Zhou X, Jiang CJ. Autonomic performance and power control on virtualized servers: Survey, practices, and trends. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(4): 631–645 July 2014. DOI 10.1007/s11390-014-1455-4

Autonomic Performance and Power Control on Virtualized Servers: Survey, Practices, and Trends

Xiaobo Zhou[1,2] (周笑波), *Senior Member*, *IEEE*, *Member*, *ACM*
and Chang-Jun Jiang[2,3] (蒋昌俊), *Member*, *CCF*, *IEEE*

[1] *Department of Computer Science, University of Colorado, Colorado Springs, U.S.A.*

[2] *The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University Shanghai 200092, China*

[3] *Department of Computer Science and Technology, Tongji University, Shanghai 201804, China*

E-mail: xzhou@uccs.edu; cjjiang@tongji.edu.cn

Received February 15, 2014; revised May 4, 2014.

Abstract Modern datacenter servers hosting popular Internet services face significant and multi-facet challenges in performance and power control. The user-perceived performance is the result of a complex interaction of complex workloads in a very complex underlying system. Highly dynamic and bursty workloads of Internet services fluctuate over multiple time scales, which has a significant impact on processing and power demands of datacenter servers. High-density servers apply virtualization technology for capacity planning and system manageability. Such virtualized computer systems are increasingly large and complex. This paper surveys representative approaches to autonomic performance and power control on virtualized servers, which control the quality of service provided by virtualized resources, improve the energy efficiency of the underlying system, and reduce the burden of complex system management from human operators. It then presents three designed self-adaptive resource management techniques based on machine learning and control for percentile-based response time assurance, non-intrusive energy-efficient performance isolation, and joint performance and power guarantee on virtualized servers. The techniques were implemented and evaluated in a testbed of virtualized servers hosting benchmark applications. Finally, two research trends are identified and discussed for sustainable cloud computing in green datacenters.

Keywords autonomic computing, joint performance and power control, virtualized server, Internet application, sustainable computing

1 Introduction

1.1 Motivation for Autonomic Control

Modern datacenters are becoming the computing platform for supporting cloud computing that aims to offer information technology capabilities over the Internet as an on-demand pay-per-use service. The key enabling technologies include server virtualization, service-oriented architecture, pay-as-you-go business model, and emerging autonomic resource management techniques.

Today the information and communication technology accounts for about 3% of global electricity usage and greenhouse gas, which is about the same as the emissions of airlines. More than half of the energy use and emissions is due to servers and datacenters. Improving system performance and reducing energy consumption are critical issues for building the next generation green datacenters. However, due to the highly dynamic nature of Internet workloads, increasing complexity of applications, and complex dynamics of shared infrastructure, datacenters face significant challenges in managing application performance while maintaining resource utilization efficiency and reducing power consumption costs.

Large-scale computing systems, exemplified by virtualized datacenters, have reached a level of complexity where the human effort required to get the systems up and running and keeping them operational is getting out of hand[1]. To manually manage the performance of the hosted applications demands extensive experience and expertise on the workload profile and on the computing system. However, the timescales over which the changes in the workload profile occur may not allow manual intervention. Furthermore, the contention of

Survey

This research was supported in part by the National Science Foundation of USA under Grant Nos. CNS-0844983 (CAREER Award) and CNS-1217979, and the National Natural Science Foundation of China under Grant No. 61328203.

shared resources among multiple client applications that are consolidated on virtualized servers has a significant impact on the application performance. The situation is further complicated by the fact that datacenters need to control the power consumption to avoid power capacity overload, to lower electricity costs, and to reduce their carbon footprint. The complexity and the scale of virtualized datacenters make it increasingly difficult for administrators to manage them. Hence, there are growing research interests in autonomic computing paradigm in the context of virtualized datacenters.

1.2 Issues and Challenges

This paper focuses on recent research on autonomous performance and power control of Internet services hosted on virtualized servers. The main challenging research issues are:

Workload and Platform Complexity. In a virtualized datacenter, the user perceived performance is the result of a complex interaction of complex workloads in a very complex underlying system[2]. Today's popular multi-tier multi-service architecture imposes complex inter-tier and intra-tier performance dependences. Recent studies[2-4] observed highly dynamic workloads of Internet services that fluctuate over multiple time scales, which can have a significant impact on the processing and power demands imposed on datacenter servers. Datacenters hosting Internet services are often built upon virtualized server clusters. Virtualization technologies such as VMware[5] and Xen[6] provide an abstraction of hardware resources to run multiple instances of independent virtual machines (VMs) in one physical computer. The benefits of virtualization include high resource utilization, performance isolation, high availability, and fast server switching[7-8]. However, complexity of server parameter configuration, bursty workloads and inherent nonlinearity of performance and power versus resource allocation introduce significant challenges to achieving accurate and agile performance and power control.

Performance Metrics. End-to-end system response time is a major performance metric of multi-tier Internet services. It is the response time of a request that flows through a multi-tier system[9]. Most research on performance control in multi-tier server systems focused on the average response time. Percentile-based response time, compared to the average response time, has the benefit that is both easy to reason about and to capture individual users' perception of performance[2,4,9-11]. For example, in a set of 100 response time values that are sorted from the best to the worst, the 90th percentile simply means the 90th value in the list. However, it is challenging to assure a percentile-based response time of requests in complex multi-tier Internet services in virtualized environments, particularly in the face of highly dynamic workloads[12]. Also, there is lack of studies of controlling the effective system throughput, which is the number of requests that meet the service level agreement on the response time. Improving effective system throughput is significant because one key goal in a datacenter is to maximize the useful work with given resources and power budget[13]. But it is very challenging as an Internet service has many configurable parameters and its operating environment is highly dynamic. To be profitable, a datacenter must achieve high utilization, and the key to this is the agility — the capacity to assign any server to any service[14].

Performance Isolation. Virtualized datacenters face an important but challenging issue of performance isolation among heterogeneous customer applications. Performance interference resulting from the contention of shared resources, e.g., the last level cache and memory bandwidth, among co-located virtual servers, has significant impact on application performance. Most existing performance isolation techniques, be hardware or software resource partitioning based, require invasive instrumentation and modification of the guest operating system or the virtualization management layer[15-16]. However, resource partitioning can be difficult and costly to implement and even if accomplished may result in inefficient resource utilization[17]. Due to portability and transparency, non-invasive performance isolation is desirable in virtualized datacenters provisioning cloud computing services, which host third-party customer applications and often use virtualization software from third-party vendors.

Joint Performance and Power Control. The widely used high-density servers impose stringent power and cooling requirements. It is necessary to precisely control power consumption of servers to avoid system failures caused by power capacity overload or overheating[18]. However, it is challenging to assure both the performance of heterogeneous applications and the power consumption cap of the underlying virtualized server clusters mainly due to the workload dynamics and the system dynamics of shared infrastructure. Recent studies applied utility optimization to coordinate power consumption and average response time[19-20]. However, they lack assurance on system stability and performance in the face of highly dynamic workloads. Those techniques for average response time may not be applicable to percentile-based response time due to its strong nonlinearity with resource allocation.

The rest of this paper is organized as follows. Section 2 surveys state-of-the-art techniques in joint performance and power control in virtualized datacenter server systems. Sections 3, 4, and 5 introduce three representative approaches based on machine learning and control techniques for autonomic performance control of multi-tier Internet services, non-invasive energy-efficient performance isolation on virtualized servers, and joint performance and power assurance in virtualized server clusters, respectively. Section 6 identifies two research trends in sustainable cloud computing in green datacenters. We conclude the paper in Section 7.

2 State-of-the-Art

2.1 Autonomic Performance Control for Multi-Tier Internet Services

There were many studies on the performance modeling and analysis of multi-tier Internet servers with queueing foundations and optimization techniques. More specifically, there were studies in profile-driven performance optimization for clusters[21-22], performance modeling of multi-tier systems[23], and dynamic virtual server provisioning for performance assurance in multi-tier clusters[4,24]. Those queueing model based techniques can provide the average response time based performance guarantee, but are not effective for the percentile-based response time guarantee.

Feedback control techniques were used alone or together with queueing models for service differentiation and performance guarantee on Internet servers. Early work focused on performance control of individual Internet servers[25-26]. Recent studies were on performance assurance in multi-tier Internet servers[10,27-30]. Those studies focused on using the average response time as the performance metric.

Percentile-based performance metric has the benefit that is easy both to reason about and to capture individual users' perception of Internet service performance[2,11]. There are a few important studies in percentile-based delay guarantee in multi-tier Internet services. An adaptive admission control designed in [11] complements, but does not apply to dynamic server provisioning in datacenters. A dynamic server provisioning method proposed in [9] is model dependent and the application profiling needs to be done offline for each workload before the server replication and allocation. A fuzzy control based server provisioning method proposed in [10] is effective under stationary system workloads, but it does not adapt to the very dynamic nature of Internet workloads. A model-dependent stochastic approximation technique can estimate the tardiness quantile of response time distribution[31]. Its PID controller was designed and tuned for a particular simulated workload. It is not adaptive to highly dynamic workloads. An approximation-based approach in [4] is effective only in the heavy-traffic case in a near-to-saturation system. An approach proposed in [32] can model the probability distributions of response time based on CPU allocations on virtual machines in a datacenter. The performance model was obtained by offline training of the collected system data. It is not adaptive online to dynamically changing workloads.

