Providing Dependability and Resilience in the Cloud: Challenges and Opportunities

Samuel Kounev[1], Philipp Reinecke[2], Fabian Brosig[1], Jeremy T. Bradley[4], Kaustubh Joshi[3], Vlastimil Babka[5], Anton Stefanek[4], and Stephen Gilmore[6]

[1] Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
{kounev,fabian.brosig}@kit.edu
[2] Freie Universitaet Berlin, Germany
philipp.reinecke@fu-berlin.de
[3] AT&T Labs Research, NJ, USA
kaustubh@research.att.com
[4] Imperial College London, UK
{jb,as1005}@doc.ic.ac.uk
[5] Charles University Prague, Czech Republic
babka@d3s.mff.cuni.cz
[6] University of Edinburgh, UK
Stephen.Gilmore@ed.ac.uk

Abstract Cloud Computing is a novel paradigm for providing data center resources as on demand services in a pay-as-you-go manner. It promises significant cost savings by making it possible to consolidate workloads and share infrastructure resources among multiple applications resulting in higher cost- and energy-efficiency. However, these benefits come at the cost of increased system complexity and dynamicity posing new challenges in providing service dependability and resilience for applications running in a Cloud environment. At the same time, the virtualization of physical resources, inherent in Cloud Computing, provides new opportunities for novel dependability and quality-of-service management techniques that can potentially improve system resilience. In this chapter, we first discuss in detail the challenges and opportunities introduced by the Cloud Computing paradigm. We then provide a review of the state-of-the-art on dependability and resilience management in Cloud environments, and conclude with an overview of emerging research directions.

1 Introduction

In today's data centers, IT services and applications are typically hosted on dedicated hardware in order to provide dependability guarantees. Server capacity is typically over-dimensioned to ensure adequate Quality-of-Service (QoS) under variable workloads and load fluctuations. The use of dedicated hardware with over-dimensioned capacity not only leads to poor resource efficiency, but also makes it hard to react to changes and conflicting demands in operating conditions, business processes or use practices. Moreover, the adoption of new applications and the increasing demand for IT services leads to an exponential growth in the number of servers and the required network infrastructure. Servers in data centers nowadays are estimated to have average utilization ranging from 5% to 20% [90, 94] which corresponds to their lowest energy-efficiency region [16]. The growing number of underutilized servers, often referred to as "server sprawl", translates into increasing data center operating costs including system management costs and power consumption costs of the server, network and cooling infrastructure. According to a study at Lawrence Berkley National Labs (2007), power consumption in data centers doubled from 2000 to 2005, and in 2006, the USA alone spent an estimated 61 TW-h in data centers. By 2025, power consumption in data centers is projected to grow by 1600% and energy will become the major factor in the Total-Cost-of-Ownership (TCO) for IT [7]. Already today, according to IDC, over 40% of data center customers report power demand outstripping supply, while cooling capacities at their threshold have become a limiting factor in deploying new systems [92]. In addition to driving costs up, the rising energy consumption of the ICT sector will have a significant impact on the global CO2 emissions. While today, ICT accounts for 2% to 4% of the global CO2 emissions, it is projected to reach 10% in 5-10 years [53]. Thus, reducing the costs of ICT and their environmental footprint while keeping a high growth rate of IT services is one of today's greatest challenges for society.

Driven by the pressure to improve energy efficiency and reduce data center operating costs, industry is looking towards Cloud Computing which is a novel paradigm for providing data center resources (computing, network and storage) as on-demand services over a private or public network in a pay-as-you-go manner. Cloud Computing is normally considered at three different levels: i) Infrastructure-as-a-Service (IaaS) where raw compute, storage, and network resources are provided, ii) Platform-as-a-Service (PaaS) where an application environment on top of the bare bones infrastructure is provided, iii) and Software-as-a-Service (SaaS) where a working application is provided (e.g., NetSuite and SalesForce.com). Cloud Computing makes it possible for enterprises to consolidate their IT resources internally or to completely outsource their IT infrastructure taking advantage of the economies of scale of a shared infrastructure. In both cases, some substantial reductions in the TCO for IT can be achieved. Virtualization plays a key role in this process since it makes it possible to significantly reduce the number of servers in data centers by having each server host multiple independent virtual machines (VMs) managed by a Virtual Machine Monitor (VMM) often referred to as a Hypervisor. By enabling the consolidation of multiple applications on a smaller number of physical servers, virtualization promises significant cost savings resulting from higher energy efficiency and lower system management costs. Moreover, virtualization facilitates system evolution by enabling adaptability and scalability of service infrastructures.

With investments of billions of dollars, the fortunes of dozens of companies, and major research initiatives staked on its success, it is clear that Cloud Computing is here to stay. Cloud-based infrastructures are rapidly becoming a destination of choice to host a variety of applications ranging from high-availability enterprise services and online TV stations, to batch-oriented scientific computations. However, it is not yet clear whether Cloud services can be a dependable alternative to dedicated infrastructure. In the context of this chapter, we consider a system to be dependable if it is able to provide availability, responsiveness and reliability in the presence of hardware or software failures. In contrast, resilience of a system is considered as the system's ability to continue providing dependable services under external perturbations such as security attacks, accidents, unexpected load spikes or fault-loads. The remainder of this chapter explores this question and is organized as follows. Section 2 describes the challenges and opportunities in providing dependability and resilience in the Cloud. In Section 3, we review the state-of-the-art on dependability, performance and security management in Cloud infrastructures. An overview of the emerging research directions in Cloud Computing is provided in Section 4.

2 Challenges and Opportunities

The increased complexity and dynamicity induced by Cloud Computing pose new challenges and opportunities in providing service dependability and resilience. On one hand, availability and privacy are serious challenges for applications hosted on Cloud infrastructure. On the other hand, a Cloud provider's economies of scale allow levels of investment in redundancy and dependability that are difficult to match for smaller operators. Furthermore, the ability to monitor large numbers of applications and users can enable 'wisdom of crowds' approaches to provide enhanced security much in the same way that network providers have been able to do with worms and DDoS attacks.

2.1 Challenges

In spite of the many benefits Cloud Computing promises, today, the lack of trust in shared virtualized infrastructures is a major showstopper for its widespread adoption. According to [8], 74% of technological and financial decision makers in the UK would not put mission-critical applications in the Cloud. Service unavailability, performance unpredictability, and security risks are frequently cited as major reasons for the lack of trust [5, 70]. Some recent stress tests conducted by Sydney-based researchers revealed that the infrastructure-on-demand services offered by Amazon, Google and Microsoft suffer from regular performance and availability issues [115]. Response times of services varied by a factor of twenty depending on the time of day the services were accessed. According to [70], concerns of organizations about service availability is the number one obstacle to the adoption of Cloud Computing. Service overload, hardware failures and software errors are among the most common causes of service unavailability as experience with Google's AppEngine, GMail and Amazon's S3 service shows [6, 114, 115].

