DYNAMIC SERVICE LEVEL AGREEMENT MANAGEMENT FOR EFFICIENT OPERATION OF ELASTIC INFORMATION SYSTEMS

Completed Research Paper

Markus Hedwig
University of Freiburg
Platz der Alten Synagoge
79085 Freiburg, Germany
markus.hedwig@is.uni-freiburg.de

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

Dirk Neumann
University of Freiburg
Platz der Alten Synagoge
79085 Freiburg, Germany
dirk.neumann@is.uni-freiburg.de

Abstract

The growing awareness that effective Information Systems (IS), which contribute to sustainable business processes, secure a long-lasting competitive advantage has increasingly focused corporate transformation efforts on the efficient usage of Information Technology (IT). In this context, we provide a new perspective on the management of enterprise information systems and introduce a novel framework that harmonizes economic and operational goals. Concretely, we target elastic n-tier applications with dynamic on-demand cloud resource provisioning. We design and implement a novel integrated management model for information systems that induces economic influence factors into the operation strategy to adapt the performance goals of an enterprise information system dynamically (i.e., online at runtime). Our framework forecasts future user behavior based on historic data, analyzes the impact of workload on system performance based on a non-linear performance model, analyzes the economic impact of different provisioning strategies, and derives an optimal operation strategy. The evaluation of our prototype, based on a real production system workload trace, is carried out in a custom test infrastructure (i.e., cloud testbed, n-tier benchmark application, distributed monitors, and control framework), which allows us to evaluate our approach in depth, in terms of efficiency along the entire SLA lifetime. Based on our thorough evaluation, we are able to make concise recommendations on how to use our framework effectively in further research and practice.

Keywords: Service Science, Adaptive provisioning, Green IT/IS, Service management, Service Level Agreements, Workload Forecast

Introduction

Effective information systems (IS) have increasingly matured to critical success factors in modern enterprises. For instance, the ability to process extensive business data quickly or the operation of innovative and powerful customer portals have become vital necessities in today's dynamic business world. In this sense, effective adaptation to new technology trends as well as agile response to changing customer requirements are key to sustainable business practices. Nonetheless, through a strict focus on functionality, IT operation expenditures have typically been neglected; however, a rising cost of operation and the awareness of its substantial environmental foot print has risen corporate IT out of its "shadowy existence" into a key challenge for the next decades (Koomey 2007).

Current market predictions estimate an annual growth of the server market by 5% (Pettey & Tudor 2010). Given this trend, information systems will most probably continue to grow in size and complexity, and thus the challenge of the steadily growing consumption of resources persists. Green IT initiatives have already taken up this problem and offer a variety of innovations and best practice to reduce the economic and environmental impact of computing (Velte et al. 2009; Schulz 2009). However, thorough optimization and enhancement efforts are often beyond the expertise of medium-sized and small enterprises, therefore requiring high initial investments.

With the emerging service world, companies have obtained a powerful alternative to their classic in-house IT operation concepts. For instance, cloud computing enables enterprises to outsource parts of their physical IT to service providers and buy back computing power on a pay-as-you-go basis. In contrast to traditional in-house operation, service providers can leverage economy of scale and scope effects. Thus, by offering their services to multiple customers simultaneously and building up expertise, they can provide corporate computing demands more cost effectively. Within the service world, Service Level Agreements (SLAs) have become the common practice to define the terms of business between two parties. On the one hand, they contain the Service Level Objectives (SLO), such as response time requirements; on the other hand, they include the financial arrangements for the service operation. While in the beginning service providers mainly offered their services on a best-effort basis, today strict binding SLA's are slowly being established in the market (Sahni & Tan 2011; Goolsby 2007) .

Nonetheless, the adaptation to the new service technology as well as the sensible utilization of IT will be one of the key challenges of enterprise computing in the next decade. Even though various technological advances as well as new software design paradigms enable highly efficient operation modes today (Bailey 2009), the feasible implementation bears further problems in the real operation. For instance, this includes the relation between the Quality of Service (QoS) and the economic parameters of a service contract (e.g. price and penalty). In this context, we present our new integrated Service Level Agreement (SLA) operation model, which contributes to the sensible and economic utilization of IT. More concretely, we designed and implemented a new approach for the operation of large enterprise information systems. In contrast to existing concepts in the field of green technology, our model does not solely aim at reducing the resource consumption. Instead, our work extends the current state of the art by incorporating both *technical and economical* parameters of SLAs into the system operation strategy. By correlating the economic value of the system (i.e. profit and penalties), cost of operation, user behavior, and performance characteristics, our integrated management approach derives profit optimal operation strategies. Accordingly, our model facilitates highly efficient operation strategies for information systems while at the same time guarantees a continuously high QoS.

In this paper we focus on large web-facing enterprise information systems that provide their services to a large number of concurrent users. The workload of these systems is typically characterized by continuously varying size (i.e., workload) and composition (i.e., workload mix) of simultaneous requests, where each single request only generates a relatively small system load. Common representatives of this group are applications such as e-commerce portals and bulletin board platforms. Inherently, these end-user driven systems often face a highly volatile workload as usage patterns are typically characterized by a strong seasonality component (e.g., time of day or day of week). In addition it is noteworthy that the systems are traditionally over-provisioned because QoS is apparently valued higher than cost savings.

In summary, these enterprise information systems often suffer from an inefficiently low average utilization of computing resources. Furthermore, these systems are typically operated on commodity hardware or on entry level servers in order to reduce the cost of operation (Short et al. 2011). While virtualization and consolidation may help significantly to mitigate some of these problems for relatively

small instances of applications, virtualization and consolidation as such do not directly affect the efficient operation of large information systems, which require the computing power of several nodes. One feasible solution for large systems is the use of resource adaptive operation modes, which adapt continuously the system size to the user demand.

However, the implementation of such systems is non-trivial. In fact, if the operated service is provided on the basis of an SLA, the problem becomes particularly hard because efficient operation modes inherently increase the risk of violating performance constraints (i.e., QoS requirements). Even worse, if we consider different SLA configurations for a single service, the problem becomes increasingly complex. For instance, a high-priced service should be managed more conservatively than a low-cost service. In Addition, our work extends the traditional static view on Service Level Objectives (SLOs) towards a fully dynamic notion of SLA compliance. By continuously evaluating the SLA compliance at run time and adapting the SLOs in real-time, the system can react to changes in its environment instantaneously. This dynamic view on SLAs significantly reduces the risk of violating SLAs at runtime based on wrong or outdated operational decisions.

To account for the heterogeneity of IT systems, our integrated management model utilizes a complex system performance model and a workload forecast model. Based on these components, our information system operation model can determine the impact of different operation strategies and find a profit optimal configuration at runtime.

The high complexity of enterprise system environments entails that any attempt trying to prove the validity of a novel approach to management that bases on data analysis alone is not reliable due to the non-linearities in system operation. Instead, the only way to validitate the goodness of a novel management approach is by relying on experiments on a real-life testbed. Accordingly, we developed a test environment to provide a reliable evaluation. Our test system comprehends of a cloud infrastructure, a benchmark application, extensive monitoring and control software, and the integrated, SLA aware management model itself. In order to provide a real-world evaluation scenario, we used a real production system workload process to generate our test workload. Our evaluations indicate that the application of our new resource management system allows reducing resource consumption of the IT systems under test conditions by up to 40 percent.

The main contribution of this work is threefold.

- We provide a new perspective on the management of enterprise information systems and introduce a novel framework that harmonizes economic and operational goals.
- Our novel dynamic SLA management model, which induces economic influence factors into the operation strategy, adapts performance goals of an enterprise information system dynamically (i.e., online at runtime).
- The integration of our model into a test environment allows us to evaluate our concept in depth, in terms of efficiency along the entire SLA lifetime. Based on our thorough evaluation, we are able to make concise recommendations on how to use our framework effectively.

The remainder of this paper is structured as follows. The next section discusses the design problem statement followed by the presentation of the related work. Subsequently, we discuss the foundation of the integrated information system management model, followed by a detailed presentation of the model. Finally, in the succeeding section, the model is evaluated. The last section summarizes the paper and provides a conclusion and an outlook on our future work.

Design Problem Statement

Keeping pace with continuous IT innovations and product life cycle enhancements has become extremely difficult and cost intensive for enterprises. As a consequence, IT landscapes often suffer from physical server sprawl, over provisioned information systems, and a lack of integrated management tools and service management frameworks (Bailey 2009).

The efficient and QoS aware operation of elastic enterprise information systems requires a thorough understanding about all factors of influence that affect the overall performance of the system. Even though the idea of resource-adaptive operation of large applications seems to be very tempting, its application is absolutely complex. This complexity stems from the fact that the overall performance of

systems depends on various, interdependent factors; changes in each factor can lead to critical performance limitations or – in the worst case – cause total system failure. In order to manage a system efficiently, the factors of influence need to be well understood. Accordingly, we can categorize the following factors of influence (Figure 1) into the groups' workload, SLA, software components, as well as the infrastructure characteristics. Based on these factors, we have to derive an **operation strategy** or operation mode respectively. In classic system design, this would be the provisioning and configuration of an appropriate infrastructure. In modern service design, the operation strategy is more complex and can be adjusted to the preferences of the provider or customer. Furthermore, depending on the focus, there might be more dominant factors, such as data quality. Nonetheless, in this work we focus on the factors determining the system performance.

