NSF GRANT # 0946935
NSF PROGRAM NAME: CMMI

Achieving Energy-Efficiency in Data-Center Industry: A Proactive-Reactive Resource Management Framework

Natarajan Gautam
Texas A&M University

Lewis Ntaimo
Texas A&M University

Abstract: Data centers consume a phenomenal amount of energy and emit a staggering amount of greenhouse gases. There exist technologies such as virtualization and dynamic voltage scaling which, if appropriately used, have tremendous potential to reduce energy consumption. However, approaches to use these resource management technologies are either highly *proactive* running optimization based on forecasts of information load, or highly *reactive* using traditional feedback control theory. Also these approaches typically consider only a single technology with little or no interaction with others. The main reason for that is the various approaches are considered at different timeframes such as daily, hourly and real-time, for example. To redress these shortcomings, we propose a unified framework that is not only aware of the decisions at all timeframes but leverages upon the best practices of both proactive and reactive mechanisms.

1. Introduction: Growing demand for data services, the availability of high-volume Internet gateways, and the relatively modest facilities requirements for server banks have led to an explosive growth in the data center industry. Practically every single organization, especially if they have a web page, whether a private company or a public undertaking, uses the services of a data center (in-house or outsourced, usually the latter) to acquire, analyze, process, store, retrieve, and disseminate data. However, data centers consume a phenomenal amount of energy and emit a staggering quantity of greenhouse gases.

Industry-wide data centers currently spend over $5 Billion annually on electricity and several new power-plants have to be built to sustain their growth, according to EPA (the United States Environmental Protection Agency). The combined greenhouse gas emissions from all data centers exceed what entire countries like Argentina or the Netherlands emit. Therefore, reducing energy consumption and thereby greenhouse gas emissions in data centers is of paramount importance from an environmental standpoint.

In addition to the environmental impact, reducing energy consumption would also result in serious economical gains. For every watt of power used by IT equipment in data centers today, another watt or more is typically expended to remove waste heat. In fact, energy costs are higher than the cost to lease the space for data centers.

Some data centers have started to address the problem of excessive heating in their units by using more efficient heat exchangers, utilizing the heat generated by data centers for other purposes, designing layouts of data centers efficiently, etc. Although these strategies are extremely important in building *green* data centers, they do not address the fundamental inefficiency issue. Most of the energy consumed by data centers is not for useful work. The utilization of an average server is about 6%, but it also generates heat while being idle.

One of the reasons for this low server-utilization is that most data centers are built up of cheap servers running a single application. A few data centers have began to address these concerns using naïve solutions such as removing dead servers, enabling power-save features, and powering off servers when not in use. But, careful design and control with accurate load forecasting and capacity planning under uncertain loads would push the envelope even further. For this we leverage upon technologies such as *virtualization* (enabling more than one application to run on one server) and Dynamic Voltage Scaling (or *DVS* which allows servers to run at different frequencies by adjusting the voltage).

It is only recently that energy management for server clusters found in data centers has gained much attention (see [1], [2] and [3]). A categorization of closely related investigations is summarized in [4] in terms of whether (i) the schemes consider just server shutdowns or allow DVS, (ii) they focus on energy management of just one server (which can be thought of as completely independent management of a server without regard to the overall workload across the system) or *S* servers,

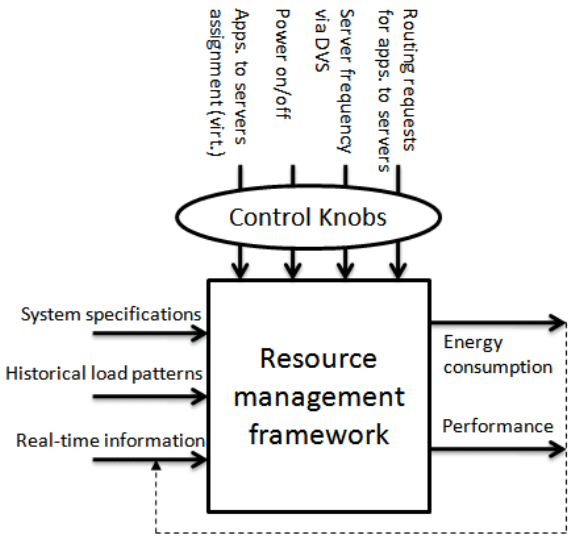
(iii) they consider different applications to be concurrently executing across these servers with different loads, (iv) they try to meet quality of service requirements imposed by these applications, and, (v) they incorporate servers availability and reliability.

