A Cyber-Physical Integrated System for Application Performance and Energy Management in Data Centers

Hui Chen *, PengCheng Xiong†, Karsten Schwan†, Ada Gavrilovska†, ChengZhong Xu*‡
*Shenzhen Institutes of Advanced Technology
Shenzhen, GuangDong 518055
†College of Computing,Georgia Institute of Technology
Atlanta, Georgia 30332
‡Department of Electrical and Computer Engineering, Wayne State University
Detroit, MI 48202

***Abstract*—Both performance and energy cost are important concerns for current data center operators. Traditionally, however, IT and mechanical engineers have separately optimized the cyber vs. physical aspects of data center operations. In contrast, the work presented in this paper jointly considers both the IT - cyber - and the physical systems in data centers, the eventual goal being to develop performance and power management techniques that holistically operate to control the entire complex of data center installations. Toward this end, we propose a balance of payments model for holistic power and performance management. As an example of coordinated data center management system, the energy-aware cyber-physical system (EaCPS) uses an application controller on the cyber side to guarantee application performance, and on the physical side, it utilizes electric current-aware capacity management (CACM) to smartly place executables to reduce the energy consumption of each chassis present in a data center rack. A web application, representative of a multi-tier web site, is used to evaluate the performance of the controller on the cyber side, the CACM control on the physical side, and of the holistic EaCPS methods in a mid-size, instrumented data center. Results indicate that coordinated EaCPS outperforms the cyber and physical control modules working separately.**

***Keywords*-Energy Efficiency, Cyber-Physical System, Control Theory**

I. Introduction

According to the latest reports [1], the electricity used in US data centers in 2010 likely accounted for between 1.7% and 2.2% of total electricity use. This, nonetheless, imposes a significant load on the electric grids and generation facilities, and such loads will sharply increase if modern cloud computing technologies continue to cause the further expansion of large-scale data center facilities. Moreover, given annual energy costs in the millions of dollars, data center operators face continuing challenges of profitability under rising energy prices, while maintaining competitively low costs for services that offer to end users the levels of performance they demand.

To reduce energy costs while also improving IT system performance, one must consider and attempt to optimize both (i) the cooling and power generation/delivery – the physical, and (ii) the IT systems running applications – the cyber – components of data center systems. The importance of such holistic action is underlined by the fact that the energy used for cooling alone can contribute up to 50% of the total energy costs seen in a traditional data center [2]. In response, this paper presents an approach to and examples of holistic data center management where income is determined by service level agreements (SLAs) that set the price paid by customers. So, the data center's operating margin depends on two factors: (1) the provided quality of service, where higher QoS levels typically imply higher charges to users, and (2) energy cost, which depends on the IT and cooling equipment's power consumption in the data center, where lower costs lead to higher profit.

There are many challenges to balance application performance and energy management in the cyber world of data centers. One challenge is that data center workload demands can be bursty and even vary significantly during the course of a single day. This raises a requirement for greater flexibility in resource provisioning, and typically rules out static provisioning methods that would likely either over-provision or under-provision resources. To solve this problem, we first describe a hybrid provisioning approach that combines predictive with reactive control strategies to dynamically provision IT resources at different time scales. This *cyber world* solution is based on three important observations.

- First, many workloads, especially web workloads in data centers, exhibit periodic patterns (daily, weekly, etc.), as illustrated in [3], [4], [5].
- Second, bursts can result in bottlenecks in certain processing components, as with the multi-tier web applications considered in our work, where the bottleneck incurred by bursts typically resides in their application server components [6].
- Third, actual demand patterns are statistical in nature, and so, there will be deviations from predicted patterns due to unforeseen factors such as flash crowds, service

978-1-4673-2154-9/12/$31.00©2012 IEEE

outages, and holidays [3].

Based on the above observations, for multi-tier web application, if we establish the relationship between the system capacity and number of application servers, then we could adjust the number of application servers according to the workload forecast predictively, in advance of workload spikes. For this reason, we rely on a predictive control strategy to help estimate the incoming workload in the near future, thereby improving the accuracy of the resource provisioning method and reducing energy waste. We then augment the predictive strategy with a reactive control strategy, in order to deal with the differences between actual workload and predictions, the goal being to reduce the number of SLA violations in the system.

The specific challenges we consider in the physical world lie in unbalanced thermal distributions, unequal current draws, and different energy efficiencies of computing devices. These physical factors can strongly affect the total energy consumption of the data center. Heterogeneity in energy efficiency of different computing devices will cause unbalanced thermal generation. Unbalanced thermal distribution will lead to increased use of cooling energy in the data center. Finally, unequal electric current draws will violate the three-phase balance principle and result in increased energy consumption.

Our prior work introduces several techniques to deal with the challenges listed above. First, in [7], we describe a spatially aware workload placement method to balance the thermal distribution in the data center, thereby reducing total cooling energy consumption. Next, based on the specific observation that there are different levels of energy efficiency in the different power domains of each single enclosure in our data center, we develop additional methods to optimize the total computing energy consumption in each enclosure [8]. For the blade-based configurations in our data center system, this can be achieved via a current-aware workload scheduling method that minimizes the energy consumption of the total enclosure under the same workloads. Initial results and measurements demonstrate the importance of considering such constraints and inputs from the physical world when determining how to provision resources.

The specific contribution of this paper is that it leverages the experiences listed above to develop holistic provisioning strategy for multi-tier web applications. The strategy uses a balance of payments model to integrate cyber resource control with physical environment controls to optimize data center profit. The paper combines the following methods.