Statistical machine learning techniques have been recently used for measuring the capacity of websites[33], for online system reconfiguration[34-36], and for coordinated admission control and autonomic resource allocation in multi-tier systems[37-41]. Rao *et al.* proposed a reinforcement learning approach for autonomic configuration and reconfiguration of virtual machines[35]. Muppala and Zhou proposed a coordinated session-based admission control with statistical learning for improving effective session throughput of multi-tier Internet applications[38]. Guo *et al.* proposed and designed a neural fuzzy control based approach for agile server parameter tuning, which combines the strengths of fast online learning and self-adaptiveness of neural networks and fuzzy control[42-43]. They further designed a genetic algorithm with multi-agent reinforcement learning for coordinated virtual machine resizing and server tuning for high system throughput and power utilization efficiency[44].

2.2 Power Management in Computing Systems and Datacenters

Web server power management utilizes techniques including dynamic voltage and frequency scaling (DVS/DVFS)[45], system shut-down, consolidation, etc. A few early and important studies proposed to reduce power consumption in Web servers by applying the DVS technique[46]. Recently, DVS was applied for maximizing the performance of power constrained high-density servers[47] and for improving power efficiency of server farms[48]. Those studies focused only on single-tier Web systems.

Power management in multi-tier systems imposes significant challenges. Applying independent DVS algorithms in a multi-tier server pipeline will lead to inefficient usage of power for assuring an end-to-end delay guarantee due to inter-tier dependency[49]. Horvath *et al.*[49] implemented a coordinated DVS policy for a three-tier Web system based on distributed feedback control and an optimization model that minimizes total power consumption while meeting end-to-end delay deadline. Wang *et al.*[50] proposed a multiple-input and multiple-output (MIMO) controller to accurately regulate the total power consumption of an en-

closure by conducting processor frequency scaling for each server while optimizing multi-tier application performance. Such controllers are designed based on offline system identification for specific workloads. They are not adaptive to abrupt workload changes though they can achieve control accuracy and system stability within a range theoretically.

Power management in virtualized datacenters is a very active research area. Power management techniques based on DVS are not easily applicable to virtualized environments where physical processors are shared by multiple virtual machines[8,18]. For example, changing the power state of a processor by DVS will affect the performance of multiple virtual machines hosting different applications. A few recent researches studied the limitation of DVS technique in virtualized environments and proposed DVS alternatives by applying "soft" techniques to exploit a hypervisor's ability to limit hardware usage by guests virtual machines[8,19]. A few other interesting approaches integrated DVS with load management for energy conservation in virtualized datacenters[18,51]. Virtualization technology also offers opportunities to consolidate workloads on fewer powerful servers for improving server resource utilization and performance isolation. For instance, virtualPower[52] provides coordinated power management in virtualized enterprise systems. The increasing power densities of datacenter servers can lead to a greater probability of thermal failover, affecting the availability of the systems and increasing the cost of additional cooling. There is a growing interest to explore power over-subscription, VM consolidation, thermal management, power proportionality and power-cost optimization in datacenter servers[53-59].

2.3 Joint Power and Performance Management on Virtualized Servers

It is a research trend that power utilization efficiency and performance control of multi-tier server systems are jointly tackled. However, it is challenging due to their correlated yet conflicting goals. There are three general approaches, power-oriented, performance-oriented, and explicit trade-off based.

Power-oriented approaches ensure that a server system does not violate a given power budget while maximizing performance of hosted applications[20,47,50,60] or increasing the number of services that can be deployed[61-62]. pMapper[63] tackles power-cost tradeoffs under a fixed performance constraint. vManage[64] performs virtual server placement to save power without degrading performance.

Performance-oriented approaches aim to guarantee a performance target while minimizing the power consumption[18,31,52,65-67]. They do not have explicit control over power consumption.

Coordinated power and performance management with explicit trade-offs has recently been studied in virtualized servers[19,68-71] and in disk drives[72]. The work in [70] proposed an interesting approach for semantics-free coordination between power and performance modules. Mistral[69] is a control architecture to optimize power consumption, performance benefit, and the transient costs in Cloud environments. Co-Con[20] is a two-level control architecture for power and performance coordination in virtualized server clusters. It gives a higher priority to power budget tracking while performance is a secondary goal. vPnP[19] coordinates power and performance in virtualized datacenters using utility function optimization. It provides the flexibility to choose various trade-offs between power and performance. However, it lacks the control automation and accuracy in the face of dynamic workloads. PERFUME[71,73] provides better control accuracy under a dynamic workload. Its follow-up work APPLEware[74] is an autonomic and scalable middleware for joint performance and power control of multi-service applications in virtualized datacenters. It features a distributed control structure that provides predictable performance and energy efficiency for large complex systems. pVOCL[75] is a power-aware virtual OpenCL framework that controls the peak power consumption and improves the energy efficiency of the underlying server system through dynamic consolidation and power-phase topology-aware placement of GPU workloads.

2.4 Sustainable Computing with Renewable Energy in Green Datacenters

As the environmental concerns and the energy consumption of datacenters continuously grow, developing sustainable datacenters is becoming an increasingly important mission for major Internet service operators[76-80]. The vast majority of the previous studies on the sustainable energy management has focused on the single datacenters. These studies aim to achieve sustainable operation driven by green energy supply partially or completely from following aspects: 1) Studies[76,81-82] focus on the energy demand side of a datacenter. 2) Studies[83-85] focus on matching energy supply of a datacenter server cluster with its energy demand. 3) Study[86] focuses on different energy storage approaches in the sustainable datacenters to improve green energy usage efficiency.

Aksanli *et al.*[76] designed an adaptive job scheduler to increase the green energy usage in a sustainable datacenter, which utilizes short term prediction of solar and

wind energy production. This scheduling method may violate the QoS requirement due to unnecessarily delaying batch jobs. Goiri *et al.* proposed GreenHadoop[81], a MapReduce framework for a datacenter powered by solar energy and using electrical grid as a backup. It aims to maximize the green energy consumption by job scheduling. It does not consider the potential opportunities to improve the green energy usage by workload distribution across distributed datacenters.

A few recent studies start to utilize green energy in distributed datacenters. Deng *et al.*[77] proposed an adaptive request routing approach to meet the operational cost, QoS, and carbon footprint goals. Zhang *et al.*[80] proposed GreenWare, a middleware system that dynamically dispatches transactional requests to distributed datacenters to maximize the use of green energy within the allowed operation budget. However, these studies only consider transactional requests that are of low cost for routing and do not consider another important category of cloud workload, i.e., batch jobs.

Recently, Liu *et al.* proposed GLB[78], a geographical load balancing approach that can significantly reduce the required capacity of green energy by using the energy more efficiently with request dispatching. GLB provides a representative workload dispatching and capacity provisioning method to minimize the system energy cost and the request delay cost. sCloud[87] differs in that it considers the workload heterogeneity and batch job migration across distributed datacenters.

3 Practice 1: Autonomic Performance Control in Multi-Tier Internet Services

Popular Internet services employ a complex multi-tier architecture, with each tier provisioning a certain functionality to its preceding tier and making use of the functionality provided by its successor to carry out its part of the overall request processing. Typically, a three-tier architecture is used in many Internet services, i.e., Web, Application and Database. For load sharing, one tier is often replicated and clustered.

Autonomous resource management is critical to performance assurance and is challenging due to rapidly growing the scale and complexity of the services and the underlying computing systems. Many recent research efforts relied on queuing-theoretic and control-theoretic approaches[9,21-23,29], based on explicit system performance models for dynamic server provisioning. However, it is challenging to accurately estimate system performance model parameters such as service time, workload distribution, and so on. Furthermore, system parameter variation of virtual servers, highly bursty workloads, and inherent nonlinearity of performance versus resource allocation introduce additional challenges to achieve accurate and agile system performance control[4,12].

End-to-end system response time is a major performance metric of multi-tier Internet services. But using the average delay as the performance metric is unable to quantify the peak-to-mean ratio of service demands[2]. Percentile-based delay metric has the benefit that is easy both to reason about and to capture individual users' perception of service quality[9,11].

However, it is very challenging to assure the percentile-based response time of requests of a multi-tier service. Compared with the average delay, a percentile-based response time introduces much stronger nonlinearity to system resource allocation. A queueing approximation-based approach[4] is effective for percentile-based response time guarantee only when the system capacity is near to saturation. Offline application profiling and training based approaches[9,32] can be time consuming and the obtained performance models are not adaptive to highly dynamic workloads[12]. Control theoretic techniques were applied for performance guarantee by performing linear approximation of system dynamics and estimation of system parameters. However, if the system configuration or workload range deviates significantly from those used for system identification, the estimated system model used for control becomes inaccurate.