Our objective in this research is to combine virtualization, DVS, powering on/off servers as well as routing of requests to servers in a single framework with the understanding that they are performed at different time domains or granularities. Further, they are also different in terms of what information is used in the decisions, some are history-dependent (such as virtualization [3] and powering on/off [5]) and others are real-time feedback based (such as DVS [6] and routing [7]).

We propose a unified framework that is both aware of the decisions at all granularities and combines proactive as well as reactive mechanisms. We describe the corresponding problem statements in Section 2. Then in Section 3 we describe several pieces that are involved in solving this large-scale problem. Section 4 describes some overall results followed by concluding remarks in Section 5.

2. Problem Description: Consider a data center with S servers and A applications. Assume that using virtualization and intelligent routing it is possible to host more than one application on each server as well as each application running on more than one server. At the *strategic* level, the problem is how to assign applications to servers. This is usually a one-time decision that lasts for a few days or weeks.

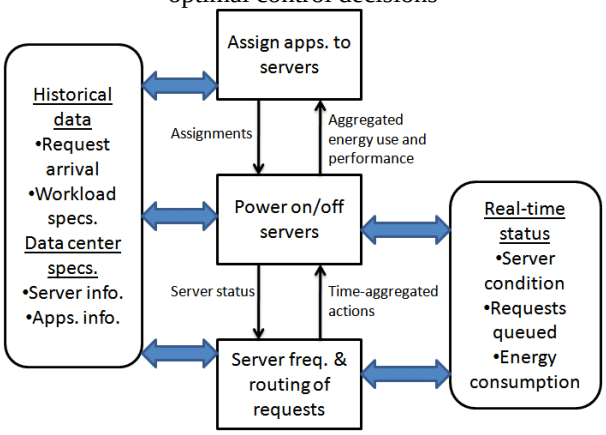
Figure 1: The unifying framework for proactive-reactive control



At the *tactical* level, the decision is whether a server must be powered on or off for a stretch of time, usually about an hour or so. The benefit of powering off a server is that it does not consume any energy or transmit any heat. The strategic assignment of applications to servers must be done right to enable this because if a server is off, then those applications must be running at one or more servers that are on so that requests for those applications are not turned away. The idea is to power down servers when the load is low (i.e. lean periods of traffic). Right away one can see the benefits of performing assignments being aware of powering on-off.

Further, at an *operational* level one can decide the frequency to run the servers thereby controlling the processing speeds. Ultimately server frequencies directly affect energy consumption. Frequencies can be modified using DVS instantaneously, so these decisions can be made in real time. Another decision that is made in real time is routing in terms of where requests, upon arrival, need to be assigned. Each request has to be assigned to one of the servers on which it is running. Clearly, decisions made regarding optimal frequencies and routing depend on the tactical decisions as well as strategic decisions and vice versa.

Figure 2: Information flow in the framework for optimal control decisions



For the decision-making process we assume that we know the mapping of energy consumption as well as processing speeds to frequencies for each server. We also assume that there is historical information regarding the request arrival pattern and processor requirements (i.e. workload) for every application. However, we need to build in some robustness to tackle the inherent uncertainty in arrival and workload characteristics for the future. The framework is described in Figure 1 and the connection of various components in Figure 2. With this in mind we next

describe a summary of the various research components considered in this research project.

3. Overview of Research Components: Our research problem is the control of a large-scale highly-stochastic system with decisions made at multiple time granularities using historic as well as real-time information. Our approach is to solve multiple sub-problems that form various research components. The key is that these sub-problems are both decision as well as information aware of the other sub-problems. We observe that this keeps the main problem tractable. More importantly, the solutions obtained by exchanging decision and information is significantly different (and more energy-efficient) than those available in the literature which are largely independent decompositions. With that motivation, we present a summary of the various sub-problems that would appear as separate research articles in appropriate journals.

