Global Power Manager (GPM) for Data Centers

Sanjay Bharadwaj, Fatemah Panahi, Maheswaran Venkatachalam
Department of Computer Science, University of Wisconsin-Madison
{sanjaysb,fatemeh,kvmakes}@cs.wisc.edu

Abstract

Reducing data center power consumption is an important challenge with the increase in the demand for their services. We propose a Global Power Manger (GPM) for data centers. Global Power Manager migrates services around the data center with the goal of minimizing energy consumption in the data center. GPM has a holistic view of the data center and can considers any power-consuming element, such as networking equipment, in making its decisions and works with any given data center topology. Our preliminary results show the correctness of the model while at the same time leaving us with a full-fledged simulator that can be used for running traces from larger data-centers.

1 Introduction

Over the last 15 years, data centers have significantly grown both in size and application. Moreover, due to the increase in popularity of cloud-based services, the cloud service providers (such as Amazon, Google, and Microsoft) have grown in size. Data centers have significant power consumption. Servers in these data centers currently account for nearly 1.5% of the total electricity consumption in the U.S. at a cost of approximately $4.5 billion per year [1]. Given the extremely high consumption, it is imperative to increase efficiency in energy consumption as much as possible.

Extensive research has been done on reducing the data center energy consumption. However, designing an effective approach for power saving is very challenging. The reason is that many components contribute to the power consumption in a data center. Therefore, there are a large number of opportunities for power saving. A nave design might lead to conflicts in the data center. For example, consolidating workloads to a single rack just because the individual servers can handle the CPU and bandwidth loads does not guarantee that the Top of Rack/aggregation switches can handle the load. The next challenge is designing a scalable model so that all data that is input to the model is gathered in real-time from different components and the decisions are made in a timely manner. Moreover, the model should utilize efficient migration schemes so that the latency in providing the services is not dramatically increased. This is specifically important in case that the cloud provider has promised a certain performance and latency levels to the customer.

We propose a Global Power Manager (GPM) for data centers to reduce power consumption. In proposing the GPM, we pursue the following goals:

- GPM should have a holistic view of the data center, and specifically consider the power consumption of the networking equipment in addition to the servers.
- GPM should leverage consolidation opportunities at all granularities (For example, servers and racks).
- GPM should make constraint aware decisions to prevent conflicts.
- GPM should maintain the data center performance and have minimal overhead.

In designing the GPM we will assume that the state of the art saving approaches exists. The GPM also will have knowledge about the different components that consume energy in the data center and the topology of the data center. Building upon the state of the art saving approaches, and considering all elements involved in the power consumption in a data center, the GPM can make informed decisions for changing certain configurations, migrating services between servers, and idling/shutting down parts of the data center in order to reduce the power consumption.

2 Model

We propose a mathematical model that tries to optimize the energy consumption in a data center by migrating services such that the overall power consumption of the data center is minimized. Our formulas are subject to constraints that make sure that the performance and latency of the service is acceptable and that saving approaches do not conflict with each other. In this section, we describe the relationships between the utilization of servers and network components to their power consumption and present the various invariants that must hold in a valid configuration. These invariants are used as constraints in an LP formulation of the scheduling problem.

The first step in modeling the scheduled state of the data center is to obtain the mapping from virtual machine instances to physical servers at a given instant. We represent this as a two dimensional array X:

$$X_{ij} = \left\{ \begin{array}{cc} 1, & ifVM_j runson S_i \\ 0, & Otherwise \end{array} \right\} \quad (1)$$

The utilization of each resource on a server is the sum of the respective utilizations of all the virtual machines running on the server. Thus the utilization of the ith server is given by:

$$U_i = \left(\frac{100}{cpuMax_i} \right) \sum_{j=1}^{m} X_{ij} * Uvm_j \quad (2)$$

$$U\_bw_i = \sum_{j=1}^{m} X_{ij} * B_j \quad (3)$$

$$U\_mem_i = \sum_{j=1}^{m} X_{ij} * M_j \quad (4)$$

In the following discussion, we refer to an application at the granularity of a virtual machine instance and if multiple applications are running on the same VM instance, their utilizations are summed up. This means that statistical inference used by the GPM for each VM instance would be the combined behavior of all the applications on the VM instance. One major reason for this decision is that the environment used in our testing and simulation is a virtualized environment and it is best to use the largest instance that can be migrated as possible to minimize the variables in the optimization problem.