- A hybrid approach uses predictive and reactive control to provision IT resources: predictive control works at coarse time scales (e.g., hours) to determine how many servers should be deployed for each tier of a multi-tier web application, using workload prediction. Reactive control handles any excess demand by adjusting the resource allocation among *virtual machines* (VMs) at finer time scales (e.g., minutes). The application control of these two methods achieves an obvious improvement in meeting SLAs, conserving energy and reducing provisioning cost.
- An electric current-aware VM placement method considers inputs from the physical environment, regarding power and current usage, when making placement decisions, thereby achieving improved energy efficiency.
- A balance of payments model integrates the cyber and physical control systems as an *energy-aware cyber-physical system* (EaCPS) to coordinate IT resource provisioning and workload placement management.

EaCPS has been implemented and evaluated in an actual instrumented testbed. Experimental evaluations demonstrate significant improvements in performance and energy savings compared to regimes with separate cyber vs. physical control systems.

The remainder of this paper is organized as follows. Section 2 presents the predictive and reactive control model for the IT resource provisioning of multi-tiers web applications. The control model for electric current-aware workload placement in the a data center's physical environment is described in Section 3. In Section 4, we propose the EaCPS – a balance of payments model – that combines the separate cyber and physical control systems. The workload identification, performance profiling of multiple system configurations, and control algorithm implementations are illustrated in Section 5. Section 6 introduces the experimental setup and results. A brief overview of related work and concluding remarks appear at the end.

II. Cyber Control For IT Resource Provisioning

We first describe the architecture of the cyber controller for resource provisioning used in our integrated solution. In a cloud environment, multiple applications will be hosted by a common pool of virtualized servers. Each application consists of several interacting components, each of which runs in a virtual machine. In order to enhance the utilization of physical severs, there are always several virtual machines consolidated in the same physical server and sharing its resources, including CPU capacity, disk access bandwidth and network I/O bandwidth. Resource allocations at run time are made by hypervisors or virtual machine monitors (VMM) (e.g., ESXi). One way to ensure that applications meet their performance target or threshold is by performing application-level configuration management. As illustrated in [6], different configuration (*w-a-d*, where *w* is the number of web servers, *a* is the number of application servers, and *d* is the number of database servers) of the application have different bottlenecks. In this particular code, when the number of users increase, the bottleneck is typically determined to be in *a* (the number of application servers). In response, one can dynamically change the number of application servers according to workload prediction. One

could also meet application performance at the node level, by dynamically adjusting the resource allocated to each virtual machine through the VMM, thereby controlling the performance of each application component.

The architecture of the cyber control is presented in Figure 1. The bottom of the figure shows the cyber control domain, which consists of the workload forecast module, configuration database, and configuration service modules. They are used by the application controller to decide the application configuration and the resource entitlement for each virtual machine, in real time. Above the cyber control domain is the managed infrastructure. Each physical server runs multiple virtual machines. Each virtual machine hosts one tier of an application, which can span multiple hosts. The node controller is delivered with the ESXi server installed on the physical server. More specifically,

- The *workload forecast* maintains the historical workload analysis results, collected from previous real data center operation data, and provides the workload prediction information to the application controller.
- The *configuration DB* stores system configuration information, such as the VM locations, the type of application components running in a VM, etc. The information can be updated by other services, such as the VM migration service.
- The *configuration service* maintains empirical data from experimental results of different application configurations, and is used to give the suggestion about the configuration of the application based on the predicted workload.
- The *application controller* collects information from the configuration service and the workload forecast to decide the application's configuration. The configuration can change depending on the predicted daily or weekly pattern of the workload, or due to bursts caused by accidental events, holidays, etc. The application controller interacts with the node controllers to adjust the resource allocation to a specific application to maintain its performance target.
- The *node controllers* reside on each physical server. They are responsible for resource allocations to each VM to meet the resource utilization target as determined by the application controller. The demanded resource entitlement depends on the workload and utilization target.

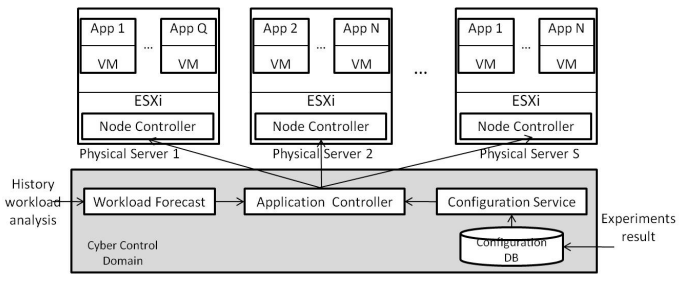

Figure 1. Cyber Control Architecture

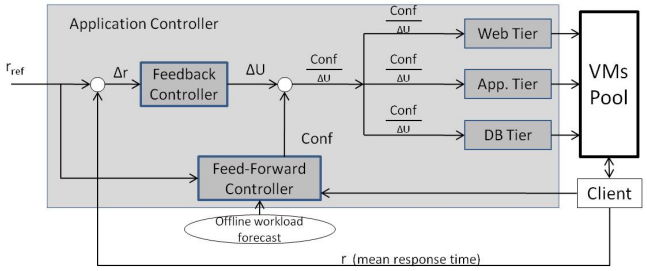

Figure 2. Feedback and feed-forward application controller

Application Controller. Figure 2 shows the architecture of the application controller, consisting of a feed-forward controller and a feedback controller. The feed-forward controller utilizes the offline workload forecast to suggest the configuration (w-a-d) of the whole application system according to the incoming workload prediction derived from historical observations, as discussed in later sections. The feedback controller is used to deal with abnormal workload burst, and to tune the resource entitlements of each application component based on the error between the performance target (r_{ref}) and the measured performance r.

