Proceedings of the ASME 2011 Pacific Rim Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Systems,MEMS and NEMS
InterPACK 2011
July 6-8, 2011, Portland, Oregon, United States

InterPACK2011-52080

SPATIALLY-AWARE OPTIMIZATION OF ENERGY CONSUMPTION IN CONSOLIDATED DATA CENTER SYSTEMS

Hui Chen*, **Mukil Kesavan, Karsten Schwan, Ada Gavrilovska,**
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332

School of Computer*
Beijing University of Posts and Telecommunications
Beijing, China 100876

Pramod Kumar
Yogendra Joshi
G.W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

Energy efficiency in data center operation depends on many factors, including power distribution, thermal load and consequent cooling costs, and IT management in terms of how and where IT load is placed and moved under changing request loads. Current methods provided by vendors consolidate IT loads onto the smallest number of machines needed to meet application requirements. This paper's goal is to gain further improvements in energy efficiency by also making such methods 'spatially aware', so that load is placed onto machines in ways that respect the efficiency of both cooling and power usage, across and within racks. To help implement spatially aware load placement, we propose a model-based reinforcement learning method to learn and then predict the thermal distribution of different placements for incoming workloads. The method is trained with actual data captured in a fully instrumented data center facility. Experimental results showing notable differences in total power consumption for representative application loads indicate the utility of a two-level spatially-aware workload management (SpAWM) technique in which (i) load is distributed across racks in ways that recognize differences in cooling efficiencies and (ii) within racks, load is distributed so as to take into account cooling effectiveness due to local air flow. The technique is being implemented using online methods that continuously monitor current power and resource usage within and across racks, sense BladeCenter-level inlet temperatures, understand and manage IT load according to an environment's thermal map. Specifically, at data center level, monitoring informs SpAWM about power usage and thermal distribution across racks. At rack-level, SpAWM workload distribution is based on power caps provided by maximum inlet temperatures determined by CRAC speeds and supply air temperature. SpAWM can be realized as a set of management methods running in VMWare's ESXServer virtualization infrastructure. Its use has the potential of attaining up to 32% improvements on the CRAC supply temperature requirement compared to non-spatially aware techniques, which can lower the inlet temperature $2 \sim 3°C$, that is to say we can increase the CRAC supply temperature $2 \sim 3°C$ to save nearly 13% -18% cooling energy.

INTRODUCTION

Internet-based services supplied by companies like Amazon, Google, and Facebook, along with new models like cloud computing are fueling the construction and operation of ever increasing numbers of data center facilities. This has not only led to a rapid rise in their energy consumption where according to EPA reports, data centers accounted for roughly 1.5% of U.S.

*This author is a visiting PhD student in Georgia Tech from BUPT electricity use in 2006, with an estimated growth rate of 12% per year [1], but in addition, companies face the fact that annual data center op-ex costs can exceed the cap-ex costs of equipment acquisition and installation. Further, corporations and governments have become concerned about the environmental influence of data centers, including their carbon dioxide (CO_2) footprints. Motivated by the "green" or sustainability needs of data centers, this paper seeks to incorporate agressive energy efficiency technologies into the ways in which data center IT loads and their cooling infrastructures are managed. It is this joint consideration of both the IT and cooling infrastructures that distinguishes our work from prior efforts typically focused on specific subsystems like processor power consumption or chip vs. board vs. node/rack level power usage. Specifically, building on earlier work in Computer Science, we assume that future data centers will use cloud computing and virtualization technologies that can dynamically migrate IT workloads across machines, e.g., to consolidate them onto smaller numbers of servers without sacrificing service quality requirements [2]. We leverage research in Mechanical Engineering that considers heat distribution and transfer in the data center, to better place equipment or to design new power-efficient racks and enclosures [3, 4], but incontrast to such prior work, our approach is based on actual measurements obtained continuously and throughout the operation of a fully instrumented small-scale data center rather than relying on thermal models gained from simulation. We adopt machine learning methods [5, 6] to support the workload distribution decision process. Our approach and research are based on evidence from both our own prior work [3, 7, 8] and other efforts that additional efficiencies can be obtained from the holistic approach pursued in our work. Thermal-aware task scheduling based on models, for instance, has been shown useful for high performance systems and applications [9, 10]. We aim to generalize such methods to deal with cloud and enterprise systems in which virtualization technologies and dynamic application behaviors offer both new challenges and opportunities in terms of runtime IT load management.

The remainder of this paper is organized as follows. Section 2 discusses the backgroud of virtualization technology and thermal mapping in data centers, which is a key subproblem of the spatially aware workload distribution solutions sought in our work. Section 3 uses reinforcement learning to solve this subproblem, and Section 4 presents a spatially aware scheduler based on the optimal policy generated by the reinforcement learning model. Section 5 describes our experimental data center's configuration and reviews initial experimental results. Section 6 concludes the paper.

BACKGROUND

In this section, we will introduce some background about the virtualization technology that we used in our experiments, and the problem we want to solve in this paper.

Virtualization in Data Center

Virtualization is not a new technology [11], but due to the present of Virutal Machine Moniters (VMMs) such as Xen, ESXi, etc. and also the hardware support in Intel and AMD processors, it re-attracts industry and academic community's attention in these years. More and more companies have devoted to applying the virtualization technology into their new products, just like Amazon's EC2, Microsoft's Azure, VMware's vCenter, and so on. It almost has been a trend to virtualize the compute resource in data center, which is driven by the following three main benefits provided by virtualization.