Among the various resources available to a virtual machine, we presently monitor the CPU and network bandwidth utilizations of each instance in our model as shown above. The CPU utilization is a natural choice to be monitored due to its direct linear correlation to the power consumption of the server. The CPU alone can account for up to 30% of the total server power consumption and directly influences other system components utilization percentages.

At this time, we do not use utilization of system memory as an optimizing requirement in the model although memory is one of the largest consumers of power in a server. The reason for this decision is that although memory has a fixed upper bound much like other resources in the system, the requirements of memory for any given application is flexible and the relationship between performance of an application and its memory allocation are not universal. Secondly, accurately measuring the useful memory utilization by any external agent such as the GPM is difficult in a real time environment since it cannot measure what sections of the memory allocated to an application are live and required in the near future. But we do not rule out the possibility of acquiring this information from the application itself. With data centers being increasingly used for cloud based services and each data center having a platform for application development, it is feasible to require all cloud based applications to provide their memory usage values to the GPM through a standard API. At this time, memory allocation to an application (in our case a VM instance) is used for modeling the migration overhead as explained later in this section.

The next requirement to model the global state is some information reflecting the network topology of the datacenter. In our present framework, it is sufficient for us to know what network switching and routing elements form a critical set to a server. We define the critical set as comprising those network components (NC below) that are vital to maintain the connectivity of the server both to the external network as well as any other section of the data center network that the server may require access to. While in reality this component can be expected to be fairly dynamic, our present model considers it a constant. It is part of our future work to analyze the definition of the critical set for different datacenter network architectures many of whose primary goal is to provide redundancy in paths available and hence our definition does not directly apply and their effect on the optimization problem. The mapping is represented by the two dimensional array N:

$$U\_nc_k = 100 * \frac{\sum_{i=1}^{n} N_{ik} * U\_bw_i}{capNC_k} \quad (5)$$

$$N_{ik} = \left\{ \begin{array}{cc} 1, & ifNC_k iscriticalto S_i \\ 0, & Otherwise \end{array} \right\} \quad (6)$$

We use this information to direct aggregation of jobs on servers based on their network location that we hope reflects their geographic layout in the datacenter. The benefits of this are twofold.