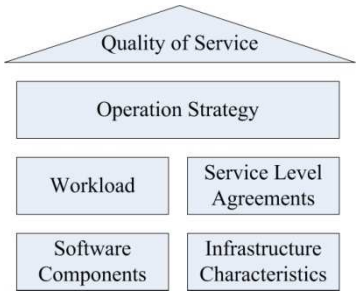

Figure 1: Factors of QoS Influence

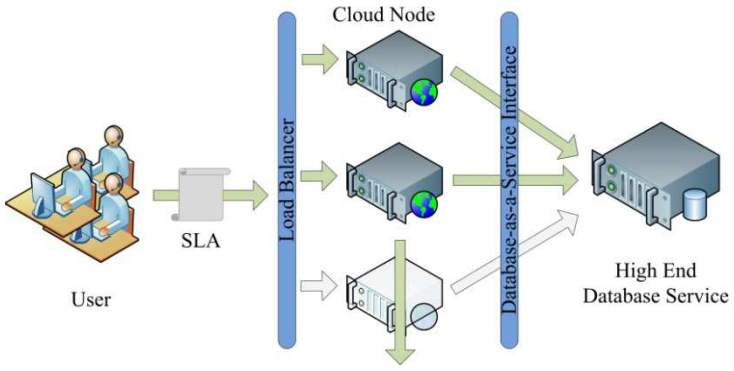

Figure 2: Elastic Application in a Cloud Computing Environment

The **infrastructure characteristics** and **software components** determine the performance characteristics of an information system. In our work, we use a state-of-the-art design for information systems in clouds (Figure 2). Elastic applications have emerged as the next generation of distributed systems. Compared to classic design concepts, they provide extensions enabling resource adaptive operation modes. In particular, this design is highly beneficial for cloud environments as it allows for utilizing the capability of the cloud to instantaneously allocate resources to systems increasing the scalability tremendously (RightScale 2011). Additionally, we utilize a database-as-a-service instead of a traditional database to avoid the problems of data synchronization. Due to their complexity, large information systems often reveal a complex performance behavior. In order to provision the optimal infrastructure configuration for any given workload, the characteristics of the software and the infrastructure must be known a-priori. System performance models are one approach to determine the expected QoS of a certain system configuration facing a certain workload level.

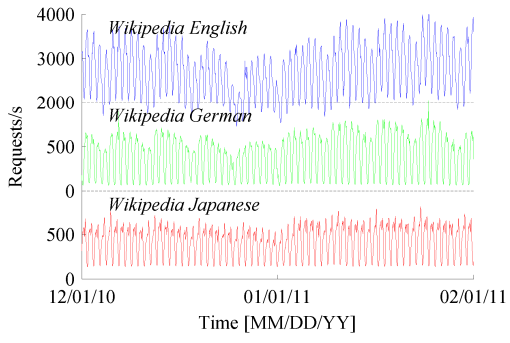

Figure 3: Wikipedia Workload Trace

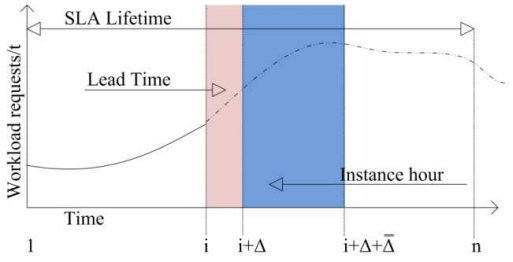

Figure 4: Reconfiguration Lead-time

Next to the system performance characteristics, the **workload** process is a key factor of influence determining whether or not a system is able to meet QoS objectives. Many end-user applications face highly volatile workloads, as the workload traces of Wikipedia (Figure 3) demonstrate. This volatility leads

to low average utilizations in static system designs (i.e., the number of resources remains constant). By dynamically adding and removing resources (e.g. servers) from the system, the average utilization of the system can be significantly increased. Nonetheless, the optimal adaptation faces many challenges. More specifically, one key challenge is to overcome the reconfiguration lead time (Figure 4). Usually, the hardware configuration cannot be adapted instantaneously; rather there occurs a delay between the initiation of reconfiguration and its availability due to configuration of the resources, synchronization and reconfiguration of the load balancers. However, due to significant lead-times in the reconfiguration, resource adaptations need to be initiated in advance. The concrete lead time depends on the level of reconfiguration. Nonetheless for the purpose of our model, we assume that the lead time is constant. Thus, in order to mitigate the risk of SLA violations or system crashes, the systems cannot be managed on the basis of the current state, but rather require near-future workload predictions. At heart, workload processes (i.e. particular workload generated by a large number of users) are stochastic processes consisting of trend and seasonal factors as well as unpredictable anomalies such as sudden appearances of users (flash-crowd effect). Consequently, workload forecast mechanisms need to be customized for the individual characteristics of the workload process.

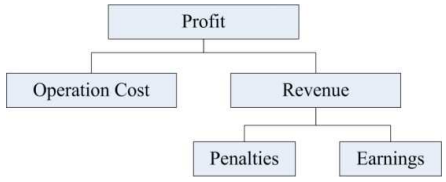

Figure 5: Provider Revenue Model

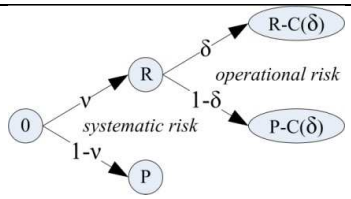

Figure 6: Risk view on SaaS Operation

The final factor of influence is the **Service Level Agreement** itself. It defines the revenue and penalties as well as the cost of operation in our scenario. Figure 5 shows the structure of our economic model. The profit is defined as the provider revenue of the system less the infrastructure cost for providing the service. The ultimate goal of our information system operation model is to increase the resource efficiency of an enterprise information system by aligning the operation strategy to these economic parameters. To facilitate this alignment, we consider the industry scenario of a Software-as-a-Service (SaaS) provider, operating her application on the infrastructure of a third party Infrastructure-as-a-Service (IaaS) provider. Clearly, the application of our SLA model is not limited to this scenario, but for the purpose of this paper, this scenario enables a full cost assessment of the impact of resource adaptive operation modes. In the later evaluation, this helps us to determine the saving potential of our model in sevice operation.

Figure 6 illustrates the risk perspective of adaptive **operation strategies**. In static operation modes, the major risks are infrastructure failures, such as power outages. This systematic risk $(1-\vartheta)$ of IT system operation can mainly be countered by using high quality equipment. During the Service Level Agreement design phase, this systematic risk is typically included in the price for the service offer. In addition to this systematic risk, resource adaptive operation modes exhibit an operational risk factor. For instance, sudden peaks in the workload process can cause sharp drops in performance causing SLA violations. Those situations could have been mitigated using larger configurations. In this sense, we define $(1-\delta)$ as the risk that a certain strategy violates the SLA. By using more resources, this risk can be reduced or even eliminated: this comes however, at the price of higher operation cost $C(\delta)$. Theoretically, the provider can select a risk level δ which maximizes his profit. Evidently, this implies that operation strategies which embody a certain risk of SLA violation might be beneficial for the provider.

Having in mind that the optimal operation strategy is not risk free, Figure 7 depicts the economic situation of the service provider. The redline represents revenue and penalty defined in the SLA, the blue line the cost of operation and the black dashed line the resulting profit. Obviously, if the provider maintains too few resources, the SLA is ultimately violated, which entails the payment of a penalty. On the other hand, if the provider adds too many resources to the system, the corresponding costs of operation increase, reducing overall profit. Assuming the provider knows the optimal operation strategy a priori (i.e. the optimal infrastructure size at any given point during the SLA lifetime), he can choose the cost minimal strategy of the optimal point of operation. Albeit, this optimal strategy would maximize the provider's profit, this strategy would induce a high risk, as minimal performance violations would directly lead to a

negative profit. Given the uncertainty during operation, a rational (at least risk neutral or risk averse) provider would choose a less risky operation strategy in the gray shaded area in order to build up some performance reserves to compensate for the operational risk of sudden changes in the environment.

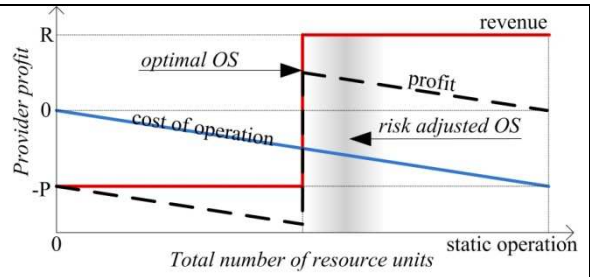

Figure 7: Profit depending resource usage

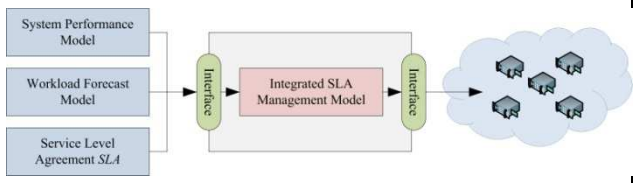

Figure 8: Integrated SLA Management Model Design

Based on the aforementioned factors of influence, we developed our integrated, SLA aware, information system management model (Figure 8). The goal of our model is to derive a cost effective operation strategy. Based on the revenue and penalty defined in the SLA and the cost of operation, our model aims to find a profit optimizing tradeoff between the cost of operation and the risk of performance violations. Our model manages the system on the base of a workload forecast model, a system performance model as well as on the specification of the SLA. The output of the aforementioned model is converted into the data format of our dynamic and integrated SLA management model and processed. Due to this modular design, our management tool supports a variety of these models (c.f. Related Work section). Additionally, our model encompasses actuator interfaces, which initiate, based on the real-time evaluation of the data, the reconfiguration of the infrastructure (i.e. add and remove servers) as well as the reconfiguration of the application (i.e. configure load balancers). In our later evaluation we will use a Fourier Transformation based forecast model and an empirical system performance model developed in our recent research.

