and Chase, 2004, Zhang et al., 2005). Given an arbitrary system state and a corresponding workload, this methodology has the capability to calculate the probability of performance failures and to identify the root causes of the performance limitation. Another widely used concepts employ queuing-models (Urgaonkar et al., 2005a, Urgaonkar et al., 2007, Urgaonkar et al., 2005b). The intuition is to reproduce the structure of an n-tier application as a queuing network and to model the processing time at each layer of the system as transition probabilities. Based on the transition time of jobs through the network, the performance can be assessed. The lengths of the queues represent potential bottlenecks in the system.

Some of these models incorporate first extension for adaptive provisioning. The TAN based model has been extended with a mechanism for short-term performance forecasting (Powers et al., 2005). With the help of time series analysis and regression, the occurrences of performance violations in the system are modeled. By extrapolating the derived process, the future system stress is forecasted. This prediction, together with the TAN, provides the base for reconfiguration decisions. Compared to the Workload Prediction Model of Tecless, the prediction algorithm of the workload is rather reactive than predictive and completely disregards observations of the distant past. Furthermore the prediction regards the workload process en bloc without isolating different factors of influence. Though the introduced adaptive provisioning model is based on similar techniques, the different nature of the factors of influence is analyzed and estimated individually, leading to a significantly higher forecasting accuracy.

The discussed queuing model has also been advanced to an adaptive provision model. (Urgaonkar et al., 2008) propose a predictive algorithm to estimate the request arrival rate in the system. Based on the workload observations of the same time of day for a distinct period, they create a request arrival rate distribution and set a high percentile as the forecast of the near future usage intensity. This value is increased by the weighted average prediction error of the last hours. The prediction error of the recent past is included into the forecast. Based on the final prediction, they use their queuing network to estimate the system stress and reconfigure the infrastructure accordingly. Though the incorporate feedback mechanism accounts for the dynamic of the workload process, the predictive element is highly static and hence cannot react to the dynamic of a workload process. The derivation of a workload distribution tends to continuously overestimate the workload level, as this concept is highly vulnerable to outliers. The Workload Prediction Model of Tecless, does not only predict the level of the workload, but also models the dynamic of the process. This allows significantly higher prediction accuracy. Furthermore the use of a queuing network requires prior modeling of the software architecture. The Bottleneck Detection Model used in Tecless derives the performance directly from observations and hence does not require this step. Overall, Tecless is independent of the software design and the user behavior. Instead, all necessary configuration parameters of the model are directly derived from observations of the system during operation.

The Tecless Model

This paper introduces Tecless, a novel model for adaptive provisioning of distributed enterprise systems. As illustrated in Figure 2, the model comprises three major units, each dealing with the special characteristics of such systems.

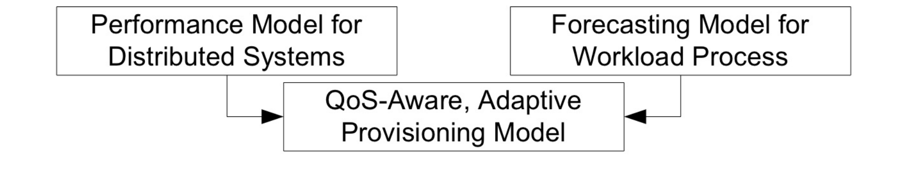

Performance Model for Distributed Systems
Forecasting Model for Workload Process
QoS-Aware, Adaptive Provisioning Model

Figure 2. Structure of the Adaptive Provisioning Model

- A **bottleneck detection model** (Malkowski et al., 2007) that specifies how a given enterprise system needs to be enhanced (e.g. by adding another application or database server) at runtime to increase the system performance. The Bottleneck Detection Model provides an accurate description of the performance characteristics of an enterprise system. By iteratively analyzing the performance of a system under different stress levels, the Bottleneck Detection Model is able to determine the maximal feasible workload of a hardware configuration as well as the resource, limiting the performance.
- A **workload prediction model** that attempts to forecast volatile user behavior: By analyzing the behavior of the past, the model predicts the level of workload in the near future. The single factors of influence are isolated and estimated separately with the help of a modified time series decomposition model. To account for unforeseen changes in the workload process, the model is complemented with a feedback mechanism. This novel methodology offers a precise forecast of the workload process with an accuracy of up to 99%.
- A **provisioning model** that brings the Workload Prediction Model and Bottleneck Detection Model together in order to continuously optimize the system configuration such that a given Quality-of-Service (QoS) level can be achieved. Based on the other two models, the provision model derives an optimal infrastructure size at any time, allowing an economical infrastructure operation.

The Bottleneck Detection Model

The following paragraphs detail the various aspects of the models. The idea of performance models for large-scale, distributed systems originates from the field of systems research in computer science. The Bottleneck Detection Model is motivated by the increasing complexity of modern architectures and the need to reconfigure hardware and software at run-time. Today, IT administrators base their decisions on monitoring data, expertise and experience.

This intuitive approach is often error-prone, which has lead to the demand of robust models for the characterizing performance behavior during operation. Particular importance has been given to the determination of the maximum performance capabilities of a system deployed on a given infrastructure as well as on the bottleneck, limiting the performance of the system.

The utilized model in this paper is based on a statistical induction model. The underlying idea of the observation-based performance model is that the complexity and interactions of the different resources of a modern architecture do not allow an exact modeling. To facilitate this complexity, the performance characteristics are directly derived from observations of the behavior of the system during its operation (Pu et al., 2007), (Malkowski et al., 2007),(Malkowski et al., 2009),and (Swint et al., 2006).