The lack of trust in shared virtualized infrastructures is a major impediment which applies both to public and private Clouds. Indeed, virtualization comes at the cost of increased system complexity and dynamicity. The increased dynamicity is caused by the introduction of virtual resources and the lack of direct control over the underlying physical hardware. The increased complexity is caused by the complex interactions between the applications and workloads sharing the physical infrastructure. The inability to predict such interactions and adapt the system accordingly makes it hard to provide dependability guarantees in terms of availability and responsiveness as well as resilience to external perturbations such as security attacks. Thus, virtualization introduces new sources of failure and threats degrading the dependability and trustworthiness of Cloud Computing infrastructures. Service providers are faced with the following challenges:

– How much resources (e.g., CPUs, main memory, storage capacity, network bandwidth) should be allocated to a new application deployed in the Cloud infrastructure and how should the application be configured to satisfy its requirements for dependability (availability and reliability) and responsiveness avoiding the pitfalls of underprovisioning or overprovisioning resources?
– How much and at what rate and granularity (e.g., CPU cycles, cluster nodes) should resources be added or removed proactively to avoid Service Level Agreement(SLA) violations or inefficient resource usage due to varying customer workloads and load fluctuations?

Moreover, the consolidation of workloads translates into higher utilization of physical resources which makes the system much more vulnerable to threats resulting from unforeseen load fluctuations, hardware and software failures, and network attacks. The Cloud provider is faced with the challenge of how to efficiently share physical resources among hosted applications in the face of highly variable and unpredictable resource demands as well as operational failures.

An environment with a few large Cloud infrastructure providers not only increases the risk of common mode outages affecting a large number of applications, but also provides highly visible targets for attackers. Community-driven sites such as [2] track outages in major Cloud providers and have documented a number of outages and security vulnerabilities over the last two years affecting hundreds of Internet sites.

Sharing of Cloud resources by entities that engage in a wide range of behaviors and employ best practices to varying degrees can expose Cloud applications to increased risk levels. For example, on April 26 2008, Amazon's Elastic Cloud (EC2) had an outage [1] across several instances due to a single customer applying a very large set of unusual firewall rules and instantiating a large number of instances at the same time, thereby triggering a performance degradation bug in Amazon's distributed firewall.

Multiple administrative domains between the application and infrastructure operators reduces end-to-end system visibility and error propagation information, thus making problem detection and diagnosis very difficult. Additionally, for competitive reasons, Cloud infrastructure providers may not provide full disclosure regarding the cause of outages or other detailed infrastructure design information, raising the question of the verifiability of claims regarding dependability.

The hosting of data on outsourced and shared infrastructure that may be in a different legal jurisdiction than the owner of the data has serious legal and privacy implications. Corporate accountability legislation such as the Sarbanes-Oxley Act (SOX) of 2002 and privacy clauses included in legislation such as the Health Insurance Portability and Accountability Act (HIPAA) of 1996 and the Telecommunications Act of 1996 create obstacles to the applicability of Cloud solutions in the financial, healthcare, and telecom industries. For example, BusinessWeek reported in Aug 2008 [57] that ITricity, a European provider of Cloud computing capacity, could not offer services to such companies until it began offering owner-hosted private Cloud services. The recently formed industrial consortium called the Cloud Security Alliance [3] includes in its charter several issues regarding the interplay of Cloud Computing and legal requirements.

2.2 Opportunities

Cloud computing enables economies of scale leading to large redundancy levels and wide geographical footprints. For example, Amazon's EC2 currently supports two regions in the US and Europe, each split into independent 'availability zones', while AT&T's Synaptic Cloud computing offering provides five 'super IDCs' located across the world. These can be leveraged through techniques such as virtual machine migration and cloning to provide better fault tolerance and disaster recovery, especially for operators of smaller applications that may not have been able to afford such capabilities.

New security and reliability services can be enabled or strengthened by virtue of being located in the Cloud. For example, popular cloud-based email services such as GMail amplify manual feedback from some users to provide automatic spam filtering for all users. Oberheide et. al. describe in [85] a Cloud-based anti-virus solution that can not only utilize multiple vendors to provide better coverage, but also compares data blocks across users to improve efficiency and provides an archival service for forensic analysis.

Managed Cloud services that include OS level support can result in improved reliability and security due to consistent centralized administration and timely application of patches and upgrades.

3 State-of-the-Art Review

In this section, we provide an overview of the state-of-the-art in resilience for Cloud Computing. We start with a discussion of resilience assessment techniques and then survey methods for managing resilience. The approaches in the first section cover the issue of how Cloud system resilience could be assessed, while the second section comprises methods that are used to manage the system with the goal of improving resilience.

3.1 Approaches for Resilience Assessment

Dependability, performance and security can be evaluated using measurements on real deployments, measurements on test-beds, simulations, and analysis of models. While a lot of work exists in these areas, approaches specifically targeting Cloud systems are still rare.

Measurement studies on real Cloud systems are undertaken in order to understand the effect of the Cloud on the application. In [38], resilience of an Infrastructure-of-a-service (IaaS) cloud is quantified as job rejection rate and response delay in situations where the cloud is subjected to changes in demand and available capacity. However, most existing studies focus on performance. The typical approach in such studies is to generate a workload and measure various performance indicators. The tools used and the indicators of interest depend on the application that the authors focus on. [31, 48, 82, 86, 111] serve as examples for evaluation of the performance of applications for scientific computing. These studies employ tools that generate a typical High Performance Computing (HPC) workload, and measure run-times. In [86], the authors also evaluate the time required for allocating and releasing virtual machines. Since flexible resource allocation is a major selling-point of Cloud systems, this aspect should not be ignored when considering overall performance. Furthermore, [86] studies the performance of disk I/O operations performed on virtualized disks. The findings in [29,31,48] show that applications running on Cloud systems have run-times that are longer and exhibit more variance than applications running on native systems. The authors of [86], however, state that extensive caching in a Cloud system may result in significantly faster disk I/O operations, compared to a native system. On the other hand, [86] also shows that a quick performance drop occurs once the cache size is exceeded.

Experimentation on test-beds is seldom performed for Cloud systems, since the complexity and costs of setting up a Cloud environment of realistic size become prohibitively large. Existing approaches thus tend to focus on special aspects of Cloud systems, in particular, specific programming models and virtualization technology. In [44], the performance of an application using MapReduce is evaluated on a small cluster of physical machines, each of which runs several virtual machines. The authors point out that performance may suffer from virtual machines competing for the physical I/O resources. As virtualization is a key component in Cloud Computing, its impact on resilience must be understood. Virtualization is rather amenable to experiments in test-beds. Existing work [42,43,73,96] focusses on studying the performance impact of configuration options and workloads through experiments with benchmarks and standard performance measurement tools. Performance indicators are typically throughput and benchmark-specific aggregated metrics.

Benchmarks for virtualized server consolidation, i.e., benchmarks measuring aggregated server performance when physical resources are virtualized and shared, include vConsolidate [14], VMMark [41] and recently SPECvirt_sc2010 [98]. Benchmarks for virtualized servers are still a subject of discussion, as there is a lack of consensus for a metric describing consolidated server performance [20,36].

The authors of [36] propose new metrics taking into account per-VM performance along with total system throughput. The authors of [20] emphasize that particularly for benchmarking database performance, the consolidation of resource-intensive workloads is of crucial importance. None of the virtual benchmarks available today measure database-centric properties adequately [20].