These two controllers work at different time scales. The feed-forward controller works at long time-scale (hours), using model-based predictive control *proactively* to tune the application system configuration (w-a-d) based on the workload daily/weekly pattern captured from historical data. We refer to this configuration as the *base workload*. The feedback controller operates at a shorter time-scale (minutes). It is invoked at run time, whenever the error between the target and measured performance exceeds some threshold value (δr). Integration of feed-forward and feedback controllers provides a more robust solution than that with either feed-forward or feedback alone.

Node Controller. The application controller interacts with node controllers deployed on each physical node. The goal of each node controller is to execute the commands from the application controller, thereby maintaining application performance targets by dynamically adjusting the resource entitlements of all virtual machines on the node. It implements two functions: the resource controllers and the arbiter. The resource controllers consider CPU and memory resources. They determine the specific resource allocations among VMs, as guided by inputs from the application controller, explained in more detail in [9].

III. Physical Control For Workload Placement

Next, we introduce the physical control for the workload placement based on the power efficiency motivation, which

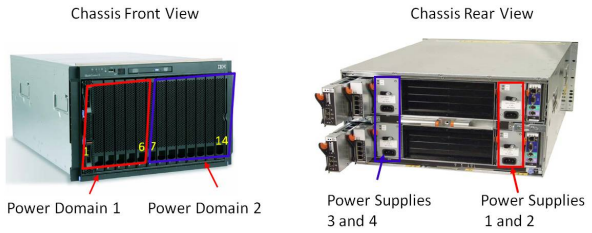

Figure 3. Construction of the IBM E Series BladeCenter

will affect the cloud owner's operation cost on energy. The bulk of the energy cost of the data center is cooling cost and the compute power consumption. Cooling cost largely depends on the thermal distribution in the data center – if there are no 'hot' spots or unbalanced thermal distribution, then the cooling cost could achieve the most savings.

Current-aware Workload Placement. In our earlier work, we developed a current-aware approach to workload placement, inspired by the three-phase balance requirement in electrical engineering. Usually, an electrical appliance consumes less power when the load is balanced across the three-phase circuit, which also ensures the safe operation of the equipment. The proposed current-aware workload placement method is based on the manufacturing structure of the blade chassis in our data center. This method may not be general to all data centers, but nonetheless, demonstrates the energy saving potential of applying the three-phase balance principle in data centers or more generally, demonstrates the importance of considering both the cyber and the physical elements of data center installations.

As shown in Figure 3, every chassis in our data center has a total of fourteen hot-swap blade server bays and a media tray in the front. The rear of the chassis contains up to 4 power modules, 2 blowers, 2 management modules, and 4 switch (I/O) modules. These components connect to the blades in the front of the chassis through the mid-plane, into which all major components are plugged. The power supplies in bays 1 and 2 provide redundant power to all the BladeCenter modules and to blade bays 1 through 6. The BladeCenter unit supports a second pair of power modules in power bays 3 and 4 that provide redundant power to blade bays 7 through 14. To provide true redundant power, BladeCenter power modules 1 and 3 must be connected to a different 200-240 Volt AC power source than power modules 2 and 4. We denote the two power domains as A and B. B supplies power to 8 blade servers, i.e., 2 more servers than A. Although A includes other BladeCenter modules such as mid-plane, blowers, I/O modules, etc., these modules consume mostly static power when the chassis power is on, which is about 390 Watts in our experiment. As a result, power domain A typically consumes more power than B when workloads are low, but when all servers reach their capacity limits, then B's power consumption will surpass A. In order to obtain the computing to power relationship for these two domains, we design an experiment that gradually increases the workload placed on each domains from 10% to 100%. Based on the experimental results, we obtain the following equations for the average computing power model of each blade. Note that these results indicate that the blade servers in B are more power efficient than those in A.

$$P_A = 0.4 * U_{cpu}(t) + 180 \quad (1)$$

$$P_B = 0.38 * U_{cpu}(t) + 120 \quad (2)$$

Next, at the chassis level, we will utilize a PI controller to advise the system how to perform server selection. Because the most power efficient state for the chassis is when the two power domains' electric currents reach the balance, we run a thread for each chassis to check the electric current difference between the two domains at a specified time interval, which is 5 minutes in our case. After obtaining the electric difference between the two domains, we use a proportional plus integral controller to calculate the workload to be migrated between these two domains. The equation for the PI controller is:

$$Workload_{migrate} = K_p c_t + K_i \int c_t dt + a \quad (3)$$

where c_t is the electric current difference between the two domains at time slot t, K_i is the integral gain, and K_p is the proportional gain, which is the proportion between CPU workload and current difference in our case obtained from stored data. Because of the different power efficiency of the two domains, the workload to current ratios for these two domains are also different. We use K_{pa} and K_{pb} to represent the ratio for domain A and B separately. K_p is then computed as:

$$K_p = \frac{K_{pa} * K_{pb}}{K_{pa} + K_{pb}} \quad (4)$$

With K_p determined, we next tune the K_i term, used to remove the oscillation brought by proportional controller. Usually a simple proportional control system either oscillates, moving back and forth around the setpoint because there is nothing to remove the error when it overshoots, or stabilizes at a too low or too high value. By adding a proportion of the average error, namely the integral term to the process input, the average difference between the process output and the setpoint is continually reduced. Therefore, eventually, a well-tuned PI loop's process output will settle down at the setpoint [10].