Related Work

Our integrated, SLA aware management model combines various different research threads into an interdisciplinary model. In the related work section, we will discuss the different aspects of performance modeling, workload forecasting, and SLA management and present the current state of the art in research.

Performance analysis of large, distributed systems is a very active research field and a variety of models have been developed. Famous representatives are for instance (Urgaonkar et al. 2008), who used queuing models for automated resource allocation in information systems or (Cohen et al. 2004) who employed machine learning to model the performance characteristics. Both models are an alternative to our empirical system performance model presented in this paper. Most of the newer contributions in this field extend the performance models to dynamic research management systems. For instance, the authors in (Gmach et al. 2009) developed a reactive migration controller for virtualized environments. However, compared to our concept, their approach is only designed for basic single-layered systems. The paper (Chandra et al. 2003) introduces a resource allocation model for shared datacenters based on a queuing network performance model and a time series workload forecast mechanism. However, they do not consider SLA's in the provisioning process. Another concept (Padala et al. 2009) is an automated control model for virtual resources. The model manages the varying resource demands by dynamically allocating resources to or migrating virtual machines. Nevertheless, in modern cloud environments this migration approach is usually not supported. In (Ardagna et al. 2007) a model has been developed to manage the resource demand of multiple concurrent systems. In contrast to our model, it optimizes the system only for a single point in time, rather than incorporating the state of the SLA's. In the paper (Lassettre et al. 2003) the authors developed a surge protection model for dynamic resource allocation. The authors in (Lim et al. 2010) developed an autonomic control model to scale elastic storage systems based on the utilization of the system.

Most of the aforementioned work only subsidiarily discusses the impact of a varying workload. However, particularly in environments with highly volatile end-user workload, this factor appears to be the most

decisive factors. The book (Feitelson 2011) provides a detailed overview of workload and workload modeling. In the paper (Urgaonkar et al. 2008), the authors present a workload forecast model based on an empirical distribution estimation of past workloads. The authors in (Gmach et al. 2007) used Fourier Transformation for data smoothing and applied time series analysis for workload prediction. In (Powers et al. 2005) the authors used time series as well as regression analysis for workload prediction. Nonetheless, workload processes differ strongly in their characteristics and thus forecast methods need to be individually adapted to the process characteristics. Accordingly, all aforementioned approaches are an alternative to our Fourier Transformation based workload forecast.

Service Level Agreements and their different aspects have been extensively discussed by the research community. The following paragraph presents an overview of related and complementing work in this field. The article (Buyya et al. 2009) provides a good overview of SLAs in the field of clouds. The authors in (Yeo & Buyya 2007) developed an integrated risk analysis scheme to analyze the effectiveness of resource management policies. Based on SLAs, they determine whether a system is capable of meeting the required objectives and whether the acceptance of a single job is economically feasible. The paper (Aib & Boutaba 2007) presents an approach to business and policy driven refinement in application hosting environments. Their featured model focuses on QoS objectives and includes a mechanism for runtime adaptation. In contrast to our work, both focus on batch processing and thus do not require to cope with dynamic workloads, performance and SLA components. In the article (Hasselmeyer et al. 2006) , the authors introduce a model for the automatic negotiation of the Service Level Agreements prior to the contract start. (Buco et al. 2004) develop a business-objective-based SLA management system over the whole lifecycle of the agreements. Similarly, the paper by (Koller & Schubert 2007) presents architecture for autonomous QoS management based on SLA specifications. The paper (Sahai et al. 2001) sketches a general scheme for Service Level Agreements which allows the autonomic management in services systems. In the same direction, the paper by (Raimondi et al. 2008) presents the implementation of an automated SLA monitoring for services. Although our model does not cover all technical and negotiation aspects of the SLA, a productive version would require such an SLA management concept. In summary, most aspects of our dynamic SLA management model haven solved individually. However, we haven't found any work combining all aspects into a single integrated model for enterprise information systems.

Foundation of the Dynamic SLA Management Model

Having explained the factors of influence and the requirements for cost effective and risk aware operation of enterprise information systems in clouds, we will now present the foundation of our dynamic and integrated SLA management model. More concretely, we will discuss our recent and ongoing work in the fields of SLAs, system performance modeling, workload forecast, and information system operation strategies.

Service Level Agreements

Service Level Agreements specify all aspects of business relations between the contract partners, such as the rights and duties of each party, contract duration as well as guarantees and warranties. The QoS requirements are distinguished into Service Level Objectives (SLOs), Monitoring Intervals and SLA assertion. SLOs consist of specific measurable characteristics of the system (e.g. availability, throughput, response time) together with a threshold value for this characteristic. Furthermore, an SLA includes monitoring intervals, which define when compliance with the SLO is checked. The SLA assertion can combine several SLOs and specifies under what conditions the SLA is violated. In addition, it defines the lifetime of the SLA as a contract and also controls the penalty payments in case of SLA violation. Different SLAs may lead to very different operation strategies.

In the following presentation, we assume a basic SLA definition. The SLO demands a response time of the information system below $\theta = 500\,ms$ for each request. The SLA between the SaaS provider and consumer specifies that this SLO target must be met for α =95% of all requests on average, measured in intervals of 1 $second$. If the provider fulfills the SLA, he receives a payment of R; if he fails to comply, he has to pay a penalty of P (in the remainder of the paper, the penalty is defined as a negative number, i.e. $P = -100$). The SLA lifetime is set to $T = 24$ hours. Next to the SLA specifications, we additionally define that our controller is applied every $d = 15$ minutes. This allows us to divide the SLA lifetime in $n = 96$ periods. We further define i as the current time index and $w = w_1, \dots, w_m \in W$ as the workload

process during the SLA lifetime. Evidently, the workload data of w_j with $j > i$ are unknown at time index i. Furthermore, we assume that the provider operates the information system on the resource of an Infrastructure-as-a-Service (IaaS) provider. The cost for an instance hour is defined as $\overline{K}$.

System Performance Model

Elastic applications are one solution towards highly efficient operation modes of large enterprise systems in clouds. Recall that the provisioning the optimal hardware configuration for any given workload, a thorough understanding of the system characteristics is necessary. Usually elastic applications do not scale linearly with the amount of hardware resources. In particular distributed systems usually show complex performance characteristics, which are caused by various, non-trivial interdependencies within and between the different layers of the system. In our recent work (Malkowski et al. 2010; Malkowski et al. 2011), we developed an observation based, empirical approach for the performance modeling of large systems. In contrast to other models, our approach solely relies on the observed system behavior during operation. More specifically, our system characteristics model monitors the workload process, system metrics, as well as the SLO relevant metrics (e.g. response time) and saves the data in an operational data store. Based on this recorded data, our performance model can predict the expected degree of SLO compliance of a certain configuration c under a workload level w. For the purpose of this paper, we used a relatively simple application design (Figure 2), which scales along the number of cloud resources $c = c_1, \dots, c_q \in C$. In our scenario, the database service is assumed to deliver constant QoS, independent of the system load. In the following presentation we define lambda $\Lambda(c, w)$ as the system performance model function providing the degree of SLO compliance for a certain workload w and configuration c.

For instance, a system with $c = 3$ resource units facing a workload level of $w = 250$ might be able respond to all requests within a response time $\Theta = 500$ ms. In this case our degree of SLO compliance is $\Lambda(3,250) = 100\%$. If we reduce the system size by one node to $c = 2$, then system most likely reaches an overload state and the degree of SLO compliance consequently drops to $\Lambda(2,250) = 50\%$.

Generally, the system performance function $\Lambda(c, w)$ has a value range between 0% and 100% and can be interpreted as a step-wise defined function. If the system is overloaded, the system performance function will tend towards 0%. If the system is only moderately utilized, the system performance function $\Lambda(c, w)$ is near 100%. The degree of SLO compliance will obviously increase with the number of resource units c and decrease with the increasing workload level w.

Figure 9 shows the performance characteristics of our benchmark application used in the evaluation. The value axis represents the degree of SLO compliance, whereas the description axis presents the workload and the configuration (i.e. number of nodes in the system). The range of high QoS is delimited with the green surface. In the area between the green and the yellow service, the system is still fully functional, although it may exhibit reduced response time. For most applications occasionally operating in the yellow surface (SLO compliance between 95% and 50%) is acceptably. If the workload drops the SLO satisfaction below the yellow surface, the SLA might be compromised. Beneath the red surface, the system will enter a critical, completely overloaded state. If the system enters this area, a total system failure is likely to occur. In our benchmark system, the smallest configuration c_1 is able to achieve a $\Lambda(c_1, 30)$ =100% SLO compliance up to 30 concurrent users. However, depending of on SLO specification, the configuration can account for up to 90 concurrent users (Compliance target $\alpha = 95\%$) or 110 concurrent users ($\alpha = 90\%$), respectively.