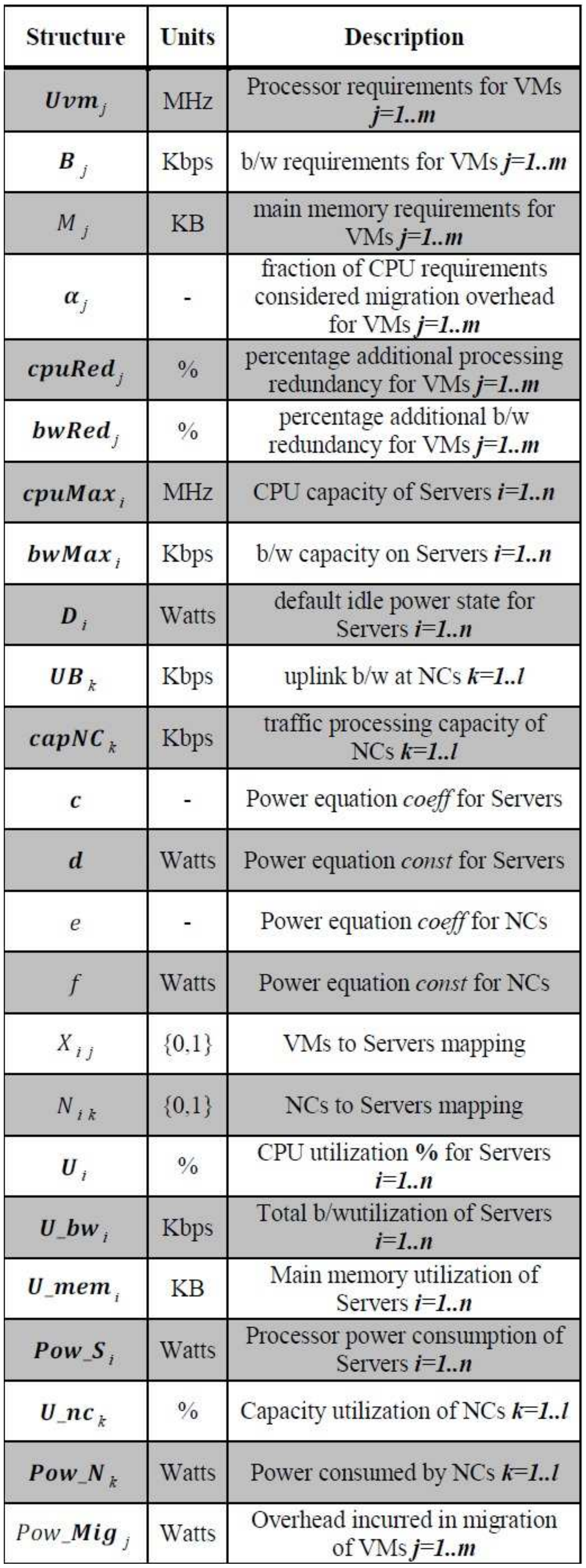
First, by aggregating jobs based on their network location we are potentially making some of the network components redundant. These switches and routers can be powered down to an acceptable idle state.

Second, and the reason we believe this is a particularly important consideration not taken into account in other works, is the fact that under the assumption the network layout resembles the physical layout of the datacenter, the flexibility of the cooling infrastructure can be leveraged to reduce power consumption further by powering down most of the cooling components in idles sections of the data center. Modern data center cooling systems have fine grained control over sections as small as racks and containers to entire warehouses. If for example, entire containers are cleared of all running jobs by aggregating them with the workload in other containers, this containers power supply (that includes the power supply to servers, switches and the cooling infrastructure) can be completely switched off.

Other structures and variables that hold state information are mentioned in the table below.

Now we present the power models for the servers and network components. Studies have shown that the power consumed by a server is linearly proportional to its processor utilization [6]. We model it as follows.

$$Pow\_S_i = \left\{ \begin{array}{cc} D_i, & U_i = 0 \\ c * U_i + d, & U_i > 0 \end{array} \right\} \quad (7)$$



Structure	Units	Description
Uvm_j	MHz	Processor requirements for VMs ***j=1..m***
B_j	Kbps	b/w requirements for VMs ***j=1..m***
M_j	KB	main memory requirements for VMs ***j=1..m***
α_j	-	fraction of CPU requirements considered migration overhead for VMs ***j=1..m***
$cpuRed_j$	%	percentage additional processing redundancy for VMs ***j=1..m***
$bwRed_j$	%	percentage additional b/w redundancy for VMs ***j=1..m***
$cpuMax_i$	MHz	CPU capacity of Servers ***i=1..n***
$bwMax_i$	Kbps	b/w capacity on Servers ***i=1..n***
D_i	Watts	default idle power state for Servers ***i=1..n***
UB_k	Kbps	uplink b/w at NCs ***k=1..l***
$capNC_k$	Kbps	traffic processing capacity of NCs ***k=1..l***
c	-	Power equation *coeff* for Servers
d	Watts	Power equation *const* for Servers
e	-	Power equation *coeff* for NCs
f	Watts	Power equation *const* for NCs
X_{ij}	{0,1}	VMs to Servers mapping
N_{ik}	{0,1}	NCs to Servers mapping
U_i	%	CPU utilization **%** for Servers ***i=1..n***
U_bw_i	Kbps	Total b/wutilization of Servers ***i=1..n***
U_mem_i	KB	Main memory utilization of Servers ***i=1..n***
Pow_S_i	Watts	Processor power consumption of Servers ***i=1..n***
U_nc_k	%	Capacity utilization of NCs ***k=1..l***
Pow_N_k	Watts	Power consumed by NCs ***k=1..l***
Pow_Mig_j	Watts	Overhead incurred in migration of VMs ***j=1..m***