IV. Cyber-Physical Coordinated Resource Management System

Consider the cyber and physical resource management systems described separately in the above two sections. These two control systems have different objectives in the

data center operation. The former focuses on application performance, relative to user experiences and the business benefits of the service. The latter leverages the different power efficiencies between the two power domains of blade servers to reduce power consumption for the whole chassis, which will lead to energy saving and operation cost reduction. The deficiencies of these two control systems are also obvious: the physical control may deteriorate the quality of service during workload relocation to save energy, where the benefits earned from energy reduction may not make up for the loss caused by SLA violations. Conversely, the cyber resource management system can control the application performance to satisfy the SLA most of the time, but it neglects the potential to save energy when resources are first being allocated. In response, this integrates the cyber and physical resource management systems to obtain an energy-aware mangement system –*EaCPS*, which not only strives to maintain a certain level of quality of service, but also exploits the chance to save energy during operation.

If the decisions of the cyber and physical control system are consistent, then it is ok for the EaCPS to make the final decision by just following what the two separate control modules suggest. An example is when the application controller needs to power on another tomcat server to handle the increasing incoming workload, and when the physical control side wants to increase the current of some domain at that time to balance the electric current of the two domains. In that case, both controllers will agree on the utility of placing the new virtual machine in the lower current power domain. There may also be disagreement between both controllers, however. An example is when the physical control module needs some virtual machine to be migrated from one power domain to the other, but the cyber control component does not permit VM migration at that time for performance reasons. The EaCPS payment model is the proposed basis for dealing with such conflicts.

A. Balance of Payments Model for EaCPS

While we have already introduced the variables and parameters used in our performance and cost modeling, for convenience, Table I summarizes the notations used in this section.

For the cost of the data center, we only consider the energy cost, which includes the cooling and computing energy cost. For cooling power, we rely on a relationship between the cooling power and the computing power in the data center, as presented in Equation 5:

$$P_t^{AC} = \frac{P_t^{comp}}{CoP(T_{sup}^{in})} \tag{5}$$

where *CoP* means *coefcient of performance*, which is the ratio of the heat removed over the work required to remove that heat. A higher CoP means more efficient cooling, and usually, the CoP increases with increase in the air temperature supplied by the CRAC, T_{sup}^{in}. Note the *CoP* is the average value for the whole data center, not for specific CRAC. So the whole energy consumption of the data center is

$$\begin{aligned} E_{total} &= (P_t^{comp} + P_t^{AC}) * t \\ &= (1 + \frac{1}{CoP(T_{sup}^{in})}) P_t^{comp} * t \end{aligned}$$

We assume that the supply air temperature is kept the same during our experiments, and denote $(1 + \frac{1}{CoP(T_{sup}^{in})})$ as a constant α. Given the power consumption model presented in the last section, the total computing power consumption for the application running in the cluster can be obtained as:

$$P_t^{comp} = \sum_{d=1}^{2} \sum_{n=1}^{N_d} \left(P_{idle,d} + \beta_d \sum_{m=1}^{M} \sum_{i=1}^{I_m} U_{im}(t) \right) \tag{6}$$

So, 'payments' for energy can be calculated as:

$$Cost = K_\psi * \alpha * \left\{ \sum_{d=1}^{2} \sum_{n=1}^{N_d} \left(P_{idle,d} + \beta_d \sum_{m=1}^{M} \sum_{i=1}^{I_m} U_{im}(t) \right) \right\} * t \tag{7}$$

In addition to the energy cost, we also need to know the income brought by the hosted applications, which is related to the performance attained from the allocated resources. We use the same performance model as presented in previous work [9], [11], which assumes that a Poisson process is a good approximation of request arrivals, and models CPUs as an M/G/1/PS queue. Based on queueing theory, the CPU resident time in the m tier is $r_{cpu,m} = \frac{u_m}{\lambda(1-u_m)}$, where $u_m = \sum_{i=1}^{I_m} U_{im}(t)/I_m$, which represent the average CPU

Table I
NOTATIONS FOR MODELING

d	power domain number:1 or 2
M	number of tiers (e.g. Web, App, DB)
I_m	number of virtual machines at tier m
N_d	number of blade servers in domain d
β_d	power coefficient of blade server in domain d
λ	average arrival rate of all transaction types
r_{cpu}	average resident time on CPU resources
r_{others}	average resident time on other resources
r	average user request response time
$P_{idle,d}$	idle blade server power cost in domain d
P_t^{comp}	computing power consumption at interval t
P_t^{AC}	cooling power consumption at time interval t
$U_{im}(t)$	CPU utilization of i^{th} VM of tier m during t
K_ψ	unit electricity price (e.g. dollars/KWH)

resource usage in tier m. The average request response time can be expressed as:

$$r = r_{cpu} + r_{others} = \frac{1}{\lambda}\left(\sum_{m=1}^{M}\frac{u_m}{1-u_m}\right) + r_{others} \quad (8)$$

where the parameters are as explained in Table I. For simplicity, we also assume that the average service time of the non-CPU resources of each request is constant, since the effect of contention for these resources on the response time is negligible, i.e., $r_{others} = \gamma$ is a constant. The charging equation is defined as:

$$f(r) = \begin{cases} \omega(1 - \exp(r - r_{ref})) & \text{if } r < r_{ref} \\ 0 & \text{if } r \geq r_{ref} \end{cases}$$

where ω is the parameter to adjust the price rate, and r_{ref} is the reference SLA. If the response time exceeds the reference threshold, then it is considered an SLA violation, which generates null value, or may even involve a penalty. Therefore, the income at a time interval *t* is:

$$Income = \sum_{t} f(\frac{1}{\lambda}\left(\sum_{m=1}^{M}\frac{u_m}{1-u_m}\right) + \gamma) \quad (9)$$

Finally, the balance of payments of the data center at any operation time interval can be expressed as:

$$Balance = \sum_{t} f\left(\frac{1}{\lambda}\left(\sum_{m=1}^{M}\frac{u_m}{1-u_m}\right) + \gamma\right) - K_\psi * \alpha * \left\{\sum_{d=1}^{2}\sum_{n=1}^{N_d}\left(P_{idle,d} + \beta_d \sum_{m=1}^{M}\sum_{i=1}^{I_m} U_{im}(t)\right)\right\} * t \quad (10)$$

Equation 10 could be used as the criterion to coordinate the cyber and physical control modules. Any control decision should keep the balance a positive value, otherwise the control suggestion should be ignored.