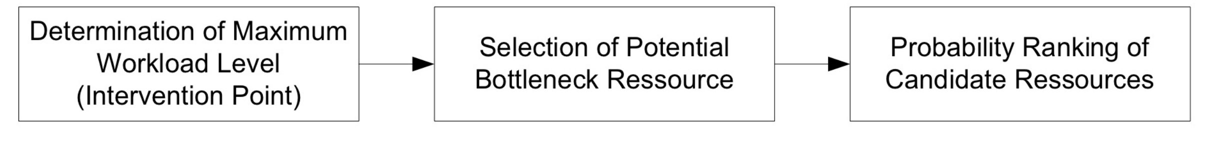

Determination of Maximum Workload Level (Intervention Point)
Selection of Potential Bottleneck Ressource
Probability Ranking of Candidate Ressources

Figure 3. Structure of the Bottleneck Detection Model

The objective of the model is to identify the bottleneck resource of the system. This is achieved by determining the maximum level of workload a system configuration is able to handle as well as the corresponding reasons for the performance limitations occurring at this maximum level. The Bottleneck Detection Model correlates the performance of the system with system level metrics of the different servers of each layer. A three-stage model (Figure 3) analyzes the behavior of the different resource and identifies the resource responsible for the performance limitations.

Determination of the Maximum Workload Level

The first step in the determination of the maximum feasible workload level is to analyze the performance behavior of the system over a wide range of workload levels $w \in WS$ (workload set). In the Bottleneck Detection Model, the term workload is defined as the number of concurrent users in the system. For every workload level w the behavior of the different resources in the servers is monitored by recording the corresponding system level metrics, such as CPU utilization, memory transfer rate or hard disk response time. With the help of a heuristic, the maximum feasible workload level is determined, which still satisfies the required QoS, whereby the QoS itself is defined as an upper limit for the end-to-end user response time.

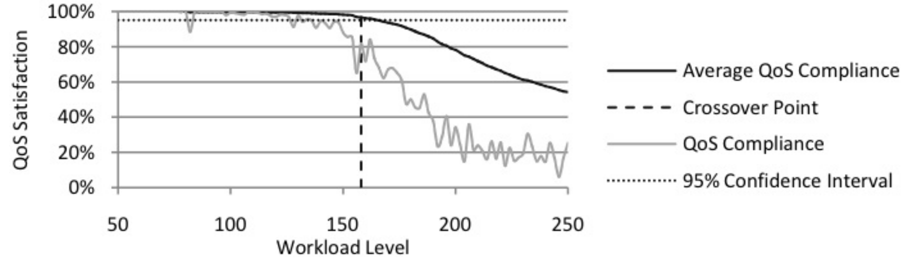


Figure 4. Illustration of the Heuristic

Figure 4 illustrates the procedure of the heuristic for the determination of the maximum workload level. The first step is to calculate the average QoS compliance of every workload level w individually (gray line). Afterwards, the heuristic starts at the lowest workload level w^{min} and iteratively adds succeeding higher workload levels w into its evaluation. In each iteration, the lower bound of the average QoS compliance of all included workload levels is determined (black line). As soon as the bound drops below 95% (dotted line), the system performance is significantly violated. This workload level (dashed line), referred as the performance crossover point c, subdivides the workload set WS into the two disjunctive subsets I, of workload levels complying with the required QoS, and I', showing significant performance limitations.

Selection of Potential Bottleneck Resources

After the determination of the maximal feasible workload level of the system, it is now necessary to determine the potential cause of the performance limitation. By correlating system level metrics to the performance behavior (i.e. system response time), the limiting factor can be derived (e.g. database hard disk, or application server memory). The procedure is briefly outlined in the following paragraph. The system level metrics are linearly correlated with the system stress. This allows the representation of their relationship with a linear function. As a result of this linear relation, the first derivative of each of these functions is approximately constant (Parekh et al., 2006). As soon as a computing resource (e.g., CPU, memory, bandwidth) becomes saturated, its corresponding metrics either become constant (e.g. CPU utilization) or tend towards infinity (e.g. disk response time). This leads to a shift in the slope of the linear function that is highly likely to be for the first time significant in the vicinity of the crossover point *c*. Statistical intervention analysis (Brockwell 2002) provides the necessary tools to examine changes in the slope of the single system level metrics. Given the approximated first derivative of each system level metric, the slope of the metrics before *(I)* and after the cross over point *c (I')* are calculated. With the help of statistical hypothesis testing, it is determined, whether there is a significant change in the behavior of the resource in the vicinity of the crossover point *c* or not. However, due to various interactions and dependencies between the single resources, not only one system level metric, but multiple different metrics are likely to exhibit this behavior. This leads to the last step.

Probability Ranking of Candidate Potential Bottlenecks

After the identification of the set of potential bottleneck resources, we subsequently perform a ranking of the candidates to find the true bottleneck of the system. This is performed by using a probability ranking based on the magnitude of change in the previously mentioned slope behavior. By this, we examine the correlation of the system level metric with the system performance more closely, facilitating the reliable identification of the bottleneck resource. For each metric a score value is determined. The resource with the maximal score value has the highest likelihood to be the bottleneck. The Bottleneck Detection Model identifies the maximum feasible workload level and the bottleneck resource. Based on these results, the system configuration can be adapted to achieve higher performance goals. More concretely, this methodology can be used to determine optimal configurations for certain levels of workload. The first step is to analyze a minimal system configuration. After the determination of its maximal workload level and the corresponding bottleneck, the bottleneck resource is replicated to achieve higher performances. By repeating this procedure iteratively, a set of workload levels and corresponding configurations can be derived. This set is later used to select on optimal configuration according to the state of the system.

The Workload Prediction Model

In the previous subsection, we presented the Bottleneck Detection Model, which derives the performance characteristics of large enterprise systems. This methodology delivers a set of configurations and their corresponding maximum workload levels, which provide the foundation for dynamically adapting the infrastructure size to the demand. However, as previously mentioned, changes in the system configuration are time-intensive, e.g. the database needs to be mirrored on additional database servers. As workload processes exhibit high volatility, reconfiguration decisions cannot be solely based on the current state of the system. Instead, the evolution of the workload process needs to be anticipated, such that QoS is continuously satisfied while at the same time the energy efficiency is increased.