Figure 1: Model variables and parameters

Where U_i is as defined previously. Our proposal rests on the assumption that D_i is significantly lower than d In our simulations as presented in the following sections, we assume D_i =0 for all servers. In practice, we recognize that datacenters are heterogeneous in terms of their hardware acquired over years and hence there may be one or more unique class of equipment that cannot be afforded to shut down. Thus the model allows for unique values that can represent any P-state that the equipment possesses to be provided to individual elements as and when appropriate. Papers such as Powernap [4] describe the means to achieve these low power states and improve efficiency. The GPM is a decision making engine that would employ different controllers to act upon its decisions thus allowing any number of similar technologies and solutions to coexist.

The power models of network components are also linear in nature as shown for switches by ChamaraGunaratne et al. [7] and for core routers by Bruce Nordman [8]. The network component utilization is the sum of the bandwidth requirements of all VM instances running on the servers in whose critical set this NC belongs. Thus NC utilization is:

$$U\_nc_k = 100 * \frac{\sum_{i=1}^{n} N_{ik} * U\_bw_i}{capNC_k} \tag{8}$$

and power consumption:

$$Pow\_N_k = \left\{ \begin{array}{cc} 0, & U\_nc_k = 0 \\ e * U\_nc_k + f, & U\_nc_k > 0 \end{array} \right\} \tag{9}$$

Here we assume the only available low power state is when it is powered off since most existing hardware requires manual effort in order to turn off only a fraction of the ports.

The final component to be modeled in our framework is the migration of jobs. Since the GPM is designed to be a scheduler, it must be sensitive to costs of and constraints on the migration of jobs. There are three primary overheads resulting from the migration of a service/application/VM instance from one physical server to another: a) additional CPU resources, b) Memory requirements, and c) Bandwidth requirements to perform the transfer. All three are in addition to the real requirements of the migrated instance itself. While many studies deal with the issue of Virtual Machine overhead their focus has been on the effect of the migration on the network bandwidth and memory utilization. Hence we handle the computation cost for the migration in terms of a percentage of the migrated VMs real CPU requirements. The reason for this is that the migration would require the source and destination to perform computations proportional to the size of the application and the CPU state prior to the migration.

This is also a convenient means of introducing a knob to control migration decisions of specific Virtual machines based on one of more of the following factors: a) a pricing model allowing the client who owns the instance to pay more to prevent migrations either directly or indirectly while asking for an SSA with lower disruptions and higher availability; b) statistical decisions based on the VMs prior behavior for example large fluctuation in resource requests may discourage the GPM from continually migrating it around; c) specific application types with lower tolerance to delays and disruptions may be exempt from forced migrations. The following is the computation overhead due to the migration of VM_j converted to an equivalent power consumption value .

$$Pow\_Mig_j = c * \frac{a_j * Uvm_j}{2} * \sum_{i=1}^{n} |X^1_{ij} - X^0_{ij}| * \frac{100}{cpuMax_i} \tag{10}$$

As mentioned earlier, we do not model memory requirements directly as a constraint. But given the assumption that each virtual machine has limited tolerance, there is a definite constraint that requires sufficient bandwidth to be available on the migration path between the source and destination servers. The size of the migration state is directly proportional to the memory footprint of the VM in the source which would be the dominant portion of the transfer for all non-data centric applications. Again this is information the GPM can store for future decisions involving the VM. We are in the process of formalizing this constraint. Some challenges are the limited amount of topological information that is present in the mapping N. There is no easy method for a scheduler to check this constraint. One possibility that seems promising is combining the GPM scheduler with a routing controller like that used for Openflow based switches that not only help in acquiring all the required information about network link states, but also enable the GPM to search among paths to find a suitable one.

