PERFUME: Power and Performance Guarantee with Fuzzy MIMO Control in Virtualized Servers

Palden Lama and Xiaobo Zhou
Department of Computer Science
University of Colorado at Colorado Springs, CO 80918, USA
{plama, xzhou}@uccs.edu

***Abstract*—It is important but challenging to assure the performance of multi-tier Internet applications with the power consumption cap of virtualized server clusters mainly due to system complexity of shared infrastructure and dynamic and bursty nature of workloads. This paper presents PERFUME, a system that simultaneously guarantees power and performance targets with flexible tradeoffs while assuring control accuracy and system stability. Based on the proposed fuzzy MIMO control technique, it accurately controls both the throughput and percentile-based response time of multi-tier applications due to its novel fuzzy modeling that integrates strengths of fuzzy logic, MIMO control and artificial neural network. It is self-adaptive to highly dynamic and bursty workloads due to online learning of control model parameters using a computationally efficient weighted recursive least-squares method. We implement PERFUME in a testbed of virtualized blade servers hosting two multi-tier RUBiS applications. Experimental results demonstrate its control accuracy, system stability, flexibility in selecting tradeoffs between conflicting targets and robustness against highly dynamic variation and burstiness in workloads. It outperforms a representative utility based approach in providing guarantee of the system throughput, percentile-based response time and power budget in the face of highly dynamic and bursty workloads.**

I. Introduction

Modern data centers apply virtualization technology to host multiple Internet applications that share underlying high density server resources for performance isolation, server consolidation, and system manageability. The widely used high density blade servers impose stringent power and cooling requirements. It is essential to precisely control power consumption of blade servers to avoid system failures caused by power capacity overload. A common technique to server power consumption control is to dynamically transition the hardware components from high power states to low-power states whenever the system power consumption exceeds a given power budget [4]. 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, power management may threaten the performance isolation of hosted applications. It is important to consider a holistic approach in controlling power and performance in virtualized data centers.

Recent studies such as [22] found highly dynamic workloads of Internet services that fluctuate over multiple time scales, which can have a significant impact on the processing demands imposed on data center servers. Furthermore, burstiness of Internet workloads has deleterious impact on client-perceived performance [18]. It is challenging to design autonomic resource provisioning techniques that are robust to dynamic variation and burstiness in workloads.

Many research studies focused on treating either power or performance as the primary control target in a data center while satisfying the other objective in a best-effort manner. Power oriented approaches [15], [19], [21], [25] disregard the SLAs of hosted applications while performance oriented approaches do not have explicit control on power consumption [2], [27]. Recently, vPnP [5] was proposed for explicit coordination of power and performance in virtualized data centers using utility function optimization. Such an approach can achieve different levels of tradeoff between power and performance in a flexible way. However, it lacks the guarantee on stability and performance of the server system especially in the face of highly dynamic and bursty workloads.

Multiple-input-multiple-output (MIMO) control technique has been applied for performance management of multi-tier applications [27] and power control of high density servers in an enclosure [25]. However, those MIMO control solutions do not provide explicit coordination between power and performance. Furthermore, they are designed based on offline system identification for specific workloads [11], [25], [27]. Hence, they are not adaptive to situations with abrupt workload changes though they can achieve control accuracy and system stability within a range theoretically.

An important goal in data centers is to meet the SLAs with customers. Many studies focused on the average end-to-end response time within a multi-tier system [1], [5], [9], [20], [23]. However, the average response time guarantee is not sufficient for many applications, in particular for interactive ones as it is unable to represent the shape of a delay curve [17]. Instead, providers of such services prefer percentile-based performance guarantee. Metric such as the 95_{th}-percentile response time has the benefit that is both easy to reason

978-1-4577-0103-0/11/$26.00

about and to capture individual users' perception of Internet service quality [16], [17], [23], [28]. But it is challenging to control the percentile-based performance, even without a power consumption cap, due to its strong non-linear relation with resource allocation and workload dynamics [13].

In this paper, we design and implement PERFUME, a system that simultaneously provides explicit guarantee on the power consumption of underlying server clusters and the performance of multi-tier applications in a prototype virtualized data center. We develop a fuzzy MIMO (FUMI) control to minimize the deviation of power and performance from their respective targets while assuring control accuracy and system stability. We recognize that it may not always be possible to simultaneously meet both power and performance targets due to bursty Internet workloads. PERFUME provides the flexibility to select varying tradeoffs between power and performance. It is capable of dealing with the complexity of multi-tier applications in a shared virtualized infrastructure due to its FUMI control that integrates fuzzy modeling logic, MIMO control and artificial neural network. The control action is taken by adjusting the CPU usage limits among individual tiers of multiple applications in a coordinated manner.

PERFUME provides performance guarantee for throughput and percentile-based response time in the face of highly dynamic and bursty workloads. It captures the nonlinearity of percentile-based performance metric such as the 95_{th}-percentile response time by applying the novel FUMI control. The FUMI control is also applied to predict the power consumption of underlying server clusters for various CPU usage limits in hosted applications. PERFUME is self-adaptive to highly dynamic workloads due to its online learning capability. It automatically learns the fuzzy model parameters at run time using a weighted recursive least-squares (wRLS) method. Compared to a standard least-squares method used in vPnP [5], wRLS method is computationally more efficient. It eliminates the need to find a suitable moving window size for collecting the data used for online learning of the system model.

We implement PERFUME on a testbed of virtualized server clusters hosting two RUBiS applications [5], [20], [23], [28]. The testbed consists of a cluster of two HP ProLiant BL460C G6 blade server modules using VMware virtual machines. Experimental results demonstrate that PERFUME's FUMI model significantly outperforms a recently applied modeling technique for web systems, ARMA (Auto Regressive Moving Average) [5], [20] in terms of prediction accuracy for throughput, percentile-based response time and power average consumption for both stationary and dynamic workloads.