Unlike virtualized server consolidation, Cloud Computing lacks well-established benchmark suites [39], although benchmarks such as TeraSort, Cloudstone or MalStone exist, and traditional high-performance computing benchmarks have been used (e.g., [31,48,86]). In [18], it is argued that the established TPC-W benchmark [97] is not appropriate for Cloud Computing, because Cloud scalability invalidates its metrics, TPC-W relies on database properties often not supported in the Cloud, and because TPC-W does not provide metrics for important Cloud properties such as scalability, pay-per-use pricing, and fault-tolerance. The authors of [18] propose desirable properties of a Cloud benchmark; similarly, [24] proposes a benchmarking framework specific to Cloud data serving systems.

Several simulation approaches for Cloud systems have been proposed. These methods differ in whether they focus on special applications or allow simulation of Cloud systems in general. The simulation framework MRPerf [112] instruments the discrete-event network simulator NS-2 [107] for studying performance and dependability of MapReduce [28]. The framework models node, network, and disk behavior in high detail and thus allows evaluating the impact of network topology choices and node/network failures, but is limited to applications that use MapReduce. In contrast, the CloudSim toolkit [22] is a discrete-event simulation toolkit for general Cloud systems. The toolkit models, among other aspects, virtual machines and VM scheduling, storage, network, and computing resources.

Analytical approaches for the evaluation of dependability and performance of Cloud systems usually focus on the impact of virtualization.

Reliability block diagrams to model system reliability at the host level have been proposed in [89]. These models do not consider the behavior of the underlying hardware and software components. More detailed models based on CTMCs are presented in [103], but these models still only capture behavior at the VM level. The two-level hierarchical approach in [56] uses fault-trees in the upper level and CTMCs in the lower level in order to capture software failures at the VMM, VM, and application level as well as hardware failures. Finally, combinatorial modelling to analyse design choices with a single physical server hosting multiple VMs was proposed in [89].

Virtualized resources shared between VM instances have a non-trivial impact on performance. Due to this overhead, traditional design-time model-based approaches, as surveyed in e.g., [15,62] may yield imprecise results when used as-is. A typical approach (e.g., [17,75]) is to construct traditional queueing network models and apply a slowdown factor to capture the effects of virtualization. Another approach is applied in [64], where artificial neural networks are used to predict performance of virtualized applications from a set of observable or controllable parameters related to CPU, memory, disk and network usage.

Prediction of resource utilization is required for dimensioning and workload placement decisions. The work in [116] focusses on predicting CPU utilization of both a VM and the Dom-0 (which hosts the network and disk drivers). The prediction model is automatically derived from a set of microbenchmarks consisting of synthetic CPU, network and disk workloads, using a robust stepwise linear regression between several metrics obtained in native and virtualized microbenchmark executions. Further parameterizations of the model are based on measurements of the application executed natively. Simple models for core utilisation and effective shared space allocation are developed in [46, 104]. The authors also note that for some shared resources (such as the cache space), online measurement and modeling is not possible today, due to a lack of appropriate performance counters.

As has long been accepted for dependability, complex systems can never be perfectly secure. Therefore, only quantitative measures allow comparisons between systems with respect to their security. While quantification of security has long been recognised as an important problem [66,67] and several approaches have been made in recent years [33, 45, 49, 65, 71, 84, 95], the area is still underexplored and subject to dispute [108]. Still, various security metrics have been proposed [12,33,49,72,101], and experimental studies have been performed [45].

For security evaluation of Cloud systems, even less work exists. In fact, quantitative security evaluation of Cloud systems is still in its infancy. The analytical approach by [91] exemplifies some of the difficulties in quantitative security assessment. In this approach, risk is computed as a weighted sum of the impact of a security incident and its probability. Both incident probabilities and impacts, however, are hard to measure. While the authors of [91] argue that probabilities can be obtained from published incidence reports and impacts can be estimated based on expert opinions, such data may be invalid due to biased report and subjective opinions. Furthermore, taking the weighted sum assumes that security is a static property, whereas it seems likely that the probability of security incidents and their impact changes over time, as both the system, the attacker, and the value of the system to the user evolve.

3.2 Approaches for Managing Dependability and Performance

There are many research challenges with respect to managing dependability and performance in Cloud systems (see Section 3). On the one hand, virtualization provides opportunities to improve these properties, on the other hand, Cloud Computing poses a complex resource allocation problem.

Virtualization for improving dependability and performance Techniques that take advantage of virtualization to improve system dependability have been the focus of recent research [25, 32, 61, 79, 102, 103]. Two lines of research can be distinguished: i) virtualization-based software rejuvenation and ii) using VM replication as a basis for failure recovery.

Software rejuvenation is a proactive fault management technique aimed at cleaning up the system's internal state to prevent occurrence of severe failures due to the phenomena of software aging or caused by transient failures [105]. A detailed introduction to rejuvenation is given in Chapter **??**. The approach has been applied to Cloud Computing and virtualization. In [102], a technique that can increase availability of application servers through the use of virtualization, clustering and software rejuvenation is presented. Analytical models are used to analyze multiple design choices when a single physical server and dual physical servers are used to host multiple VMs. It is shown that by integrating virtualization, clustering and software rejuvenation, it is possible to benefit from increased availability, manageability and savings from server consolidation through virtualization without decreasing uptime of critical services. A similar approach based on automated self-healing techniques claimed to induce zero downtime for most of the cases is proposed in [79]. Software aging and transient failures are detected through continuous monitoring of system data and performability metrics of the application server. A further virtualization-based rejuvenation technique for application servers using stochastic models was proposed in [103]. The authors present a stochastic model of a single physical server used to host multiple virtual machines (VMs) configured with the proposed technique. The model is intended as a general model capturing the application server characteristics, failure behavior, and performability measures. Finally, in [61], the authors present a technique called warm-VM reboot for fast rejuvenation of VMMs that enables efficiently rebooting only a VMM by suspending and resuming VMs without accessing the memory images. The technique is based on two mechanisms, on-memory suspend/resume of VMs and quick reload of VMMs. The technique is claimed to reduce downtime and prevent the performance degradation due to cache misses after the reboot. In [105], stochastic models that help to detect software aging and determine optimal times to perform rejuvenation are proposed. Models are constructed using workload and resource usage data collected from the UNIX operating system over a period of time. The measurement-based models are intended to help development of strategies for software rejuvenation triggered by actual measurements.

Accounting for failures by dynamically creating replicas is a common strategy to improve overall dependability. For instance, [87] uses regeneration of new data objects to account for reduction in redundancy and the Google File System [37] similarly creates new file "chunks" when the number of available copies is reduced below a threshold. Even commercial tools such as VMWare High Availability (HA) [110] allow a virtual machine on a failed host to be reinstantiated on a new machine. However, the placement of replicas becomes especially challenging when they are components in a multitier application. Recent work on performance optimization of multitier applications (e.g., [26,51,80,106]) address the performance impact of resource allocation on such multitier applications, but does not combine performance modeling with availability requirements and dynamic regeneration of failed components. The tradeoff between availability and performance is always present in dependability research since increasing availability (by using more redundancy) typically increases response time. Examples of work that explicitly address this issue include [27] and [93], both of which consider the problem of when to invoke a (human) repair process to optimize various metrics of cost and availability defined on the system. In both cases, the "optimal policies" that specify when the repair was to be invoked (as a function of system state) were computed off-line through solution of Markov Decision process models of the system.