3 GPM Scheduler

The GPM scheduler is designed so as to solve a minimizing object function given as input the present state of the data center as described by the variables and data structures presented in the previous section.

The objective of the GPM is to minimize total datacenter power and accordingly, based on our model, the following is the objective function:

Minimize:

$$\sum_{i=1}^{n} Pow\_S_i + weight * \sum_{k=1}^{l} Pow\_N_k + \sum_{j=1}^{m} Pow\_Mig_j \tag{11}$$

In the above equation, the parameter *weight* is used to factor in the savings in power resulting in the consolidation of cooling requirements based on the relationship between spatial locality and network locality in the particular data center architecture. The various constraints that have become evident in the discussions above are formalized as follows:

$$\sum_{i=1}^{n} X_{ij} = 1 \tag{12}$$

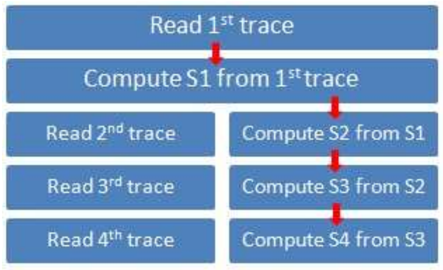


Figure 2: Simulation execution path

This represents the obvious constraint that each VM_j must exist on only one server.

$$\sum_{j=1}^{m} X_{ij} * Uvm_j * \left(1 + \frac{cpuRed_j}{100}\right) < cpuMax_i \tag{13}$$

This represents the fact that each Server i must not be over-subscribed having given due consideration to additional resources that each VM may have at its peak.

$$\sum_{j=1}^{m} X_{ij} * B_j * \left(1 + \frac{bwRed_j}{100}\right) < bwMax_i \tag{14}$$

This similarly represents the bandwidth caps of each Server.

While the above are very strict constraints since they prepare for all VMs to be at their peak utilization of respective resource at the same time, any other scheme requires more data. Ideally the GPM can provide statistical values of redundancy requirements for each VM and on top of that also predict what fraction of that cumulative redundancy will ever be needed simultaneously. This second component will be added in due course.

And finally, the constraint on the over-subscribed network components:

$$\sum_{i=1}^{n} N_{ik} * sum^{m}_{j=1} X_{ij} * B_j * \left(1 + \frac{bwRed_j}{100}\right) < UB_k \tag{15}$$

3.1 More Constraints

While the above equations are our contribution, many related works have very competent models that focus on complementary issues such as power caps that form a whole new set of constraints that can be seamlessly integrated with our model.

4 Simulation

We simulate the GPM scheduler using an existing data center trace . The simulator is written in Perl. It is designed to run a series of sequential scheduling problems each with a new set of data for the utilization variables obtained from the data trace. The simulation can be visualized as follows:

Each snapshot offers three sets of data: the identities of the VMs running in the system, the VM utilization values averaged for the duration between the previous snapshot and the present snapshot, and the VM to server mapping. The first step of the simulation is to use the first available snapshot in the trace to generate an optimized schedule. Each successive step of the simulation acquires 2 sets of data; namely the identities of the VMs still running and the VM utilization values from the corresponding snapshot from the trace. The third set of data required for the scheduling decision -the present VM to server mapping- is obtained from the previous simulated schedule. The new schedule that is obtained is compared with respect to estimated power utilization with the corresponding snapshot. Thus, the simulation rightly utilizes real server traces to produce a scheduling of the servers for a hypothetical data center during a contiguous period of time that the trace is available.