Compared to vPnP [5], PERFUME delivers significantly improved performance, in terms of power, throughput and response time assurance with respect to the given targets in the face of highly dynamic and bursty workloads. This is due to its modeling accuracy, self-adaptiveness and control theoretical foundation. Note that vPnP was originally applied to a single tier, which was identified as the bottleneck of a Web application. However, in practice, the bottleneck tier can switch between multiple tiers depending on workload patterns [1]. For fair comparison, we extend the vPnP implementation to a multi-tier application. We also demonstrate that PERFUME delivers consistent performance for various flexible control options that tradeoffs between power and performance guarantee.

In the following, Section II discusses related work. The PERFUME system architecture is presented in Section III. Section IV describes the modeling of power and performance control. Section V discusses the design of FUMI control. Section VI provides the testbed implementation details. Section VII presents the experimental results and analysis. We conclude the paper with future work in Section VIII.

II. Related Work

Power management in computing systems is an important and challenging research area. There were many studies in power management in embedded mobile devices and Web servers. For instance, the Dynamic Voltage Scaling (DVS) technique was applied to reduce power consumption in Web servers [3] and to improve power efficiency of server farms [4].

Today, popular Internet applications have a multi-tier architecture forming server pipelines. Applying independent DVS algorithms in a pipeline will lead to inefficient usage of power for assuring an end-to-end delay guarantee due to the inter-tier dependency [7]. Wang et al. [25] proposed a MIMO controller to regulate the total power consumption of an enclosure 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 situations with abrupt workload changes though they can achieve control accuracy and system stability within a range theoretically.

Modern data centers apply virtualization technology to consolidate workloads on fewer powerful servers for improving server utilization, performance isolation and flexible resource management. Traditional power management techniques are not easily applicable to virtualized environments where physical processors are shared by multiple virtual machines. For instance, changing the power state of a processor by DVS will inadvertently affect the performance of multiple virtual machines belonging to different applications [19], [27].

It is a trend that power and performance management of virtualized multi-tier servers are jointly tackled. However, it is challenging due to the inherently conflicting objectives.

Power-oriented approaches aim to ensure that a server system does not violate a given power budget while maximizing the performance of hosted applications [15], [19], [21], [26], [25] or increasing the number of services that can be deployed [6]. pMapper [24] tackles power-cost tradeoffs under a fixed performance constraint. vManage [10] performs VM placement to save power without degrading performance. Co-Con [26] is a novel two-level control architecture for power and performance coordination in virtualized server clusters. It gives a higher priority to power budget tracking and performance is a secondary goal.

Performance-oriented approaches aim to guarantee a performance target while minimizing the power consumption [2], [8], [11], [14], [16], [27]. However, they do not have explicit control over power consumption.

Coordinated power and performance management with explicit trade-offs is recently studied in virtualized servers [5], [9]. Mistral [9] is a control architecture to optimize power consumption, performance benefit, and the transient costs incurred by adaptations in virtualized server clusters. vPnP [5] coordinates power and performance in virtualized servers using utility function optimization. It provides the flexibility to choose various tradeoffs between power and performance. However, it lacks the guarantee on system stability and performance, especially under highly dynamic workloads.

There are a few important studies in percentile-based delay guarantee in multi-tier Internet services. A dynamic server provisioning approach proposed in [23] 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 approach proposed in [12] is effective under stationary system workloads, but it does not adapt to the very dynamic nature of Internet workloads. A stochastic approximation technique proposed in [16] can estimate the tardiness quantile of response time distribution. But it is model dependent for a particular simulated workload. An approach proposed in [28] can model the probability distributions of response time based on CPU allocations on virtual machines in a data center. The performance model was obtained by offline training based on data collected from the system. It is not adaptive online to dynamically changing workloads. We designed a neural fuzzy control that is adaptive to highly dynamic workloads [13]. But there was no power control and power and performance tradeoff capability.

The trade-off flexibility and system stability requirements in the face of highly dynamic and bursty workloads, together with the percentile-based response time guarantee, demand novel techniques for autonomous performance and power control.

III. PERFUME System Architecture

Hardware throttling is too rigid for power control in virtualized environments because reducing CPU frequency of a server inevitably affects the performance of all virtual machines running on that server. Unlike previous approaches which regulate a server-level or enclosure-level power consumption, PERFUME controls the power consumption of a virtual resource pool while assuring the performance of multi-tier applications hosted in it. A resource pool is a logical abstraction that groups the CPU and memory resources provided by underlying server clusters. We regulate the power consumption by applying CPU usage limits on VMware virtual machines hosted on a cluster of blade servers. It constrains the utilization of underlying physical processors thereby regulates power consumption. It is feasible due the idle power management of modern processors, which can achieve substantive savings when a processor is idle compared to it is active.

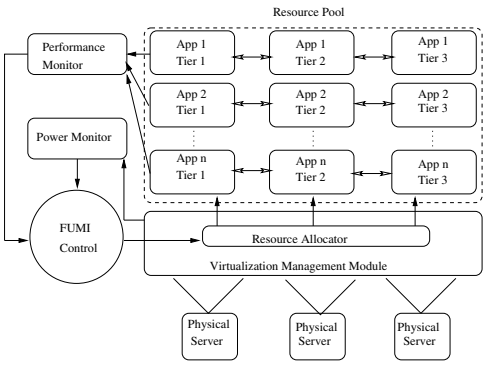


Fig. 1. The system architecture of PERFUME.