As far as failure recovery mechanisms are concerned, in [32], the authors introduce an extensible grammar that classifies the states and transitions of VM images and can be used to create rules for recovery and high availability exploiting virtualization for simplified fault tolerance. In [25], a fail-over technique based on asynchronous VM replication is proposed that asynchronously propagates changed state to a backup host at frequencies as high as forty times a second, and uses speculative execution to concurrently run the active VM slightly ahead of the replicated system state. In case of a failure, automatic fail-over with only seconds of downtime is provided while preserving host state such as active network connections. Finally, in [81], a proactive fault tolerance technique for Message Passing Interface(MPI) applications is presented exploiting Xen's live migration mechanism to migrate an MPI task from a health-deteriorating node to a healthy one without stopping the MPI task during most of the migration. Experimental results demonstrate that live migration hides migration costs and limits the overhead to only a few seconds. Some further general approaches for leveraging virtualization to improve system dependability are surveyed in [69,89]. In [78], a high-level approach for autonomic management of the system availability including real-time evaluation, monitoring and management is sketched. The authors suggest using analytical models parameterized using monitoring data collected during operation. The approach, however, is targeted at static system architectures and assumes that the underlying availability models are built manually at system design time.

Self-adaptive capacity and power management in virtualized data centers including trade-offs. We first describe general approaches to a self-adaptive capacity and power management. Afterwards, approaches specifically targeted at virtualized environments are reviewed.

A number of self-adaptive approaches have been proposed that automatically adapt resource allocations in response to changes in application workloads in a way that utility is maximized. Existing work mostly focuses on performance as QoS property and utility functions are based on assigning rewards for satisfied SLAs and penalties for violated SLAs, e.g., [10, 26, 68, 76]. In recent years, given the rising cost of energy, capacity management strategies aiming at improving the power usage effectiveness have received increasing attention, e.g., [23,50,109].

Existing approaches to self-adaptive capacity management are typically based on: i) control theory feedback loops, ii) machine learning techniques or iii) general utility-based optimization techniques. Approaches based on feedback loops and control theory, e.g., [9,13], can normally guarantee system stability by capturing the transient system behavior [13]. Machine learning techniques, without a need for an a priori analytical model of the system, base their learning sessions on live systems. Such techniques have been used to tackle resource allocation problems [100] as well as the coordination of multiple autonomic managers [47]. In utility-based approaches, the system is typically modeled by means of a performance model embedded within an optimization framework aiming at optimizing multiple criteria such as different QoS metrics [50, 77, 109].

Utility-based optimization frameworks differ in the way in which they trigger adaptations. There are reactive and proactive approaches. The former react on certain events observed in the system, the latter try to anticipate the future system behavior and thus require forecasting mechanisms. For workload forecasting, established time series analysis techniques [21] are used, e.g., Brown's quadratic exponential smoothing or general AutoRegressive - Moving Average (ARMA) models have been implemented in [77] and [23, 52], respectively. Regarding performance modeling, existing work mainly uses predictive performance models that capture the temporal system behavior (e.g., queueing networks) where the platform is normally abstracted as a "black-box" (e.g., [10,23,80,106,119]). Applications are modeled by a single queue with a single workload class [23] or multiple workload classes [80]. In [119], multi-tier applications are modeled using queueing networks where one queue represents one tier. All these models are solved analytically, e.g., in the latter case based on mean-value analysis (MVA). In [51], layered queueing models (LQNs) are solved by means of simulation. A different approach uses fuzzy-logic models to model the resource needs of an application for a given workload intensity [117]. The fuzzy-logic models need to be trained under dynamically changing workloads.

Resource allocation problems have been studied in the literature, frequently using techniques including bin packing (e.g., [19, 50]), multiple knapsack problems, and multi-dimensional knapsack problems [55]. For dynamic resource allocation applications, previous studies address this problem using linear optimization techniques [54] or non-linear optimization strategies based on simulated annealing [113], fuzzy logic [117], or other heuristics [11]. There are approaches to formulate the optimization problem as a network flow problem [68], to solve it with genetic algorithms [77], or to automatically change deployments using profiles capturing experts' knowledge of scaling different types of applications [118]. The above studies differ in the objective of the optimization and the type of applications on which they focus.

In virtualized environments, due to the introduction of virtual resources, the resource allocation problem is more complex. The studies in [74, 88] validate a performance inference in virtualized environments. There are strategies that explicitly make use of VMM configurations. For instance, the authors of [76,83] propose to exploit the min, max and shares parameters (respectively CPU priorities) for VM placement and power consolidation in data centers. In [35], the power-to-frequency relationship of dynamic voltage and frequency techniques is leveraged to distribute available power among the servers in order to get maximum performance. Some recent work on capacity management in Cloud infrastructures, based on LQN models, considers both performance and power as well as adaptation costs [50, 51]. To estimate the power consumption, utilization-based models from previous studies [63] are used. The following adaptation actions are considered: adapt a VM's CPU capacity, add/remove a VM, live-migrate a VM between hosts, and shut down/restart a host [50]. For the optimization there are two algorithms: a bin packing algorithm optimizing the power/performance tradeoff and an A* graph search algorithm that takes adaptation costs as well as search costs into account. The case study shows promising results, however, it is based on a simple multi-tier application with read-only transactions and a fixed web tier.

4 Emerging Research Directions

In this section, we outline emerging research directions targeting resilience and dependability management in Cloud infrastructures. At first, we discuss the question how the flexible allocation mechanisms available in virtualized environments can be used to tackle scalability and consolidation issues. Afterwards, we capture the research challenge of finding representative predictive models and model parameters. Finally, we examine the trade-off decisions between performance and energy consumption and highlight the need for self-aware management techniques that enable a continuous application of management activities during system operation.

4.1 A Question of Scale

By 2015, it is predicted that more than 75% of computer infrastructure will be purchased from virtualized service providers [34]. Such services are hosted in Cloud environments with computation and network resources multiplexed between many distinct services. Although functionally, services may not impact each other, there is good evidence to suggest that performance stress from one virtual machine can indeed be noticed by another virtual machine instance [30].

Cloud administrators, like software developers, are increasingly responsible for the reliable and performance-driven provision of these software and hardware services. They face difficult quantitative scalability questions, often focused around service-level response-time goals. Being able to create accurate predictive models of such services is a major challenge in performance engineering and stochastic analysis.

Clearly servers could be over-provisioned in an effort to obtain high throughput, availability or resilience for all services. However, this is not a viable solution. The economics of virtualized service provision dictate that a sufficient level of shared or multiplexed computation is in fact a requirement. The energy consumed for unnecessary servers and extra air-conditioning will render a policy of server over-provisioning unsustainable financially, even if in doing so it was able to satisfy a strict service level requirement.