Note that while current cloud operators do not provide interfaces to report SLA violations, as used in the algorithms described above, one contribution of this work is the demonstration of the utility of such interfaces, including to cloud data center operators. Numerous other efforts have demonstrated the utility of richer APIs between applications and the hosting platform [12], and select commercial products and stardatization efforts support flexible management interfaces for exchange of such information [13], [**?**].

V. Implementation

Workload Identification. The benchmark application used in this paper is a modified version of the Rice University Bidding System (RUBiS) [14], an online auction benchmark modeled after eBay. It has 22 transaction types, such as browsing for items and viewing user information. In our testbed, the servlet version of the application server is deployed. A RUBiS database is initialized with sufficient content for meaningful application behavior.

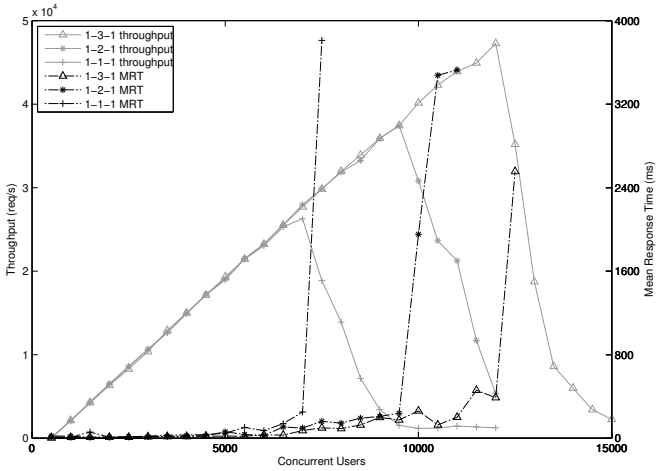

Figure 4. Application Performance under different application system configurations(w-a-d)

Two workload generators are used in experiments, one is the default RUBiS client emulator producing stationary workloads. Its deficiency is that the relative frequency of the different transaction types remains constant over time. The other generator is a custom workload generator that can replay transaction traces collected from production systems. The workload traces used in our experiments are obtained from the Internet Traffic Archives [15]. We then use the method introduced in [3] to identify and discretize patterns in the forecasted workload demand. It uses the dynamical programming algorithm to find a small number of time intervals and representative demand for each, also keeping the deviation from actual demand minimal. The final result is to represent the daily pattern in workloads by discretizing their demands into consecutive, disjoint time intervals with a single representative demand value in each interval. The workload pattern obtained from the trace file is shown in the next section.

Performance Profiling. In order to obtain empirical data for the application controller, we run a series of experiments to profile the performance of different application configurations (w-a-d), as shown in Figure 4. We observe that the performance of different w-a-d configurations is largely influenced by the number of application servers namely *a*. Based on these experiments, the system capacity *c*, i.e., the number of concurrent users can be supported for a given system configuration, represented as follows:

$$C(a) = \begin{cases} c \leq 7000 & \text{if } a = 1 \\ c \leq 9500 & \text{if } a = 2 \\ c \leq 12000 & \text{if } a = 3 \\ ... \end{cases}$$

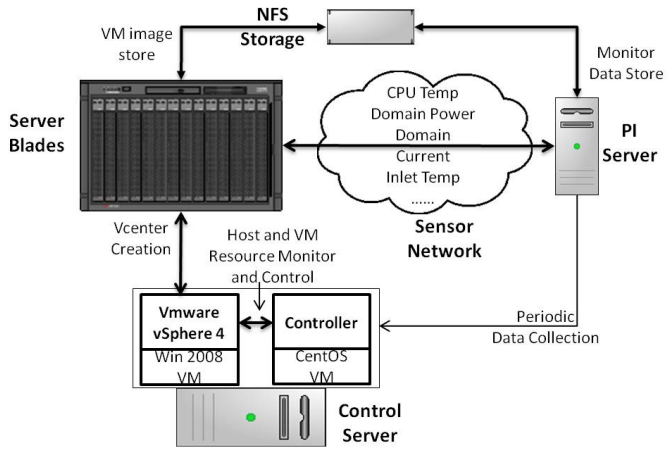

Figure 5. Experimental Infrastructure

Request response times will increase immediately when the number of concurrent users exceeds the system capacity limits. In addition, we collect resource utilization data during the experiments, and observe that the web and database servers are all under low CPU utilization, while the utilization of the application servers increases. It reaches the peak point when the experiment proceeds to the capacity limit, then stays around the peak point until the end of experiment.

Algorithm Implementation. For brevity, we summarize the implementation of the algorithms as follows. The cyber, i.e., application controller continuously performs two checks – one to adjust, if necessary, the number of application servers a, the second to adjust VMs' resource allocations on individual nodes, depending on the difference between the expected and real workload.

The physical CACM controller also consists of two phases. In the first (allocation) phase, the incoming workload is assigned to the more power efficient domain if the current difference is lower than the reference point, else it is assigned to the domain that has lower current. In the second (adjust) phase, the main objective is to reduce energy usage and inter-domain current imbalance. To do this, the algorithm first turns off idle VMs, then idle hosts, and finally, makes decisions to migrate VMs. A VM or host is considered idle if its time period of zero CPU usage exceeds some value. If migration is necessary, the candidate migration VMs are all selected from the domain that has the larger current, and the destination host is chosen from the other domain, as long as the current imbalance is above a preset threshold value.