Figure 1 illustrates the system architecture of PERFUME. The computer system under control is a virtualized blade server cluster hosting multiple multi-tier applications. Each tier of an application is deployed at a virtual machine created from a resource pool, which logically abstracts the CPU resources provided by the underlying physical server clusters. The power monitor, using VMware's Intelligent Power Management Interface (IPMI) sensors, periodically measures the average power consumption of the server cluster at the resource pool level and sends the value to the FUMI control. Measuring power at the resource pool level allows FUMI control to achieve a fine-grained power control, which is desirable in a virtualized data center that uses different resource pools for different customers. The performance monitor periodically measures the throughput as well as the percentile-based response time of each multi-tier application and sends the performance values to the controller. FUMI control determines the CPU usage limits on various tiers of multiple applications to regulate per-application performance and the total power consumption of the server cluster. It is based on the fuzzy model predictive control theory. The resource allocator actuates the control action to limit the CPU usage of each virtual machine by using the VMware's virtualization management module.

IV. Modeling of Power and Performance Control

To achieve effective control over power consumption and performance of multi-tier applications, PERFUME applies fuzzy modeling to estimate the relationship between the performance and CPU usage limits on the virtual machines that deploy the applications. Both the throughput and the percentile-based response time are used as performance metrics. We also apply the same fuzzy modeling technique to predict the power consumption of the virtual resource pool for different virtual machines' CPU limit. A key strength of fuzzy model is its ability to represent highly complex and nonlinear systems by a combination of inter-linked subsystems with simple functional dependencies.

We construct an initial fuzzy model by applying subtractive clustering technique on data collected from the system. Each obtained cluster represents a certain operating region of the system, where input-output data values are highly concen-

trated. Clustering process partitions the input-output space and determines the number of fuzzy rules and the shape of membership functions. Then, we apply an adaptive network based fuzzy inference system to further tune the fuzzy model parameters. It uses a neural network learning technique for the purpose. At run time, we apply a wRLS algorithm to learn the model parameters for the PERFUME's self-adaptiveness in the face of highly dynamic workloads.

A. The Fuzzy Model

We consider a number of multi-tier applications hosted in a virtual resource pool as a MIMO system. The inputs to the system are CPU usage limits set at various tiers of the applications. The outputs of the system are the measured performance of each application and the average power consumption of the shared resource pool. We obtain two separate models for power and performance of the system, respectively. The system is approximated by a collection of MIMO fuzzy models as follows:

$$\mathrm{y}(k+1) = \mathrm{R}(\xi(k), \mathrm{u}(k)). \tag{1}$$

Let $\mathrm{y}(k)$ be the output variable and $\mathrm{u}(k) = [u_1(k), .., u_m(k)]^T$ be the vector of current inputs at sampling interval k. The regression vector $\xi(k)$ includes current and lagged outputs:

$$\xi(k) = [\mathrm{y}(k), .., \mathrm{y}(k - n_y))]^T \tag{2}$$

where n_y specifies the number of lagged values of the output variable. Note that a regression vector may also include lagged inputs to achieve even better accuracy of power and performance prediction. R is a rule based fuzzy model consisting of K fuzzy rules. Each fuzzy rule is described as follows:

R_i: If $\xi_1(k)$ is $\Omega_{i,1}$ and .. $\xi_\varrho(k)$ is $\Omega_{i,\varrho}$ and $u_1(k)$ is $\Omega_{i,\varrho+1}$ and .. $u_m(k)$ is $\Omega_{i,\varrho+m}$ then

$$\mathrm{y}_i(k+1) = \zeta_i \xi_i(k) + \eta_i \mathrm{u}(k) + \phi_i. \tag{3}$$

Here, Ω_i is the antecedent fuzzy set of the i^{th} rule which describes elements of regression vector $\xi(k)$ and the current input vector $\mathrm{u}(k)$ using fuzzy values such as 'large', 'small', etc. ζ_i and η_i are vectors containing the consequent parameters and ϕ_i is the offset vector. ϱ denotes the number of elements in the regression vector $\xi(k)$. Each fuzzy rule describes a region of the complex non-linear 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 rules. That is,

$$\mathrm{y}(k+1) = \frac{\sum_{i=1}^{K} \beta_i (\zeta_i \xi_i(k) + \eta_i \mathrm{u}(k) + \phi_i)}{\sum_{i=1}^{K} \beta_i} \tag{4}$$

where the degree of fulfillment for the i^{th} rule β_i is the product of the membership degrees of the antecedent variables in that rule. Membership degrees are determined by fuzzy membership functions associated with the antecedent variables. The model output is expressed in the form of

$$\mathrm{y}(k+1) = \zeta^* \xi_i(k) + \eta^* \mathrm{u}(k) + \phi^*. \tag{5}$$

The aggregated parameters ζ^*, η^* and ϕ^* are the weighted sum of vectors ζ_i, η_i and ϕ_i respectively.

B. On-line Adaptation of the Fuzzy Model

Internet workloads to a data center vary dynamically in arrival rates as well as characteristics [22]. This results in significantly varying resource demands at multiple tiers of Internet applications. It is time-consuming and may even be infeasible to obtain a static system model that can provide sufficient prediction accuracy of power and performance for all possible variations in the workload. Hence, the system control models need to adapt on-line in the face of dynamic workloads. We apply a wRLS method to adapt the consequent parameters of the fuzzy model obtained. The technique updates the model parameters as new measurements are sampled from the runtime system. The recursive nature of the wRLS method makes the time taken for this computation negligible for a control interval that is more than 10 seconds. It applies exponentially decaying weights on the sampled data so that higher weights are assigned to more recent observations.