This is one of the major challenges facing Cloud Computing, how can many services be multiplexed in a virtualized environment and guarantee service level agreements imposed upon those services while minimizing the energy costs and maximizing the revenue of the overall cloud environment.

4.2 Parameter Sweeping

Here are some examples of the quantitative scalability questions and requirements that a virtualized environment might face. Maintaining a predictive model of a Cloud environment will mean both sustaining an accurate behavioral model of the services and virtualized architecture but also addressing the key scalability and configuration issues, for example:

- How many servers does a Cloud cluster need in order to execute 4000 jobs every minute at least 95% of the time?
- Under the predicted traffic profile, at what rate can a Cloud environment hibernate its servers to save energy, given the time penalty involved in power-cycling a host and relocating virtual instances?
- How many virtual machines can be launched on a host (for the same/different service) while maintaining a service level requirement of 96.7% of service requests actioned within 0.88 seconds?

These are all examples of performance evaluation questions where the result is contingent on specific model parameters. Potentially, small fluctuations in a set of key parameters in the model will have an enormous effect on the overall performance and even functional behavior of the whole system. Discerning which parameters have the most effect on a given performance goal is a question of sensitivity analysis and can be a highly computationally intensive task even for small models.

Where such questions are not asked, or not rigorously answered, the consequences are very familiar. Systems are delivered which fail to win the trust of users because their performance is too unpredictable. Those systems which do deliver the required level of service often have excessively high running costs because their architects over-provisioned the hardware requirements in an attempt to mask failings due to uncertain software performance. There is a growing understanding that the running costs of a system greatly outweigh the development costs and that it is false economy to buy more hardware to cut software costs.

For these reasons, precise query-driven performance evaluation of computer systems and specifically virtualized computer systems is an important practical concern. In the next section will highlight some of these energy-computation tradeoffs in the context of a simple multi-client, multi-server environment. Achieving this for a more complex Cloud environment with many possible services will require a step change in modelling and analysis approaches.

4.3 Trade-off between Energy Consumption and Performance

We demonstrate the sort of energy/performance trade-off on a simple massively parallel client–server system. It serves to demonstrate the synergy of several critical issues that will need to be considered in a more complex model of a Cloud environment: scalability analysis via parameter sweeping, energy modelling and server hibernation.

The model consists of a large number of clients and a large number of servers cooperating together. The clients access the servers in two stages: first the client requests some data of the server and then the client receives the data from the server in response; the client goes on to process this data individually before restarting. The servers, in addition to serving clients, can hibernate to save energy and can also break. Broken servers are repaired. The details of this stochastic model and analysis can be found in Stefanek et al. [99]. A reward architecture is deployed to keep track of energy consumption and a fluid analysis technique [40] is used to calculate a service level agreement.

In this client/server model, we might be interested in the optimal number of servers that have to be employed in order to guarantee given performance requirements while minimizing the associated running costs. The performance requirements are often given in terms of a *Service Level Agreement* (SLA) for each client. In the context of this model, a suitable SLA might require that a client finishes its first request cycle within a given time period with a given high probability, for example within time 4.0 seconds with probability at least 0.9. Considering only the configurations that satisfy such an SLA, the *feasible configurations*, we can look for those that minimize the energy expended over the operation of the system.

Figure 1 is generated by the Grouped PEPA Analyzer (GPA) tool [4] and shows an example where we vary the number of servers and the rate with which they are hibernated. For each configuration we calculate the energy used and plot a point on the surface only if that configuration satisfies the SLA requirement mentioned above. We are able to find the configuration (84 servers and a hibernation rate of 0.37) which minimizes the energy consumption in the system. Intuitively, increasing the number of servers and decreasing the hibernation rate increases the probability of a client finishing early, but also raises the energy cost of running the system. Although, at this stage we are not capturing issues such as virtualization, multiple services or server classes in the model, this example illustrates the power of predictive modeling in being able to identify so-called sweet spots in operation.

4.4 Self-Aware Systems

As discussed in the previous sections, managing system resources in Cloud environments to ensure acceptable end-to-end application QoS and efficient resource utilization is a challenge. Modern enterprise software systems have highly distributed architectures composed of loosely-coupled services that operate and evolve independently, and are subjected to time-varying workloads.

The presented challenges call for novel systems engineering methodologies enabling the engineering of so-called *self-aware software systems* [58, 60]. The latter should have built-in online QoS prediction and self-adaptation capabilities

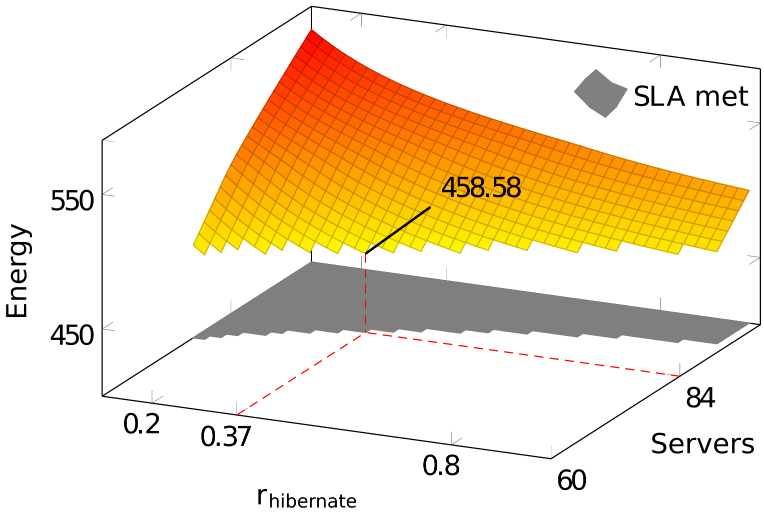

Figure 1. Global optimization of the energy consumption of the server components from [99]. Only configurations satisfying the SLA are shown in colour.

used to enforce QoS requirements in a cost- and energy-efficient manner. Self-awareness in this context is defined by the combination of three properties that systems should possess:

– *Self-reflective*: aware of their software architecture, execution platform and the hardware infrastructure on which they are running as well as of dynamic changes that occur during operation,
– *Self-predictive*: able to predict the effect of dynamic changes (e.g., changing user workloads) as well as predict the effect of possible adaptation actions,
– *Self-adaptive*: proactively adapting as the environment evolves in order to ensure that their non-functional requirements (e.g., availability, performance and reliability) and respective SLAs are continuously satisfied in a cost- and energy-efficient manner.

Self-aware systems engineering is a newly emerging research area at the intersection of several computer science disciplines including software architecture, computer systems modeling, autonomic computing, distributed systems, and more recently, Cloud Computing and Green IT [59].

5 Conclusion

We provided an overview of the research challenges and opportunities in providing dependability and resilience in Cloud Computing environments. State-of-the-art approaches for resilience assessment and for managing dependability, performance and security were presented, including approaches to self-adaptive capacity and power management in virtualized data centers. The identification of the existing gaps led to an overview of the emerging research directions. It is still an open question, how a set of services should be multiplexed in a virtualized environment while SLAs are guaranteed in such a way that the revenue of the overall Cloud environment is maximized. In particular, modeling the trade-offs between energy consumption/costs and application QoS remains a challenge.