3.1 Neural Fuzzy Control Based Approach to Percentile-Based Response Time Assurance

In our previous studies[10,88], we proposed a model-independent rule-based fuzzy logic controller that utilizes heuristic knowledge for performance assurance on multi-tier servers. It uses a set of pre-defined rules and fuzzy membership functions to perform control actions in the form of dynamic server provisioning adjustment. This kind of controllers has some drawbacks. First, it is designed manually on trial and error basis using heuristic control knowledge. There is no specific guideline for determining important design parameters such as the input scaling factors, the rule base, and the fuzzy membership functions. Second, those design parameters are not self-adaptive, so not effective in the face of highly dynamic workloads. To avoid ill effects of modeling inaccuracy and to enhance system agility and self-adaptiveness, we considered the integration of model-independent control with fast learning neural networks.

In recent studies[12,89], we proposed and developed a self-adaptive server provisioning method based on an integrated neural fuzzy controller for the percentile-based response time guarantee in virtualized multi-tier server clusters.

Fig.1 shows the block diagram of the server provisioning approach with a self-adaptive neural fuzzy control. The controller aims to bound a percentile-based response time T_d to a specified target T_{ref}. It has two inputs: error denoted as $e(k)$ and change in error denoted as $\Delta e(k)$. Error is the difference between the target and the measured value of the delay in the k-th sampling period. The output is the resource adjustment $\Delta m(k)$ for the next sampling period.

We designed the neural fuzzy controller using a general four-layer fuzzy neural network. The layers of the neural network and their interconnections provide the functionality of membership functions and rule base of the controller. Unlike a rule-based fuzzy controller, the membership functions and rules in the neural fuzzy controller dynamically construct and adapt themselves as the neural network grows and learns. The controller is self-adaptive to system and workload dynamics and tunes its parameters in an agile manner online. The fuzzy neural network adopts fuzzy logic rules as follows:

R_r: *IF* x_1 *is* A_1^j .. *and* x_n *is* A_n^j, *THEN* y *is* b_r

where R_r is the r-th fuzzy logic rule, x_i is an input, either to be $e(k)$ or $\Delta e(k)$, and y is the rule's output. A_i^j is the j-th linguistic term associated with the i-th input variable in the precondition part of the fuzzy logic rule R_r. Linguistic terms are fuzzy values such as "positive small", "negative large". They describe the input variables with some degree of certainty, determined by their membership functions $u_{A_i^j}$. The consequent part or outcome of the rule R_r is denoted as b_r. Each rule contributes to the controller output, the resource allocation adjustment $\Delta m(k)$, according to its firing strength. The decomposition of $\Delta m(k)$ to the different tiers is performed in proportion to the per-tier delay observed from the controlled system.

The neural fuzzy controller combines fuzzy logic reasoning with the learning capabilities of an artificial neural network. Initially, there are only input and output nodes in the neural network. The membership and the rule nodes are generated dynamically through the structure and parameter learning processes.

In the structure learning phase, to avoid the newly generated membership function being too similar to the existing ones, we use the similarity measure approach[90] to check the similarity of two membership functions. If the measure is less than a pre-specified value, the new membership function is adopted.

In the parameter learning phase, the learning is used to adaptively modify the consequent part of existing fuzzy rules and the shape of membership functions to improve the controller performance in the face of highly dynamic workloads. It is achieved by minimizing an energy function defined as the difference between the target and measured percentile-based end-to-end delays. The learning algorithm recursively obtains a gradient vector in which each element is defined as the derivative of the energy function with respect to a parameter of the network. This is done by a chain rule method[12].

3.2 Representative Results

We implemented the neural fuzzy control based server provisioning method in a testbed of virtualized three-tier server clusters. As the related work in [9], the database tier is not replicated in our testbed. Virtualization of the cluster is enabled by VMWare's vSphere 4.1 Enterprise edition. Each server is hosted inside a virtual machine (VM). The configuration of each VM for the web and application tiers is 1 vCPU, 2 GB RAM and 15 GB hard disk space. The guest operating system used is Ubuntu Linux version 10.04. Load balancers are used to distribute requests among VMs at the web and application tiers. An Apache module, *mod_proxy_balancer*, is used for load balancing while taking into account session affinity.

The neural fuzzy controller interacts with the VM manager through the vSphere 4.1 API. We used an open-source multi-tier application benchmark RUBiS[①] in the case study. RUBiS implements the core functionality of an eBay like auction site. We instrumented the clients to submit workloads of various mixes with time-varying intensity and measure per-tier utilization and percentile-based response time.

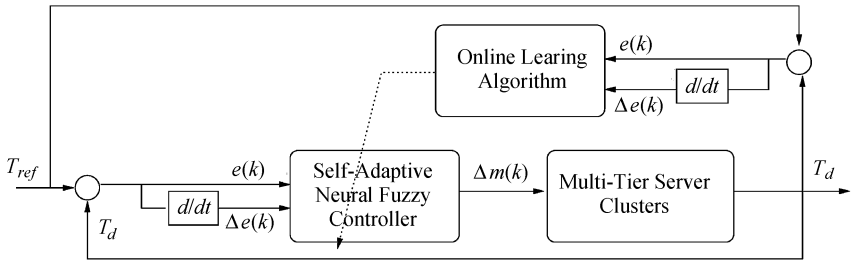


Fig.1. Block diagram of a self-adaptive neural fuzzy controller.

[①]RUBiS — Rice University bidding system. http:/rubis.owz.org, May 2014.

For performance evaluation, we applied a dynamic workload with sudden step-changes similar to what used in [9]. As shown in Fig.2(a), initially, the workload consists of a bidding mix of 200 concurrent users. After 20 minutes, the workload intensity is doubled to 400 concurrent users with browsing workload mix. Another 20 minutes later, the workload intensity is decreased to 300 concurrent users. The control interval used in the case study is 3 minutes. The reported results are from a single run.

Fig.2(b) shows that the self-adaptive neural fuzzy controller is able to guarantee the 95th-percentile delay target of 2 seconds within a few sampling intervals. The multi-tier system is initially provisioned with one virtual server at each tier. As the controller starts allocating virtual servers at the web and application tiers, it applies online learning to tune its neural network structure and parameters based on the measured percentile-based end-to-end delay of requests. We observe that the 95th-percentile delay approaches the target of 2 seconds within the first 12 minutes of the experiment as a result of agile server allocations. The oscillation of the delay around its target is mainly due to the fact that neural fuzzy controller needs to learn how to control the system by exploring different server allocations. As time progresses, the controller becomes more effective in achieving the end-to-end delay guarantee. There is a spike in the measured delay at the 20th minute and a drop in the measured delay at the 40th minute due to the sudden changes in the workload intensity and mix. The neural fuzzy controller effectively adds or removes virtual servers to/from different tiers to bring the end-to-end delay close to the target, and achieves the delay guarantee in an agile manner.

Fig.2(c) shows the change in the allocation of virtual servers at various tiers. The controller allocates servers at individual tiers in proportion to the per-tier delay measurement in a self-adaptive manner. Note that the server allocation adjustments are only distributed between the web and application tiers as the database tier is not replicated.

Summary. In this practice[89], we found that compared with a rule-based fuzzy controller and a proportional-integral controller, the neural fuzzy controller based approach delivers superior self-adaptive performance assurance in the face of highly dynamic workloads. It is robust to variation in workload intensity, characteristics, and change in delay target and server switching delays.

4 Practice 2: Performance Isolation on Virtualized Servers

Performance isolation among heterogeneous applications is an important but challenging issue in virtualized servers. It provides the base for predictable application performance. There are invasive techniques based hardware and software resource partitioning[15-17,91] to avoid performance interference. Others including our recent work use scheduling to mitigate interferences at system-level[92]. While resource partitioning might not be available on commodity hardware and operating systems, interference mitigation is often not sufficient to enforce strict isolation, especially when co-located applications have heterogeneous and time-varying demands.

4.1 Non-Invasive Energy-Efficient Approach to Performance Isolation

In study [93], we took the challenge to design a *non-invasive* performance isolation mechanism that is completely built on top of proprietary virtualization software (e.g., VMware) and commodity hardware (e.g., Intel X86 processors). To this end, we proposed, NINEPIN, a performance isolation mechanism based on a novel hierarchical control framework. The core of NINEPIN is the idea of tracking application-related performance metric and dynamically provisioning (or

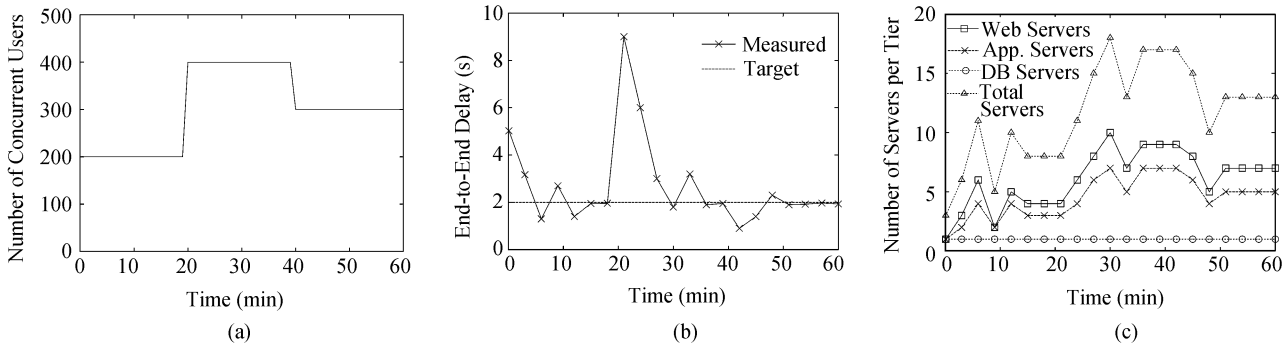


Fig.2. Performance of the neural fuzzy control in a virtualized testbed hosting RUBiS application. (a) A sudden change workload. (b) 95th delay assurance. (c) Server allocation.

compensating) resources to maintain consistent performance. Besides performance isolation, we took further steps to optimize overall performance and minimize energy consumption.