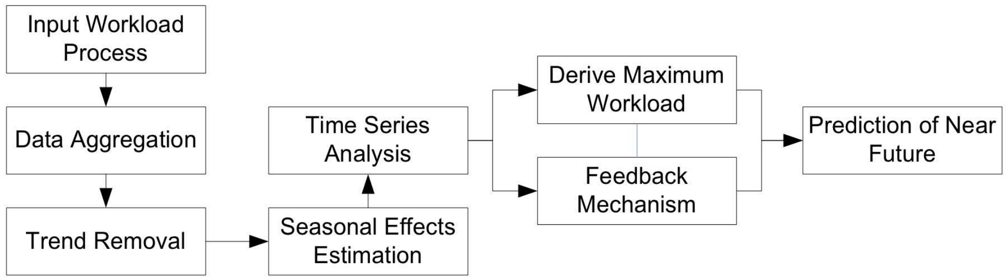


Figure 5. Structure of the Workload Prediction Model

This subsection presents a novel time series approach for analyzing and forecasting workload processes of end-user enterprise systems. Our analysis of this processes showed strong correlations of the workload level with the time of day and day of week. Furthermore, we identified long-term factors of influence which imply that the workload processes is not fully stationary. The presented Workload Prediction Model is able to forecast non-stationary workload processes, as long as the historic data allow deriving the long term dynamics of the process. Sudden changes in the process behavior are handled with a feedback mechanism. Compared to other approaches in this domain (Powers et al., 2005, Urgaonkar et al., 2008), this model can actually identify and model all significant factors of influence. Our resulting model allows predicting the near future. Together with a feedback mechanism, this novel concept is able to predict workload process with an accuracy of up to 99% in the mentioned domain. Figure 5 illustrates the structure of the Workload Prediction Model. The first step is the aggregation of the data to remove the short-term variation. Afterwards a modified time series analysis decomposition model is used to estimate the different factors of influence on the process. This includes the trend removal, the estimation of seasonal effects and the analysis of the residual process. As the decomposition model predicts the average expected workload, the final step is to calculate the maximum expected workload in the near future. This upper bound is necessary to account for the short-term stochastic variation of the process. To provide the model with capability to react to unforeseen changes, the model is complemented with a feedback mechanism.

Properties of the Workload Processes

In the scenario of end-user systems, the workload is generated by various clients with individual usage patterns. Users can start at new season at any time, execute an arbitrary transition path, and terminate their session. Modeling all clients individually is infeasible and hence the prediction model is designed to work on aggregated data. All clients are aggregated into a single source. This levels out individual effects and reduces the complexity to a manageable level. In our Workload Prediction Model, we assume that usages patterns (i.e. composition of workload) of the aggregated user communities are constant over time. This allows neglecting transaction type information.

The database of the Workload Prediction Model provides a training dataset based on the past observations. The database is used to estimate the various parameters of the model. The Workload Prediction Model requires a time series of the total number of request per period in the training dataset. This leads to two configurable parameters: The length of single period Δ and the total training dataset length $n\Delta$. The advantage of a longer timeframe $n\Delta$ is the reduction of the standard error of the later estimators. However, we have to take into account that usage patterns can vary in the long term (influence of non-stationary factors). This might bias the parameter estimation. Consequently choosing the dataset length is a tradeoff between accuracy of the estimators and the capability of the model to adapt changes in the process behavior. The second parameter Δ (single period length) should be long enough to eliminate the short-term stochastic variation. The optimal period lengths of both parameters need to be empirically evaluated, as their optima strongly depend on the characteristics of the individual processes. The aggregated input processes consists of $i=1,\dots,n$ data points r_i, each representing the total number of requests during a single period. Each data point r_i starts at the time t_i and ends at $t_{i+1} = t_i + \Delta$.

$$r_i = m_i + s_i + Z_i \qquad (1)$$

After the definition of the model input, we are now able to define the formal model. The decomposition model (1) of the workload process r_i consists of a long term trend component m_i, a seasonal element s_i and the residual stochastic process Z_i. By definition of the decomposition model, the trend and seasonal component are considered as deterministic elements. The following subsection presents appropriate methods to estimate the single factors of influence directly from the workload process. The feedback mechanism and the determination of the maximum expected workload are subsequently discussed.

Removal of the Long-Term Trend

Classical decomposition models suggest a top-down approach to analyze time series data (Brockwell and Davis, 2002). The first step is the removal of the trend of the process, which originates from changes in the user behavior (e.g. increasing usage intensity) or the growth of the user community. Classical decomposition modeling suggests the use of regression. However, the use of univariate regression is ineligible for the trend isolation in our scenario, as workload processes often exhibit more complex dynamics. An alternative methodology is the smoothing with a long-term moving average as suggested by (Kreiß and Neuhaus, 2006). The idea is to remove the long term trend from the observed process without knowing the true factors of influence. It assumes that the effect of the long term factors is already significantly represented in the preceding data points. To avoid interference with seasonal effects, the length of the moving average window needs to be longer than the seasonal periods. Therefore it is set to one week, whereby the number of data points per week is given by n_w.

$$m_i = -\frac{1}{n_w}\sum_{j\in L_i} r_i \qquad L_i = \{ j \in N \mid t_j < t_i \wedge t_j > t_{i-n_w} \} \qquad (2)$$

Equation (2) calculates the average workload level of the preceding week for every data point r_i. This value yields the first component of the decomposition model. Furthermore it can be shown that this methodology satisfies the first property of weak stationarity (Kreiß and Neuhaus, 2006).