3.1 Hydrostatic model of overall system: Using the expected values for all the random quantities we formulate a deterministic optimization problem that minimizes the total energy consumed by the data center system. The formulation considers all four elements that we control, namely assignment of applications to servers (via virtualization), determining whether a server must be powered on or off, assigning optimal frequencies to run the servers (via DVS) and allocating requests to servers (i.e. routing). Due to the hydrostatic or fluid nature of the model, we use average values for the forecasted demand and workload in each time period. However, by striving for a lower utilization we ensure that the quality of service is satisfactory. It is crucial to point out that the real-time decisions of frequency and routing are made at an aggregate level for a finite period of time with the understanding that while being implemented in real time, these would be time-averaged values.

3.2 Stochastic-programming methods to handle uncertainties in hydrostatic model: In the hydrostatic model we used average values of the forecasted demand to make our decisions. However, web-based systems are highly non-stationary or have very large variability. Thus the average values forecasted could be significantly different from average values realized. Thus we formulate a stochastic optimization program that models estimated mean arrival rates and estimated mean workload as random quantities so that the constraint on quality of service is a chance-constraint. This results in a more robust decision, especially at the strategic and tactical levels. Furthermore, the decisions implicitly also take into account second-order effects, not just the mean values.

3.3 Graph-theoretic approach for strategic applications-to-servers assignment: Using the insights developed in the hydrostatic models we consider the strategic problem of deciding which applications should be virtualized on which servers in a data center. We begin by creating a bipartite graph with the *A* applications and *S* servers. An arc between an application and a server implies that that application is virtualized on that server. Our initial choice of this graph is done by summarizing arc costs and node demands based on the insights and parameters of the hydrostatic models. Subsequently we use several techniques including branch-and-bound and genetic algorithm to improve upon the initial solution by being fully aware of the tactical and operational decisions.

3.4 Distributed frequency and routing control using population games: Given the strategic assignment of applications-to-servers, in a given time period we first determine the minimal set of servers that should be powered on (this is the tactical decision). For real-time control of the frequency as well as routing we consider a game between the servers the router that gets incoming requests. The servers determine their frequencies in a distributed fashion using a pricing mechanism dictated by the router with constraints on quality of service. In return, based on the frequencies adopted the router determines prices as well as makes a decision regarding the fraction of requests for a particular application to go to a particular server. We show that the system converges to a Nash-equilibrium quickly and hence can be effectively used in a distributed algorithm. Understandably, this is still a fluid or hydrostatic model with deterministic parameters.

3.5 Optimal control policies at the tactical and operational levels: Here we consider a stochastic arrival process as well as stochastic workload requirement. We first model a centralized queueing scenario so that all servers run at the same frequency (unless they are powered off) and whenever a server completes a job, it picks up the first job waiting in line. Further, there is a cost for powering on and off a server. We show that the optimal policy for such a system is hysteretic in terms of powering servers on and off, and, threshold-type for determining frequencies. Next we extend this model by using a centralized controller to determine routing (with a queue at each server) as well as powering servers on and off. However, every server that is powered on determines its own frequencies. We show that the optimal policy at each server is a load-dependent threshold policy.

3.6 Stochastic fluid-flow model to determine optimal thresholds for real-time control: Although we know that the optimal policy at each server is

load-dependent threshold policy, we still need to determine what the optimal threshold values are. For this we develop a stochastic fluid-flow model where sources randomly input traffic into a fluid queue based on the loads experienced. We develop differential equations for the buffer contents and use appropriate boundary conditions to determine the first passage time between thresholds. Then we formulate and solve a non-linear program for the optimal thresholds so that expected energy consumption is minimized subject to satisfying quality of service constraints. For this we need stationary probabilities for using various frequencies which can be obtained using a semi-Markov process analysis that leverages upon the first passage times.

4. Synopsis of Results: Detailed analysis and results based on the six research components 3.1 to 3.6 will appear in various manuscripts. The reader is encouraged to view the PI's website as that is where the manuscripts would appear when they are ready. Nevertheless, to give a flavor of the results we present some key findings here. The most fundamental question that we seek to answer is if it is necessary at all to consider a unified framework. Our results show that there is a significant difference in terms of optimal decisions between considering the pieces independently without any information exchange versus when there is information exchange. Not to mention, the overall energy savings and improvement in performance that is attained as a result of the unified framework with all the appropriate information exchanged.