Fig.3 shows the architecture of NINEPIN. The computer system under control is a virtualized server hosting multiple applications running in VMs. NINEPIN forms a control loop that continuously monitors application performance of co-located VMs and adjusts their resource allocation to guarantee individual performance targets and maximize overall system utility. The key component in NINEPIN is the two-level hierarchical controller, which contains the level-1 utility optimizer and the level-2 model predictive controller. At every control interval, the optimizer searches the space of possible resource allocations of the co-located VMs and tries to maximize the overall system utility. The optimization depends on a fuzzy MIMO performance model to generate hypothetical resource allocations and predict the performance of individual applications. The predicted performance is evaluated by the system utility function for optimality check. The individual performance that leads to the optimal system utility is then fed to the level-2 as the performance targets for the model predictive controller.

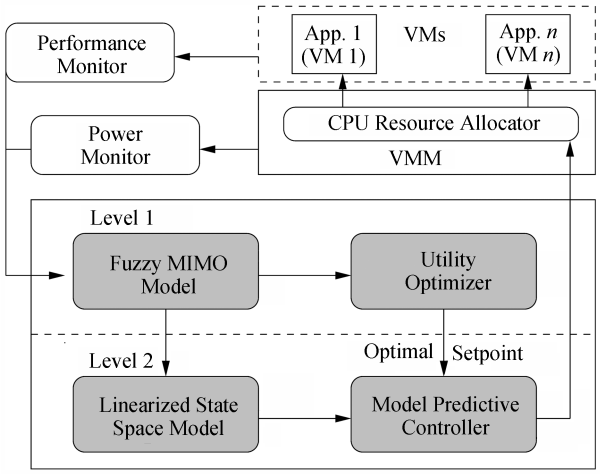


Fig.3. Architecture of NINEPIN.

Modeling Application Performance. The key to the effectiveness of NINEPIN is the fuzzy model that captures the complex relationship between resource allocations and performance. The MIMO model accepts a vector of resource allocations for individual applications and outputs the vector containing the predictions of their performance. The modeling is difficult as the relationship between resource and performance exhibits significant nonlinearity under interferences and time-varying workload. We used a collection of fuzzy logic to approximate the complex relationship. Given resource allocation $\boldsymbol{u}(k)$, a fuzzy logic is described as:

If $\xi_1(k)$ *is* $\Omega_{i,1}$ *and* .. $\boldsymbol{\xi}_\varrho(k)$ *is* $\Omega_{i,\varrho}$ *and* $u_1(k)$ *is* $\Omega_{i,\varrho+1}$ *and* .. $u_m(k)$ *is* $\Omega_{i,\varrho+m}$ *then*

$$y_i(k+1) = \boldsymbol{\zeta}_i \boldsymbol{\xi}_i(k) + \boldsymbol{\eta}_i \boldsymbol{u}(k) + \boldsymbol{\phi}_i. \quad (1)$$

Here, Ω_i is the antecedent fuzzy set of the i-th rule which describes elements of regression vector $\boldsymbol{\xi}(k)$ and the current input vector $\boldsymbol{u}(k)$ using fuzzy values such as "large", "small". $\boldsymbol{\zeta}_i$ and $\boldsymbol{\eta}_i$ are vectors containing the consequent parameters and $\boldsymbol{\phi}_i$ is the offset. ϱ denotes the number of elements in the regression vector $\boldsymbol{\xi}(k)$. Each fuzzy rule describes a region of the complex nonlinear system model using a simple functional relation given by the rule's consequent part. The model output is calculated as the weighted average of the linear consequents in the individual predictions $y_i(k+1)$. We built a similar fuzzy MIMO model for energy usage, which takes current resource allocation as input and outputs predictions of energy consumption.

Online Model Adaption. The fuzzy performance model should be robust enough to accommodate the changes in the interference and workload. The online model adaptation is performed when a significant error in the prediction of performance is detected. This avoids the overhead of frequent adaptation and computationally expensive re-optimization. NINEPIN applies a weighted recursive least squares method to adapt the consequent parameters of the fuzzy MIMO model as new measurements are sampled from the runtime system. It applies exponentially decaying weights on the sampled data so that larger weights are assigned to more recent observations. Then, it re-computes the optimal performance targets based on the updated fuzzy model.

System Utility Optimizer. The primary goal of the optimizer is to maximize the system utility with respect to the performance of all hosted applications. To provide predictable and satisfying application performance, we define the utility function of individual applications based on their service level objectives (SLOs) and propose a *unified* utility function for the performance of all heterogeneous applications. Specifically, we defined utility as the percentage of effective data completed by transactional workloads or batching jobs that meet their corresponding SLOs. For transactional workloads, it is the ratio of data delivered by SLO-compliant requests to the total data. For batching jobs, it could be the data from finished jobs or the progress of unfinished jobs. NINEPIN considers instruction per cycle (IPC) as a good measure of performance for computation-bound batching jobs, and calculates their utility as the ratio of the co-hosting IPC

to the sole IPC. The system utility function is the sum of individual application utilities plus the utility of energy, though preference is given to the performance by discounting the weight of energy utility.

Model Predictive Control. The control goal is to steer the system into a state of optimum target tracking, while penalizing large changes in the control variables. It minimizes the deviation of application performance from their respective targets received from level-1 optimizer. The model predictive controller decides the control actions at every control period k by minimizing the following cost function:

$$V(k) = \sum_{i=1}^{H_p} ||\boldsymbol{r} - \boldsymbol{y}(k+i)||_P^2 + \sum_{j=0}^{H_c-1} ||\Delta \boldsymbol{u}(k+j)||_Q^2. \quad (2)$$

Here, $\boldsymbol{y}(k)$ is a vector containing the performance measure of each application. The controller uses the linearized state-space model to predict each application's performance over H_p control periods, called the prediction horizon. It computes a sequence of control actions $\Delta \boldsymbol{u}(k), \Delta \boldsymbol{u}(k+1), .., \Delta \boldsymbol{u}(k+H_c-1)$ over H_c control periods, called the control horizon, to keep the predicted performance close to their pre-defined targets $\boldsymbol{r}$. P and Q are the weighting matrices whose relative magnitude provides a way to trade off tracking accuracy for better stability in the control actions. NINEPIN linearizes the fuzzy model to a state-space linear time variant model and transforms the MIMO control to a standard quadratic programming problem.

4.2 Representative Results

We implemented NINEPIN on our university cloud testbed running VMware ESX 4.1. The testbed consists of HP ProLiant BL460C servers, each with dual Intel Xeon E5530 quad-core processors and 32 GB memory. We deployed the SPEC CPU2006 benchmark and the RUBiS multi-tier benchmark as the heterogeneous applications into different VMs. NINEPIN is implemented as a third-party resource management middleware that interacts with VMware's vSphere API. Since the benchmarks are primary CPU-bound, we set NINEPIN to control the CPU allocations. We compared NINEPIN with two other approaches, *Default* and *Q-Clouds*. Default refers to the static resource allocation imposed by VMware without tracking individual application performance. Q-Clouds is a representative performance isolation scheme recently proposed[94].

Fig.4 compares performance isolation in different approaches and draws the normalized application performance relative to running solo. We co-located homogeneous programs from the SPEC CPU2006 benchmark to study the effectiveness of NINEPIN in dealing with complex interference relationships. Fig.4(a) shows that applications running under NINEPIN have much more consistent performance compared with other approaches. NINEPIN successfully delivers stable performance for individual applications with average variations less than 5%. Besides predictable performance, NINEPIN also has significant performance improvement over Default and Q-Clouds by on average 38.6% and 25.2%, respectively (Fig.4(b)). Fig.4(c) reveals that NINEPIN's advantage is due to the flexible allocation of the CPU resource to applications by tracking the performance of individual applications.

Summary. The novel hierarchical control framework of NINEPIN aligns performance isolation goals with the incentive to regulate the system towards optimal operating conditions. The framework combines machine learning based self-adaptive modeling of performance interference and energy consumption, utility optimization based performance targeting, and a robust model predictive control based target tracking. Experimental results in [93] demonstrate that NINEPIN outperforms Q-Clouds, improving the overall system utility and reducing energy consumption.