First, Virtualization could consolidate multiple virtual systems into few physical servers. Compared to previous 5-20% average resource utilization in data centers [12], virtualization not only improve the resource utilization of physical servers, but also save the equipments, power, floor space, cooling, and management cost of data center operators. The use of virtual machines also provides other features like security, performance and fault isolation. The second benefit is utility computing services delivered by virtual machines, which make the computing widely available as water and electricity, and also customers are charged by usage, Amazon's EC2 is a successful example for this case. The last but not least benefit is the automatic resource management enabled by live virtual machine migration [13, 14], which allows for flexible resource allocation among the virtual machines to meet application performance requirements and even the system's high-availability that was traditionally only availabe through customized hardware or software [15].

Whereas the above benefits of virtualization, IDC estimated that 2010 will be the first year that more than 50% installed applications will run inside a virtual machine and predicted that more than 23% of all servers shipped in 2014 will be actively supporting virtual machine technology. In addition, more than 70% of all server workloads installed on new shipments in 2014 will reside in a virtual machine [16].

Determining The Data Center Thermal Map

A typical data center is laid out with a hot/cold aisle arrangement, by installing the racks and perforated floor tiles on a raised floor to which cold air is delivered by Computer Room Air Conditioner (CRAC) units [17]. The cooling air enters the racks from their front panel, removes heat while flowing through these racks, and exits from the rear of the racks. The heated air forms hot aisles behind the racks, from which it is extracted back to the CRAC unit's intakes, which in most cases, are positioned above the hot aisles.

The thermal maps of data centers arranged in this fashion are affected by many factors, including the assignments of computational tasks to data center nodes, the power consumption and the thermo-mechanical properties of IT devices, the CRAC's cooling capacity, and others [18,19]. Further, there are air circulation issues, such as the fact that due to the complex nature of airflows inside the data center, some of the hot exhaust air from server outlets will recirculate into the inlets of other servers. More generally, because the thermal distribution implicitly correlates with the energy cost of the data center, it is important to improve it to enhance energy efficiency or maximize the center's computational capabilities. Improperly designed or operated data centers may suffer from overheated servers and potential system failures, or from overcooled system and energy waste.

Thermal maps and their accurate prediction are useful for determining the thermal changes caused by different workload distributions, thus forming the basis of the SPAWM – spatially aware workload management – methods being developed in our work. [20, 21] employ a cross heat interference matrix to characterize heat recirculation between nodes, using CFD simulations to calculate the cross-interference co-efficients. This static co-efficient matrix is then used to advise workload distribution methods. However, in practice, cross heat interference is dynamic, affected by the air velocities and temperature supplied by the CRAC, thereby challenging the assumption that the amount of air and heat recirculated from the outlet of one server to the inlet of another is static and stable. In response, [5, 6] derived a machine learning method to construct an approximate thermal map, which used a trained neural net model to predict the output temperature under certain environmental inputs, using CFD simulation results to help train the neural net. We use the same approach, but employ actual experimental data to train the neural net. This then becomes the basis for the reinforcement learning methods described next.

A REINFORCEMENT LEARNING MODEL FOR SPAWM

We begin this section with an introduction to reinforcement learning and its applicability to data center workload distribution management, followed by a more precise formulation of the problem solved in this paper.

Reinforcement Learning and Its Applicability to SpAWM

Reinforcement Learning (RL) focused on how an agent ought to behave in a dynamic environment so as to maximize long term rewards defined by some high level goal. It refrers to a collection of trial-and-error methods by which an agent learns to make good decision through a sequence of interactions with envrionment. RL offers two advantages: (1) it does not require a prior model of either the system being considered or of the environmental dynamics, and (2) it is able to capture the delayed effects of a decision making task [22].

The purpose of RL is to search such a policy, which directs the agent to the best action it could take in the current state. Every action in a state is measured by a value function that estimates the future cumulative rewards obtained from taking this action. The reward information is propagated backward temporally in repeated interactions, eventually leading to an approximation of the value function. The optimal policy is essentially choosing the action that maximizes the value function in each state. The interactions consist of exploitations and explorations. Exploitation is to follow the existed policy; in contrast, exploration is the selection of random actions to capture changes in the environment so as to enable the refinement of existing policy [23].

Consider the whole data center as the agent environment, and the workload dispatch controller is the agent that interacts with the environment. The states are bladecenter resource utilization and inlet temperature, and possible changes to resource utilization form the set of actions. Each time the controller assigns an incoming workload to a specific node, it receives monitoring values describing the change of inlet temperatures. After sufficiently many interactions, the controller obtains good estimations for the assignment decisions at some given state. Starting from an arbitrary initial setting, the controller is able to balance the thermal distribution by following the optimal policy. Through exploration, the controller can modify its workload assignment policy according to the dynamics of incoming workloads.

RL theory is developed from Markov Decision Process (MDP) or Semi-MDP [24]. Formally, for a set of environment states S and a set of actions A, the MDP is defined by the transition probability $P_a(s,s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ and an immediate reward function $R = E[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s']$. At each step t, the agent perceives its current state $s_t \in A(s_t)$, the agent transits to the next state s_{t+1} and receives an immediate reward r_{t+1} from the environment. The value function of taking action a in state s can be defined as:

$$Q(s,a) = E\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right\} \tag{1}$$

where $0 \leq \gamma < 1$ is a discount factor helping $Q(s,a)$'s convergence.

Formulation of Workload Distribution Control as a RL Task

We describe an incoming workload by its projected additional utilization of the data center's computing resources. As a result, the workload distribution task is naturally formulated as a continuing discounted MDP, where the goal is to optimize the holistic energy cost. We define the reward function based on the BladeCenter's inlet temperature. The state space are the CPU utilization and inlet temperature of every BladeCenter, both of which are fully observable by the monitoring infrastructure.

Actions are the changes to the BladeCenter's utilization, and there are two kinds of actions that can lead to such changes: (1) dispatching the incoming workload, and (2) migrating existing workload. The workload distribution task is formulated as follows.