6 Acknowledgements

The work of the first author was funded by the German Research Foundation (DFG) under grant No. KO 3445/6-1. Jeremy Bradley and Anton Stefanek are supported by the UK Engineering and Physical Sciences Research Council on the AMPS project (reference EP/G011737/1).

References

1. Amazon web services discussion forums. `http://developer.amazonwebservices.com/connect/thread.jspa?threadID=21401&tstart=15`. Retrieved in June 2009.
2. Cloud computing incidents database. `http://wiki.cloudcommunity.org/wiki/CloudComputing:Incidents_Database`. Retrieved June 2009.
3. Cloud security alliance. http://www.cloudsecurityalliance.org. Retrieved in January 2011.
4. Grouped PEPA analyzer. `http://doc.ic.ac.uk/~as1005/GPA`.
5. IDC Enterprise Panel, August 2008, N=244, 2008.
6. Amazon S3 Availability Event. `http://status.aws.amazon.com/s3-20080720.html`, July 2008.
7. Virtualisierung bremst Energiebedarf. Computer Zeitung Nr. 52, Dec. 2008.
8. Cloud or Fog? The business realities of cloud computing for UK enterprises. NTT Comm., 2009.
9. T. Abdelzaher, K. Shin, and N. Bhatti. Performance guarantees for web server end-systems: A control-theoretical approach. In *IEEE Transactions on Parallel and Distributed Systems*, pages pp.80–96, 2002.
10. B. Abrahao, V. Almeida, and J. Almeida. Self-adaptive SLA-driven capacity management for internet services. In *Int. Workshop on Distributed Systems: Operations and Management*, 2006.
11. B. Addis, D. Ardagna, B. Panicucci, and L. Zhang. Autonomic management of cloud service centers with availability guarantees. *Cloud Computing, IEEE International Conference on*, 0:220–227, 2010.
12. J. Almasizadeh and M. A. Azgomi. Intrusion process modeling for security quantification. In *International Conference on Availability, Reliability and Security*, pages 114–121, Los Alamitos, CA, USA, 2009. IEEE Computer Society.
13. J. Almeida, V. Almeida, D. Ardagna, Õ. Cunha, C. Francalanci, and M. Trubian. Joint admission control and resource allocation in virtualized servers. *Jour. of Par. and Distr. Comp.*, 70(4):344 – 362, 2010.
14. P. Apparao, R. Iyer, X. Zhang, D. Newell, and T. Adelmeyer. Characterization & analysis of a server consolidation benchmark. In *Proc. of 4th Intl. ACM Conference on Virtual execution environments*, VEE '08, pages 21–30, 2008.

15. S. Balsamo, A. Di Marco, P. Inverardi, and M. Simeoni. Model-based performance prediction in software development: a survey. *Software Engineering, IEEE Transactions on*, 30(5):295 – 310, May 2004.
16. L. Barroso and U. Hölzle. The Case for Energy-Proportional Computing. IEEE Computer, vol. 40, 2007.
17. F. Benevenuto, C. Fernandes, M. Santos, V. Almeida, J. Almeida, G. Janakiraman, and J. Santos. Performance models for virtualized applications. In *Frontiers of High Performance Computing and Networking - ISPA 2006 Workshops*, volume 4331 of *LNCS*, pages 427–439. Springer, 2006.
18. C. Binnig, D. Kossmann, T. Kraska, and S. Loesing. How is the weather tomorrow?: towards a benchmark for the cloud. In *Proc. of the 2nd Intl. Workshop on Testing Database Systems*, DBTest '09, pages 9:1–9:6. ACM, 2009.
19. N. Bobroff, A. Kochut, and K. Beaty. Dynamic Placement of Virtual Machines for Managing SLA Violations. pages 119 –128, 2007.
20. S. Bose, P. Mishra, P. Sethuraman, and R. Taheri. Performance evaluation and benchmarking. chapter Benchmarking Database Performance in a Virtual Environment, pages 167–182. Springer-Verlag, Berlin, Heidelberg, 2009.
21. G. Box, G. M. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting & Control (3rd Edition)*. Prentice Hall, 3rd edition, February 1994.
22. R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. In *Proc. of the 7th High Performance Computing and Simulation (HPCS 2009) Conference*, volume abs/0907.4878, 2009.
23. Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. In *SIGMETRICS Int. Conf.*, 2005.
24. B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking cloud serving systems with ycsb. In *Proc. of the 1st ACM Symposium on Cloud computing*, SoCC '10, pages 143–154. ACM, 2010.
25. B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield. Remus: high availability via asynchronous virtual machine replication. In *NSDI'08*, 2008.
26. I. Cunha, J. Almeida, V. Almeida, and M. Santos. Self-adaptive capacity management for multi-tier virtualized environments. pages 129 –138, may. 2007.
27. H. de Meer and K. S. Trivedi. Guarded repair of dependable sys. *Theoretical Comp. Sci.*, 128:179–210, 1994.
28. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51:107–113, January 2008.
29. J. Dejun, G. Pierre, and C.-H. Chi. EC2 performance analysis for resource provisioning of service-oriented applications. In *Proc. 3rd Workshop on Non-Functional Properties and SLA Mgmnt in Service-Oriented Computing*, 2009.
30. N. J. Dingle. An empirical study of the scalability of performance analysis tools in the cloud. In *26th UK Performance Engineering Workshop (UKPEW 2010)*, pages 9–16, June 2010.
31. Y. el Khamra, H. Kim, S. Jha, and M. Parashar. Exploring the Performance Fluctuations of HPC Workloads on Clouds. In *2nd IEEE international conference on cloud computing technology and science (CloudCom)*, 2010.
32. E. Farr, R. Harper, L. Spainhower, and J. Xenidis. A Case for High Availability in a Virtualized Environment (HAVEN). In *Int. Conf. on Availability, Reliability and Security*, 2008.