We express the fuzzy model output in Eq. (4) as follow:

$$\mathrm{y}(k+1) = X\theta(k) + e(k) \tag{6}$$

where e(k) is the error value between actual output of the system (i.e., measured performance or power) and predicted output of the model. $\theta = [\theta_1^T \theta_1^T .. \theta_p^T]$ is a vector composed of the model parameters. $X = [w_1 X(k), w_2 X(k), .., w_p X(k)]$ where w_i is the normalized degree of fulfillment or firing strength of i^{th} rule and $X(k) = [\xi_i^T(k), \mathrm{u}(k)]$ is a vector containing current and previous outputs and inputs of the system. The parameter vector $\theta(k)$ is estimated so that the following cost function is minimized. That is,

$$Cost = \sum_{j=1}^{k} \lambda^{k-j} e^2(j). \tag{7}$$

Here λ is a positive number less than one. It is called "forgetting factor" as it gives higher weights on more recent samples in the optimization. This parameter determines in what manner the current prediction error and old errors affect the update of parameter estimation. The parameters of fuzzy model are updated by the wRLS method.

V. FUMI Control Design

We apply the fuzzy model predictive control principle to design the FUMI control. FUMI control is well suited for power and performance control in virtualized server clusters due to its capability to solve constrained MIMO control problems of complex non-linear systems. It determines control actions by optimizing a cost function, which expresses the control objectives and constraints over a time interval. Since the system model is nonlinear, FUMI control linearizes the fuzzy model at the current operating point in order to avoid non-convex, time-consuming optimization. We formulate the power and performance assurance of virtualized multi-tier applications as a predictive control problem. Then, we present detailed steps to transform the control formulation to a standard quadratic programming problem, which allows us to design and implement the control algorithm based on an effective quadratic programming method.

A. FUMI Control Formulation

FUMI control aims to minimize the deviation of power consumption and performance of multi-tier applications from their respective targets. It decides the control actions at every control period k by minimizing the following cost function:

$$\begin{aligned} V(k) &= \sum_{i=1}^{H_p} ||r1 - y_1(k+i)||_P^2 + \sum_{i=1}^{H_p} ||r2 - y_2(k+i)||_Q^2 \\ &+ \sum_{j=0}^{H_c-1} ||\Delta \mathrm{u}(k+j)||_R^2. \end{aligned} \quad (8)$$

Here, $y_1(k)$ is the power consumption of the resource pool. $y_2(k)$ is a vector containing the percentile-based end-to-end response time or the throughput of each application. The controller predicts both power and performance over H_p control periods, called the *prediction horizon*. It computes a sequence of control actions $\Delta \mathrm{u}(k), \Delta \mathrm{u}(k+1), .., \Delta \mathrm{u}(k+H_c-1)$ 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 limits imposed on various tiers of the multi-tier applications. P and Q are the tracking error weights that determine the trade-off between power and performance. The third term in Eq. (8) represents the control penalty and is weighted by R. This term 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. The constraint is formulated as:

$$\sum_{j=1}^{M} (\Delta u_j(k) + u_j(k)) \leq U_{max} \quad (9)$$

where M is the number of applications hosted in a resource pool and U_{max} is the total CPU capacity of the resource pool.

B. Transformation to Quadratic Programming

To transform the MIMO control problem to a standard quadratic programming problem, we linearize the fuzzy model and represent it as a state-space linear time variant model in the following form:

$$\begin{aligned} x_{lin}(k+1) &= A(k)x_{lin}(k) + B(k)\mathrm{u}(k). \\ \mathrm{y}(k) &= C(k)x_{lin}(k). \end{aligned} \quad (10)$$

The state vector for the state-space description is defined as

$$x_{lin}(k+1) = [\xi_i^T(k), 1]^T. \quad (11)$$

The matrices $A(k)$,$B(k)$ and $C(k)$ are constructed by freezing the parameters of the fuzzy model at a certain operating point $\mathrm{y}(k)$ and u(k) as follows. First, we calculate the degree of fulfillment β_i for the current inputs (i.e CPU usage limits) chosen for the system and compute the aggregated parameters ζ^*, η^* and ϕ^*. Comparing Eq. (5) and Eq. (10), the state matrices are computed as follows:

$$A = \begin{bmatrix} \zeta^*_{1,1} & \zeta^*_{1,2} & .. & .. & .. & \zeta^*_{1,\varrho} & \phi^*_1 \\ 1 & 0 & .. & & & 0 & 0 \\ & & \vdots & & & & \\ 0 & 1 & \vdots & & & 0 & 0 \\ \vdots & \vdots & \ddots & & & \vdots & \vdots \\ \zeta^*_{2,1} & \zeta^*_{2,2} & .. & .. & .. & \zeta^*_{2,\varrho} & \phi^*_2 \\ 0 & \vdots & \ddots & & & \vdots & \vdots \\ \zeta^*_{p,1} & \zeta^*_{p,2} & .. & .. & .. & \zeta^*_{p,\varrho} & \phi^*_p \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & .. & 0 & .. & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \eta^*_{1,1} & \eta^*_{1,2} & .. & \eta^*_{1,m} \\ 0 & .. & .. & 0 \\ \vdots & & & \vdots \\ \eta^*_{2,1} & \eta^*_{2,2} & .. & \eta^*_{2,m} \\ 0 & .. & .. & 0 \\ \vdots & & & \vdots \\ \eta^*_{p,1} & \eta^*_{p,2} & .. & \eta^*_{p,m} \\ 0 & .. & .. & 0 \\ 0 & .. & .. & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & .. & .. & .. & .. & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & .. & .. & .. & .. & 1 & 0 \end{bmatrix}$$

where ζ^*_{ij} is the j^{th} element of aggregate parameter vectors ζ^* for application i. Similarly, η^*_{ij} is the j^{th} element of aggregate parameter vectors η^* for application i.

Next, we express the objective of FUMI control, defined by Eq. (8), as a quadratic program:

$$\text{Minimize } \frac{1}{2}\Delta \mathrm{u}(k)^T H \Delta \mathrm{u}(k) + c^T \Delta \mathrm{u}(k) \quad (12)$$

subject to constraint $\Omega \Delta \mathrm{u}(k) \leq \omega$.