Analysis of Seasonal Effects

The isolation of the seasonal influence in the process is the next step in the decomposition model. It deals with the removal of all factors with periodical recurring influence on process. In the context of our model two effects can be identified: the weekday and the time of day. Regarding that the length of the calibration dataset comprises at most the data from a few months, yearly effects cannot be estimated. Principally, the effect of the time of day and day of week can be estimated separately. However, evaluations have shown that - given a sufficient long training dataset - the simultaneous estimation of both effects deliver statistically better results.

$$s_i = \frac{1}{|w_i|}\sum_{j\in W_i}(r_j - m_j) \qquad W_i = \{ j = 1,\ldots,n \mid w(j) = w(i) \} \qquad (3)$$

Equation (3) expresses our simultaneous estimator, where the help-function *w(i)* returns a unique index representing the time of day and weekday. For each index the average number of requests is calculated. This factor is estimated on the base of the trend free data and renders the second component in the decomposition model.

Analysis of the Residual Stochastic Process with Time Series Analysis

The last element of the decomposition model deals with the residual stochastic variation of the process. The residuals are given as the difference between the observations and the trend and seasonal component $(Z_i=r_i-m_i-s_i)$. Time Series Analysis provides various tools for the process type identification (Hamilton, 1994). Nevertheless, the selection of an appropriate model requires in-depth prior analysis of the data. Furthermore, the application of time series analysis at least requires that the process is weakly stationary, which demands the following two properties. The first property requires a constant mean $E(Z_i)=\mu$, which is automatically met having applied the moving average smoother (Kreiß and Neuhaus, 2006). The second property demands that the correlation between any two data points is constant for any given distance. The compliance to this property is hard to show. However, a thorough analysis of the results can be used as ex-post justification of this property.

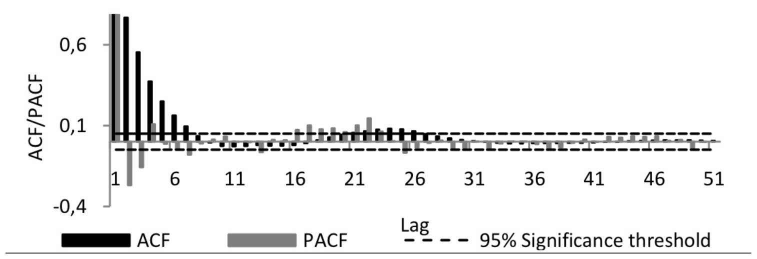


Figure 6. ACF and PACF of Residual Wikipedia Workload Process

Figure 6 shows the autocorrelation function (ACF) and partial autocorrelation function (PACF) of a sample workload trace. The exponential decrease of the ACF and the first four significant lags of the PACF indicate an autoregressive (AR) behavior of order 4. The order of the process strongly depends on *Δ* and on the characteristics of the process and hence has to be determined for every examined workload trace. Generally the fitting of a model is an optimization problem. However, in case of an AR process the Yule-Walker estimate constitutes a feasible alternative, as it allows the arithmetical computation of the model parameters. Albeit the standard error is very high, it is sufficient in this scenario, as the parameters are estimated from several hundred data points. The standard error of the Yule-Walker estimators of large datasets has approximately the same standard error as the maximum likelihood estimators.

$$\hat{\Phi} = \hat{\gamma}(p)\hat{\Gamma}_p^{-1} \text{ with } \hat{\Gamma}_p = [\hat{\gamma}(k-J)]_{k,j=1}^{p} \text{ and } \gamma_z(h) = Cov\left(Z_i + Z_{i+h}\right) \qquad (4)$$

The formulation of Yule-Walker parameter estimation is given in (4). This leads to final model, given in (5).

$$\hat{Z}_i = E(Z_i) = \hat{\phi}_1 Z_{i-1} + ... + \hat{\phi}_p Z_{i-p} \qquad (5)$$

$$\hat{r}_i = \hat{Z}_i + \hat{m}_i + \hat{s}_i \qquad (6)$$

This formulation allows predicting the next value in the residual series Z_i based on the most recent observations of the process. By adding the trend and the seasonal component (6) the average expected workload in the near future is forecasted. Depending on the size of the intervalΔ , the most recent values are not included in the prediction. However, these values are considered through the later discussed feedback mechanism.

Determining the Maximum Number of Concurrent Users

Up to this point our scheme predicts the average number of requests per single period. However, the Bottleneck Detection Model bases its analysis on the number of concurrent users in the system. By relying on the QoS requirement, being end-to-end response time < 1s (c.f. the last section) and utilizing the fact that the end-to-end response time of the system tends towards the maximal accepted response time in the case of performance violations, the number of concurrent jobs in the system can straightforwardly be derived. This, however, entails an overestimation of the workload during the uncritical phases of the system.

$$\hat{r}_i^c = \frac{\delta}{\Delta}\hat{r}_i \qquad (7)$$

Having in mind that the model primarily depends on the accurate prediction of the number of concurrent users during the critical states, this issue does not alter the results. Accordingly, we have the approximated number of concurrent jobs in the system is given by (7).

Determining the Maximum Expected Workload

Hitherto, the Workload Prediction Model estimates the average expected number of concurrent jobs in the system. Due to the stochastic nature of the workload process, the level of workload will exhibit very high short-term variations. To ensure a continuous QoS satisfaction, the system configuration needs to handle the maximum expected workload in each single period. With the help of statistical inferences, the maximum expected workload level, which is not exceeded with a certain probability, is determined. Assuming independence of single requests allows us to model the workload process as a queuing model. Independent arrival processes are usually modeled with an exponential mean time distribution (Kleinrock, 1975). By definition, in this scenario, the total number of requests during a period is Poisson distributed with a variance of the square root of the expected value. In case the number of events during the period is large enough (>100), then the Poisson distribution tends toward normal distribution.

$$\hat{u}_i^c = \hat{r}_i^c + 2.33 + \sqrt{\hat{r}_i^c} \qquad (8)$$

This nice property enables us to formulate the maximum expected workload level. To determine the workload, which is not exceeded with a probability of 99%, the normal distribution confidence interval is calculated (8).