The next step is to measure its simulated performance against the real scheduler in terms of total power consumption. These values are calculated according to the model introduced in previous sections for different values of each parameter and plotted. More about the data used in our simulation is provided in the following section along with their results.

5 Resutls

The data used in our simulation was received from our university data center. Since we were interested primarily in virtualized components, we were constrained to work with just 12 servers running VMware ESX hypervisor. During the 1 week duration that our trace was collected, a total of 239 unique virtual machine instances were observed. The data trace consists of the average CPU, Network, and Memory utilizations of each of these virtual machines during the interval between the previous samples reporting and the present samples reporting. This sampling interval is 30 minutes. In this preliminary analysis, we report the effects of the various parameters in our model along with their performance with respect to the real schedules that the virtual machines were made to run on. The results show cautious optimism with some parameter configurations providing consistent reductions in power consumption of over 40%. Figure 3 shows one such simulation where some of the parameters include a migration overhead of 0.5 and a capacity redundancy requirement of 50%. But the primary goal of this effort was to implement a working simulator that could be fed much larger traces when available. While we must warn that any results presented here cannot be used to validate the performance of the model on even medium scale data centers, the fact that the model behaves as expected is encouragement to perform these tests on larger data traces.

As seen above, our simulated schedules usually have a very large number of migrations. This is because, given that a large number of the virtual machines have low CPU utilization values, their migration overhead as calculated in our model is low and hence the solver is free to migrate as many virtual machines as needed to try and eliminate a single server. The migration overhead fraction was kept at 0.2 in this run. But the situation did not change even when

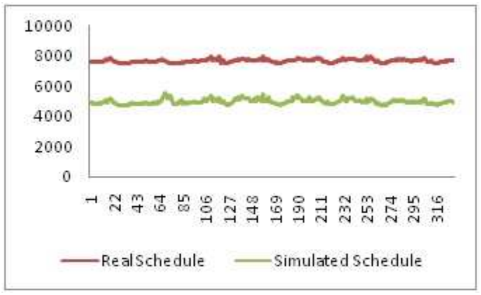


Figure 3: Estimated power consumption (in watts) for each of the 334 traces

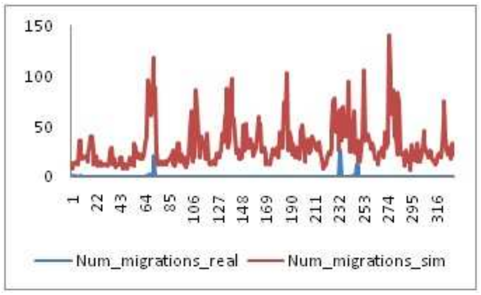


Figure 4: Number of migrations between successive schedules

the fraction was increased to 1 since out of the 239 virtual machines, over a 100 of them have utilization values below a 100MHz (as compared to the availability of 8 cores running at 2.7 GHz on each server) thus making their overhead insignificant to the overall objective function.

Except redundancy constraint parameters, none of the other tunable parameters had adverse impact on the simulated schedules the primary problem being the very low utilization values of these virtualized servers.

6 Related Work

Considerable research has been done in this area. Some researchers focus on understanding where in the data center the power is consumed. They then try to find sections in the data center that have a considerable potential for power saving [2, 3]. Servers consume more than 50% of the total power in a data center [2]. Therefore, most of the research in this area has been focused on reducing the server power consumption. However, as the server consolidation techniques mature, the marginal benefit that consolidation will give us will be minimal and thus we should focus on the other sections of the data center for power savings as well. For example, in these works, they ignore power consumption of the networking equipments which is about 5% of the total data center power consumption. In this work, we try to desing a model that can easily take into account different energy consuming components in the data center.