The matrices Ω and ω are chosen to formulate the constraints on CPU resource usage as described in Eq. (9). Here, $\Delta \mathrm{u}(k)$ is a matrix containing the CPU usage limits on each virtual machine over the entire control horizon H_c. And,

$$H = 2(R_{1u}^T P R_{1u} + R_{2u}^T Q R_{2u} + R). \quad (13)$$

$$c = 2[R_{1u}^T P^T (R_{1x} Ax(k) - r1) + R_{2u}^T Q^T (R_{2x} Ax(k) - r2)]^T. \quad (14)$$

The matrices R_{1u}, R_{1x} are associated with the performance models of hosted applications and matrices R_{2u}, R_{2x} are associated with the power model of the resource pool.

$$R_{iu} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{H_p-1} \end{bmatrix} \quad R_{ix} = \begin{bmatrix} CB & 0 & .. & 0 \\ CAB & CB & .. & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{H_p-1}B & CA^{H_p-1}B & .. & CA^{H_p-H_c}B \end{bmatrix}$$

C. FUMI Control Interface

Figure 2 shows the interface between the FUMI online learning and MIMO control components. The starting point of the continuous chain of interaction is the fuzzy model of multi-tier applications hosted in virtualized servers. The model is initially obtained offline as described in Section IV. The optimizer of FUMI Control decides the CPU resource allocation, $u(k)$, at each interval to minimize the deviation of power consumption and performance of multi-tier applications from their respective targets, denoted by ref. The online learning algorithm wRLS adapts the fuzzy model automatically in response to dynamic workloads. It learns the fuzzy model parameters by utilizing the current and previous measurements of actual power consumption and system performance $y(k+1)$, and the control actions $u(k)$.

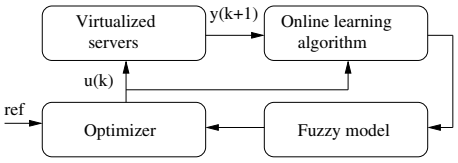


Fig. 2. Interface between FUMI control and learning components.

VI. PERFUME Implementation

A. The Testbed

We have implemented PERFUME on a testbed consisting of two HP ProLiant BL460C G6 blade server modules and a HP EVA storage area network with 10 Gbps Ethernet and 8 Gbps Fibre/iSCSI dual channels. Each blade server is equipped with Intel Xeon E5530 2.4 GHz quad-core processor and 32 GB PC3 memory. Virtualization of the cluster is enabled by an enterprise-level virtualization product, VMware ESX 4.1. VMware's vSphere module controls the disk space, memory, and CPU share in MHz allocated to the virtual machines. It also provides an API to support the remote management of virtual machines. We create a resource pool from the virtualized server cluster to host multi-tier applications. PERFUME system architecture is shown in Figure 1. Each tier of an application is hosted inside a VMware virtual machine with 2 VCPUs, 4 GB RAM and 15 GB hard disk space. The guest operating system used is Ubuntu Linux version 10.04.

As many related studies [5], [20], [23], [28], our work uses an open-source multi-tier application benchmark RUBiS in the testbed. RUBiS implements the core functionality of an eBay like auction site: selling, browsing and bidding. We configure the RUBiS clients to generate workloads of different mixes as well as workloads of time-varying intensity.

B. PERFUME Components

1) Power Monitor: The average power consumption of the server cluster is measured at the resource pool level by using a new feature of VMware ESX 4.1. Since the power consumption of resource pool depends on that of its underlying physical hosts, VMware gathers such data through IPMI sensors. The power monitor program runs on a separate virtual machine and collects power measurement data by using VMware vSphere API.
2) Performance Monitor: PERFUME uses a sensor program provided by RUBiS client for performance monitoring. We modify the sensor to measure the client-perceived percentile-based end-to-end response time and throughput over a period of time. The number of requests finished during a control interval is throughput.
3) FUMI Controller: The controller first updates the fuzzy models based on power and performance data measured online at every control interval of 30 seconds. This control interval is chosen by considering the trade-off between noise in the sensor measurements and faster response of the controller. Then, it invokes a quadratic programming solver, *quadprog*, in MATLAB to execute the control algorithm described in Section V. The solution of the control algorithm in terms of VM CPU usage limits is sent to the resource allocator.
4) Resource Allocator: VMware's vSphere module is used to impose CPU usage limits on the virtual machines supporting different tiers of the hosted applications. The controller issues commands to the resource allocator by using VMware vSphere API 4.0.

VII. Performance Evaluation

A. Modeling Accuracy of the FUMI Control

The accuracy of the fuzzy model has significant impact on effective resource allocation for joint control of power and performance in PERFUME. Thus, we first evaluate the accuracy of the fuzzy models obtained by the FUMI control approach for power and performance prediction.

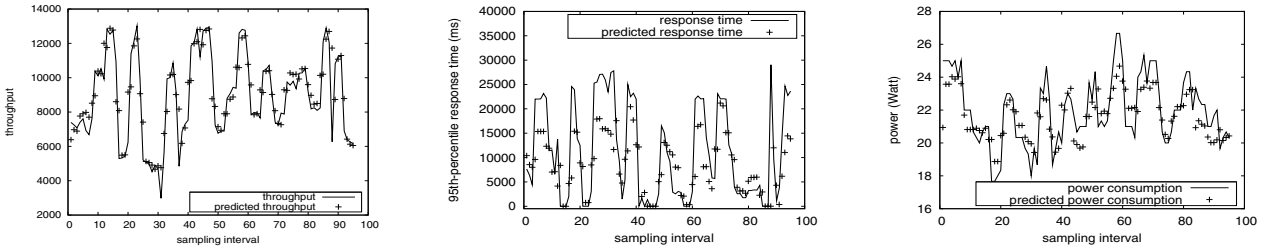
We obtain a system model for predicting the performance of one RUBiS application and the total power consumption of both RUBiS applications hosted in the virtual resource pool. The FUMI control module conducts the initial system modeling based on offline power and performance measurements collected from the testbed. The data is collected by randomly allocating various CPU usage limits on each tier of the two RUBiS applications. Each application has a workload of a browsing mix of 1000 concurrent users. Applying fuzzy subtractive clustering on the collected data, we obtain a fuzzy model consisting of four rules with different input membership functions and consequent parameters. Then, we tune the consequent parameters by applying a neural network based training, which converges within 24 iterations.