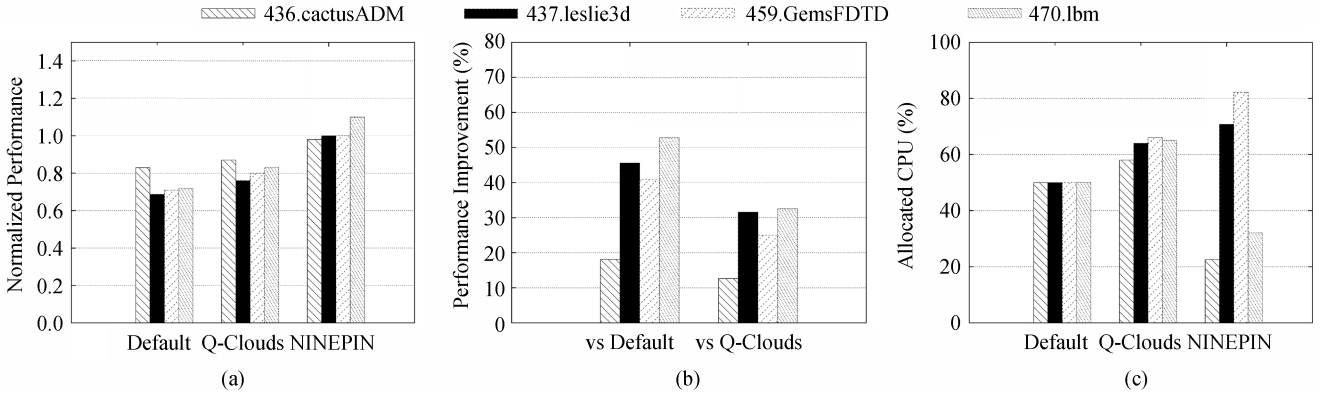


Fig.4. Performance isolation with Default, Q-Clouds, and NINEPIN. (a) Normalized performance. (b) Performance improvement. (c) Resource allocations.

5 Practice 3: Joint Performance and Power Guarantee in Virtualized Servers

Modern datacenters apply virtualization technology to host multiple Internet services that share underlying high density blade server resources for performance isolation, server consolidation, and system manageability. High density blade servers impose stringent power and cooling requirements. It is necessary to precisely and agilely control power consumption of servers to avoid system failures caused by power capacity overload or overheating. A common technique to server power consumption control is to dynamically transit hardware components from high-power states to low-power states whenever the system power consumption exceeds a given power budget[48]. However, it has significant influence on the performance of hosted applications as it may result in violation of service level agreements (SLAs) in terms of response time and throughput required by customers. Furthermore, such an approach is not easily applicable to virtualized environments where physical processors are shared by multiple virtual machines. Changing the power state of a processor will affect the performance of multiple virtual machines belonging to different applications. Thus, it may threaten the performance isolation property of virtualization technologies. It is important to consider a holistic approach in controlling power and performance in virtualized datacenters.

Many research studies focused on treating either power or performance as the primary control target in a datacenter while satisfying the other objective in a best-effort manner. Power-oriented approaches[8,48,50,59-60] disregard the SLAs of hosted applications while performance-oriented approaches do not have explicit control on power consumption[18,31,65-67]. Co-Con[20] and vPnP[19] were designed for explicit coordination of power and performance in virtualized datacenters using utility function optimization. Such approaches can achieve different levels of trade-off between power and performance in a flexible way. However, they lack the guarantee on stability and performance of the server system in the face of highly dynamic workloads. That could lead to state-flapping[1], a scenario where oscillations occur between system states that result in poor power and performance assurance. Percentile-based performance assurance and multi-service architectures further impose significant complexity and challenges.

5.1 Fuzzy MIMO Control Based Approach

In studies of [71,73], we proposed and developed a prototype control framework PERFUME for joint power and performance management on virtualized server clusters. Fig.5 illustrates its system architecture. The computer system under control is a virtualized blade server cluster hosting multiple multi-tier applications. Each tier of an application is deployed in a virtual machine (VM) created from a resource pool, which logically abstracts resources provided by the underlying physical blade servers.

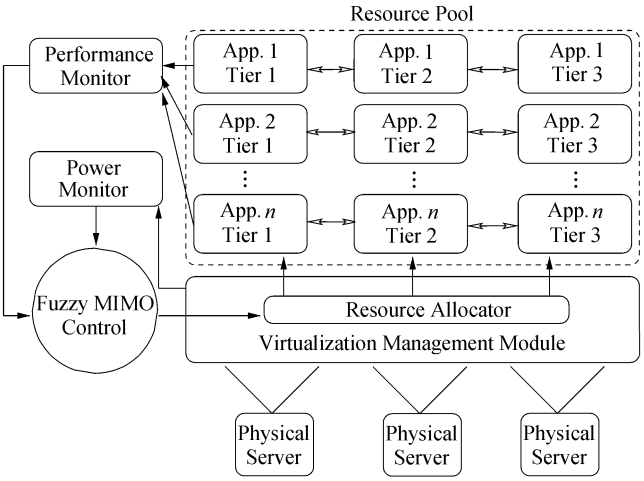


Fig.5. PERFUME's system architecture.

The power monitor periodically measures the average power consumption of the virtualized server cluster at the resource pool level and sends the data to the control module. The performance monitor periodically measures the average throughput and the average response time of each application and sends the performance values to the control module. The control module determines the CPU usage limits on various tiers to regulate per-application performance and the total power consumption of the virtualized server cluster. The resource allocator actuates control actions to limit the CPU usage of each VM.

It is important but very challenging to determine the quantitative and dynamic relationship between the controlled variables (i.e., power and performance) and the manipulated variables (e.g., VM CPU limit), due to the inherent nonlinearity of performance and power with resource allocation, workload dynamics, and the shared virtualized infrastructure. PERFUME applies fuzzy MIMO modeling to estimate the relationship between the performance and CPU usage limits on the VMs. It also applies the modeling technique to predict the power consumption of the resource pool for different VMs' CPU limits. A key strength of fuzzy model is its capability to represent highly complex and nonlinear systems by a combination of inter-linked subsystems with simple functional dependencies. The model predictive MIMO control incorporates the optimization of various objectives in deciding the control actions. It is able to manipulate multiple variables of a system for

controlling multiple outputs while considering the impact of their complex interactions.

The core of the prototype framework is the fuzzy MIMO control module that aims to minimize the deviation of power consumption and achieved performance of multi-tier applications from their respective targets. It determines the control actions at every control period k by minimizing the cost function:

$$V(k)=\sum_{i=1}^{H_p}||r1-y_1(k+i)||_P^2+\sum_{i=1}^{H_p}||r2-y_2(k+i)||_Q^2+\sum_{j=0}^{H_c-1}||\Delta \boldsymbol{u}(k+j)||_R^2. \quad (3)$$

Here, $y_1(k)$ is the power consumption of the resource pool of the underlying server cluster. $y_2(k)$ is a vector containing the performance of each application, i.e., response time or throughput. The controller predicts both power and performance over H_p control periods, called the *prediction horizon*. It computes a sequence of control actions $\Delta \boldsymbol{u}(k), \Delta \boldsymbol{u}(k+1), \ldots, \Delta \boldsymbol{u}(k+H_c-1)$ over H_c control periods, called the *control horizon*, to keep the predicted power and performance close to their pre-defined targets $r1$ and $r2$ respectively. The control action is the change in CPU usage limit imposed on the VMs. P and Q are the tracking error weights that determine the trade-off between power and performance. The weights can be set by a datacenter administrator. The third term in (3) represents the control penalty and is weighted by R. It penalizes big changes in control action and contributes towards high system stability. The control problem is subject to the constraint that the sum of CPU usage limits assigned to all multi-tier applications must be bounded by the total CPU capacity of the resource pool.

5.2 Representative Results

We linearized the obtained fuzzy model as a state-space linear time variant model. We transformed the MIMO control problem to a standard quadratic programming problem, and solved the quadratic programming problem based on the MATLAB. We implemented PERFUME on a testbed consisting of two HP ProLiant BL460C G6 blade server modules. The power monitor and the fuzzy MIMO control modules interact with the VMware VMM via vSphere API 4.1. As vPnP[19], PERFUME hosts two RUBiS multi-tier benchmark applications in the testbed experiments for performance comparison.

A key feature of PERFUME is its ability to assure joint power and performance guarantee with flexible trade-offs while assuring control accuracy and system stability. The trade-offs between inherently conflicting power and performance objectives can be specified by a datacenter administrator. The system stability is measured in terms of relative deviation of power and performance from their respective targets, as defined in vPnP[19]. We experimented with power-preferred, performance-preferred and balanced control options under a highly dynamic workload[12]. Fig.6(a) shows the changes in the number of concurrent users.