A final coordination step in the EaCPS system evaluates the decisions coming from the cyber and physical side, and uses the balance of payments (BoP) model to make a final decision.

VI. Evaluation

Testbed Architecture. The architecture of the energy-aware management system is depicted in Figure 5. There are four main parts of the system: IBM BladeCenter, control server, PI server, and NFS storage.

- *BladeCenter*: its configuration and structure are described in Section III. All blades are virtualized with the VMware ESXi 4.0 hypervisor, and the management of our virtualized datacenter prototype is under VMware vSphere. All virtual machines in the BladeCenter are running ubuntu (64bit) Linux.
- *Control Server*: the control server accommodates two virtual machines. The first runs Windows 2008 and the VMware vSphere Client, which collects VM- and host-level information, and it also supports the execution of control commands such as VM migration, on/off,etc. The second VM runs CentOS and executes the resource allocation controller based on collected cyber information, as well as physical information such as CPU temperature, inlet temperature, chassis power and current draw, all gathered through the PI server.
- *PI server*: this server collects environmental information via a dedicated sensor network deployed in our data center, such as the inlet temperature of each BladeCenter, CPU temperature of each CPU, power, current and voltage information of each power strip outlet, PDU outlet, etc. The PI system is a commercial product provided by OSIsoft company[16].
- *NFS Storage*: to enable 'hot' VM migration, NFS storage is used to store all virtual machine images. This storage is accessible to all blade servers and the control server.

Cyber Control of Application Performance. We first present the results comparing the use of only the cyber control system, compared to a statically provisioned system.

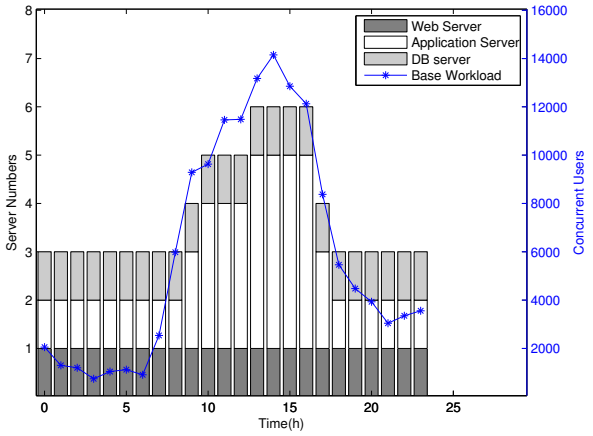

Figure 6. Dynamic System Configuration Change According to Incoming Base Workload

The results in Figures 6 and 7 show the benefits of using the cyber controller. The application controller adjusts the application's configuration dynamically according to the predicted base workload. As illustrated in Figure 6, the application system configuration dynamically changes with the base workload, according to the performance profiling of each system configuration identified in the last section. During the most intensive workload period, the number of application servers increases to four. As expected, Figure 7 shows that the performance of the dynamically changing system is much better than that of the statically configured one. The dynamic system's throughput has a similar curve as the base workload trace, while the throughput of the static system drops earlier, as soon as the workload reaches the system capacity limit. The statistical analysis of request response times shows that for an SLA of 100ms, the cyber controller helps reduce SLA violations of the application system by 25.5%.

Physical Control for Power Efficiency. Next, we evaluate the energy savings and performance impact of using only the CACM controller. In this experiments, we use 11 VMs, 6 of which are used for the RUBiS application, while the other 5 VMs are running a micro-benchmark that generates a specified CPU workload during some specified time period. In order to make the energy difference obvious, we make sure these VMs consume 80% of their CPU entitlements for the entire 30 minute duration of the experiment. The RUBiS benchmark uses the same workload trace as introduced in the previous section, but the system configuration for RUBiS is 1-4-1, namely, there are 4 static tomcat servers to process requests. We run the experiments two times for different scenarios.

- *Scenario 1:* we place all VMs in power domain A, which has 6 physical servers. One server is dedicated to the web tier virtual machine, each of the remaining servers host two virtual machines, one for RUBiS, the other one is running the mico-benchmark. There is no migration during the experiment. We denote this scenario as 'Static'.
- *Scenario 2:* The initial state is the same as in Scenario 1, but we start the CACM controller, which adjusts the placement of VMs during the experiment. This scenario is called 'CACM only'.

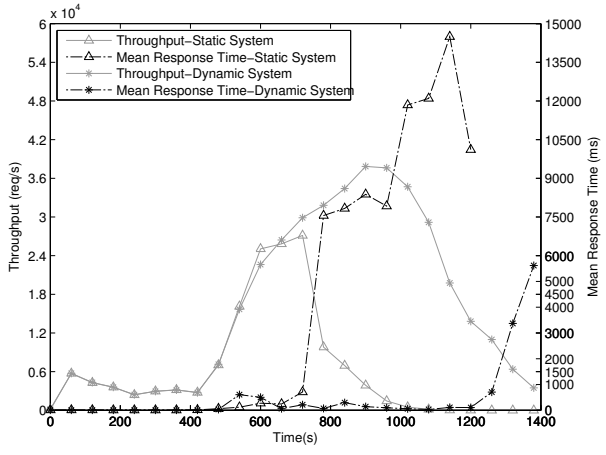

Figure 7. Application Performance Comparison between Cyber Control and Static Under-Provisioned System