Figures 3(a) and 3(b) show that FUMI models can accurately predict the performance in terms of throughput and the 95_{th}-percentile response time for various CPU allocations at different sampling intervals. Figure 3(c) shows that the FUMI model is able to accurately predict the average power consumption for various CPU allocations at different sampling intervals. The accuracy is measured by the normalized root mean square error (NRMSE), a standard metric for deviation. Figures 3(a), 3(b) and 3(c) show that the checking data and FUMI prediction are very close, with the NRMSE 12.5%, 17.6% and 15.2% in the three scenarios respectively.

B. Self-adaptiveness of the FUMI Control

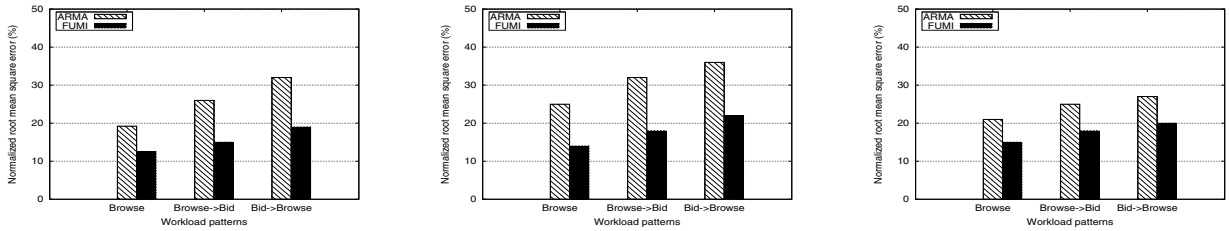
We evaluate the self-adaptiveness of FUMI model by measuring its prediction accuracy when the workload is changed from browsing mix of 1000 concurrent users to bidding mix of 500 concurrent users and vice versa. The prediction accuracy is again quantified by the normalized root mean square error. We compare our results with a popular and recently used technique for modeling Internet systems, ARMA [5], [20].

Figures 4(a) and 4(b) show that FUMI model outperforms ARMA model in predicting performance of a multi-tier application for both stationary and dynamic workloads. On average, the improvement in performance prediction accuracy for the throughput and 95_{th}-percentile end-to-end response time are 35% and 43%, respectively. The improvement in



(a) Throughput. (b) The 95_{th}-percentile response time. The power consumption.

Fig. 3. The prediction accuracy of performance and power by FUMI models.



(a) Throughput. (b) The 95_{th}-percentile response time. The power consumption.

Fig. 4. The prediction accuracy comparison between FUMI and ARMA models under a dynamic workload.

power prediction accuracy is shown in Figure 4(c). Compared to the ARMA modeling FUMI modeling has an additional benefit, that is, it eliminates the need to find a suitable moving window size for collecting the data used for online learning of the performance and power control models.

It is difficult to obtain a static system model that can provide sufficient prediction accuracy of power and performance for all possible variations in highly dynamic and bursty workloads. The fuzzy models obtained by the FUMI approach are self-adaptive in the face of dynamically varying workloads. This is due to the fact that FUMI control integrates the strengths of fuzzy logic, MIMO control and artificial neural network and achieves fast online learning using a wRLS method. FUMI updates the model parameters as new measurements are sampled from the runtime system.

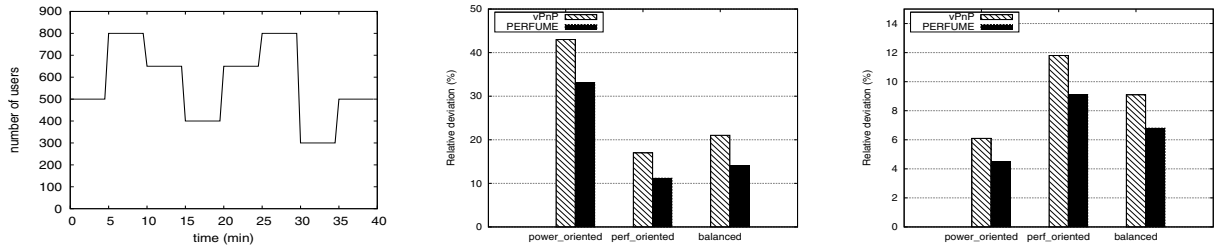
C. Power and Performance Assurance of PERFUME

1) Flexible tradeoffs: A key feature of PERFUME is its ability to assure joint power and performance guarantee with flexible tradeoffs while assuring control accuracy and system stability. The tradeoffs between inherently conflicting power and performance objectives can be specified by a data center administrator. The system stability is measured in terms of relative deviation of power and performance from their respective targets, as defined in vPnP [5]. We experiment with power-preferred, performance-preferred and balanced control options under a highly dynamic workload [13]. Figure 5(a) shows the dynamic changes in the number of concurrent users.

PERFUME achieves the specified tradeoffs by tuning the tracking error weights, P and Q, in the MIMO control objective defined by Eq. (8). Figure 5(b) compares the control accuracy of vPnP with PERFUME in assuring the throughput target for various tradeoffs between power and performance. Our results demonstrate that, compared to 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 Figure 5(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 control accuracy for both power and performance assurance.