The state space. In the workload distribution task, the state space is defined to be the set of BladeCenter nodes' resource utilization, currently simply measured as CPU utilization values. So, each node i's state at time period t can be expressed as $< u_i(t) >$, and the whole data center's state can be noted as:$\langle u_1(t), \ldots u_i(t) \ldots, u_n(t) \rangle$, where $u_i(t)$ are the i^{th} BladeCenter's CPU utilization. The value of $u_i(t)$ should be in the range of $[0,1]$.

The actions. In the workload distribution process, we must consider the energy cost of the entire system. This means that if we must increase the power consumption of some computing device for the incoming workload, we then need to consider whether we can reduce cooling cost by not increasing the max inlet temperature of those compute nodes. One simple way of doing so is to find the "coldest" node to accomodate the incoming workload, but more generally, we must also balance node utilization in order to obtain high performance for the incoming workload. Conversely, if the incoming workload will not affect cooling cost, we should consider ways to reduce computing power costs, e.g., by consolidating workload and idling nodes.

The reward function. The long term cumulative reward is the optimization target of RL. In the workload distribution task, the desired workload assignments are the ones that optimize the whole system energy cost, including the computing device and the CRAC's energy cost. Cooling energy is modeled by its *coefficient of performance* (COP), 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^{in}_{sup}. So, the CRAC units will operate more efficiently by raising the temperature of the air supplied to the room. The cooling costs of the data center at slot t can be calculated as:

$$P^{AC}_t = \frac{P^{comp}_t}{CoP(T^{in}_{sup})} + P_{fan} \quad (2)$$

where P_{fan} is the total power required to drive the CRAC fans, which can be considered a linear function of fan speed. P^{comp}_t is the summary of the power consumption of all nodes, calculated by:

$$P^{comp}_t = \sum_{i=1}^{n} (\omega_i + \alpha_i u_i(t)) \quad (3)$$

where ω_i denotes power consumption of BladeCenter i at idle state, and α_i represents extra power consumption at full utilization for each BladeCenter. These two parameters will be given in our BladeCenter power profile in Section 5. From the calculation of P^{AC} and P^{Comp}, we can find that the T^{in}_{sup} is the important coefficient which influence the whole cooling cost. So we define the reward as:

$$reward = T^{in}_{redline} - max\left\{T^{in}_i\right\}, i \in [1,n] \quad (4)$$

where $T^{in}_{redline}$ is the maximum safe BladeCenter inlet temperature, which minus the maximum observed BladeCenter inlet temperature is the temperature T_{adj} to which we can adjust the CRAC. If T_{adj} is negative, it indicates that a BladeCenter inlet exceeds our maximum safe temperature. In response, we lower T_{sup} to bring the servers back to safe operation range, which means the reward actually is a kind of penalty, because lowering the T_{sup} will lead to the increase of cooling cost. Based on the equation the bigger the reward the higher T_{sup} we could set for CRAC, which means more cooling energy could be saved. So this reward function just tells us whether we can save or lose energy, where the concrete number can be calculated by the given P^{AC} and P^{comp} functions.

Solutions to the RL Task

The particular RL algorithm we used here is known as temporal-difference (TD) methods, which is an experience-based solution based on the theory that the average of the sample $Q(s,a)$ values collected approximates the actual value of $Q(s,a)$ given a sufficiently large number of samples [25]. A sample is in the form of (s_t, a_t, r_t), the $Q(s,a)$ is updated incrementally at each time a new sample is collected:

$$\Delta Q(s_t, a_t) = \alpha * [r_t + \gamma * Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (5)$$

where α is the learning rate, which decays to zero to ensure convergence, γ is the discount factor between 0 and 1 expressing the present value of expected future reward. The Q values are usually stored in a look-up table and updated during the traning process by writing new values to the corresponding entries in the table. In the workload distribution task, the RL-base agent issues workload dispatching or VM migration actions following an ε-greedy policy. With a small probability ε, the agent picks a random action, and follows the best policy it has found for most of the time. Starting from any initial policy, the agent will gradually refine the policy based on the feedback perceived at each step.

THE DESIGN AND IMPLEMENTATION OF SPAWM

In this section, we introduce SPAWM, a RL-based spatially aware workload management agent. Including too many servers

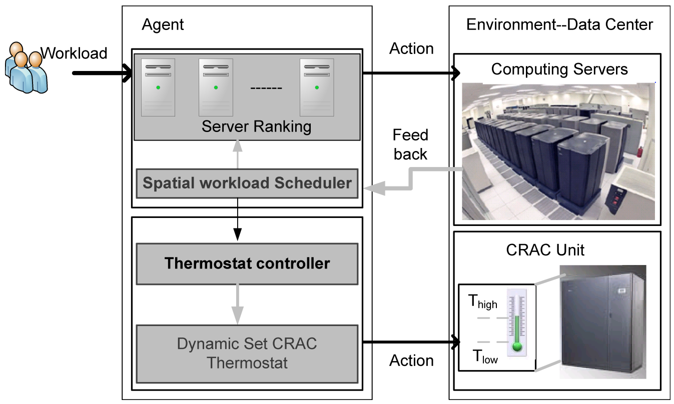

FIGURE 1. SPAWM ARCHITECTURE

in the RL problem creates challenges to the adaptability and scalability of SPAWM, which we address by employing model-based RL methods with two levels of approximation.

Overview

SPAWM is designed as a standalone daemon residing in VMware's vCloud hypervisor. It takes advantage of the control interface provided by vCloud to migrate VMs between the servers, which helps achieve the uniform thermal distribution in the data center to save cooling energy, and also considers workload balance among all active nodes as the second priority objective. Fig. 1 illustrates the architecture of SPAWM and the CRAC control environment. In this paper, we mainly focus on the interaction between SPAWM and the data center environment. The thermostat controller for the CRAC is being designed in complementary work carried out by our mechanical engineering collaborators. Its role is to receive thermal adjustment commands or alerts based on which it then reconfigures the CRACs. The current realization of SPAWM assumes that an incoming workload can be run on any machine in the data center.