PERFUME achieves the specified trade-offs by tuning the tracking error weights, P and Q, in the MIMO control objective defined by (3). Fig.6(b) compares the control accuracy of vPnP with PERFUME in assuring the throughput target for various trade-offs between power and performance. The results demonstrate that, compared with vPnP, PERFUME delivers average improvement of 30% in performance assurance in terms of relative deviation for various control options. We obtained similar results with the average improvement of 25% for relative deviation in power consumption with respect to its power cap target, as shown in Fig.6(c). Note that the control accuracy of the power-preferred option is the highest for power assurance but the lowest for throughput assurance. Whereas, the control accuracy of the performance-preferred option is the highest for throughput assurance and the lowest for power assurance. The balanced control option shows good con-

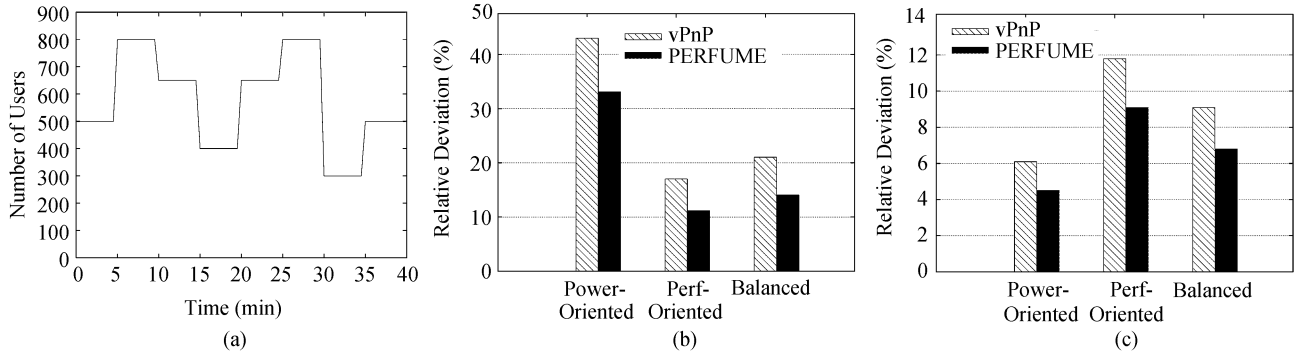


Fig.6. Power and performance assurance with flexible control options under a highly dynamic workload. (a) Highly dynamic workload. (b) Throughput. (c) Power consumption.

trol accuracy for both power and performance assurance.

Summary. This practice[73] shows PERFUME can effectively control both the throughput and percentile-based response time of multi-tier applications due to its novel self-adaptive fuzzy modeling that integrates the strengths of fuzzy logic, MIMO control, and artificial neural network.

6 Research Trends in Sustainable Cloud Computing in Green Datacenters

6.1 Improving Energy Efficiency in Datacenters

Energy efficiency is a fundamental consideration when managing computing infrastructures and services. This is due to the economic issues derived from increasing energy rates and the environmental impacts. Often a large amount of energy consumed by a computing infrastructure is the result of inefficiencies in its operation and administration. Therefore, lots of efforts towards energy efficient datacenter operation focus on improving the efficiency in one or all of the three major datacenter components: computing infrastructure, cooling facility, and power supply equipment.

One research trend focuses on improving energy efficiency of the computing infrastructure for sustainable cloud computing. Cloud computing can improve resource utilization efficiency by server virtualization and consolidation that allows cloud providers to run multiple workloads from different customers on the same computing infrastructure. Sustainable cloud computing needs to deal with energy consumption of the underlying computing infrastructure as well as performance requirement of the provisioned services. Cloud systems are multi-tenant and workloads are heterogeneous, e.g., response time critical applications such as e-transactional Web applications and batch-style applications such as MapReduce jobs. As cloud systems become more complex, there is an increasing number of resources shared between tenants and multi-tenant interference has to be mitigated for performance predictability. There are great interests in designing power and heterogeneity aware algorithms for energy efficiency and performance guarantee in multi-tenant cloud environments.

6.2 Applying Renewable Energy for Sustainability in Datacenters

Today, major cloud service operators have taken various initiatives to operate their datacenters with renewable energy partially or completely. Google, Facebook, and Apple have started to build their own green power plants to support the operation of their sustainable datacenters. Researchers envision that datacenters at cluster level can be completely powered by renewable energy, e.g., solar and wind, and be self-sustainable. Most green power plants use wind turbines and/or solar panels for power generation. Unlike traditional energy resources, the availability of renewable energy varies widely during the times of a day, seasons of the year, and the geographical locations of the power plants. Such intermittency makes it very hard for datacenters to effectively use renewable energy.

On the other hand, the power demand of a datacenter is highly dependent on the resource requirements of hosted workloads. The availability and cost of the power supply, e.g., renewable energy supply and electricity price, is often dynamic over time. Thus, autonomic resource provisioning and workload management can have great impacts on renewable energy consumption and cost reduction for improving sustainability, e.g., scheduling deadline non-critical service demands in a manner that follows the availability of the renewable energy generation, and co-locating heterogeneous workloads on the shared computing infrastructure for resource utilization efficiency and energy efficiency.

7 Conclusions

Virtualized datacenter servers, the platform for supporting cloud computing, allow diverse applications to share the underlying server resources. Due to the highly dynamic nature of Internet workloads, increasing complexity of applications, and complex dynamics of shared infrastructure, there are significant challenges in managing application performance while maintaining resource utilization efficiency and reducing power consumption costs. This paper surveys representative approaches to autonomic performance and power control on virtualized servers. To this end, we introduced three approaches we recently designed for autonomic resource management on virtualized servers based on machine learning and control, and identified two research trends in autonomic and sustainable cloud computing in green datacenters.

References

[1] Huebscher M C, McCann J A. A survey of autonomic computing: Degrees, models, and applications. *ACM Computing Surveys*, 2008, 40(3), Article No.7.

[2] Mi N, Casale G, Cherkasova L, Smirni E. Burstiness in multi-tier applications: Symptoms, causes, and new models. In *Proc. the 9th ACM/IFIP/USENIX Int. Middleware Conference* (*Middleware*), Dec. 2008, pp.265-286.

[3] Caniff A, Lu L, Mi N, Cherkasova L, Smirni E. Fastrack for taming burstiness and saving power in multi-tiered systems. In *Proc. the 22nd Int. Teletraffic Congress* (*ITC*), Sept. 2010.

[4] Singh R, Sharma U, Cecchet E, Shenoy P. Autonomic mix-aware provisioning for non-stationary data center workloads. In *Proc. the 7th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2010, pp.21-30.

[5] Sugerman J, Venkitachalam G, Lim B H. Virtualizing I/O devices on VMware workstation's hosted virtual machine monito. In *Proc. USENIX Annual Technical Conference*, June 2001.

[6] Barham P, Dragovic B, Fraser K *et al.* Xen and the art of virtualization. In *Proc. the 19th ACM Symposium on Operating Systems Principles* (*SOSP*), Oct. 2003, pp.164-177.

[7] Menascé D A, Bennani M N. Autonomic virtualized environments. In *Proc. IEEE Int. Conference on Autonomic and Autonomous Systems*, July 2006.

[8] Nathuji R, Schwan K. Virtualpower: Coordinated power management in virtualized enterprise systems. In *Proc. the 21st ACM Symposium on Operating Systems Principles* (*SOSP*), Oct. 2007, pp.265-278.

[9] Urgaonkar B, Shenoy P, Chandra A *et al.* Agile dynamic provisioning of multi-tier Internet applications. *ACM Trans. Autonomous and Adaptive Systems*, 2008, 3(1): 1-39.

[10] Lama P, Zhou X. Efficient server provisioning for end-to-end delay guarantee on multi-tier clusters. In *Proc. the 17th IEEE Int. Workshop on Quality of Service* (*IWQoS*), July 2009.

[11] Welsh M, Culler D. Adaptive overload control for busy Internet servers. In *Proc. the 4th USENIX Symposium on Internet Technologies and Systems* (*USITS*), Mar. 2003.

[12] Lama P, Zhou X. Autonomic provisioning with self-adaptive neural fuzzy control for end-to-end delay guarantee. In *Proc. IEEE/ACM Int. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (*MASCOTS*), Aug. 2010, pp.151-160.

[13] Vaid K. Datacenter power efficiency: Separating fact from fiction (keynote). In *the USENIX Workshop on Power Aware Computing and Systems* (*HotPower*), Oct. 2010.

[14] Greenberg A, Hamilton J R, Jain N *et al.* Vl2: A scalable and flexible data center network. In *Proc. ACM SIGCOMM*, Aug. 2009, pp.51-62.

[15] Tam D K, Azimi R, Soares L B, Stumm M. RapidMRC: Approximating L2 miss rate curves on commodity systems for online optimizations. In *Proc. the 14th Int. Conference on Architecture Support for Programming Language and Operating System* (*ASPLOS*), March 2009, pp.121-132.

[16] Zhang X, Dwarkadas S, Shen K. Towards practical page coloring-based multicore cache management. In *Proc. the 4th ACM European Conference on Computer Systems* (*EuroSys*), April 2009, pp.89-102.

[17] Xie Y, Loh G H. Pipp: Promotion/insertion pseudo-partitioning of multi-core shared caches. In *Proc. the 36th Int. Symposium on Computer architecture* (*ISCA*), June 2009, pp.174-183.

[18] Wang Y, Wang X, Chen M, Zhu X. PARTIC: Power-aware response time control for virtualized Web servers. *IEEE Trans. Parallel and Distributed Systems*, 2011, 22(2): 323-336.