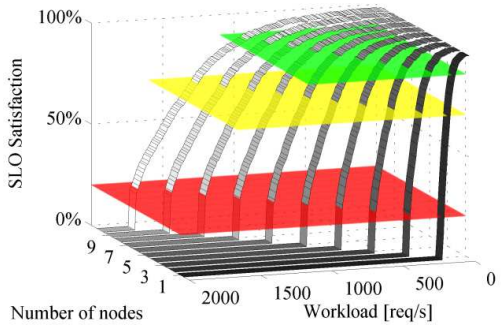

Figure 9: System Performance Curves

Workload Forecast

Modern elastic applications allow resource adaptive operation without compromising system stability. Nevertheless, significant reconfiguration lead times demand to reconfigure the system in advance

(Hedwig et al. 2009). In our recent research (Hedwig et al. 2010), we developed a Fourier Transformation based workload forecast model. The core idea of our model is to decompose the workload process with the help of the Discrete Fourier Transformation into its single spectra components. In particular, an end-user generated workload exhibit predominant seasonal factors of influence (e.g. time of day). By identifying and isolating these components, we can predict the near futures of the workload process with high accuracy. Due to space constraints we omit the detailed presentation of the forecast algorithm and define Omega (*1*) as the workload forecast function. Based on the past workload observations $\vec{u}$, we estimate the near future of the workload process from the current index i to the end of the SLA lifetime n.

$$\vec{\omega} =< \omega_i, \dots, \omega_n >= \Omega(\vec{u}, i, n) \quad (1)$$

$$\dot{W} = \begin{array}{c} \\ w_1 \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccc} i & \cdots & i+\tau \end{array} \\ \begin{pmatrix} \mathrm{P}(w_1 = \omega_i) & \cdots & \mathrm{P}(w_1 = \omega_{i+\tau}) \\ \vdots & \ddots & \vdots \\ \mathrm{P}(w_m = \omega_i) & \cdots & \mathrm{P}(w_m = \omega_{i+\tau}) \end{pmatrix} \end{array} \quad (2)$$

Our forecast model, as well as most mechanisms in literature, usually delivers point estimates for the workload in the near future. However, the later dynamic SLA management model demands distribution estimates in order to facilitate a more robust operation. Thus, we complemented our forecast model with empirical prediction error estimation. Based on the comparison between the forecast and the later observation, the accuracy of the prediction is determined. The prediction error estimation is defined in the form of a matrix (*2*), whereby τ refers to the forecast horizon of the error estimation. The single elements of the matrix contain the probability to face a workload level w in period i.

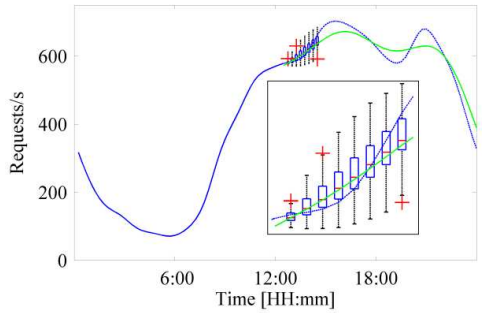

Figure 10: SLA Lifetime Forecast

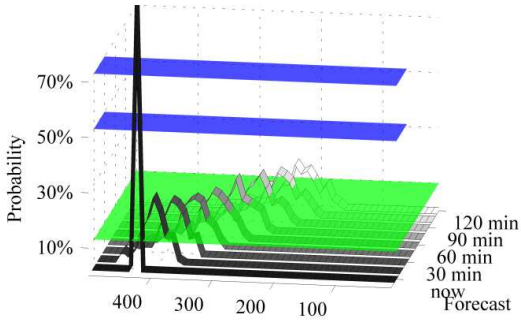

Figure 11: Workload Forecast Error Distribution

Figure 10 shows the application of the workload forecast and the error distribution on a Wikipedia workload trace (Mituzas 2011). The blue line presents the observed workload process and the green line the forecast. For the first two hours of the forecast, the prediction error distribution is depicted by the box plots. Without a detailed analysis, we see that the workload forecast provides a good forecast of the process until the end of the SLA lifetime. Figure 11 depicts the workload forecast error distribution. The first peak represents the current workload level; the following curves show the prediction distribution of the forecast periods. For the near future the prediction is very accurate, though the accuracy decreases with the increasing forecast horizon. Both components (ω and $\dot{W}$) are later used by the dynamic SLA management model to derive efficient operation strategies online.

Naïve Operation Strategies

In this section, we provide a brief overview on common operation strategies in order to show the origin of our dynamic SLA management model. We assume that the respective controllers are executed n times over the SLA lifetime, whereby the index of the current period is defined as i. Based on the most recent monitoring data, the controllers determine an optimal configuration $s \in C$ for the next period. However, as the system may have a significant lead-time Δ (defined as a multiple of the controller execution interval), the reconfiguration decisions will first be available in $i + \Delta$. The following strategies are presented in their most basic version.

$$s_{i+\Delta} = \text{constant} \quad = S^{\text{static}}(\emptyset) \quad (3)$$

The most basic operation strategy is a static system operation mode that keeps the hardware configuration unchanged over the whole SLA lifetime (*3*). The static system operation mode reflects a

classical operation strategy that is naturally extremely costly, as the system needs to be adjusted to the peak workload. In the later evaluation, we will use the static operation mode as baseline.

$$s_{i+\Delta} = c = S^{\text{reactive}}(c^*, \alpha, w) = \begin{cases} \text{scale up:} & c = c + 1 \quad \Lambda(c^*, w) < \alpha \\ \text{scale down:} & c = c - 1 \quad \Lambda(c^* - 1, w) \geq \alpha \\ & \emptyset \quad \text{else} \end{cases} \quad (4)$$

The basic reactive operation mode (*4*) follows an observation-based operation strategy. The basic intuition is that the system configuration will be scaled up if the current configuration c^* cannot satisfy the SLO at a desired compliance target of α. This reactive controller reflects the current state of the art of cloud providers. For instance Amazon Simple Queue Service (Amazon 2011) stores incoming requests in a queue. If the queue exceeds a predefined length, the system is scaled up. Although this controller type significantly reduces the resource consumption, it incorporates the inherent risk of performance violation as systems often cannot be reconfigured ad-hoc and the reconfiguration decision is solely based on the current workload level w.

$$s_{i+\Delta} = c = S^{\text{predictive}}(\alpha, \Delta, \omega) = \arg\min_{c \in C}\{c | \Lambda(\omega_{i+\Delta}, c) \geq \alpha\} \quad (5)$$

While the reactive operation modes base configuration decisions upon the current situation of the system, the predictive operation mode (*5*) uses the workload forecasts ω to make the decision. The intuition is the same as for the basic reactive operation mode. Scaling up becomes necessary if the current configuration c^* cannot meet the SLO target compliance in the near future, during which the configuration is fixed due to the lead time. In those cases it is necessary to increase the system configuration now, such that the advanced configuration at time $i + \Delta$ is available to satisfy the SLA accordingly.

Dynamic and Integrated SLA Management Model

Based on the background of the previous section, we now present our dynamic and integrated SLA management model. The core idea of our model is to feed (i) the economic parameters of the SLA document (i.e. revenue and penalty) and (ii) the cost of operation into the operation strategy of an enterprise information system. However, as the criterion of complying (violating, respectively) with an SLA is binary, the inclusion of the economic parameters into the online management strategy is difficult to achieve. In order to identify the optimal tradeoff between the cost of operation and the risk of violating the SLA, we developed a pricing and valuation scheme for the different cost factors on the period level *i*. This allows us to assess the economic impact of different configuration options in real time. This in turn allows us to derive the efficient configuration for any period *i* in real time given the recent and anticipated future performance. In this section, we will first define our management model. Based upon this dynamic model, we introduce the core of our SLA management model. Subsequently, we will discuss different configuration options with respect to their applicability.