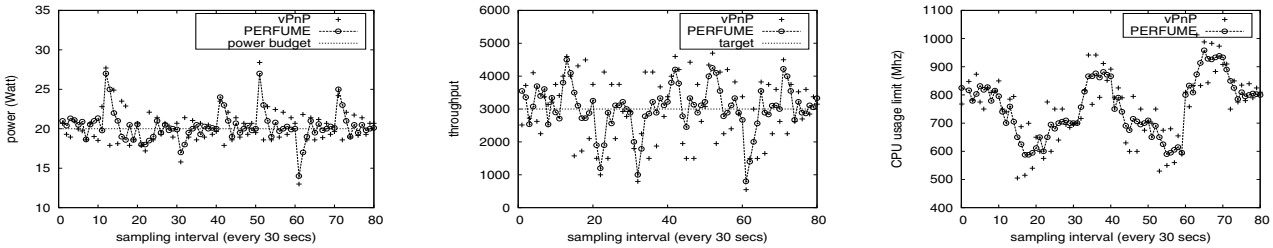
2) System stability: We now take a closer look at the system stability of PERFUME under the highly dynamic workload. We experiment with the power-performance balanced control option. Figures 6 (a) and (b) illustrate that PERFUME offers more accurate assurance of power and performance targets compared to vPnP in [5]. We show the results for only one of the hosted RUBiS applications. Similar results were obtained for the other. Note that the spikes in average power consumption and throughput at various intervals are due to the limitation of purely reactive approach under abrupt changes in the workload intensity. However, PERFUME is able to quickly adapt itself and control both power consumption and throughput so that they converge to the steady state. On the other hand, results show there are more significant oscillations in power and performance assurance due to the lack of control accuracy and system stability guarantee in vPnP. There is an improvement of 25% and 32% in relative deviation of power consumption and throughput respectively.

Figure 6 (c) compares the total CPU usage limits allocated by vPnP and PERFUME at various sampling intervals. On average, PERFUME uses similar amount of CPU resources



(a) A highly dynamic workload. (b) Throughput. (c) Power consumption.

Fig. 5. Power and performance assurance with flexible control options under a highly dynamic workload.



(a) Power consumption. (b) Throughput. (c) CPU allocation.

Fig. 6. Comparison between PERFUME and vPnP for power and performance assurance under a highly dynamic workload.

as vPnP. However, there is significantly less fluctuations in resource allocation. The modeling accuracy, self-adaptiveness and control theoretic foundation of FUMI control enables PERFUME to achieve system stability and accurate control for both power and performance of multi-tier applications in the face of highly dynamic workloads.

D. Robustness of PERFUME under Bursty Workloads

We evaluate the robustness of PERFUME under a bursty workload. We use an approach proposed in [18] to inject burstiness into the arrival process of RUBiS clients according to the index of dispersion. The dispersion index modulates the think times of users between submission of consecutive requests. We set the index of dispersion to 4000 and the maximum number of concurrent users to 1000. Figure 7 (a) shows the bursty workload in which the number of active users in a RUBiS application fluctuates over a period of 200 seconds.

Figures 7(b) and 7(c) illustrate that, compared to vPnP, PERUME is able to provide better assurance of average power consumption and throughput targets in the face of the bursty workload. We choose a sampling interval of 20 seconds for both approaches. A smaller sampling interval provides better responsiveness to workload fluctuations, but increases the sensitivity towards random noise. Note that the variations in the average power consumption and throughput are mainly due to burstiness in the workload and the control actions (CPU allocations) taken at each sampling interval. The robustness of PERFUME under bursty workloads is attributed to the fact that its control actions are based on a more accurate model of the system and a sound control theoretic foundation. Moreover, PERFUME is more adaptive to variations in workload due to its fast online learning algorithm. We observe that compared with vPnP, there is the improvement of 32% and 44% in terms of relative deviation of power and throughput by PERFUME.

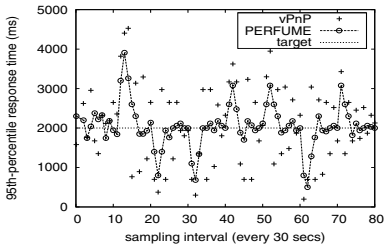
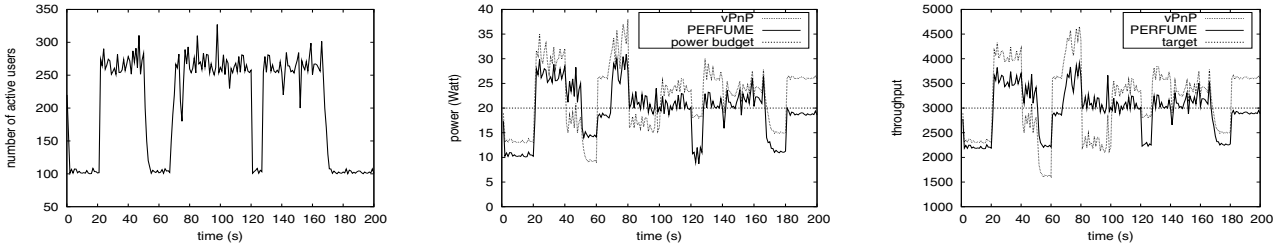


Fig. 8. The 95_{th}-percentile response time in vPnP and PERFUME.

E. Percentile-Based Response Time Guarantee in PERFUME

We now demonstrate the capability of PERFUME in accurately achieving the 95_{th}-percentile response time guarantee in a multi-tier application. Note that PERFUME is able to provide any percentile based delay guarantee.