33. F. C. Freiling. Dependability metrics. chapter Introduction to security metrics, pages 129–132. Springer-Verlag, Berlin, Heidelberg, 2008.
34. S. Gaisbauer, J. Kirschnick, N. Edwards, and J. Rolia. VATS: Virtualized-aware automated test service. In *QEST'08, Fifth International Conference on the Quantitative Evaluaiton of Systems*, pages 93–102, St Malo, September 2008. IEEE.
35. A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *Int. Joint Conf. on Measurement and modeling of computer systems*, 2009.
36. A. Georges and L. Eeckhout. Performance metrics for consolidated servers. In *Proceedings of HPCVirt 2010*. Association for Computing Machinery (ACM), 2010.
37. S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. In *Proc. of the 19th ACM Symp. on Operating Systems Principles*, 2003.
38. R. Ghosh, F. Longo, V. Naik, and K. Trivedi. Quantifying resiliency of iaas cloud. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, pages 343 –347, 31 2010-nov. 3 2010.
39. R. Grossman. The case for cloud computing. *IT Professional*, 11(2):23 –27, March–April 2009.
40. R. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411(22–24):2260–2297, April 2010.
41. B. Herndon, P. Smith, L. Roderick, E. Zamost, J. Anderson, V. Makhija, B. Herndon, P. Smith, E. Zamost, and J. Anderson. Vmmark: A scalable benchmark for virtualized systems, 2006.
42. N. Huber, M. von Quast, F. Brosig, and S. Kounev. Analysis of the Performance-Influencing Factors of Virtualization Platforms. In *OTM 2010 Conferences - Distributed Objects, Middleware, and Applications (DOA'10)*. Springer Verlag, 2010.
43. N. Huber, M. von Quast, M. Hauck, and S. Kounev. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. In *1st Intl. Conf. on Cloud Computing and Service Science (CLOSER 2011)*, 2011.
44. S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu, and X. Shi. Evaluating mapreduce on virtual machines: The hadoop case. In *Proc. of 1st Intl. Conference on Cloud Computing*, CloudCom '09, pages 519–528. Springer-Verlag, 2009.
45. M. Ihde and W. H. Sanders. Barbarians in the gate: An experimental validation of nic-based distributed firewall performance and flood tolerance. In *Proc. of Intl. Conference on Dependable Systems and Networks*, pages 209–216, 2006.
46. R. Iyer, R. Illikkal, O. Tickoo, L. Zhao, P. Apparao, and D. Newell. Vm3: Measuring, modeling and managing vm shared resources. *Computer Networks*, 53(17):2873 – 2887, 2009. Virtualized Data Centers.
47. J. Kephart et al. Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs. In *Int. Conf. on Autonomic Computing*, 2007.
48. K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. In *Proc. of 2nd IEEE Intl. Conference on Cloud Computing Technology and Science (CloudCom)*, pages 159–168, 2010.
49. A. Jaquith. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Professional, 2007.
50. G. Jung, M. Hiltunen, K. Joshi, R. Schlichting, and C. Pu. Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures. In *ICDCS*, 2010.

51. G. Jung, K. Joshi, M. Hiltunen, R. Schlichting, and C. Pu. Generating adaptation policies for multi-tier applications in consolidated server environments. In *Intl. Conf. on Autonomic Computing*, 2008.
52. G. Jung, K. Joshi, M. Hiltunen, R. Schlichting, and C. Pu. A cost-sensitive adaptation engine for server consolidation of multitier applications. In *Int. Conf. Middleware*, 2009.
53. J. Kaplan, W. Forrest, and N. Kindler. Revolutionizing Data Center Energy Efficiency. McKinsey, 2008.
54. A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko, and A. Tantawi. Dynamic placement for clustered web applications. In *International Conference on World Wide Web*, 2006.
55. H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.
56. D. S. Kim, F. Machida, and K. S. Trivedi. Availability modeling and analysis of a virtualized system. In *IEEE Pacific Rim International Symposium on Dependable Computing*, 2009.
57. R. King. How cloud computing is changing the world. In Businessweek on the World Wide Web. `http://www.businessweek.com/technology/content/aug2008/tc2008082_445669.htm`, Aug 2008. Retrieved in June 2009.
58. S. Kounev. Engineering of Next Generation Self-Aware Software Systems: A Research Roadmap. In *Emerging Research Directions in Computer Science. Contributions from the Young Informatics Faculty in Karlsruhe*. KIT Scientific Publishing, July 2010. ISBN: 978-3-86644-508-6.
59. S. Kounev, F. Brosig, and N. Huber. Descartes Research Project. http://www.descartes-research.net/. Last retrieved in February 2011.
60. S. Kounev, F. Brosig, N. Huber, and R. Reussner. Towards self-aware performance and resource management in modern service-oriented systems. In *Proceedings of the 7th IEEE International Conference on Services Computing (SCC 2010), July 5-10, Miami, Florida, USA*. IEEE Computer Society, 2010.
61. K. Kourai and S. Chiba. A fast rejuvenation technique for server consolidation with virtual machines. In *IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, pages 245–255, 2007.
62. H. Koziolek. Performance evaluation of component-based software systems: A survey. *Performance Evaluation*, 67(8):634 – 658, 2010. Special Issue on Software and Performance.
63. S. Kumar, V. Talwar, V. Kumar, P. Ranganathan, and K. Schwan. vManage: loosely coupled platform and virtualization management in data centers. In *Int. Conf. on Autonomic computing*, 2009.
64. S. Kundu, R. Rangaswami, K. Dutta, and M. Zhao. Application performance modeling in a virtualized environment. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th Intl. Symposium on*, pages 1 –10, 2010.
65. C. Lamprecht, A. van Moorsel, P. Tomlinson, and N. Thomas. Investigating the efficiency of cryptographic algorithms in online transactions. *International Journal of Simulation: Systems, Science & Technology*, 7(2):63–75, 2006.
66. B. Littlewood, S. Brocklehurst, N. E. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2(2-3):211–230, 1993.
67. B. Littlewood and L. Strigini. Redundancy and diversity in security. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *ESORICS*, volume 3193 of *Lecture Notes in Computer Science*, pages 423–438. Springer, 2004.
68. Z. Liu, M. S. Squillante, and J. L. Wolf. On maximizing service-level-agreement profits. In *Proc. of 3rd ACM conference on Electronic Commerce (EC'01)*, 2001.

69. S. Loveland, E. M. Dow, F. LeFevre, D. Beyer, and P. F. Chan. Leveraging virtualization to optimize high-availability system configurations. *IBM Syst. J.*, 47(4):591–604, 2008.
70. M. Armbrust et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
71. B. B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan, and K. S. Trivedi. Modeling and quantification of security attributes of software systems. In *Proc. of Intl. Conf. on Dependable Systems and Networks (DSN'02)*, pages 505–514, 2002.
72. J. Magott and M. Woda. Evaluation of soa security metrics using attack graphs. In *Proc. of 3rd Intl. Conf. on Dependability of Computer Systems (DepCoS-RELCOMEX)*, pages 277–284, 2008.
73. Y. Mei, L. Liu, X. Pu, and S. Sivathanu. Performance measurements and analysis of network i/o applications in virtualized cloud. *IEEE Intl. Conf. on Cloud Computing*, 0:59–66, 2010.
74. Y. Mei, L. Liu, X. Pu, and S. Sivathanu. Performance Measurements and Analysis of Network I/O Applications in Virtualized Cloud. In *International Conference on Cloud Computing*, 2010.
75. D. A. Menasce. Virtualization: Concepts, applications, and performance modeling. In *Proceedings of International CMG Conference*, 2005.
76. D. A. Menasce and M. N. Bennani. Autonomic virtualized environments. In *ICAS '06: Proceedings of the International Conference on Autonomic and Autonomous Systems*, 2006.
77. H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, and L. Yuan. Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers. In *Intl. SCC*, 2010.
78. K. Mishra and K. Trivedi. Model based approach for autonomic availability management. In *Service Availability*, volume 4328 of *LNCS*, pages 1–16. 2006.
79. L. Moura Silva, J. Alonso, P. Silva, J. Torres, and A. Andrzejak. Using virtualization to improve software rejuvenation. In *Int. Symposium on Network Computing and Applications (NCA 2007)*, 2007.
80. M. N. Bennani and D. A. Menasce. Resource Allocation for Autonomic Data Centers using Analytic Performance Models. In *International Conference on Automatic Computing*, 2005.
81. A. B. Nagarajan, F. Mueller, C. Engelmann, and S. L. Scott. Proactive fault tolerance for hpc with xen virtualization. In *ICS '07*, pages 23–32, 2007.
82. J. Napper and P. Bientinesi. Can cloud computing reach the TOP500? In *Proc. Combined Workshops on UnConventional High Performance Computing Workshop plus Memory Access Workshop (UCHPC-MAW'09)*, pages 17–20, 2009.
83. R. Nathuji and K. Schwan. VirtualPower: coordinated power management in virtualized enterprise systems. In *ACM SIGOPS Symposium on Operating systems principles*, 2007.
84. D. M. Nicol, W. H. Sanders, and K. S. Trivedi. Model-based evaluation: From dependability to security. *IEEE Trans. Dependable Secur. Comput.*, 1(1):48–65, 2004.
85. J. Oberheide, E. Cooke, and F. Jahanian. Cloudav: N-version antivirus in the network cloud. In *Proc. of the 17th USENIX Security Symposium*, July 2008.
86. S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing. In *1st Intl. Conf. on Cloud Computing*. ICST Press, 2009.