Dynamic SLA Model

As mentioned in the related work section, SLAs have become the standard for defining the terms of operation in IT infrastructures. SLAs specify the revenue for maintaining the services, penalties for QoS violation, monitoring metrics, and corresponding thresholds. However, today SLAs are predominantly treated in a static way. Accordingly, SLAs are translated into a static set of requirements at design time of the IT services. During runtime, the system is provisioned according to these requirements. Thus, the SLOs, defined in the SLA, are merely used for monitoring the system and determining if and if so, when the SLA is violated. The previous operation modes aim at satisfying the SLO compliance target α, defined in the SLA, at any point in time. By continuously providing a performance level above α, SLA violations are avoided. However, SLAs are typically giving more flexibility to providers, as they only require fulfilling the SLO target over the entire SLA lifetime. For instance, if the initial target α is not met at the beginning of the SLA lifetime, system operation should aim at providing better QoS during the remaining lifetime such that the SLA is still satisfied. For instance, if the SLA requires a response time below one second in $\alpha = 95\%$ of all cases, the SLA can still be met if the response time of the system was initially below 95 % (say 94 %) by offering a better response time during the remaining lifetime (say 96%). In this case the SLO target is set to 96% in order to achieve a 95% SLO compliance on average and thus comply with the SLA.

$$\alpha = \delta\beta_i + (1-\delta_i)\gamma_i \Leftrightarrow \gamma = \frac{(\alpha - \delta\beta_i)}{1-\delta_i} \quad (6)$$

Our SLA model exploits this idea by optimizing over the SLA lifetime. Based on the specified SLO compliance target α, our model continuously evaluates the current degree of compliance β (i.e. the average SLO compliance up to the current period) and determines an updated SLO target γ for the period i online. If the system complies with the new target, the SLA will be fulfilled. Equation (*6*) provides the formal definition: Essentially, we define $\delta = i/n$ as the fraction of the elapsed SLA lifetime. The target SLO compliance α must be equal to the weighted current SLO compliance β and the SLO compliance in the remaining lifetime γ. Solving this equation for γ gives us the required SLO compliance for the remaining SLA lifetime. This approach facilitates updating the performance target during operation continuously succeeding in a more fine grained and sensible operation of the system.

$$\bar{\bar{\beta}} = \frac{\beta * (i-1) + \sum_{j=i}^{i+\Delta-1} \Lambda(c_j, \omega_j)}{i+\Delta-1} \quad (7)$$

$$\bar{\bar{\gamma}} = \frac{\alpha - \delta' * \bar{\bar{\beta}}}{1-\delta'}, \delta' = \frac{i+\Delta-1}{n} \quad (8)$$

$$\hat{\beta}^{c,w} = \frac{\bar{\bar{\beta}} * (i+\Delta-1) + \Lambda(c,w)}{i+\Delta} \quad (9)$$

$$\hat{\gamma}^{c,w} = \frac{\alpha - \delta'\hat{\beta}^{c,w}}{1-\delta''}, \delta'' = \frac{i+\Delta}{n} \quad (10)$$

Nonetheless, reconfiguration decisions need to be initiated in advance and thus we cannot only rely on the current state of the SLO compliance. Instead we have to estimate the state of the system in period $i+\Delta$. Again, we assume a constant reconfiguration lead time. Equation (*7*) estimates the expected degree of SLO compliance based on the current degree of compliance β and the expected degree of compliance $\Lambda(c_j, \omega_j)$ in the time between i and $i+\Delta$. Similar to equation (*6*), the expected required SLO compliance for the remaining SLA lifetime $\bar{\bar{\gamma}}$ is calculated with the relative elapsed time in $i+\Delta-1$. The goal of the our SLA model is to assess the impact of different configuration decisions. Thus, in equation (9) and (10), the impact of different configurations c for the decision period $i+\Delta$ is determined. The variable $\hat{\beta}^{c,w}$ provides the expected degree of SLA compliance in period $i+\Delta$ for the configuration c if the system faces a workload level w. The corresponding $\hat{\gamma}^{c,w}$ value is calculated in equation (*9*) with the elapsed time of the period $i+\Delta$. If $\hat{\gamma}^{c,w}$ is larger than 100%, the corresponding system configuration would have a fatal impact on the SLA, as the new SLO target could not be satisfied at all. In this case our dynamic SLA management model assumes a non avertable SLA violation.

Online Economic Assessment of the SLA State

As previously mentioned, the online derivation of a cost efficient and risk minimal operation strategy is a non-trivial problem. Usually, the assessment whether the SLA is fulfilled or violated is binary being determined at the end of the SLA lifetime. However, in order to ensure an economic-aware operation strategy, the value of the different configuration options must be evaluated online in order to derive a feasible operation strategy. Thus, our dynamic SLA management model evaluates the impact of the different configuration decisions in real time on the basis of past performance and on anticipated future user behavior. Furthermore, we designed a scheme to distribute the revenue R and penalty P among the single periods i of the SLA lifetime. This procedure allows us to derive a profit optimal operation strategy which is aware of the risk of violating the contract. By calculating the expected profit of all configuration options, weighted by the probability of facing a workload level w, we can determine the profit optimal and risk adjusted configuration for the decision period $i+\Delta$. The model is designed in a way, that it employs larger configurations (in terms of resource units) for high penalties and high revenues contracts to avoid SLA violations. In contrast, if the resources are expensive or the penalty is low, the dynamic SLA management model chooses very inexpensive operation strategies. Clearly these operation strategies entail a higher risk of SLA violations but more importantly increase the expected profit in return.

The core idea of our SLA management concept is to distribute the total revenue R on the single periods n equally. Furthermore, we define that the full revenue R/n of a period is earned if the system complies with the SLO target α. Nevertheless, as slight performance violations might be reasonable from an economic point of view, we permit that the configurations may only partially fulfill their goal in single periods. As a consequence, these periods only earn a corresponding fraction of the revenue. Conversely, we also allow that a period can earn more than the assigned revenue by overachieving the SLA target. This approach incentivizes the system to build up performance reserves.

$$\hat{R} = R\left(1 - \delta' * \frac{\bar{\bar{\beta}}}{\alpha}\right) \quad , \delta' = \frac{i + \Delta - 1}{n} \quad (11) \qquad r^{c,w} = \frac{\hat{R}}{n - i - \Delta} * \frac{\Lambda(c,w)}{\bar{\bar{\gamma}}} \quad (12)$$

In order to account for the uncertainties in the system, we continuously update the distributable revenue among the remaining periods with the help of the expected SLA compliance $\bar{\beta}$ before the decision period. By determining the ratio between the achieved compliance and target and multiplying this value with the elapsed SLA lifetime, we can derive the distributable revenue $\hat{R}$ for the remaining lifetime (*11*). In order to determine the revenue share of the period $i + \Delta$, we divide the remaining revenue by the number of remaining periods *(12)*. In contrast to our original formulation, we define that the full revenue of the period is earned, if the system complies with the online target $\bar{\bar{\gamma}}$. By multiplying this value with the expected SLO compliance $\Lambda(c, w)$ of a configuration c and a workload w, the expected revenue can be calculated.

$$\Pi = -\frac{P}{n}(i + \Delta) \quad (13) \qquad \pi^{c,w} = -\frac{P}{n}\frac{\left(\hat{\gamma}^{c,w} - \hat{\beta}^{c,w}\right)}{1 - \hat{\gamma}^{c,w}} * i \quad (14) \qquad p^{c,w} = \begin{cases} 0 & \Pi > \pi^{c,w} \\ \Pi - \pi^{c,w} & \text{else} \end{cases} \quad (15)$$

The valuation of the penalty is different from the revenue valuation. In our penalty valuation model, we assume that the system should achieve a 100% SLO compliance in each period. All deviations are, similar to the revenue valuation, priced by a fractional penalty. However, as the SLO target α accounts for limited performance shortcomings, each period is given a violation allowance of $-P/n$. If the performance allowance is not required it can be saved for later periods. Equation (*13*) defines the performance violation allowance buffer. Equation (*14*) presents the penalty caused by the system up to the current period i including the online SLO impact $\hat{\gamma}^{c,w}$ of the current decision period. Finally, equation (*15*) states that only positive penalties are relevant for the configuration decisions. This prevents, that the system eventually attempts to increase the profit of a configuration by "earning" negative penalties.

In order to derive the profit optimal configuration, the cost of operation for the period of consideration must be included in the decision process. However, cloud resources are commonly priced on an hourly base and hence we use the attributable price of the cloud resource for the period $K = \bar{K}/\bar{\Delta}$ by dividing the cost of an instance hour $\bar{K}$ by the number of periods the instance hour can be used: $\bar{\Delta} = 1/\Delta$. By multiplying this cost factor with the number of resources c, we can derive the cost impact of a configuration.

$$\text{Target Function:} \quad \vec{s'} = <s'_i, \dots, s'_n> = \Phi(\gamma, j, n) = \underset{c_i, \dots, c_t \in C}{\operatorname{argmin}} \sum_{t=j}^{n} c_i \quad (16a)$$

$$\text{Subjected to:} \quad \frac{1}{n-j}\sum_{t=j}^{n} \Lambda(c_t, \omega_t) \geq \gamma \quad (16b)$$

$$\text{Optional:} \quad \Lambda(c_t, \omega_t) > \alpha^{\text{lb}}, \forall_{t=j,\dots,n} \quad (16c)$$

In our resource adaptive operation mode together with our dynamic SLA management model, the provisioning decision in one period affects the online performance goal γ of the remaining period. In order to include this factor into our operation strategy, we forecast the workload process until the end of the SLA lifetime and select an appropriate configuration for all future periods. Equations (*16a-c*) provide the formal definitions of our provisioning model. The target function (*16a*) minimizes the total number of resource during the remaining periods $n - j$ of the SLA lifetime. The first constraint (*16b*) ensures that the resulting configuration vector is able to satisfy the online SLA target γ. The optional constraint (*16c*) allows defining a lower bound SLA target α^{lb}. This way, the dynamic SLA management model will not select a system configuration which provides less performance than the lower bound. This optimization problem can be solved in linear time utilizing a basic heuristic. Starting with the largest configuration $c^{\max}$, we remove one resource unit in each iteration from the period with the lowest relative SLO performance loss. Thus, we require at most $c^{\max} * n$ iterations to solve the optimization problem.

$$\bar{\Gamma}^{c,w} = K * (|\Theta(\hat{\gamma}^{c,w}, i+1, n)| - |\Theta(\bar{\bar{\gamma}}, i+1, n)|) \quad (17) \qquad \Gamma^{c,w} = \begin{cases} \bar{\Gamma}^{c,w} & \text{else} \\ 0 & \bar{\Gamma}^{c,w} < 0 \\ P & \bar{\Gamma}^{c,w} = \emptyset \\ P & \bar{\Gamma}^{c,w} > R + P \end{cases} \quad (18)$$

By calculating the difference between the original performance goal $\bar{\bar{\gamma}}$ and the new performance goal $\gamma^{w,c}$, we can estimate the operation cost impact on the future periods (*17*). By multiplying the resulting resource difference with the resource cost K, we obtain the economic impact $\bar{\Gamma}^{c,w}$ of the configuration c for the future periods. Nonetheless, depending on the resulting Γ, case differentiations may become necessary (*18*). The second condition defines that cost saving effects may not be included in the decision problem.