Model Creation and Scalability

As introduced in previous section, the basic RL method uses the $Q(s,a)$ value table to decide which action is the optimal one for the current state, so the number of Q values grows exponentially with the state space. For our problem, because CPU utilization is a continuous integer between [0,1], the state space will be an infinite set. Obviously, the Q value table method is not appropriate for our situation. Instead of updating each $Q(s,a)$ value directly from the immediate reward recently collected, we employ environment models to generate a simulated experience for value function estimation. These environment models are implemented as a data structure that capture the relationship between current state, action and the observed reward. The model can be trained from previous collected samples in the form of (s_t, a_t, r_{t+1}). Once trained, a model is able to predict the r values for unseen state-action pairs.

Turning back to the problem considered in this paper, we must find the relationship between resource utilization, workload assignment, and thermal distribution. In such a multi-variable situation, a neural network is a good choice for determining this relationship from the recorded sample data. To do so, we selected standard multi-layer feed-forward back-propagation neural network (NN) with sigmoid activations and output to represent the environment model. This is because of the NN's ability to generalize from linear to non-linear relationships between the environment and the real-valued immediate reward. More importantly, it is easy to control the structure and complexity of the network by changing the number of hidden layers and the number of neurons in each layer. This flexibility facilitates the integration of a supervised learning algorithms with RL for better convergence. The performance of model-based RL algorithms depends on the accuracy of the environment model in generating simulated samples. Thus, the training samples used to train the model should be representative. In the implementation of the Q function, an NN-based function approximator replaces the tabular form. The NN function approximator takes the state-action pairs as input and outputs the approximated Q value. It directs the selection of workload assignment actions based on the ε-greedy policy.

Algorithm 1 shows the SPAWM online algorithm. It is designed to run forever until being stopped. At each workload assignment interval, SPAWM records the previous state and observes the actual immediate reward obtained. The next action is selected by ε-greedy policy according to outputs of the function

Algorithm 1: The SPAWM online algorithm

Initialize Q_{nn} to trained function approximator.
Initialize $t \leftarrow 0, a_t \leftarrow nop$.
Repeat
    $S_t(u_{1,t}, ..., u_{i,t}, ..., u_{n,t}) \leftarrow$ get_current_state();
    $W_t \leftarrow$ identify_workload();
    For i=1 to n
        $r_{i,t} \leftarrow Q_{nn}(u_{1,t}, ..., u_{i,t} + W_t, ..., u_{n,t})$;
    $a_t \leftarrow max(r_i)$;
    $r_{t,real} \leftarrow$observed_reward();
    $update\_R_{model}(S_t, a_t, r_{t+1}, R_{model})$;
    $update\_Q_{nn}(R_{model}, Q_{nn})$;
    $t \leftarrow t+1$;
util SPAWM is stopped.

approximator Q.

The use of environment models offers two advantages for RL tasks. First, model-based RL is more data efficient. With limited samples, the model is able to shed insight on unobserved rewards. Especially in online policy adaptation, the model is updated every time with new collected samples. The modified model generates simulated experiences to update the value function, and hence expedites policy adaptation. Second, the immediate reward models can be reused in a similar environment. In online adaptation, once SPAWM identifies that the incoming workload's resource requirement is similar to a previous workload, the corresponding model is re-used. Instead of starting from scratch, the reuse of previous models is equivalent to starting from guided domain knowledge, which again improves the whole data center's thermal distribution.

In model-based RL, the scalability problem is alleviated by the model's ability to cope with relative scarcity of data in large scale problems. The conventional table-based Q values can be updated using the batch of experiences generated by the environment model. However, the table-based Q representation requires a full population using the rewards simulated by the model. This is problematic when the RL problem scale up. In SPAWM, we use netural network to generate the approximation value function, which helps to reduce the time in updating the value function in every workload assignment.

Neural Network Based Learned Value Function

There are several off-the-shelf neural network development libraries, enabling us to leverage these techniques rapidly. We selected the Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network Library (PyBrain) [26], which contains algorithms for neural networks and reinforcement learning, allowing us to combine the two to build our model and value function approximators to cope with the large dimensionality state space. We use the neural net to predict how heat is generated and flows within the data center. There will be N inputs to the model, which denote every BladeCenter's CPU utilization, and the outputs will be the inlet temperature of every BladeCenter. Between the input layer and the output layer, there are L internal or hidden layers. Each layer contains a set of of elements known as neurons. Each neuron j accepts N_j inputs from the previous layer, applies a weighting factor $w_{j,a}$ to each input I_a, and uses the sum of the weighted inputs as the I-value for its activation function f. The result of this function y_i is passed to neuron in the next layer. After we get the output from the netural network predictor, we could then use the Eqn. 4 to calculate the approximate reward of the input state. The RL model can then use such an approximate reward value to decide which action to choose.