Feedback Mechanism

Recall that the idea of the Workload Prediction Model is to analyze past user behavior in order to predict near future workload. However, in case of unforeseen changes in the usage patterns, e.g. flash-crowds (Urdaneta et al., 2007), this model ultimately fails. To overcome this limitation, the model is complemented with a basic feedback mechanism, which continuously compares the model prediction with the observed data. This allows the model to reactively deal with short-term non-stationary process factors.

$$f_i^c = \begin{cases} (r_{i-1}^c - \hat{r}_{i-1}^c) & if\ (r_{i-1}^c - \hat{r}_{i-1}^c) > 0 \\ 0 & else \end{cases} \quad (9)$$

If the former prediction is lower than the corresponding observation, then the difference of both is added to the next workload prediction. The feedback mechanism (9) is the last element of the Workload Prediction Model. The final model enables forecasting the expected number of concurrent users in the system.

$$\tilde{\hat{r}}_i^c = \hat{u}_i^c + f_i^c \qquad (10)$$

The maximum expected workload (10) will be used in the Adaptive Provisioning Model in the next section. The value can be interpreted as an upper bound of the number of concurrent jobs in the system (which is not exceeded by at least a 99% probability). At the bottom line, the hardware configuration needs at least to handle this number of concurrent user.

Putting Things Together in the Adaptive Provisioning Model

The Bottleneck Detection Model provides a set of configurations and corresponding maximum workload levels. The Workload Prediction Model forecasts the expected maximum number of concurrent jobs in the system for the next period. The final step is to combine both models into the Adaptive Provisioning Model, which aims at selecting the minimal system configuration that satisfy the QoS at any time

Let $c_i \in C$ be a valid hardware configuration for the examined system, whereby c_i is defined as the smallest feasible configuration. The variable c_i^{max} defines the maximal feasible workload level of the configuration c_i which satisfies the QoS requirements. Furthermore, the configurations are sorted according to $j<i \Rightarrow c_j^{max<}\ c_i^{max}$.

$$c^* = \arg\min_{c_i \in C}\{c_i^{\max} \mid c_i^{\max} > \tilde{r}_i^c\} \quad (11)$$

Equation (11) constitutes the core of the Adaptive Provisioning Model. Depending on the predicted maximum workload level, the smallest sufficient system configuration is selected. During operation, the Adaptive Provision Model continuously supervises the workload process. If the present configuration is insufficient to handle the expected system stress, the model switches to the next stronger configuration c^*_{i+1}. If a weaker configuration c^*_{i-1} is also capable of handling the expected maximum workload, the capacity of the system is reduced to the next smaller configuration.

Case Study on Wikipedia Workload Traces

Following the presentation of the theoretical background, this section evaluates the performance of the Adaptive Provisioning Model. The model is assessed in a simulation of a real world environment based on generic data sources. The Wikimedia Foundation (Wikimedia, 2009) provides detailed workload traces (Mituzas, 2009) of their projects. Being ranked 8th on the list of globally most accessed websites (Alexa, 2009) and with a distributed infrastructure of more than 350 servers (Wikimedia, 2008), Wikipedia is a representative example for large information systems. In the following analysis, the traces of the German (DE), English (EN) and Japanese (JA) versions of Wikipedia are analyzed.

Figure 7 shows a sample dataset of one week for the three Wikipedia projects. Obviously, there is a very high variation in the workload traces over time. For example, the minimum requests per second of Wikipedia Germany (DE) are as low as 45 requests per second at night, while during daytime it goes up to 660 requests per second. Specifically, the German and Japanese versions have a very low utilization at night, as their target user community is geographically concentrated.

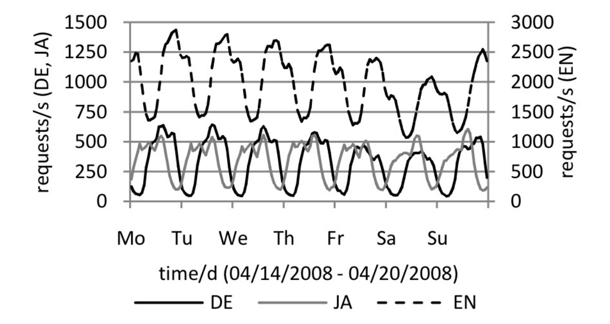


Figure 7. Wikipedia Workload Traces

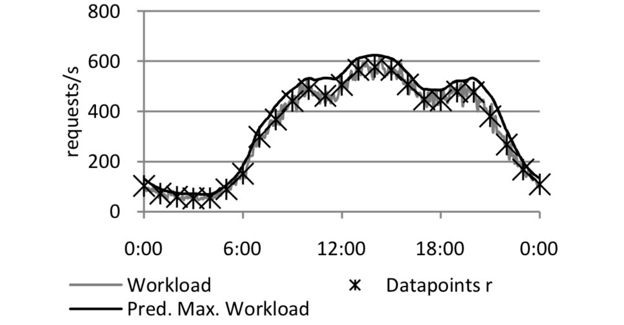


Figure 8. Workload Prediction Model

Hence, adapting the hardware configuration to the demand is likely to offer high saving potentials. Nevertheless the shape and level of the single traces vary over time. For example the characteristics of weekdays are different to weekends. Also occurrences like public holidays, have an impact on the workload trace.