The third constraint controls that in those cases where there is no feasible solution (i.e. no operation strategy is able to achieve the SLA), the future impact is priced with the penalty P. Furthermore, we included a feasibility constraint: If the cost of future operation exceeds the revenue penalty span, continuing the operation is economical infeasible.

$$d(c) = \sum_{w \in W} \left(\dot{W}_{w,i+\Delta} * (r^{c,w} + p^{c,w} + \Gamma^{c,w}) \right) + k^{c} \quad (19) \qquad s_{i+\Delta} = \underset{c}{\operatorname{argmax}}\ d(c) \quad (20)$$

Similar to the naïve operation strategies, the optimal configuration is selected on the base of the valuation of the current system state. First, for all configurations c and all workload levels w with a positive probability of occurrence, the expected profit is calculated on the base of the expected revenue, penalty, and cost of operation as well as their impact on the future periods (*19*). Since the workload is uncertain, the outcome of each workload level is weighted with its probability. The configuration with the highest expected profit is selected for the decision period $i + \Delta$ (*20*). Nonetheless, due to unforeseen changes and the systematic risk of performance violations, the continuation of the information system operation might be economically infeasible. In case all configurations c have an expected profit of P (i.e. there is no feasible way of fulfilling the contract), the dynamic SLA management model assumes that the SLA cannot be saved at all regardless of any activities. In this case the system switches to the smallest configuration c_1 to reduce the loss by saving the cost of operation.

Modifications

The previous section presented the dynamic SLA management model in detail. Although our model is designed for a risk neutral, rational operation strategy, there are several options to adapt the model to the individual needs of the service provider. The first option is the inclusion of a lower bound SLO threshold. The presented version of the integrative SLA model is designed to switch to the smallest configuration c_1 facing a certain SLA violation. Although this might be an economically rational decision, this behavior may contradict other soft factors such as reputation of the provider. Thus, our first proposed modification is to switch to the predictive operation mode (*5*) once the SLA is ultimately violated. This is attained by setting the SLA target to the lower bound target α^{lb}. This guarantees an acceptable QoS but reduces the cost of operation for the doomed contract.

$$h'(t,c) = \begin{cases} 1 & \sum_{w \in W} \Lambda(w,c) W_{w,t} < \gamma \\ 0 & \text{else} \end{cases} \quad (21) \qquad h(c) = \sum_{t=\mathrm{i}+\Delta}^{\mathrm{i}+\Delta+\bar{\Delta}-1} h'(t,c) \quad (22) \qquad \bar{\bar{k}}(c) = K * \left(\sum_{c'=c_1}^{c} \frac{\bar{\Delta}}{h(c')} \right) \quad (23)$$

Today's cloud providers usually charge their resources on an hourly base. However, our model is usually executed in smaller time intervals and determines the expected optimal configuration in accordance with the lead time Δ. As a consequence, our SLA model might add additional resources to the system, which are only required for a limited time (i.e. significantly lower than the length of an instance hour). To compensate this effect, our model can be complemented with an economic feasibility assessment function of the provisioning decision. The core idea is that the cost for each instance hour is shared between the periods using this resource (i.e. the periods $i + \Delta$ to $i + \Delta + \bar{\Delta}$). For example, if one additional resource unit is only required in one period i, this period is charged with the price of the full instance hour $\bar{K}$. If the resource is feasible for all periods of the instance hour, each period is charged with the fractional cost K. Hence we have to determine whether the new configuration is not only feasible for the predicted period, but also for the following periods during the lifetime of the instance hours. Based on the workload forecast distribution, equation (*21*) verifies if the configuration c can sustain the workload in period t. Subsequently, equation (*22*) determines the number of periods, during instance hour lifetime $\bar{\Delta}$, this resource c is accessible. Finally, equation (*23*) defines the fractional resource costs of the current period. This valuation of the different configurations allows the model to identify the optimal configuration and significantly reduces the number of configuration changes.

$$\bar{\bar{P}} = P - K \sum_{t=1}^{i} |c_t| \quad (24) \qquad \Lambda'(c,w) = \begin{cases} \lambda & \lambda \le \gamma \\ \gamma & \lambda \ge \gamma \end{cases}, \lambda = \Lambda(c,w) \quad (25) \qquad \Lambda'(c,w) = \begin{cases} \lambda & \lambda \le \gamma \\ \lambda + (\lambda - \gamma) * \dfrac{n-i}{n} & \lambda \ge \gamma \end{cases}, \lambda = \Lambda(c,w) \quad (26)$$

The profit of the provider is defined as his revenue less the cost of operation. In particular in scenarios with low penalties and unforeseen high workload, the dynamic SLA management model may decide to give up on the SLA target. Nevertheless, if this happens in the middle of the SLA lifetime, the deliberate violation of the SLA might be an unfavorable option as the previous efforts to comply with the SLA

already caused significant cost of operations. To incorporate this aspect into the management strategy, the penalty is continuously increased by the expenditures used for additional resources (*24*). This procedure avoids abortions during the SLA lifetime.

The last modification of our model deals with the fractional revenue determination of each period. In its initial design, the model allows to earn additional revenue by over-fulfilling the SLA target in each period. In some scenarios this behavior might be unwanted. Equation (*25*) provides an alternative option by limiting the maximum revenue to the revenue assigned to the period by redefining the bounds of the degree of SLO compliance function Λ. The second option (*26*) provides a moderate alternative as it continuously shifts from the original version to the modified version (*25*) during the SLA lifetime. These incentivize the SLA model to build up performance reserves at the beginning and switch to a more cost effective operation mode at the end of the SLA lifetime.

Evaluation

The evaluation of our dynamic SLA management model, based on real production system workload traces, was carried out in a self-developed test infrastructure (i.e., cloud testbed, elastic application, distributed monitors, and control framework) (Malkowski et al. 2011). In the evaluation, we systemically analyzed the impact of different SLA properties (e.g. revenue, penalty, QoS requirements) and the various configuration options.