87. C. Pu, J. Noe, and A. Proudfoot. Regeneration of replicated objects: A technique and its eden implementation. In *Proc. of the 2^{nd} Int. Conf. on Data Engineering*, pages 175–187, 1986.
88. X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu. Understanding Performance Interference of I/O Workload in Virtualized Cloud Environments. In *Intl. Conf. on Cloud Computing*, 2010.
89. H. V. Ramasamy and M. Schunter. Architecting Dependable Systems Using Virtualization. In *Workshop on Architecting Dependable Systems: Supplemental Volume of the DSN'07*, 2007.
90. K. Rangan. The Cloud Wars: $ 100+ Billion at Stake. Technical Report, Merrill Lynch, May 2008.
91. P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. *IEEE Intl. Conf. on Cloud Computing*, pages 280–288, 2010.
92. J. Scaramella. Worldwide Server Power and Cooling Expense, 2006-2010 Forecast. IDC, 2006.
93. K. G. Shin, C. M. Krishna, and Y.-H. Lee. Optimal dynamic control of resources in a distributed system. *IEEE Trans. on Softw. Eng.*, 15(10):1188–1198, 1989.
94. L. Siegele. Let It Rise: A Special Report on Corporate IT. The Economist, 2008.
95. S. Singh, W. H. Sanders, D. M. Nicol, and M. Seri. Automatic verification of distributed and layered security policy implementations. Technical Report UILU-ENG-08-2209 (CRHC-08-05), Univ. of Illinois at Urbana-Champaign, 2008.
96. S. Sivathanu, L. Liu, M. Yiduo, and X. Pu. Storage management in virtualized cloud environment. *Cloud Computing, IEEE International Conference on*, 0:204–211, 2010.
97. W. D. Smith. TPC-W: Benchmarking an ecommerce solution. `TPCWhitepaper, http://www.tpc.org/tpcw/`, 2001.
98. SPEC Virtualization Committee. SPECvirt_sc2010. `http://www.spec.org/virt_sc2010`, 2010.
99. A. Stefanek, R. Hayden, and J. T. Bradley. Fluid analysis of energy consumption using rewards in massively parallel Markov models. In *Intl. Conf. on Performance Engineering (ICPE) 2011, March 14-16, 2011*, 2011. (To appear).
100. G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani. A Hybrid Reinforcement Learning Approach to Autonomic Resource Allocation. In *Int. Conf. on Autonomic Computing*, 2006.
101. The Center for Internet Security. The cis security metrics v1.0.0, May 2009.
102. T. Thein, S.-D. Chi, and J. S. Park. Availability modeling and analysis on virtualized clustering with rejuvenation. *Int. Journal of Computer Science and Network Security*, 8(9):72–80, 2008.
103. T. Thein and J. S. Park. Availability analysis of application servers using software rejuvenation and virtualization. *J. Comput. Sci. Technol.*, 24(2):339–346, 2009.
104. O. Tickoo, R. Iyer, R. Illikkal, and D. Newell. Modeling virtual machine performance: challenges and approaches. *SIGMETRICS Perform. Eval. Rev.*, 37:55–60, January 2010.
105. K. Trivedi, K. Vaidyanathan, and K. Goseva-Popstojanova. Modeling and analysis of software aging and rejuvenation. In *Proceedings of the 33rd Annual Simulation Symposium*, pages 270 –279, 2000.
106. B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi. An analytical model for multi-tier internet services and its applications. In *SIGMETRICS Int. Conf.*, 2005.

107. Various authors. The Network Simulator ns-2. http://www.isi.edu/nsnam/ns/. (Retrieved in May 2010).
108. V. Verendel. Quantified security is a weak hypothesis: A critical survey of results and assumptions. In *NSPW '09: Proceedings of the New Security Pradigms Workshop 2009*, pages 37–50, New York, NY, USA, 2009. ACM.
109. A. Verma, P. Ahuja, and A. Neogi. pMapper: power and migration cost aware application placement in virtualized systems. In *Int. Conf. on Middleware*, 2008.
110. VMWare. Vmware high availability (ha), restart your virtual machine. Accessed May 2009. http://www. vmware.com/products/vi/vc/ha.html.
111. E. Walker. Benchmarking Amazon EC2 for high-performance scientific computing. *;Login*, 33(5):18–23, October 2008.
112. G. Wang, A. R. Butt, P. Pandey, and K. Gupta. A Simulation Approach to Evaluating Design Decisions in MapReduce Setups. In *Intl. Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 1–11. IEEE, 2009.
113. X. Wang, D. Lan, G. Wang, X. Fang, M. Ye, Y. Chen, and Q. Wang. Appliance-based autonomic provisioning framework for virtualized outsourcing data centre. In *ICAC*, 2007.
114. S. Wilson. Appengine outage. CIO Weblog, http://www.cio-weblog.com/50226711/appengine_outage.php, June 2008.
115. B. Winterford. Stress tests rain on Amazon's cloud. IT News, http://www.itnews.com.au/News/153451,stress-tests-rain-on-amazons-cloud.aspx, Aug. 2009.
116. T. Wood, L. Cherkasova, K. Ozonat, and P. Shenoy. Profiling and modeling resource usage of virtualized applications. In *Proc. of 9th ACM/IFIP/USENIX Intl. Conf. on Middleware*, pages 366–387, 2008.
117. J. Xu, M. Zhao, J. Fortes, R. Carpenter, and M. Yousif. On the use of fuzzy modeling in virtualized data center management. In *International Conference on Autonomic Computing*, 2007.
118. J. Yang, J. Qiu, and Y. Li. A Profile-Based Approach to Just-in-Time Scalability for Cloud Applications. In *International Conference on Cloud Computing*, 2009.
119. Q. Zhang, L. Cherkasova, and E. Smirni. A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In *Int. Conference on Autonomic Computing*, 2007.