Results of the Workload Prediction Model

Figure8 shows the application of the Workload Prediction Model on the Wikipedia DE project for a single day (09/23/2008). The gray line shows the simulated input processes. The marks represent the aggregated data points r_i used for the model calibration. The prediction model is setup to predict the maximum workload level in the next 15 minutes. The period length Δ is set to one hour and the calibration dataset comprises the data of the preceding twelve weeks. Empirical evaluations showed that this parameter setup delivers reliable and accurate results. The black line depicts the maximum predicted workload per second or, according to the argumentation, the number of concurrent jobs in the system. Overall, the model is able to predict the workload level accurately.

$$q_{0.05} = \frac{1}{n}\sum_{j=1}^{m} h'_j , h'_j = \begin{cases} 1 & \dfrac{r_j - \tilde{r}^{c}_j}{r_j} \leq 0.05 \\ 0 & else \end{cases} \qquad (12)$$

To assess the prediction quality, the following key figure $q_{0.05}$ (12) is defined. It gives the percentage of predictions, which did not exceed the true outcome by 5% of the relative workload, whereby n donates the number of model evaluations during the regarded time frame. This gives a tangible measure for the accuracy of the prediction.

Table 2. Application of the Bottleneck Detection Model

Project	DE	JA	EN
$q_{0.05}$	98.75%	99.20%	98.40%

Table 2 summarizes the results of the prediction quality for one week (09/23/2008 – 09/29/2008), whereby the model has been recalibrate every second day. All three workload traces have been accurately predicted by the model.

Results of the Bottleneck Detection Model

The Workload Prediction Model provides the first input for the Adaptive Provisioning Model. The second input is given by the Bottleneck Detection Model. The following table presents exemplary results of the Bottleneck Detection Model on a RUBiS Benchmark (RUBiS, 2004). The experiments were conducted on the Emulab (White et al., 2002). Details of the experiment setup can be found in (Malkowski et al., 2007). The RUBiS benchmark is a test bed auction site modeled after eBay. In this experiment it is configured as a three-tier application, consisting of a web server, an application server and a database server. The web and application server were installed on high end systems, the database server on a low- end machine. The applied scenario consisted of a workload composition with 30% of the requests causing a database write, while the remaining 70% are sole read operations. Table 3 presents the results of the detection process. The notation (W-A-D) refers to the number of replication in each tier of the system.

Table 3. Application of the Bottleneck Detection Model

Configuration (W-A-D)	Maximum Workload Level/ Crossover point c	Bottleneck Resource
1-4-1	844	Application Server CPU
1-6-1	1400	Database Server CPU
1-8-1	1370	Database Server CPU
1-8-2	1640	Application Server CPU

The Bottleneck Detection Model correctly identifies the limiting resource in each scenario. In the 1-6-1 configuration the database server becomes the bottleneck resource. Replicating the application layer does not increase the performance. Only the replication of the database increases the overall performance of the system. These results give an idea of the performance characteristics of large systems depending on their hardware configuration.

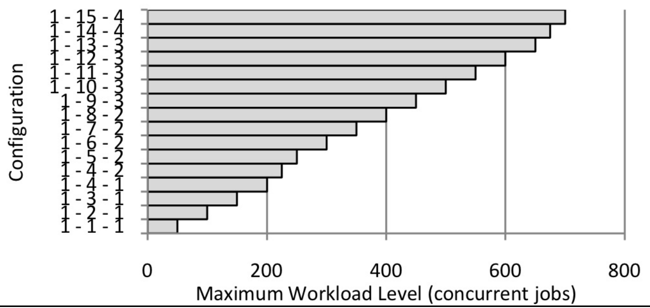


Figure 9. Configuration and Workload Levels

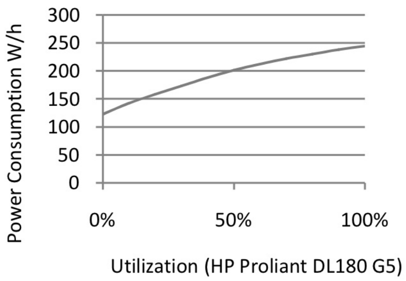


Figure 10. Utilization/ Power Consumption

For the further evaluation of the Adaptive Provisioning Model, a synthetic set of configurations of the Wikipedia infrastructure and their corresponding maximum workload levels is assumed. Figure 9 shows the ascending set of configurations and their corresponding workload levels used in the further evaluation.

Utilization depended Power Consumption

For a realistic estimation of the saved energy, a reference server is selected. Figure 10 shows the utilization dependent power consumption of an HP Proliant DL180 G5 server (SPEC, 2008). The energy intake of servers partially depends on their utilization. By removing servers form the infrastructure, the utilization of the remaining machines is increased. Hence part of the energy saved by switching of machines is consumed through the higher utilization of the remaining servers. This behavior needs to be regarded in the assessment.

Adaptive Provisioning

The final step is the application of Tecless. Based on the results of the Workload Prediction Model and the Bottleneck Detection Model a sufficient small hardware configuration is selected at any time.

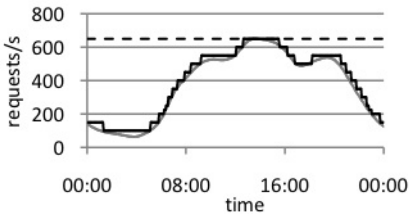


Figure 11. Adaptive Provisioning

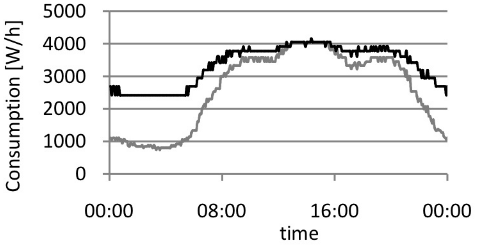


Figure 12. Power Consumption

Figure 11 depicts the application of the model on the same reference day as in (Figure 8). The lower gray line donates the prediction of the maximum expected workload. As the provisioning can only be modified in discrete steps, the black line shows the maximum workload level of the selected configuration according to (Figure 9). The upper, light gray line shows the static provisioning, which is set to handle 650 requests/s. It is evident, that only during peak times the maximal setup of the systems is needed. Figure 12 shows the resulting energy consumption of the system. The gray line shows the power consumption of the static system configuration. During the night, the energy intake is constant on a high level as the system utilization is close to idle. Especially during nights, the adaptive provisioning contributes significant savings, while during the peak times the consumption is equal.