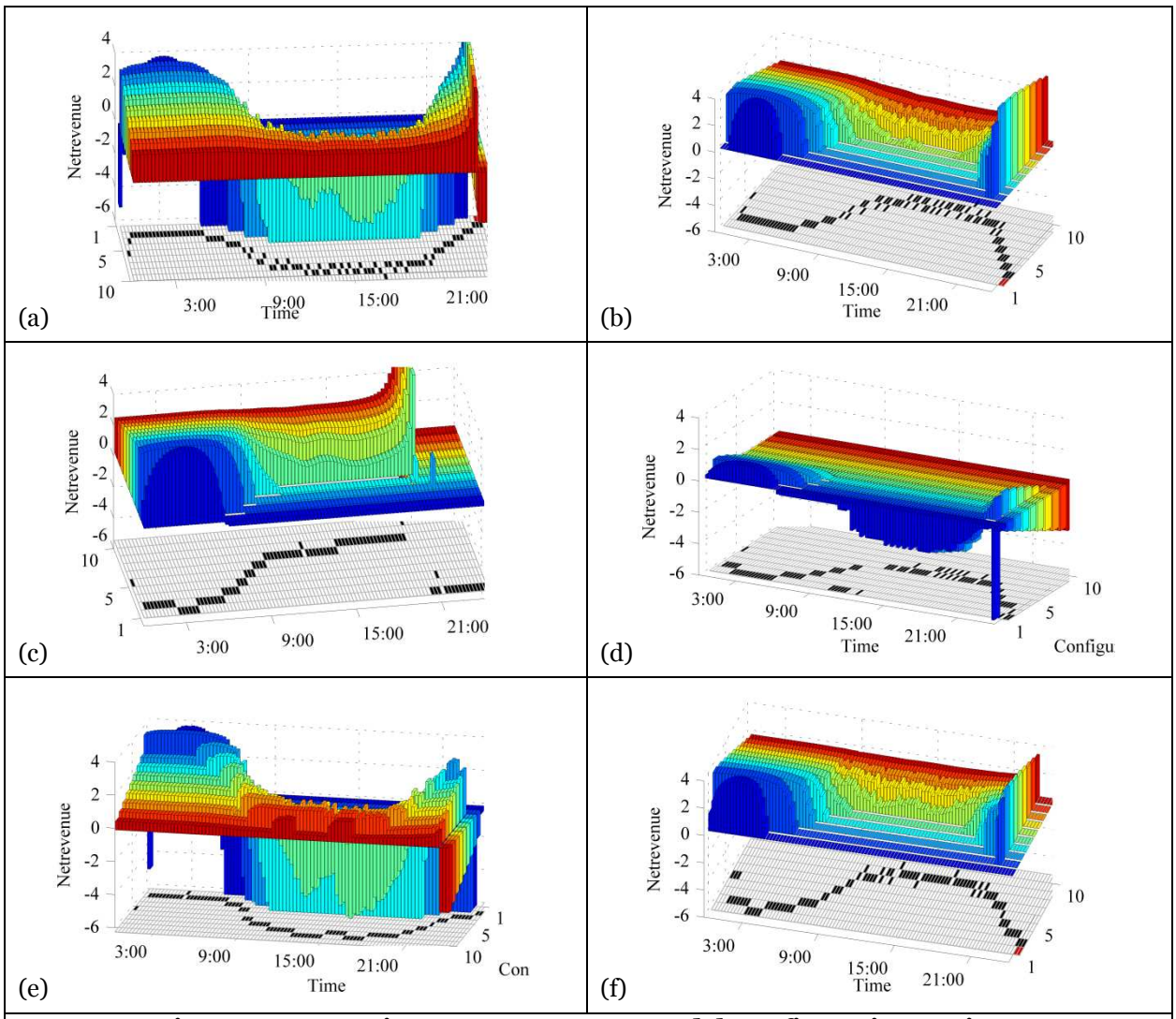

Figure 12: Dynamic SLA Management Model Configuration Options

Figure 12 provides insights into the decision logic of our dynamic SLA management model. While the bars show the expected profit of the different configurations over time, the lower surface depicts the provisioned configuration. In this scenario, we use the workload process of a single day of Wikipedia Germany (Figure 10). The revenue R and the penalty P are set to \$300. The infrastructure cost $\overline{K}$ is \$1.2 and the maximum infrastructure size is limited to $c^{\max} = 10$. The cost of operation is oriented at the common price for mid-range servers of public cloud provider offers. The SLA has a lifetime T of 24 hours. The reconfiguration lead time is set to two periods or 30 minutes respectively. Furthermore the SLA target α is set to 95%. In the following presentation, we use the original model formulation with the lower bound SLA control modification ($\alpha^{\text{lb}} = 70\%$).

Figures (a) and (b) show the characteristics of our base line scenario, where the negative components are trunctuated in figure (b). In this moderate scenario, we can study the behavior of the dynamic SLA management model in a *normal* scenario. Figure (a) shows that during peak times the weaker configurations would violate the SLA and hence are negative. The green mid range configuration exhibits only minor performance limitations and is thus only valued slightly negative. Figure (b) shows the same graphs from another perspective. We can clearly see how the weaker configurations become infeasible during daytime (higher workload) and later become feasible again with the decreasing workload.

In figure (c) and (d), we reduced the SLA target to $\alpha = 80\%$ and deactivated the lower bound control. Due to the relatively high revenue compared to the QoS requirements, the SLA is successfully fulfilled after 85% of the SLA lifetime. In figure (d), we additionally reduced the revenue to $R = 120$ and the penalty to $P = 0$. The SLA specification in this scenario is close to the economic infeasibility as the costs of operation nearly consume the total revenue. However, in absence of the lower bound control, our rational model decides not to provision resources during the expensive peak time. Due to the weak SLA definition, the dynamic SLA management model is still able to meet the SLA target and operate profitably. Nonetheless in this case the system is operated with a minimum amount of resources during the peak time. Evidently, this is no realistic scenario, but facilities the potential of our model.

In figure (e), we activated the node feasibility assessment modification. Evidently, this helps to significantly reduce the fast change of configurations. Furthermore, we can see the profit impact of this modification. As soon as the node is used for multiple periods, the resource costs per period decreases (as the node cost per instance hour is shared over multiple periods) and the specific configurations become more profitable. Finally, figure (f) shows the base line scenario with a longer reconfiguration lead-time ($\Delta = 4$). The dynamic SLA management model tends to provision more resources in the single periods. For instance, we can see this effect in the middle of the SLA lifetime. The configuration c_5 is valuated higher than in the baseline scenario (b). This behavior is caused by the lower confidence of workload forecast and the difficulties to maintain the QoS.

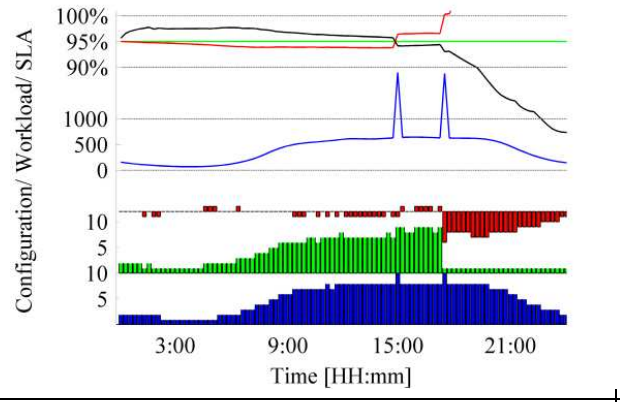

Figure 13: Low-cost Scenario with Disturbance

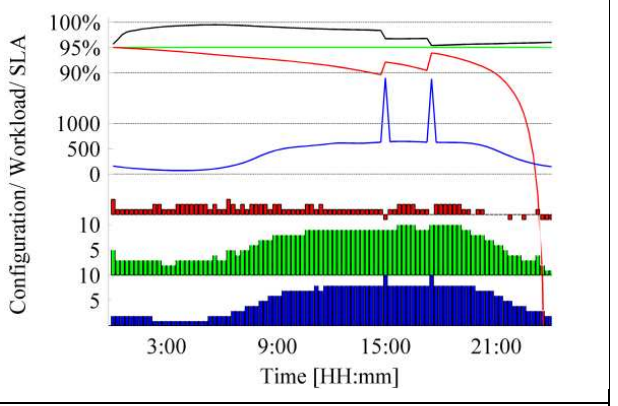

Figure 14: High Revenue Scenario with Disturbance

In the following, we present the behavior of the dynamic SLA management model in the case of an unforeseen, but regularly, recurring disturbance in the workload process. Hence, this disturbance is included in the workload forecast error distribution as a minor outlier. Figure 13 presents a low-cost scenario with a revenue and penalty of 150. Figure 14 depicts the high revenue scenario with a revenue

and a penalty of 300. The upper red line represents γ and the black line β, whereas the green line is the original performance target α. The blue line below depicts the workload process. The blue graph depicts the optimal configuration determined a priori, the green bar the provisioning of our model and the red the differences between the both.

Both figures illustrate the behavior of our SLA management model. At the beginning in both the scenarios, the SLA is overachieved, which can be seen by the decreasing value of β. Nevertheless, in the high price scenario more resources are provisioned and β decreases at a fast rate. During both peaks, the online target γ increases. Subsequent to both violations the amount of provisioned resources increases. However, following the second peak, the SLA of the lost-cost scenario is non recoverable and violated, whereas the SLA in the high price scenario can still be met, by overachieving the SLA in the remainder of the SLA lifetime. As our model is configured to behave strictly profit optimizing, it switches the system to a minimal configuration. In summary, this scenario shows the potential of our dynamic SLA management model. While the low cost service is operated with a weak and cost effective infrastructure, the high price service is operated with a more powerful configuration, which can sustain unforeseen workload peaks.

Table 1: Results of the dynamic SLA management model on four weeks of operation

Wikipedia Version	Revenue	Penalty	Alpha	Degree SLA Compliance	SLO Minimum	SLO Maximum	Average Resources	Profit	Successful SLA
German	$300	$-300	0.95	96.3%	73.6%	99.1%	6.07	$125.12	28
	$3000	$-3000	0.95	97.9%	737%	100%	8.69	$2749.73	28
	$300	$-300	0.80	93.5%	69.24%	100%	5.24	$149.09	28
	$300	$-300	static	97.7%	90.2%	100%	10	$12	28
English	$300	$-300	0.95	95.2%	76.0%	98.6%	8.52	$54.62	28
	$3000	$-3000	0.95	96.3%	76.1%	99.6%	9.74	$2719.49	28
	$300	$-300	0.80	91.8%	73.5%	97.8%	7.48	$84.58	28
	$300	$-300	static	96.7%	92.0%	99.6%	10	$12	28
Japanese	$300	$-300	0.95	95.9%	83.8%	100%	6.40	$115.67	28
	$3000	$-3000	0.95	97.4%	77.5%	100%	8.53	$143.61	28
	$300	$-300	0.80	92.8%	71.7%	99.6%	5.43	$2843.61	28
	$300	$-300	static	98.5%	100%	100%	10	$12	28

Table 1 shows a broad range of results of the dynamic SLA management model with different configurations on different workload traces. As the German, English and Japanese Workload strongly differ in their average workload levels, all processes have been normalized such that the maximum workload does not exceed 1000 requests per second. As benchmark, we included the results of a static operation. The cost for one cloud resource per hour was set to $1.20 and the delay time was set to 30 minutes. The first scenario is balanced between the economic parameters and the performance goal. The second scenario is a high revenue scenario and the third has a low performance target. The table presents the results of 28 days from the mid of February to the mid of March in 2011. Our model successfully fulfills the SLA in all scenarios in every run. In particular in the low cost scenarios, our model helps to significantly reduce the cost of operation. The lower SLO minimum originates from the switch to the lower bound control on fulfillment of the SLA. In the high revenue scenario, the SLA management model provisions significantly more resources to the system, as the costs of operation are minor. The saving on the English trace is significantly lower, as the system traces have a higher base load.