[19] Gong J, Xu C Z. vPnP: Automated coordination of power and performance in virtualized datacenters. In *Proc. IEEE Int. Workshop on Quality of Service* (*IWQoS*), June 2010.

[20] Wang X, Wang Y. Co-Con: Coordinated control of power and application performance for virtualized server clusters. In *Proc. the 17th IEEE Int. Workshop on Quality of Service* (*IWQoS*), July 2009.

[21] Stewart C, Shen K. Performance modeling and system management for multi-component online services. In *Proc. the 2nd USENIX Symposium on Networked Systems Design and Implementation* (*NSDI*), May 2005, Vol.2, pp.71-84.

[22] Villela D, Pradhan P, Rubenstein D. Provisioning servers in the application tier for e-commerce systems. *ACM Trans. Internet Technology*, 2007, 7(1): Article No.7.

[23] Diao Y, Hellerstein J L, Parekh S *et al.* Controlling quality of service in multi-tier Web applications. In *Proc. the 26th IEEE Int. Conference on Distributed Computing Systems* (*ICDCS*), July 2006.

[24] Liu X, Heo J, Sha L, Zhu X. Queueing-model-based adaptive control of multi-tiered Web applications. *IEEE Transactions on Network and Service Management*, 2008, 5(3): 157-167.

[25] Abdelzaher T F, Shin K G, Bhatti N. Performance guarantees for Web server end-systems: A control-theoretical approach. *IEEE Trans. Parallel and Distributed Systems*, 2002, 13(1): 80-96.

[26] Lu Y, Abdelzaher T F, Lu C, Sha L, Liu X. Feedback control with queueing-theoretic prediction for relative delay guarantees in Web servers. In *Proc. the 9th IEEE Real-Time and Embedded Technology and Applications Symposium* (*RTAS*), May 2003, pp.208-218.

[27] Kamra A, Misra V, Nahum E M. Yaksha: A self-tuning controller for managing the performance of 3-tiered Web sites. In *Proc. the 12th IEEE Int. Workshop on Quality of Service* (*IWQoS*), June 2004, pp.47-56.

[28] Lama P, Zhou X. aMOSS: Automated multi-objective server provisioning with stress-strain curving. In *Proc. IEEE Int. Conference on Parallel Processing* (*ICPP*), Sept. 2011, pp.345-354.

[29] Padala P, Hou K Y, Shin K G *et al.* Automated control of multiple virtualized resources. In *Proc. EuroSys Conference* (*EuroSys*), April 2009, pp.13-26.

[30] Padala P, Shin K G, Zhu X *et al.* Adaptive control of virtualized resources in utility computing environments. In *Proc. EuroSys Conference* (*EuroSys*), March 2007, pp.289-302.

[31] Leite J C B, Kusic D M, Mossé D, Bertini L. Stochastic approximation control of power and tardiness in a three-tier Web-hosting cluster. In *Proc. the 7th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2010, pp.41-50.

[32] Watson B J, Marwah M, Gmach D *et al.* Probabilistic performance modeling of virtualized resource allocation. In *Proc. the 7th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2010, pp.98-108.

[33] Rao J, Xu C. Online measurement of the capacity of multi-tier Websites using hardware performance counters. In *Proc. the 28th IEEE Int. Conference on Distributed Computing Systems* (*ICDCS*), June 2008, pp.705-712.

[34] Guo Y, Lama P, Rao J, Zhou X. V-cache: Towards flexible resource provisioning for clustered applications in IaaS clouds. In *Proc. the 27th IEEE Int. Parallel and Distributed Processing Symposium* (*IPDPS*), May 2013, pp.88-99.

[35] Rao J, Bu X, Xu C *et al.* VCONF: A reinforcement learning approach to virtual machines auto-conguration. In *Proc. the 6th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2009, pp.137-146.

[36] Rao J, Wei Y, Gong J, Xu C Z. DynaQoS: Model-free self-tuning fuzzy control of virtualized resources for QoS provisioning. In *Proc. the 19th Int. Workshop on Quality of Service* (*IWQoS*), June 2011.

[37] Muppala S, Chen G, Zhou X. Multi-tier service differentiation: Coordinated resource provisioning and admission control. In *Proc. the 18th IEEE Int. Conference on Parallel and Distributed Systems* (*ICPADS*), December 2012, pp.69-76.

[38] Muppala S, Zhou X. Coordinated session-based admission control with statistical learning for multi-tier Internet applications. *Journal of Network and Computer Applications*, 2011, 34(1): 20-29.

[39] Muppala S, Zhou X, Zhang L, Chen G. Regression-based resource provisioning for session slowdown guarantee in multi-

tier Internet servers. *Journal of Parallel and Distributed Computing*, 2012, 72(3): 362-375.

[40] Tesauro G, Jong N K, Das R, Bennani M N. A hybrid reinforcement learning approach to autonomic resource allocation. In *Proc. the 3rd IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2006, pp.65-73.

[41] Zhang Q, Cherkasova L, Smirni E. A regression-based analytic model for dynamic resource provisioning of multi-tier Internet applications. In *Proc. the 4th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2007, Article No.27.

[42] Guo Y, Lama P, Zhou X. Automated and agile server parameter tuning with learning and control. In *Proc. the 18th IEEE Int. Parallel and Distributed Processing Symposium* (*IPDPS*), May 2012, pp.656-667.

[43] Guo Y, Lama P, Jiang C, Zhou X. Automated and agile server parameter tuning by coordinated learning and control. *IEEE Transactions on Parallel and Distributed Systems*, 2014, 25(4): 876-886.

[44] Guo Y, Zhou X. Coordinated VM resizing and server tuning: Throughput, power efficiency and scalability. In *Proc. the 20th IEEE Int. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (*MASCOTS*), August 2012, pp.289-297.

[45] Unsal O, Koren I. System-level power-aware design techniques in real-time systems. *Proc. IEEE*, 2003, 91(7): 1055-1069.

[46] Elnozahy M, Kistler M, Rajamony R. Energy conservation policies for Web servers. In *Proc. the 4th USENIX Symp. Internet Technologies and Systems* (*USITS*), March 2003.

[47] Lefurgy C, Wang X, Ware M. Server-level power control. In *Proc. the 4th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2007, Article No.4.

[48] Gandhi A, Harchol-Balter M, Das R, Lefurgy C. Optimal power allocation in server farms. In *Proc. the 11th ACM SIGMETRICS*, June 2009, pp.157-168.

[49] Horvath T, Abdelzaher T, Skadron K, Liu X. Dynamic voltage scaling in multitier Web servers with end-to-end delay control. *IEEE Trans. Computers*, 2007, 56(4): 444-458.

[50] Wang X, Chen M, Fu X. MIMO power control for high-density servers in an enclosure. *IEEE Trans. Parallel and Distributed Systems*, 2010, 21(10): 1412-1426.

[51] Wang X, Chen M, Lefurgy C, Keller T W. Ship: Scalable hierarchical power control for large-scale data centers. In *Proc. the 18th Int. Conference on Parallel Architectures and Compilation Techniques* (*PACT*), September 2009, pp.91-100.

[52] Nathuji R, Isci C, Gorbatov E. Exploiting platform heterogeneity for power efficient data centers. In *Proc. the 4th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2007, Article No.5.

[53] Das R, Kephart J O, Lenchner J, Hamann H. Utility-function-driven energy-efficient cooling in data centers. In *Proc. the 7th IEEE Int. Conference on Autonomic computing* (*ICAC*), June 2010, pp.61-70.

[54] Fu X, Wang X, Lefurgy C. How much power oversubscript is safe and allowed in data centers? In *Proc. the 8th IEEE Int. Conf. Autonomic computing* (*ICAC*), June 2011, pp.21-30.

[55] Gmach D, Rolia J, Cherkasova L. Resource and virtualization costs up in the cloud: Models and design choices. In *Proc. IEEE/IFIP Int. Conference on Dependable Systems and Networks* (*DSN*), June 2011, pp.395-402.

[56] Goiri F, Le K, Guitart J *et al.* Intelligent placement of datacenters for Internet services. In *Proc. IEEE Int. Conference on Distributed Computing Systems* (*ICDCS*), June 2011, pp.136-142.

[57] Meng X, Isci C, Kephart J *et al.* Efficient resource provisioning in compute clouds via VM multiplexing. In *Proc. the 7th Int. Conference on Autonomic Computing* (*ICAC*), June 2010, pp.11-20.

[58] Thereska E, Donnelly A, Narayanan D. Sierra: Practical power-proportionality for data center storage. In *Proc. the 6th EuroSys Conference* (*EuroSys*), April 2011, pp.169-182.

[59] Urgaonkar P, Urgaonkar B, Neely M J *et al.* Optimal power cost management using stored energy in data centers. In *Proc. ACM SIGMETRICS*, June 2011, pp.221-232.

[60] Raghavendra R, Ranganathan P, Talwar V *et al.* No power struggles: Coordinated multi-level power management for the data center. In *Proc. the 13th ACM ASPLOS*, March 2008, pp.48-59.

[61] Fan X, Weber W D, Barroso L A. Power provisioning for a warehouse-sized computer. *ACM SIGARCH*, 2007, 35(2): 13-23.