Table 4. Application of the Bottleneck Detection Model

Project		DE	JA	EN
Average Server Utilization	Static Configuration	49.8%	51.8%	68.1%
	Adaptive Provisioning	79.0%	83.6%	93.7%
Average Power Consumption	Static Configuration	3.46 kw/h	3.99 kw/h	18.55 kw/h
	Adaptive Provisioning	2.59 kw/h	2.95 kw/h	14.17 kw/h
	Improvement	25.3%	26.0%	23.6%

Table 4 summarizes the results for all three examined projects. The adaptive provisioning increases the average server utilization of activated severs by up to 30%. Based on the utilization of the different machines and the utilization based power consumption of the reference server, the energy intake can be estimated for the static and the adaptive provisioning. On average, the energy consumption of the systems is decreased by 25%. In conclusion, the Tecless performed well on the Wikipedia Workload Traces. It is able to provide the same QoS with a significantly lower energy consumption.

Conclusion and Outlook

This paper presented Tecless, a novel methodology for adaptive provisioning of large enterprise systems. Tecless accounts for strong usage intensity volatilities during operation and the complexity of distributed systems. The main two objectives of the adaptive provisioning model are to guarantee a continuous QoS and to minimize the power consumption of the system. Energy conservation is achieved by adapting the size of the hardware infrastructure to the demand. Redundant servers are removed from the infrastructure and switched off because this is the most efficient way to conserve the electrical base load of the servers.

The foundation of the adaptive provisioning model consists of both the workload prediction model and the bottleneck detection model. The bottleneck model specifies how a given enterprise system needs to be scaled (e.g. by adding another application or database server) at runtime to increase the overall system performance. In other words, the bottleneck detection model provides an accurate description of the performance characteristics of an enterprise system. By iteratively analyzing the performance of a system under different stress levels, the bottleneck detection model is able to determine the maximal feasible workload of a hardware configuration as well as the resource limiting the overall performance. However, reconfiguration decisions are time intensive due to the activation delay of servers. Hence, modifications of the infrastructure need to be triggered in advance, which requires to forecast the system stress.

The workload prediction model attempts to forecast the workload process exactly. By analyzing the behavior of past user behavior, the model predicts the level of workload in the near future. The single factors of influence are isolated and estimated separately with the help of a modified time series decomposition model. To account for unforeseen changes in the workload process, the model is equipped with a feedback mechanism. This novel methodology offers a precise forecast of the workload process with an accuracy of up to 99 percent.Evaluations of the adaptive provisioning model on generic data have shown an energy saving potential of up to 25 percent, without significantly influencing the QoS. Given the steady increasing operation cost of datacenters, this methodology can help to slow down this growth process. Consequently, Tecless might reduce the impact on the environment and improve the economical efficiency of enterprise systems.

The evaluations of the adaptive provisioning model are very promising. The next step is the implementation of the model in a testbed. This will further confirm the validity of the model and investigate potential implementation challenges. Furthermore, we plan some extensions of the model. In the current version, the feedback mechanism is solely based on the prediction error. The use of more reactive machine learning feedback mechanism might enable the usage of the model in systems, which show stronger irregularities in their workload process.

References

Alexa (2009) Alexa Top 500 Sites. Amazon.com Inc.

Alonso, G., Casati, F., Kuno, H. & Machiraju, V. (2002) *Web Services,* Springer, Berlin ; London.

Bash, C. B., Patel, C. D. & Sharma, R. K. (2006) Dynamic thermal management of air cooled data centers. In: *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. ITHERM '06. The Tenth Intersociety Conference on*, pp. 8 pp.-452.

Brockwell, P. J. & Davis, R. A. (2002) *Introduction to Time Series and Forecasting,* Springer, New York.

Brunschwiler, T. M., Bruno (2008) Thermal Management of Vertically Integrated Packages. In: *Handbook of 3D Integration*(Ed, Dr. Philip Garrou, D. C. B. D. P. R.), pp. 635-649.

Clidaras, J. S., David W; Hamburgen, William (2008) Water-Based Data Center (Ed, Office, U. P. T.), pp. GOOGLE INC, USA.

Cohen, I. & Chase, J. S. (2004) Correlating instrumentation data to system states: A building block for automated diagnosis and control. In: *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, pp.

Dhiman, G. & Rosing, T. S. (2006) Dynamic power management using machine learning. In: *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pp. ACM, San Jose, California.

EPA (2008) Energy Star Program Requirements for Computers Version 5.0. pp. EPA.

Ghanbari, S., Soundararajan, G., Chen, J. & Amza, C. (2007) Adaptive Learning of Metric Correlations for Temperature-Aware Database Provisioning. In: *Proceedings of the Fourth International Conference on Autonomic Computing*, pp. IEEE Computer Society.

Hamilton, J. D. (1994) *Time Series Analysis,* Princeton University Press, Princeton, NJ.

Hedwig, M., Malkowski, S., Bodenstein, C. & Neumann, D. (2010) Datacenter Investment Support System (DAISY). In: *Proceedings of the 43rd Annual Hawaii International Conference on System Sciences (HICSS-43)*, pp. Computer Society Press, Hawaii, USA.

ITP (2007) Data Center Energy Efficiency: Turning Challenges into Opportunities. In: *Issue Focus: Meeting Data Center Energy Challenges*, Vol. Energy Matters, pp.

Kersten, R. (2007) In: *Sun microsystems*, pp.

Kleinrock, L. (1975) *Queueing Systems. Volume 1: Theory,* Wiley-Interscience.