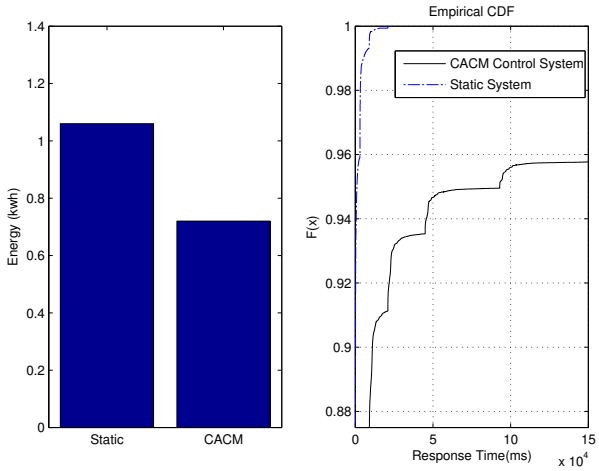

Figure 8. CACM Physical Controller Performance

For the experimental results shown in Figure 8, we observe that in this 30min experiment, the CACM controller reduces the energy consumption by about 32% (0.34 kwh) compared to the static scenario. However, this reduction comes with a significant drop in application performance. These results demonstrate that the use of the CACM controller, which considers the physical inputs only (i.e., current imbalance), can deteriorate performance, although it helps reduce energy consumption. This indicates the need for a solution like the balance of payments model in EaCPS. We present the results from the use of EaCPS next.

Coordinated Cyber-Physical Control System. We add two additional experimental scenarios to the previous section's experiments, and analyze the effectiveness of the coordinated cyber- and physical system controls:

- *Scenario 3:* all VMs are placed into power domain A. One difference from scenario 1 is that initially, we use only 3 VMs for RUBiS, and the same number of mico-benchmark VMs (5). Next, we enable the application controller, which dynamically changes the number of virtual machines for the RUBiS application during the experiment, according to the changes in base workload. We call this scenario 'App only'.
- *Scenario 4:* the initial state is the same as in Scenario 1, but we start the application and CACM controllers, as well as the coordinator module. We call this scenario 'EaCPS'.

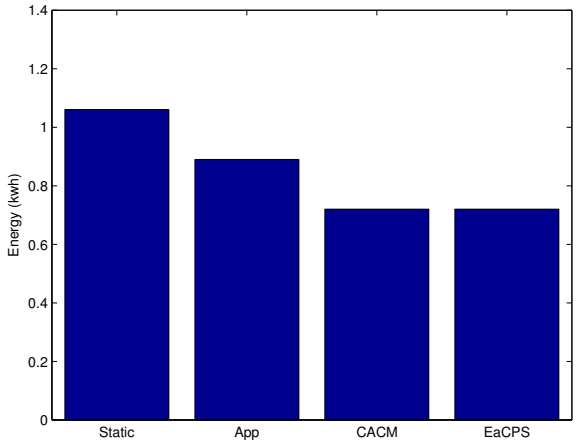

Figure 9. Power Consumption Comparison Among the Four different Scenarios

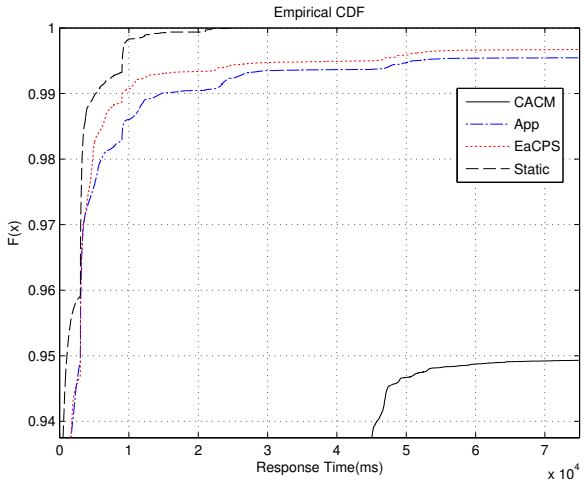

Figure 10. Application Performance Comparison Among the Four different Scenarios

From Figure 9 , 10, we see that the 'App only' scenario can result in up to 16% energy savings compared to the 'Static' scenario. This is caused by the resource over-provisioning in the static scenario compared to the 'App only' experiment. Clearly, this comes with some impact on performance, but the performance in the App only scenario still has 90% more requests with less than 200ms response time, which may be acceptable to typical cloud customers.

We next compare energy consumption and application performance for all four scenarios, as shown in Figures 9 and 10. The energy consumption data in Figure 9 is the consumption of the whole chassis during half an hour. We observe that the static scenario has the highest energy consumption, followed by the App scenario. The CACM and EaCPS scenarios have nearly the same energy consumption. These results, therefore, demonstrate the importance of using a physical control system for reducing energy usage for the whole system.

Conversely, the over-provisioned and the cyber control system result in better performance and quality of service control, as illustrated with the Static and App Only curves in Figure 10. As explained in Section III, the CACM control can deteriorate application performance while seeking energy savings, and this is evident in Figure 10. The EaCPS scenario, however, not only results in energy savings similar to CACM, but also in performance comparable to Static and App Only. In fact, as shown in the figure, EaCPS has better performance than the App Only scenario. This is somewhat counter intuitive. One possible explanation is that there are less migrations in the EaCPS scenario than in the App scenario, because the coordination controller cancels some migrations, thereby resulting in better performance for EaCPS. These results demonstrate both the feasibility and importance of integrating cyber and physical control mechanisms in data center management to achieve improved energy-efficiency and maintain desired performance levels.

VII. Related Work

One area of related work is the study of application performance management in virtualized server environments [11], [9], [4], [5]. Much of this research is focused on applying control theoretic methods to data center resource allocation management. [9] use an adaptive feedback controller in the resource partition among the components of the application to optimize application performance. [11] integrate feed-forward prediction and feedback reactive control for dynamically tuning virtual machine capacity.