[62] Govindan S, Choi J, Urgaonkar B *et al.* Statistical profiling-based techniques for effective power provisioning in data centers. In *Proc. EuroSys Conference* (*EuroSys*), April 2009, pp.317-330.

[63] Verma A, Ahuja P, Neogi A. pMapper: Power and migration cost aware application placement in virtualized systems. In *Proc. the 9th ACM/IFIP/USENIX Int. Middleware Conference* (*Middleware*), December 2008, pp.243-264.

[64] Kumar S, Talwar V, Kumar V *et al.* vManage: Loosely coupled platform and virtualization management in data centers. In *Proc. the 5th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2009, pp.127-136.

[65] Jiang C, Xu X, Wan J *et al.* Power aware job scheduling with QoS guarantees based on feedback control. In *Proc. the 18th IEEE IWQoS*, June 2010.

[66] Kusic D, Kephart J O, Hanson J E *et al.* Power and performance management of virtualized computing environments via lookahead control. In *Proc. IEEE Int. Conference on Autonomic computing* (*ICAC*), June 2008, pp.3-12.

[67] Le K, Bianchiniy R, Martonosiz M, Nguyeny T D. Cost- and energy-aware load distribution across data centers. In *Proc. Workshop on Power Aware Computing and Systems* (*HotPower*), October 2009.

[68] Cheng D, Guo Y, Zhou X. Self-tuning batching with DVFS for improving performance and energy efficiency in servers. In *Proc. the 21st IEEE/ACM Int. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (*MASCOTS*), August 2013, pp.40-49.

[69] Jung G, Hiltunen M A, Joshi K R *et al.* Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In *Proc. IEEE Int. Conf. Distributed Computing Systems* (*ICDCS*), June 2010, pp.62-73.

[70] Kansal A, Liu J, Singh A *et al.* Semantic-less coordination of power management and application performance. *ACM SIGPOS*, 2010, 44(1): 66-70.

[71] Lama P, Zhou X. PERFUME: Power and performance guarantee with fuzzy mimo control in virtualized servers. In *Proc. the 19th IEEE Int. Workshop on Quality of Service* (*IWQoS*), June 2011.

[72] Riska A, Smirni E. Autonomic exploration of trade-offs between power and performance in disk drives. In *Proc. the 7th IEEE Int. Conference on Autonomic Computing* (*ICAC*), June 2010, pp.131-140.

[73] Lama P, Zhou X. Coordinated power and performance guarantee with fuzzy mimo control in virtualized server clusters. *IEEE Transactions on Computers*, 2014. (to be appeared)

[74] Lama P, Guo Y, Zhou X. Autonomic performance and power control for co-located web applications on virtualized servers. In *Proc. the 21st ACM/IEEE Int. Workshop on Quality of Service* (*IWQoS*), June 2013, pp.63-72.

[75] Lama P, Li Y, Aji A *et al.* pVOCL: Power-aware dynamic placement and migration in virtualized GPU environments. In *Proc. the 33rd IEEE Int. Conference on Distributed Computing Systems* (*ICDCS*), June 2013.

[76] Aksanli B, Venkatesh J, Zhang L, Rosing T. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. In *Proc. the 4th USENIX Workshop on Power Aware Computing and Systems* (*HotPower*), October 2011, Article No.5.

[77] Deng N, Stewart C, Gmach D *et al.* Adaptive green hosting. In *Proc. the 9th IEEE Int. Conference on Autonomic Computing* (*ICAC*), Sept. 2012, pp.135-144.

[78] Liu Z, Lin M, Wierman A *et al.* Greening geographical load balancing. In *Proc. ACM SIGMETRICS*, June 2011, pp.233-244.

[79] Ren S, He Y. COCA: Online distributed resource management for cost minimization and carbon neutrality in data centers. In *Proc. Supercomputing* (*SC*), Nov. 2013, Article No.39.

[80] Zhang Y, Wang Y, Wang X. Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In *Proc. the 12th ACM/IFIP/USENIX Int. Conference on Middleware* (*Middleware*), December 2011, pp.143-164.

[81] Goiri I, Le K, Nguyen T D *et al.* Greenhadoop: Leveraging green energy in data-processing frameworks. In *Proc. the 7th ACM European Conference on Computer Systems* (*EuroSys*), April 2012, pp.57-70.

[82] Liu S, Ren S, Gang Q *et al.* Profit aware load balancing for distributed cloud data centers. In *Proc. the 27th IEEE Int. Parallel and Distributed Processing Symposium* (*IPDPS*), May 2013, pp.611-622.

[83] Gmach D, Rolia J, Bash C *et al.* Capacity planning and power management to exploit sustainable energy. In *Proc. the 6th IEEE Int. Conference on Network and Service Management* (*CNSM*), October 2010, pp.96-103.

[84] Li C, Zhou R, Li T. Enabling distributed generation powered sustainable high-performance data center. In *Proc. the 19th IEEE Int. Symposium on High-Performance Computer Architecture* (*HPCA*), Feb. 2013, pp.35-46.

[85] Wang Y, Chen R, Shao Z, Li T. Solartune: Real-time scheduling with load tuning for solar energy powered multicore systems. In *Proc. the 19th IEEE RTCSA*, Aug. 2013, pp.101-110.

[86] Wang D, Ren C, Sivasubramaniam A *et al.* Energy storage in datacenters: What, where, and how much? In *Proc. ACM SIGMETRICS*, June 2012, pp.187-198.

[87] Cheng D, Jiang C, Zhou X. Heterogeneity-aware workload placement and migration in distributed sustainable datacenters. In *Proc. the 28th IEEE Int. Parallel and Distributed Processing Symposium* (*IPDPS*), May 2014.

[88] Lama P, Zhou X. Efficient server provisioning with control for end-to-end delay guarantee on multi-tier clusters. *IEEE Trans. Parallel and Distributed Systems*, 2012, 23(1): 78-86.

[89] Lama P, Zhou X. Autonomic provisioning with self-adaptive neural fuzzy control for percentile-based delay guarantee. *ACM Transactions on Autonomous and Adaptive Systems*, 2013, 8(2): Article No.9.

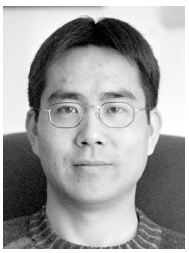
[90] Lin C, Lee C S G. Real-time supervised structure/parameter learning for fuzzy neural network. In *Proc. IEEE Int. Conference on Fuzzy Systems*, March 1992, pp.1283-1291.

[91] Lin J, Lu Q, Ding X *et al.* Gaining insights into multicore cache partitioning: Bridging the gap between simulation and real systems. In *Proc. the 14th Int. Symp. High Performance Computer Architecture* (*HPCA*), Feb. 2008, pp.367-378.

[92] Rao J, Wang K, Zhou X, Xu C Z. Improving virtual machine scheduling in NUMA multicore systems. In *Proc. the 19th IEEE Int. Symposium on High Performance Computer Architecture* (*HPCA*), Feb. 2013, pp.306-317.

[93] Lama P, Zhou X. NINEPIN: Non-invasive and energy efficient performance isolation in virtualized servers. In *Proc. IEEE/IFIP Int. Conference on Dependable Systems and Networks* (*DSN*), June 2012.

[94] Nathuji R, Kansal A, Ghaffarkhah A. Q-clouds: Managing performance interference effects for QoS-aware clouds. In *Proc. the 5th ACM European Conference on Computer Systems* (*EuroSys*), April 2010, pp.237-250.



Xiaobo Zhou is a professor and the chair of computer science, University of Colorado, Colorado Springs. He received the B.S., M.S., and Ph.D. degrees in computer science from Nanjing University, in 1994, 1997, and 2000, respectively. He was a post-doctoral researcher at the University of Paderborn in 2000. His research lies broadly in computer network systems, more specifically, autonomic and sustainable computing in datacenters, Cloud computing, server virtualization, scalable Internet services and architectures. His research was supported in part by the US NSF and NSF of China. He was a recipient of the US NSF CAREER AWARD in 2009, and the University Faculty Award for Excellence in Research in 2011. He has served as a general chair of ICCCN 2014, ICCCN 2012, a program chair of CCGrid 2015 and ICCCN 2011, a program vice chair of CCGrid 2014, GLOBECOM 2010, ICCCN 2009, HPCC 2008, and EUC 2008. He serves on the editorial board of the Elsevier's Computer Communications. He is a member of the ACM and a senior member of the IEEE.

Chang-Jun Jiang is a professor with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai. He received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1995 and conducted post-doctoral research at the Institute of Computing Technology, Chinese Academy of Sciences, in 1997. He is a council member of China Automation Federation and Artificial Intelligence Federation, the director of Professional Committee of Petri Net of China Computer Federation, and the vice director of Professional Committee of Management Systems of China Automation Federation. He was a visiting professor of Institute of Computing Technology, Chinese Academy of Science, a research fellow of the City University of Hong Kong, and an information area specialist of Shanghai Municipal Government. His current areas of research are concurrent theory, Petri net, and formal verification of software, concurrency processing and intelligent transportation systems. He is a general chair of ICCCN 2014. He is a member of the CCF and IEEE.