Koomey, J. G. (2007) Estimating Total Power Consumption by Servers in the U.S. and the World. pp. Lawrence Berkeley National Laboratory, Berkely, CA.

Kreiß, J.-P. & Neuhaus, G. (2006) *Einführung in die Zeitreihenanalyse,* Springer, Berlin, Heidelberg, New York.

Kurp, P. (2008) *Commun. ACM,* **51,** 11-13.

Linderman, R., Brunschwiler, T., Smith, B. & Michel, B. (2007) High-performance thermal interface technology overview. In: *Thermal Investigation of ICs and Systems, 2007. THERMINIC 2007. 13th International Workshop on*, pp. 129-134.

Malkowski, S., Hedwig, M., Parekh, J., Pu, C. & Sahai, A. (2007) Bottleneck Detection Using Statistical Intervention Analysis. In: *Managing Virtualization of Networks and Services*, pp. 122-134.

Malkowski, S., Hedwig, M. & Pu, C. (2009) Experimental Evaluation of N-tier Systems: Observation and Analysis of Multi-Bottlenecks. In: *IISWC '09: Proceedings of the 2009 IEEE 12th International Symposium on Workload Characterization*, pp. IEEE Computer Society.

Mituzas, D. (2009) wikistats. pp.

Moore, J., Chase, J., Ranganathan, P. & Sharma, R. (2005) Making scheduling "cool": temperature-aware workload placement in data centers. In: *Proceedings of the annual conference on USENIX Annual Technical Conference*, pp. USENIX Association, Anaheim, CA.

Parekh, J., Jung, G., Swint, G., Pu, C. & Sahai, A. (2006) Issues in Bottleneck Detection in Multi-Tier Enterprise Applications. In: *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, pp.

Powers, R., Goldszmindt, M. & Cohen, I. (2005) Short term performance forecasting in enterprise systems. In: *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 801-807. ACM, Chicago, Illinois, USA.

Pu, C., Sahai, A., Parekh, J., Gueyoung, J., Bae, J., Cha, Y.-K., Garcia, T., Irani, D., Lee, J. & Lin, Q. (2007) An Observation-Based Approach to Performance Characterization of Distributed n-Tier Applications. In: *Proceedings of the 10th IEEE International Symposium on Workload Characterization (IISWC)*, pp. 161-170. Boston, MA.

Puttaswamy, K. & Loh, G. H. (2006) Thermal analysis of a 3D die-stacked high-performance microprocessor. In: *Proceedings of the 16th ACM Great Lakes symposium on VLSI*, pp. ACM, Philadelphia, PA, USA.

Ranganathan, P., Leech, P., Irwin, D. & Chase, J. (2006) Ensemble-level Power Management for Dense Blade Servers. In: *Proceedings of the 33rd annual international symposium on Computer Architecture*, pp. IEEE Computer Society.

RUBiS (2004), pp.

Schulz, G. (2009) *The Green and Virtual Data Center,* CRC/Auerbach Publications.

Sharma, R. K., Bash, C. E., Patel, C. D., Friedrich, R. J. & Chase, J. S. (2005) *IEEE Internet Computing,* **9,** 42-49.

Sharma, R. K., Shih, R., Bash, C., Patel, C., Varghese, P., Mekanapurath, M., Velayudhan, S. & Manu Kumar, V. (2008) On building next generation data centers: energy flow in the information technology stack. In: *Proceedings of the 1st Bangalore annual Compute conference*, pp. ACM, Bangalore, India.

Singh, A., Hayward, B. & Anderson, D. (2007) Green IT Takes Center Stage. pp. springboard research.

SPEC (2008) SPECpower_ssj2008 Results pp. Standard Performance Evaluation Corporation

Sun (2009) Project Blackbox. pp. Sun microsystems.

Swint, G., Jung, G., Pu, C. & Sahai, A. (2006) Automated Staging for Built-to-Order Application Systens. In: *Network Operations and Management Symposium*, Vol. 10th IEEE/IFIP, pp.

Urdaneta, G., Guillaume, P. & van Steen, M. (2007) Wikipedia Workload Analysis. pp. Vrije Universiteit, Amsterdam, Netherlands.

Urgaonkar, B., Pacifici, G., Shenoy, P. & Spreitzer, M. (2005a) An analytical model for multi-tier internet services and its applications. In: *SIGMETRICS Perform. Eval. Rev.*, pp.

Urgaonkar, B., Pacifici, G., Shenoy, P. & Spreitzer, M. (2007) Analytic modeling of multitier Internet application. In: *ACM Trans. Web*, pp.

Urgaonkar, B., Shenoy, P., Chandra, A. & Goyal, P. (2005b) Dynamic Provisioning of Multi-tier Internet Applications. In: *Second International Conference on Autonomic Computing, 2005*, pp.

Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P. & Wood, T. (2008) ACM Trans. Auton. Adapt. Syst. In: *Agile dynamic provisioning of multi-tier Internet applications*, pp.

Velte, T., Velte, A. & Elsenpeter, R. (2008) *Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line,* McGraw-Hill Osborne Media.

Viswanathan, L. P. & Monie, E. C. (2006) Reinforcement temporal difference learning scheme for dynamic energy management in embedded systems. In: *VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on*, pp. 6 pp.

White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C. & Joglekar, A. (2002) An Integrated Experimental Environment for Distributed Systems and Networks. In: *ACM SIGOPS Operating Systems Review*, pp.

Wikimedia (2008) Wikipedia:Server. pp. Wikimedia Foundation.

Wikimedia (2009) Wikipedia:About. pp. Wikimedia Foundation.

Won, Y., Kim, J. & Jung, W. (2008) *Multimedia Systems,* **13,** 409-428.

Zhang, S., Cohen, I., Goldszmidt, M., Symons, J. & Fox, A. (2005) Ensembles of Models for Automated Diagnosis of System Performance Problems. In: *International Conference on Dependable Systems and Networks*, pp.