Another area of related work is power reduction in data centers [17], [18], [3], where most such work is dedicated to optimizing resource allocation to save energy or maximize the utility of the data center. Typical work combines power regulation methods like DVFS or turning on/off idle machines with the characteristic of workloads for the target clusters.

Some previous work integrates performance and power into one optimization objective in data center operation [19], [20], [21], [22], [23], using an approach that combines thermal energy management with workload placement, to reduce cooling energy cost. Our work complements such efforts in focusing on other features of the physical environment to reduce energy consumption, notably the different power efficiencies between the two power domains used in modern blade servers.

VIII. Summary and Conclusions

This paper presents separate cyber vs. physical control systems. The cyber system controls the configuration of a dynamic multi-tier application, to manage application performance in lieu of workload changes. The physical control system performs electric current-aware workload placement according to the physical chassis environment. A balance of

payment model is used to combine the two systems, to find the best trade-off points between application performance and energy management. We use the RUBiS benchmark to evaluate the cyber control, physical control, and CPS coordinator modules in a mid-size fully instrumented data center. Results show that the CPS integrated system has the most energy savings with nearly equal application performance, compared to the static system configuration with over-provisioned resources. The outcome is that it is essential to integrate the cyber and physical systems used in data centers to achieve both sustainable energy savings and acceptable levels of application performance.

ACKNOWLEDGMENT

We wish to appreciate the anonymous reviewers who helped improve the quality of the paper. This work is partially supported by the ZTE Corporation and Chinese Government Fund.

REFERENCES

[1] J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Business Report, 2011. [Online]. Available: http://www.analyticspress.com/datacenters.html

[2] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmal, and R. Friedrich, "Smart cooling of data centers," *ASME Conference Proceedings*, vol. 2003, no. 36908b, pp. 129–137, 2003.

[3] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, and M. Marwah, "Minimizing data center sla violations and power consumption via hybrid resource provisioning," in *Second International Green Computing Conference (IGCC'11)*, 2011.

[4] J. Rao, X. Bu, K. Wang, and C.-Z. Xu, "Self-adaptive provisioning of virtualized resources in cloud computing," in *SIGMETRICS*, 2011, pp. 129–130.

[5] J. Rao, X. Bu, C.-Z. Xu, and K. Wang, "A distributed self-learning approach for elastic provisioning of virtualized cloud resources," in *MASCOTS*, 2011, pp. 45–54.

[6] C. Pu, A. Sahai, J. Parekh, G. Jung, J. Bae, Y.-K. Cha, T. Garcia, D. Irani, J. Lee, and Q. Lin, "An observation-based approach to performance characterization of distributed n-tier applications," in *Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization*, ser. IISWC '07, Washington, DC, USA, 2007, pp. 161–170.

[7] H. Chen, P. Kumar, M. Kesavan, K. Schwan, A. Gavrilovska, and Y. Joshi, "Spatially-aware optimization of energy consumption in consolidated datacenter systems," *ASME Conference Proceedings of InterPack'11*, 2011.

[8] H. Chen, M. Song, A. Gavrilovska, K. Schwan, M. Kesavan, and J. Song, "CACM: current-aware capacity management in consolidated server enclosures," in *Second International Green Computing Conference (IGCC'11),Work in Progress*, 2011.

[9] P. Xiong, Z. Wang, S. Malkowski, Q. Wang, D. Jayasinghe, and C. Pu, "Economical and robust provisioning of n-tier cloud workloads: A multi-level control approach," in *ICDCS*, 2011, pp. 571–580.

[10] Control system-wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Control_system

[11] Z. Wang, Y. Chen, D. Gmach, S. Singhal, B. J. Watson, W. Rivera, X. Zhu, and C. Hyser, "Appraise: application-level performance management in virtualized server environments," *IEEE Transactions on Network and Service Management*, vol. 6, no. 4, pp. 240–254, 2009.

[12] S. Kumar, V. Talwar, V. Kumar, P. Ranganathan, and K. Schwan, "vManage: Loosely Coupled Platform and Virtualization Management in Data Centers," in *6th International Conference on Autonomic Computing and Communications (ICAC)*, 2009.

[13] "IBM Tivoli Software," www.ibm.com/software/tivoli.

[14] Rubis homepage. [Online]. Available: http://rubis.ow2.org/

[15] The internet traffic archive. [Online]. Available: http://ita.ee.lbl.gov/

[16] The pi system. [Online]. Available: http://www.osisoft.com/Default.aspx

[17] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, no. 0, pp. –, 2011.

[18] V. Petrucci, E. V. Carrera, O. Loques, J. C. B. Leite, and D. Mossé, "Optimized management of power and performance for virtualized heterogeneous server clusters," in *CCGRID*, 2011, pp. 23–32.

[19] Y. Chen, D. Gmach, C. Hyser, Z. Wang, C. Bash, C. Hoover, and S. Singhal, "Integrated management of application performance, power and cooling in data centers," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, april 2010, pp. 615 –622.

[20] L. Parolini, N. Tolia, B. Sinopoli, and B. H. Krogh, "A cyber-physical systems approach to energy management in data centers," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '10. New York, NY, USA: ACM, 2010, pp. 168–177.

[21] X. Wang and Y. Wang, "Coordinating power control and performance management for virtualized server clusters," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 245 –259, feb. 2011.

[22] I. Rodero, E. K. Lee, D. Pompili, M. Parashar, M. Gamell, and R. J. Figueiredo, "Towards energy-efficient reactive thermal management in instrumented datacenters," in *GRID*, 2010, pp. 321–328.

[23] Q. Tang, S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 1458 –1472, nov. 2008.