Figure 8 shows that compared to vPnP [5], PERFUME delivers significantly improved control accuracy and performance assurance for highly non-linear percentile-based response times. For this experiment, we set the 95_{th}-percentile response time target of a RUBiS application as two seconds and apply the highly dynamic workload shown in Figure 5(a). We observe that compared with vPnP, there is the improvement of 40% in terms of relative deviation by PERFUME. This is mainly due to two reasons. First, PERFUME is able to obtain the performance model with better accuracy, even in case of highly non-linear percentile-based performance metric. Second, its FUMI control provides more accurate control and system stability due to its integration of fuzzy logic, MIMO control and artificial neural network.



(a) Bursty workload. (b) Power consumption. (c) Throughput.

Fig. 7. Power and Performance assurance under a bursty workload generated by 1000 users.

VIII. Conclusion

Modern data centers face significant and multi-facet challenges in performance and power management for meeting service level agreements, resource utilization efficiency and low power consumption. PERFUME provides holistic self-adaptive performance and power control in a virtualized server cluster. As demonstrated by experimental results based on a testbed implementation, its main contributions are the precise control of power consumption of virtualized blade servers avoiding system failures caused by power capacity overload or overheating, effective control of both throughput and percentile-based response time of multi-tier applications, and guarantee of power and performance targets with flexible tradeoffs while assuring control accuracy and system stability. Our future work will integrate power-aware consolidation techniques [11], [24] with PERFUME and explore autonomous performance and power control for building green data centers.

Acknowledgement: This research was supported in part by NSF CAREER award CNS-0844983 and research grants CNS-0720524 and CNS-0824448.

References

[1] X. Bu, J. Rao, and C.-Z. Xu. A reinforcement learning approach to online web system auto-configuration. In *Proc. IEEE Int'l Conf. on Distributed Computing Systems (ICDCS)*, 2009.

[2] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. In *Proc. ACM SIGMETRICS*, pages 303–314, 2005.

[3] M. Elnozahy, M. Kistler, and R. Rajamony. Energy conservation policies for web servers. In *Proc. the USENIX Symp. on Internet Technologies and Systems (USITS)*, 2003.

[4] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *Proc. ACM SIGMETRICS*, 2009.

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

[6] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini. Statistical profiling-based techniques for effective power provisioning in data centers. In *Proc. the EuroSys Conference*, 2009.

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

[8] C. Jiang, X. Xu, J. Wan, J. Zhang, X. You, and R. Yu. Power aware job scheduling with qos guarantees based on feedback control. In *Proc. IEEE Int'l Workshop on Quality-of-Service (IWQoS)*, 2010.

[9] G. Jung, M. A. Hiltunen, K. R. Joshi, R. D. Schlichting, and C. Pu. Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In *Proc. IEEE Int'l Conf. on Distributed Computing Systems (ICDCS)*, 2010.

[10] S. Kumar, V. Talwar, V. Kumar, P. Ranganathan, and K. Schwan. vmanage: Loosely coupled platform and virtualization management in data centers. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC)*, 2009.

[11] D. Kusic and J. O. Kephar. Power and performance management of virtualized computing environments via lookahead control. In *Proc. IEEE Int'l Conf. on Autonomic computing (ICAC)*, 2008.

[12] P. Lama and X. Zhou. Efficient server provisioning for end-to-end delay guarantee on multi-tier clusters. In *Proc. IEEE Int'l Workshop on Quality of Service (IWQoS)*, 2009.

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

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

[15] C. Lefurgy, X. Wang, and M. Ware. Server-level power control. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC)*, 2007.

[16] J. C. B. Leite, D. M. Kusic, D. Mossé, and L. Bertini. Stochastic approximation control of power and tardiness in a three-tier web-hosting cluster. In *Proc. IEEE Int'l Conf. on Autonomic computing (ICAC)*, 2010.

[17] N. Mi, G. Casale, L. Cherkasova, and E. Smirni. Burstiness in multi-tier applications: Symptoms, causes, and new models. In *Proc. ACM/IFIP/USENIX Int'l Middleware Conference*, 2008.

[18] N. Mi, G. Casale, L. Cherkasova, and E. Smirni. Injecting realistic burstiness to a traditional client-server benchmark. In *Proc. IEEE Int' Conference on Autonomic Computing (ICAC)*, 2009.

[19] R. Nathuji and K. Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In *Proceedings of ACM Symp. on Operating Systems Principles (SOSP)*, pages 265–278, 2007.

[20] P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant. Automated control of multiple virtualized resources. In *Proc. of the EuroSys Conference (EuroSys)*, pages 13–26, 2009.

[21] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No power struggles: Coordinated multi-level power management for the data center. In *Proc. ASPLOS*, 2008.

[22] R. Singh, U. Sharma, E. Cecchet, and P. Shenoy. Autonomic mix-aware provisioning for non-stationary data center workloads. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC)*, pages 21–30, 2010.

[23] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood. Agile dynamic provisioning of multi-tier Internet applications. *ACM Trans. on Autonomous and Adaptive Systems*, 3(1):1–39, 2008.

[24] A. Verma, P. Ahuja, and A. Neogi. pmapper: power and migration cost aware application placement in virtualized systems. In *Proc. ACM/IFIP/USENIX Int'l Middleware Conference*, 2008.

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

[26] X. Wang and Y. Wang. Co-con: Coordinated control of power and application performance for virtualized server clusters. In *Proc. IEEE Int'l Workshop on Quality of Service (IWQoS)*, 2009.

[27] Y. Wang, X. Wang, M. Chen, , and X. Zhu. Partic: Power-aware response time control for virtualized web servers. *IEEE Trans. on Parallel and Distributed Systems*, 21(4), 2010.

[28] B. J. Watson, M. Marwah, D. Gmach, Y. Chen, M. Arlitt, and Z. Wang. Probabilistic performance modeling of virtualized resource allocation. In *Proc. IEEE Int'l Conf. on Autonomic computing (ICAC)*, 2010.