Other efforts focus on increasing the efficiency in individual components of the data center, such as servers or network equipment, for reducing the overall power consumption [4, 5]. [4] achieves energy saving in servers by putting servers in Nap while they are idle. While idle, servers consume about 60% of their peak power consumption. Therefore, this paper proposes the Nap state in which servers consume much less energy, but are also able to wake up quickly and respond to the queries in a way that does not adversely affect the latency. [5] proposes a similar approach for network equipment. Even though current network equipment do not have power states, this paper shows that we can achieve considerable power saving by having such a feature. Moreover, we can achieve energy saving by adapting the rate of operation in a switch to the workload of that switch. While reducing power consumption on individual components can bring us considerable energy savings, these efforts do not consider different power saving mechanisms that exist in a data center and how they interact with each other.

To mitigate this shortcoming, another group of related works attempt to consider the different power-consuming elements in their decision making process for workload consolidation [6]. [6] attempts to coordinate the state of the art power saving approaches, from increasing efficiency in firmware to workload consolidation, in order to achieve better savings. Lack of coordination between different elements can lead to inefficiency and/or system instability. For example, consider the case that the Virtual Machine Controller (VMC), which is in charge of workload consolidation, is not coordinated with group cappers which define the max power capacity in a group of servers. In this case, VMC might consolidate more workload in a collection of servers than allowed by the group power budget, which will cause system instability. In order to allow for a coordinated approach, this paper uses a feedback loop in which different approaches/components provide feedback to each other. Our model is very similar to this model. However, in [6] the focus is less on completeness and more on coordinating possibly conflicting solutions to improving power efficiency. Their model differs from our model in that their focus is on constraining the objective function - that only considers CPU power to not exceed bound set by other data center power-management components such as power cappers for racks and CPU frequency modulators. In that sense, their work is complementary to ours as we would benefit from incorporating the additional constraints that they have proposed in order to improve the correctness of our model.

7 Discussion

An important consideration for real time scheduler would be its efficiency in running as that would be reflected in its responsiveness to state changes in the data center. This is a concern about GPM that has prompted us to think about ways in which the optimization problem can be broken down into an iterative solution with each step having only a fraction of the number of variables as the entire problem would have. One obvious way of doing this would be to produce iteratively new states in which the scheduling is performed in decreasing granularity but with increasing scope. For example, the first iteration would look at aggregations of server loads within individual racks. The next iteration would attempt aggregation of rack workloads within layer 2 sub-tree and so on. But any such scheme will be difficult to test and validate since their primary assumption is that the datacenter is massive and it would be difficult to obtain workload and scheduling traces from large enterprise datacenters.

8 Conclusion

In conclusion, we have succeeded in creating a model that reflects in depth the major contributors to a data centers power consumption and have demonstrated the power savings it can produce for even a small set of servers as evidenced in our results above. Another priority for us was the implementation of the simulator and a running of the solving of the objective function for real data traces to test the models correctness that we have been able to demonstrate. All simulation and project files and references can be found at the project wiki: http://globalpowermanager.pbworks.com

References

[1] U.S. environmental Protection Agency. Report to congress on server and data center energy efficiency. Technical report.*ENERGY STAR Program* (2007).

[2] S. Pelley, D. Meisner, T. Wenisch, and J. VanGilder. Understanding and abstracting total data center power. In *Proceedings of the 2009 Workshop on Energy Efficient Design (WEED)* (2009).

[3] J. Karidis, J. Moreira, and J. Moreno. True Value: Assessing and optimizing the cost of computing at the data center level. *Conference on Computing Frontiers* (2009).

[4] D. Meisner, B. T. Gold, and T. F. Wenisch. PowerNap: Eliminating server idle power. In *Proceeding of the 14th international conference on architectural support for programming languages and operating systems* (2009).

[5] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI'08* (2008).

[6] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No Power Struggles: Coordinated Multi-level Power Management for the Data Center. In *Proceedings of ASPLOS* (2008).

[7] C. Gunaratne, K. Christensen and B. Nordman. Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed, *International Journal of Network Management* (2005).

[8] B. Nordman and E. Davies. Energy Engineering for Protocolsand Networks, *IETF70 Vancouver Technical Plenary* (2007).