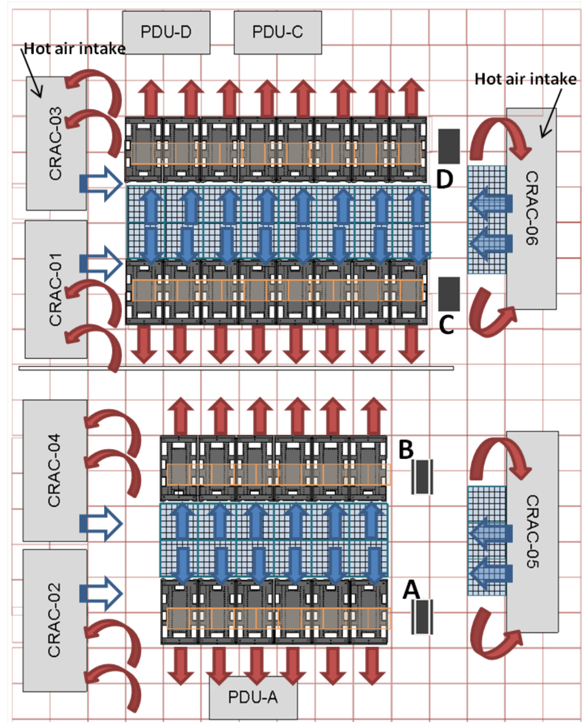

FIGURE 2. DATA CENTER LAYOUT

EVALUATION

This section introduces our experiment data center layout and the BladeCenter's power profile. We then present the training results of the neural network, which is used to calculate the value function of the RL model. Last, we analyze the output of our RL model experiment.

Data Center Layout

We study a typical medium-sized data center which has four rows as shown in Fig. 2. The data center has alternating "hot" and "cold" aisles. Cooling air from the computer room air conditioner (CRAC) reaches the front of the racks in the cold aisles via an air supply plenum under a raised floor through perforated floor tiles. The heated exhaust air is recirculated from the hot aisles at the rear of the racks back to the CRAC. Since there is some additional hot air flow back to the front of the racks, it is difficult to accurately predict thermal distribution among these BladeCenters. Finally, according to the available resources in our data center, we use 4 racks in row C, 6 racks in row D. there are totally 10 racks are used for our experiment, each rack has 6 BladeCenters, and each BladeCenter has 14 blade servers, each with 4 cores. So, there are 840 servers/3360 cores running in our experiment.

A sensor network is used to monitor the inlet temperature, fan speed, and CPU temperature of every BladeCenter, and from

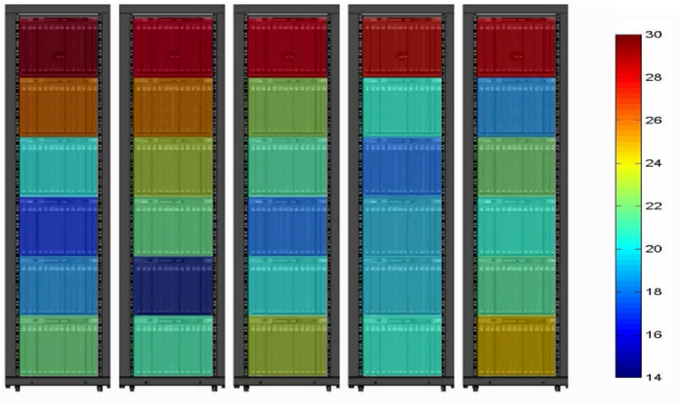

FIGURE 3. THERMAL DISTRIBUTION AMONG RACKS

the collected data, we observe that the top BladeCenter usually is the hottest one, as presented in Fig. 3, and lower temperatures observed at different rack heights across the cold aisle. This is caused by the tile air discharge velocity decrease along the rack heights, so the top BladeCenter is usually wrapped by the hot air which is to be exhausted by CRACs. Also the cold air reaches the top BladeCenter at a weak speed that is another reason for such thermal distribution in the data center. Further, because the cold air velocity varies across the cold aisle, the coldest node in each rack is different. These facts indicate that data center layout is an important factor to be considered when disseminating the incoming workload, thereby motivating the spatially aware workload distribution methods advocated in our work.

Power Profile of BladeCenter

In our data center, all racks are occupied by the IBM 8677-3XU BladeCenter, each BladeCenter has 14 BladeServers (8850-Y15), which has 2 x Dual-Core Opteron 270 / 2 GHz and RAM 4 GB. There are 4 power line cords for each BladeCenter, so we can only find out the relationship between power and resource utilization to the whole BladeCenter. In order to determine this correlation, we instrumented the whole BladeCenter with the benchmark application we wrote to get the power usage at different CPU utilization levels. The idle BladeCenter power consumption (ω) and the extra power consumption at full utilization for the BladeCenter (α) were computed through the collected data. This results is presented in Fig. 4, also we get the fitting function :

$$P_t = 2070 + 5.44 * u_t, \quad u_t \in [0, 100] \tag{6}$$

Neural Network Training

Neural Network Architecture: We used a two-layer feed-forward neural network to do the function fitting, with a sigmoid

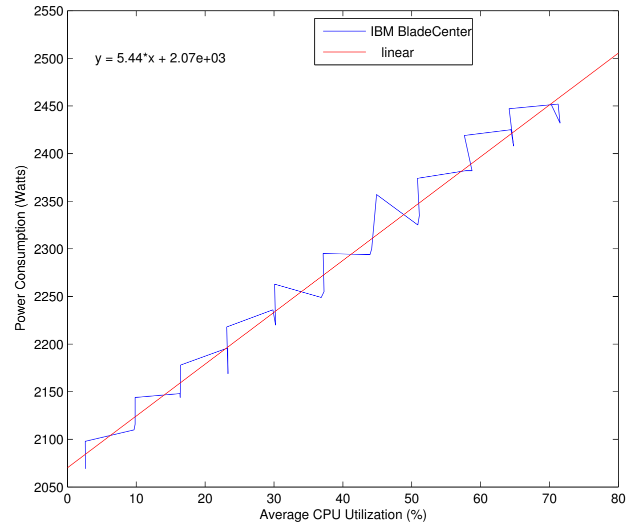

FIGURE 4. POWER PROFILE OF IBM BLADECENTER E

transfer function in the hidden layer and a linear transfer function in the output layer as showed in Fig. 5. The number of hidden neurons is set to 20, which is a tradeoff between the computing efficiency and network training performance. The output neurons is set to 6 which are responsible for the predicted inlet temperature output of each BladeCenter.