Conclusion

In this paper we presented a novel dynamic SLA management model for the sustainable and efficient operation of elastic information systems in cloud environments. Based on a system performance model and workload forecast model, our new management concept enables highly efficient operation modes. Our model extends the current state of the art by not only managing the system based on the QoS specifications of the SLA, but also according to economic parameters, such as the revenue, penalties, and

the cost of cloud resources. In contrast to (most) other SLA management concepts, our model does not necessarily aim to comply with the SLA at all times, but may instead choose a strategy that maximizes profit in the long-term. Furthermore, the dynamic character of our SLA model allows the flexible adaptation of performance goals at runtime, therefore mitigating the risk of performance violations. In summary our model bridges the gap between cloud technology and the economic value of a service. It provides a methodology to automatically manage service offers according to their value and is, therefore, particularly useful for services offered in different QoS classes with different price models.

Conceptually, our model is an effort with the aim of integrating all aspects of a Service Level Agreements (e.g., monitoring metrics and economic parameters) with runtime monitoring data. To accommodate the heterogeneity of enterprise information systems, our model is designed modularly, enabling different configurations according to the properties of the system. During operation, our model systematically processes and analyzes all factors of influence such as the performance and workload data as well as the current SLA state. In particular, for critical high-price services, our novel SLA model has proven to outperform other basic concepts. While basic controllers are conceptually able to provide any cost-effective operation mode, our model is able to adapt the operation strategy automatically based on the true economic value of the system. Thus, depending on the economic situation, it mitigates the risk of performance violations compared to rigorous cost-driven adaptive operation modes. We showed that the our model allows flexible information system operation with up to a 40 percent lower cost of operation compared to static operation modes and a significantly lower risk of SLA violations compared to other adaptive resource management systems.

We planned various extensions for our SLA management model. Recently, Amazon Web Services LLC introduced a new spot pricing scheme for cloud resources (Amazon 2011b). In our future work, we plan to extend our work to reflect these recent developments by supporting dynamic resource prices. More concretely, this enables service providers to operate the system in times with lower resource cost more risk-aware and take higher operational risks during peak time. Furthermore, we intend to extend our model to manage the resource requirements of multiple competitive systems. Based on the current SLA state of different services and their economic parameters, this extension should optimally allocate resources to the different services. For instance, if we have two services, each requiring 4 nodes, and 9 nodes in total, the management model should automatically assign the residual node to the service with the higher economic risk. In its current implementation the management model is risk neutral, aiming to maximize the expected profit. In our future work, we plan to integrate an individual risk function in order to let the operator self-select his risk-affinity level. Furthermore, the revenue and penalty parameters have only been arbitrarily specified. In our future work, we plan to use the dynamic management concept to estimate the expected cost of operation and the risk of an SLA violation and thus determine the optimal and risk adjusted price and penalty combination for a service.

References

Ardagna, D., Trubian, M. & Zhang, L., 2007. SLA based resource allocation policies in autonomic environments. *Journal of Parallel and Distributed Computing*, 67(3), 259-270. Available at: http://dx.doi.org/10.1016/j.jpdc.2006.10.006.

Aib, I. & Boutaba, R., 2007. On Leveraging Policy-Based Management for Maximizing Business Profit. *IEEE Transactions on Network and Service Management*, 4(3), 25-39. Available at: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4489642.

Amazon, 2011. Amazon Simple Queue Service. Available at: http://aws.amazon.com/sqs/.

Bailey, M., 2009. The Economics o f Virtualization: Moving Toward an Application-Based Cost Model. Available at: http://www.vmware.com/files/pdf/Virtualization-application-based-cost-model-WP-EN.pdf.

Buco, M.J. et al., 2004. Utility computing SLA management based upon business objectives. *IBM Systems Journal*, 43(1), 159-178. Available at: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5386770.

Buyya, R. et al., 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599-616. Available at: http://dx.doi.org/10.1016/j.future.2008.12.001.

Chandra, A., Gong, W. & Shenoy, P., 2003. Dynamic Resource Allocation for Shared Data Centers. *Quality of Service — IWQoS 2003*, Volume 270, 381-398. Available at: http://www.springerlink.com/content/h56r570l4u707466.

Cohen, I. et al., 2004. Correlating instrumentation data to system states: A building block for automated diagnosis and control. *In OSDI*.

Feitelson, D.G., 2011. *Workload Characterization and Modeling Book*, Available at: http://www.cs.huji.ac.il/~feit/wlmod/.

Gmach, D. et al., 2009. Resource pool management: Reactive versus proactive or let's be friends. *Computer Networks*, 53(17), 2905-2922. Available at: http://dx.doi.org/10.1016/j.comnet.2009.08.011.

Gmach, D. et al., 2007. Workload Analysis and Demand Prediction of Enterprise Data Center Applications. *2007 IEEE 10th International Symposium on Workload Characterization*, 171-180. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4362193.

Goolsby, K., 2007. How to rebound from a failed outsourcing relationship. *Outsourcing Journal*. Available at: www.outsourcing-journal.com/jan2007-rebound.html.

Hasselmeyer, P. et al., 2006. Towards Autonomous Brokered SLA Negotiation. *Exploiting the Knowledge Economy Issues Applications Case Studies*, 3. Available at: http://www.mendeley.com/research/towards-autonomous-brokered-sla-negotiation/.

Hedwig, M., Malkowski, S. & Neumann, D., 2010. TOWARDS AUTONOMIC COST-AWARE ALLOCATION OF CLOUD RESOURCES. *ICIS 2010 Proceedings*. Available at: http://aisel.aisnet.org/icis2010_submissions/180.

Hedwig, M., Malkowski, S. & Neumann, D., 2009. Taming Energy Costs of Large Enterprise Systems Through Adaptive Provisioning. In *ICIS '09: Proceedings of the Eight IEEE/ACIS International Conference on Computer and Information Science*. Phoenix, AZ, USA: IEEE Computer Society.

Koller, B. & Schubert, L., 2007. Towards autonomous SLA management using a proxy-like approach. *International Journal of Multiagent and Grid Systems*, 3(3), 313-325. Available at: http://www.mendeley.com/research/towards-autonomous-sla-management-using-a-proxylike-approach/.

Koomey, J.G., 2007. ESTIMATING TOTAL POWER CONSUMPTION BY SERVERS IN THE U . S . AND THE WORLD. *World*.

Lassettre, E. et al., 2003. *Dynamic Surge Protection: An Approach to Handling Unexpected Workload Surges with Resource Actions that Have Lead Times*,

Lim, H.C., Babu, S. & Chase, J.S., 2010. Automated control for elastic storage. In *Proceeding of the 7th international conference on Autonomic computing*. ICAC '10. New York, NY, USA: ACM, pp. 1-10. Available at: http://doi.acm.org/10.1145/1809049.1809051.

Malkowski, S. et al., 2010. CloudXplor: a tool for configuration planning in clouds based on empirical data. *Symposium on Applied Computing*, 391-398. Available at: http://portal.acm.org/citation.cfm?id=1774172.

Malkowski, S.J. et al., 2011. Automated control for elastic n-tier workloads based on empirical modeling. In *Proceedings of the 8th ACM international conference on Autonomic computing - ICAC '11*. New York, New York, USA: ACM Press, p. 131. Available at: http://dl.acm.org/citation.cfm?id=1998582.1998604.

Mituzas, D., 2011. Wikistats. Available at: http://dammit.lt/wikistats/.

Padala, P. et al., 2009. Automated control of multiple virtualized resources. *Proceedings of the fourth ACM european conference on Computer systems - EuroSys '09*, 13. Available at: http://portal.acm.org/citation.cfm?doid=1519065.1519068.

Pettey, C. & Tudor, B., 2010. Gartner Says Energy-Related Costs Account for Approximately 12 Percent of Overall Data Center Expenditures. Available at: http://www.gartner.com/it/page.jsp?id=1442113.

Raimondi, F., Skene, J. & Emmerich, W., 2008. *Efficient online monitoring of web-service SLAs*, New York, New York, USA: ACM Press. Available at: http://dl.acm.org/citation.cfm?id=1453101.1453125.

RightScale, 2011. RightScale. Available at: http://www.rightscale.com/.

Sahai, A., Durante, A. & Machiraju, V., 2001. *Towards Automated SLA Management for Web Services*, Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.7978.

Sahni, M. & Tan, L.F., 2011. APEJ CIOs will be the Rising Stars in the Corporate Hierarchy as they Drive ROI-Led Transformation Initiatives in 2011, says IDC. Available at: http://www.idc.com/AP/pressrelease.jsp?containerId=prSG22698011.

Schulz, G., 2009. *The Green and Virtual Data Center*, Boston, MA, USA: Auerbach Publications.

Short, J.E., Bohn, R.E. & Baru, C., 2011. How Much Information - Report on Enterprise Server Information. Available at: How Much Information? 2010.

Urgaonkar, B. et al., 2008. Agile dynamic provisioning of multi-tier Internet applications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(1). Available at: http://portal.acm.org/citation.cfm?id=1342172.

Velte, T., Velte, A. & Elsenpeter, R.C., 2009. *Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line*, New York, NY, USA: McGraw-Hill, Inc

Yeo, C.S. & Buyya, R., 2007. *Integrated Risk Analysis for a Commercial Computing Service*, IEEE. Available at: http://www.computer.org/portal/web/csdl/doi/10.1109/IPDPS.2007.370241.