In the literature while considering methods to make assignments for virtualization, or deciding whether or not to power down servers or real-time control actions such as DVS and routing, assumptions are made regarding various characteristics. Although these assumptions are intuitively appealing, our results indicate that they are not necessarily true. For example, one typically assumes that correlated applications would be virtualized on the same server. But our results indicate no such patterns. All these aspects along with the need to provide high performance result in a fairly non-intuitive set of decisions. This clearly makes a strong case for the need to study these systems in a common framework.

One of the key methodological contributions is the control architecture that combines proactive and reactive components. In particular, the benefit of proactive decisions is to achieve long-term goals without over-reacting to short-term fluctuations. But the downside is that when systems are non-stationary or have poor predictability, significant improvements can be made to the decisions. On the other hand, reactive techniques tend to perform better under these circumstances but are too short-sighted to achieve excellent aggregate results. To illustrate how our techniques that leverage upon both proactive and reactive techniques, consider Table 1 which we explain next.

We consider a data center that has already adopted some easy fixes such as pruning down the number of servers and adopting virtualization based on load sharing. This is the state of the practice. We illustrate what can be realized by simply performing power on/off and DVS (with a naïve equal split routing). As shown in Table 1, we consider web-server workload traces publicly available and perform power on/off and DVS using three strategies: pure forecasting (proactive); pure feedback (reactive); and our method. Notice first of all how all the three methods can realize significant energy savings compared to the state of the practice. Next note that our method trades off between performance and energy savings.

Table 1: Comparing our method against proactive and reactive methods; "%saved" implies percentage of energy saved and "QoS met" implies satisfying quality of service needs.

Trace	**Method**	**% saved**	**QoS met?**
ClarkNet-HTTP	Our method	37	Yes
	Pure forecast	25	Yes
	Pure feedback	17	No
GWA-T-2Grid5000	Our method	26	Yes
	Pure forecast	28	No
	Pure feedback	34	No
GWA-T-10SHARCNET	Our method	50	Yes
	Pure forecast	31	Yes
	Pure feedback	31	Yes

5. Concluding remarks: Our ongoing research to reduce energy consumption to the maximum extent possible has yielded positive results both for data center industry as an application domain as well as in terms of a science for multi-level and multi-dimensional control. In particular, for the latter contribution we have developed a methodology that shows that by appropriately passing information across levels, and by considering the benefits of long-term and short-term advantages, it is possible to attain huge savings at the same time not compromise on performance. In effect we have shown that it is possible to squeeze the most from a system by effectively planning, managing uncertainty and undertaking a holistic approach. We

believe that these results can be effectively used in reducing energy consumption while maintaining excellent performance in many other systems besides data centers.

6. Acknowledgements: We would like to acknowledge NSF for supporting this research study through the grant CMMI-0946935. We also thank our students Julian Gallego, Young Myoung Ko, Cesar Rincon Mateus and Ronny Polansky for all the hard work. We appreciate working with our collaborators Dr. Eduardo Perez and Dr. Srinivas Shakkottai.

7. References:

[1] M. Elnozahy, M. Kistler, and R. Rajamony, "Energy Conservation Policies for Web Servers," Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems, March 2003.

[2] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Kelle, "Energy Management for Commercial Servers," IEEE Computer, vol. 36, np. 12, pp. 39–48, 2003.

[3] A. Chandra, P. Goyal, and P. Shenoy, "Quantifying the Benefits of Resource Multiplexing in On-Demand Data Centers," Proceedings of First ACM Workshop on Algorithms and Architectures for Self-Managing Systems, June 2003.

[4] Y. Chen, A. Das, W.B. Qin, A. Sivasubramaniam, Q. Wang and N. Gautam, "Managing Server Energy and Operational Costs in Hosting Centers," ACM SIGMETRICS Performance Evaluation Review, vol. 33, no. 1, 303-314, 2005.

[5] M. Elnozahy, M. Kistler, and R. Rajamony, "Energy-Efficient Server Clusters," Proceedings of the Second Workshop on Power Aware Computing Systems, February 2002.

[6] T. Abdelzaher, K. G. Shin, and N. Bhatti, "Performance guarantees for Web server end-systems: A control-theoretical approach," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 1, 2002.

[7] A. Wierman, L.H. Andrew and A. Tang, "Stochastic analysis of power-aware scheduling," Proceedings of Allerton 2008.