Training Data Collection: In order to collect the training data for the neural network, we run four different kinds of benchmark applications in our data center, using the workload traces provided by the public Google Cluster Data [27]. During the 7-hour experiment, all related IT resource utilization and physical environment data were collected and then processed to determine the inlet air temperature and power utilization of each BladeCenter, CPU utilization of each blade server, etc. Because the whole data is too large, we just take one rack for example, Fig. 6 depicts the change of temperature and workload of all the BladeCenters (BCs) in the rack C2 during the experiment. We could find that in the non spatially-ware situation, the workload continually be assigned to the 'hottest' BladeCenter – BC6 – in our example, which makes the inlet temperature of BC6 remains the hottest one during the experiment.

Training Performance: The collected data are then used to train the neural network model to find out the complex cor-

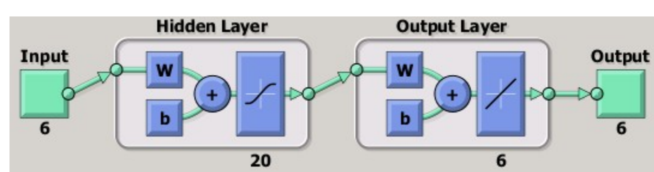

FIGURE 5. NEURAL NETWORK ARCHITECTURE

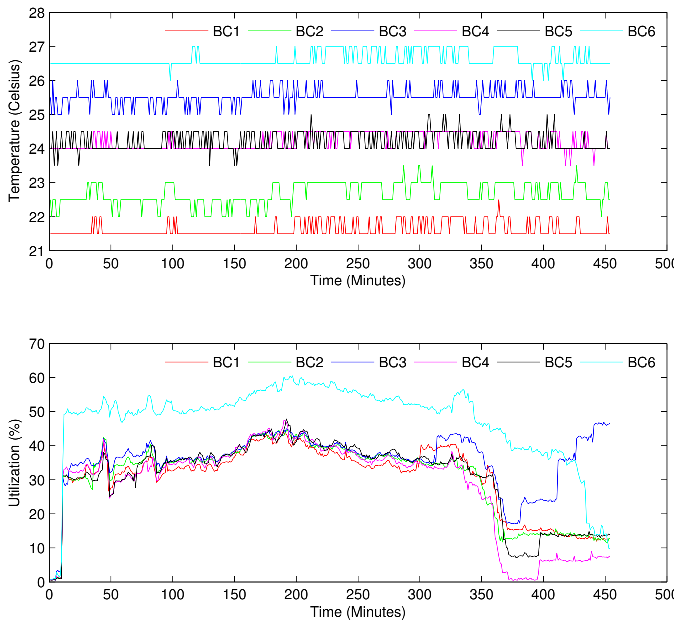

FIGURE 6. TEMPERATURE AND WORKLOAD CHANGE

relation between the CPU utilization and inlet air temperature. The input of the neural network is the real CPU utilization data we collected in the experiment, these data went through the neural network then were transferred to the predicted inlet temperature data of each BladeCenter. The comparison between the real data and predicted data is showed in Fig.7, we could find the outputs of neural network and real data nearly overlapped. The trained performance is presented in Fig. 8. R is used to measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship. In our training result,$R = 0.99209$, which indicates that the prediction of the neural network has high precision. Now we can nearly exactly infer the inlet air temperature of the BladeCenter from the BladeCenter's CPU utilization, which makes it possible to use the RL model as stated in Section 3 to advise the placement of incoming workloads.

System Performance Analysis

We next simulate the workload distribution among one rack of BladeCenters to test our spatially aware RL model. In order to fully utilize the data we have collected from the system, all workload information for BladeCenter C2 has been sampled out from the whole system workload log file, and only the resource utilization increasing period is used in this experiment, because we need to compare the spatially aware RL model to the default VMware center's dynamic power management (DPM) policy. The metric we use to compare here is the reward calculated by Eqn. 4. The workload is the same as in the last experiment,

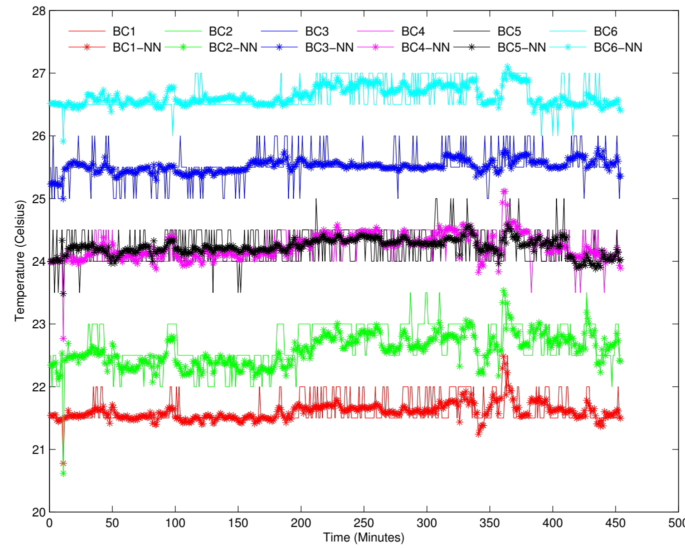

FIGURE 7. NEURAL NETWORK PREDICTION AND REAL DATA

only converted to a CPU resource request in the simulation.

We choose one state in the real experiment as the beginning state of our RL model, then start to deal with the same incoming resource requirement, i.e., for the CPU resource. The simulation result is illustrated in Fig. 9. The observed reward is the reward we calclulated from the actual experiment log data. Because the resource allocation policy in VMware's VMM places more emphasis on workload balance, one of the BladeCenters

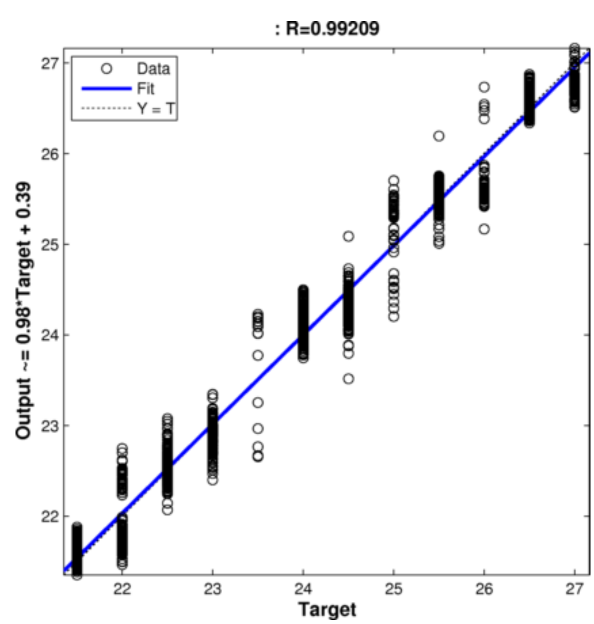

FIGURE 8. NEURAL NETWORK TRAINED PERFORMANCE

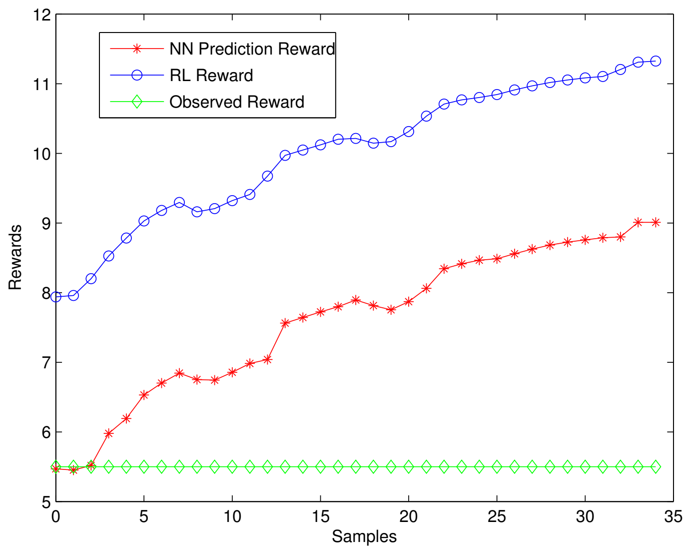

FIGURE 9. REWARDS UNDER DIFFERENT POLICY

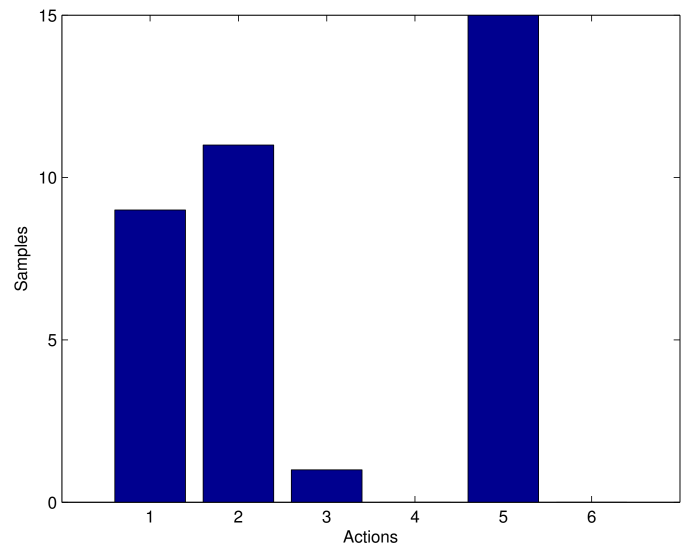

FIGURE 10. BEST ACTIONS UNDER RL POLICY

has the highest temperature for a long time, so that the observed reward did not change during this time period. If we had instead used the neural network to predict the next step inlet temperatures, and then chosen the best one to execute the workload as described in Algorithm 1, the reward is greater than the default policy in VMware's ESX server. The RL model shows more inprovement based on the neural network model, because it considers the future reward in the next state, which means that the reward is larger than the one based only on the neural network. This results in an average improvement of roughly 32%, approximately 2.4°C, the outcome being nearly 13% -18% cooling power savings in the data center,which is calculated through Eqn. 2.

We also monitor the actions chosen by the RL model at every step. We found that it changed frequently, largely dependent on the initial state and the incoming workload. But we can find that the RL model rarely assigns workload to BladeCenter 6, which usually is the hotest one in the rack, and BladeCenter 1, 2 and 5 have more chances to be chosen as the destination of the incoming workload. Compared to Fig. 3, we can find that BladeCenter 1, 2 and 5 usually are colder than other BladeCenters. This assignment strategy results in cooling energy savings since workload placement on a colder node has less influence on cooling energy cost than on a hotter node.

CONCLUSIONS AND FUTURE WORK

This paper presents SPAWM, a spatially aware workload and thermal management system for data centers. The system uses online monitoring to continuously observe a data center's cooling and IT subsystems, and then applies reinforcement learning and neural network techniques to implement optimizations that deal with data center workload, thermal distribution, and cooling facilities. SPAWM approximates a uniform thermal distribution in the rack through thermally aware workload placement, which can reduce the max inlet temperature 2–3°C. In turn, this permits CRAC fan speeds to be slowed down to reduce cooling energy.

Current results with SPAWM demonstrate the potential utility of the approach, and future work will deploy SPAWM in the vCloud enterprise software deployed in a small-scale, fully instrumented data center operated by our groups. The idea is to construct an online control module that can update its policies at runtime and along with changes in the data center's cyberphysical environment.

ACKNOWLEDGMENT

Thanks go to Chad Huneycutt and Vikneshan Sundaralingam who helped build the monitoring and experiment infrastructure. We appreciate the anonymous reviewers who helped improve the quality of the paper.

REFERENCES

[1] EPA, 2006. Report to congress on server and data center energy efficient public law 109-431. Tech. rep., U.S. Environmental Protection Agency ENERGY STAR Program, Dec. `http://www.energystar.gov/`.

[2] Nathuji, R., Isci, C., Gorbatov, E., and Schwan, K., 2008. "Providing platform heterogeneity-awareness for data center power management". *Cluster Computing,* ***11***(3), pp. 259–271.

[3] Samadiani, E., Amur, H., Krishnan, B., Joshi, Y., and Schwan, K., 2010. "Coordinated optimization of cooling and it power in data centers". *Journal of Electronic Packaging,* ***132***(3), p. 031006.

[4] Tolia, N., Wang, Z., Ranganathan, P., Bash, C., Marwah, M., and Zhu, X., 2009. "Unified power and cooling management in server enclosures". In Proceedings of the ASME/Pacific Rim Electronic Packaging Technical Conference and Exhibition (InterPACK '09).

[5] Moore, J. D., Chase, J. S., and Ranganathan, P., 2006. "Weatherman: Automated, online and predictive thermal mapping and management for data centers". In Proceedings of the 3rd International Conference on Autonomic Computing, ICAC 2006, Dublin, Ireland, 13-16 June, pp. 155–164.

[6] Shrivastava, S. K., VanGilder, J. W., and Sammakia, B. G., 2007. "Data center cooling prediction using artificial neural network". *ASME Conference Proceedings,* ***2007***(42770), pp. 765–771.

[7] Krishnan, B., Amur, H., Gavrilovska, A., and Schwan, K., 2010. "Vm power metering: feasibility and challenges". *SIGMETRICS Performance Evaluation Review,* ***38***(3), pp. 56–60.

[8] Nathuji, R., Somani, A., Schwan, K., and Joshi, Y., 2008. "CoolIT: coordinating facility and it management for efficient datacenters". In Proceedings of the 2008 conference on Power aware computing and systems, HotPower'08, USENIX Association, pp. 15–15.

[9] Banerjee, A., Mukherjee, T., Varsamopoulos, G., and Gupta, S. K. S., 2010. "Cooling-aware and thermal-aware workload placement for green hpc data centers". In Proceedings International Conference on Green Computing Conference (IGCC2010).

[10] Varsamopoulos, G., Banerjee, A., and Gupta, S. K. S., 2009. "Energy efficiency of thermal-aware job scheduling algorithms under various cooling models". In Proceedings of International Conference on Contemporary Computing, ICCC, pp. 568–580.

[11] Goldberg, R. P., 1974. "Survey of Virtual Machine Research". *IEEE Computer Magazine,* ***7***, June, pp. 34–45.

[12] Vogels, W., 2008. "Beyond server consolidation". *ACM Queue,* ***6***, January, pp. 20–26.

[13] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., and Warfield, A., 2005. "Live migration of virtual machines". In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2, NSDI'05, USENIX Association, pp. 273–286.

[14] Nelson, M., Lim, B.-H., and Hutchins, G., 2005. "Fast transparent migration for virtual machines". In Proceedings of the annual conference on USENIX Annual Technical Conference, ATEC '05, USENIX Association, pp. 25–25.

[15] Cully, B., Lefebvre, G., Meyer, D. T., Feeley, M., Hutchinson, N. C., and Warfield, A., 2008. "Remus: High availability via asynchronous virtual machine replication. (best paper)". In NSDI, p. 161.

[16] IDC, 2010. Worldwide virtual machine software 20102014 forecast: A first look. Website, Mar.

[17] Patel, C. D., Bash, C. E., Sharma, R., Beitelmal, M., and Friedrich, R., 2003. "Smart cooling of data centers". *ASME Conference Proceedings,* ***2003***(36908b), pp. 129–137.

[18] Schmidt, R. R., Cruz, E. E., and Iyengar, M. K., 2005. "Challenges of data center thermal management". *IBM J. Res. Dev.,* ***49***, July, pp. 709–723.

[19] Schmidt, R., and Iyengar, M., 2009. "Thermodynamics of information technology data centers". *IBM J. Res. Dev.,* ***53***, May, pp. 449–463.

[20] Tang, Q., Mukherjee, T., Gupta, S. K. S., and Cayton, P., 2006. "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters". In Proceedings of International Conference on Intelligent Sensing and Information Processing (ICISIP2006).

[21] Pakbaznia, E., and Pedram, M., 2009. "Minimizing data center cooling and server power costs". In Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design, ISLPED '09, ACM.

[22] Tesauro, G., Jong, N. K., Das, R., and Bennani, M. N., 2007. "On the use of hybrid reinforcement learning for autonomic resource allocation". *Cluster Computing,* ***10***(3), pp. 287–299.

[23] Rao, J., Bu, X., Xu, C.-Z., Wang, L., and Yin, G., 2009. "Vconf: a reinforcement learning approach to virtual machines auto-configuration". In Proceedings of the 6th international conference on Autonomic computing, ICAC '09, ACM, pp. 137–146.

[24] Gosavi, A., 2009. "Reinforcement learning: A tutorial survey and recent advances". *INFORMS J. on Computing,* ***21***, April, pp. 178–192.

[25] Sutton, R. S., and Barto, A. G., 1998. *Reinforcement Learning: An Introduction*. MIT Press.

[26] Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., Rückstieß, T., and Schmidhuber, J., 2010. "Pybrain". *Journal of Machine Learning Research*.

[27] Google, 2009. Google cluster data,. Website. URL http://code.google.com/p/googleclusterdata/.