Datacenter Modeling and Simulation with Focus on Energy Efficiency and Green Energy Integration

Baris Aksanli, Jagannathan Venkatesh, Tajana Rosing
University of California, San Diego
Submitted to: IEEE Computer Special Issue on Modeling and Simulation of Smart and Green Computing Systems
E-mail: {baksanli, jvenkate, tajana}@ucsd.edu

***Abstract*— Improving energy efficiency of datacenters and studies of green energy integration into the datacenters are important research problems. Recent studies include design of power management mechanisms, scheduling policies, and efficient cooling methods. However, deploying a new policy in a datacenter and verifying its effects is time consuming and often infeasible. Therefore, it is important to effectively and accurately model and simulate datacenter environments. This paper provides a survey and comparison of recent simulation platforms for datacenters. Additionally, it presents a case study of how datacenter simulation enables design and evaluation of algorithms for using green energy to improve the datacenter energy efficiency.**

Key words: Datacenter, energy efficiency, green energy, simulation

I. INTRODUCTION

Recent improvements in computer and network architectures have made internet-based applications and cloud computing systems popular. Some companies, such as Google, Amazon, Facebook, Microsoft, have multiple datacenters with thousands to millions of servers geographically distributed. Servers comprise a large portion of the overall energy cost in IT [1], with networking infrastructure, such as switches and routers, being a relatively minor component. In addition, there are a number of non-IT elements that contribute to the high energy costs of the datacenters. Examples include power distribution units (PDU) which provide power to the IT elements, uninterruptible power supplies (UPS) used for emergency situations, and computer room air conditioners (CRAC) that keep the datacenter temperature within determined limits to ensure server reliability. A recent study shows that the total energy consumption of all datacenters in the world has increased by 56% from 2005 to 2010 [2]. Energy efficiency will continue to be important in the future as the price of brown energy, the energy produced by non-renewable resources, will rise as a result of additional taxes placed on carbon emissions [3]. Energy efficient solutions, ranging from utilizing green energy sources, such as solar and wind, to optimizing HW, SW and system design for energy efficiency, will continue to be important.

As it is hard to deploy a new policy or mechanism and observe its effects on a datacenter in a controlled and a repeatable manner, it is often desirable to model it and evaluate results using simulation. Simulating a datacenter saves time and effort spent on configuring a real environment for testing [4]. Additionally, as real machines are not used for testing purposes when simulating, their computational power can be allocated to applications that bring company profit. Simulation also enables evaluation of mechanisms that may not be implementable without a high investment, such as analyzing the cost and benefits of adding and managing electricity from photovoltaics.

The main goal of our paper is to provide a survey and comparison between the current simulation platforms for evaluating energy efficiency of data center environments and to illustrate how these simulators can be used. Each has its own design goals and uses different models to accomplish the goals. For instance, GreenCloud [5] is helpful for evaluation of different power saving methods, such as dynamic voltage and frequency scaling (DVFS) and dynamic power management (DPM). In comparison, the main goal of SQS [6] is to estimate the job response time and evaluate scheduling mechanisms, which can be helpful when migrating VMs for the purpose of clustering workloads to save energy.

In this work, in addition to the survey of existing simulation platforms, we also present a case study illustrating how a datacenter simulator such as GENSim can be used for design and evaluation of green energy integration into datacenters. We analyze different green energy integration methods, such as using all of green energy first to reduce the overall need for brown energy, vs. using a contracted amount of brown energy and leveraging green energy to run additional batch jobs. This comparison might be important for datacenters which normally buy a fixed amount of brown energy each month to ensure that they have enough of reasonably priced electricity to meet their nominal user demand as the peak energy costs can be dramatically higher. However, if there is a sufficient amount of green energy available, then the peak electricity demands might be completely and consistently offset by the green energy. Additionally, we demonstrate how green energy prediction can improve the overall efficiency of datacenters by provisioning the green energy availability accurately for longer time intervals. After the analysis, we compare all the methods in terms of our energy efficiency metrics. Our simulation results also show that the highest energy efficiency is obtained when green energy prediction is used for execution of additional batch jobs that can be started with the available green energy.

In the next sections we describe different datacenter simulation platforms, provide their comparison and finally illustrate how they can be used for a case study in green energy integration into datacenters.

II. DATACENTER SIMULATION PLATFORMS

Simulators used for evaluating datacenter energy efficiency have to provide accurate estimates of power consumption and performance in order to estimate the energy costs. There are many metrics that can help quantify energy efficiency. Quantifying the ratio of power consumed by IT elements as compared to the overall power supplied to the datacenter, Power Usage Efficiency (PUE) [1], is commonly used, but does not provide any information regarding datacenter performance. Measuring work done per energy consumed highlights that increasing performance within a given power budget can also be a way to improve the energy efficiency, in addition to decreasing the power costs. Hence, accurate estimate of the performance of a datacenter is as important as quantifying its energy consumption. Some important performance metrics include average response time of the requests, number of service level agreement (SLA) violations, and others. Evaluating the effects of power saving policies, such as DVFS, or job scheduling algorithms on power consumption, helps with quantifying the benefits to energy efficiency due to lowering of power costs. Lastly, new metrics related to green energy efficiency have been proposed in [7] .

Simulators also differ in the types of workloads they consider. Barosso and Holze [1] classify datacenter workloads into two categories: latency-centric service requests and throughput-oriented batch jobs. A service request requires response that is in the order of milliseconds. However, a batch job does not have tight response time constraints, but instead it needs to maintain a certain level of throughput. Most simulators to date consider only one or the other type of workload, while datacenters normally run a mix of the two.

In the next subsections we describe state-of-the-art simulators and show how they can be used to evaluate and improve datacenter energy efficiency.

A. Multi-tier Datacenter Simulator (MDCSim)

MDCSim evaluates datacenter energy efficiency by quantifying the power cost on per-server basis and providing a good estimate of throughput and response times of datacenter jobs [4]. This simulator uses a simple linear, utilization based power model for servers:

$$P = P_{idle} + CPU_{util} * (P_{peak} - P_{idle}) \quad (1)$$

However, MDCSim uses a much more detailed model to evaluate the total response time of service jobs. It models the multi-tiered nature of datacenter workloads. In a datacenter, a job might require services of different servers, each executing a specific process. A set of servers processing a specific type of request forms a tier. Each tier may have different service rates; hence requests might have different response times in each tier. MDCSim models communication between tiers, plus scheduling and resource allocation algorithms per each tier. It reports response times and throughput of service requests.

A tier is represented as a collection of servers. A single server may belong to only a single tier and has a single CPU, memory, a disk, and network connectivity. M/M/1 queue is used to describe server performance. However, this model, as we show in the next subsection, does not accurately represent realistic workloads. Authors do use a prototype datacenter to run some sample workloads to estimate parameters such as interarrival time and service time of requests for each tier. MDCSim reports simulation errors of less than 10% for both latency and throughput of each tier, and of 3% for server power consumption. As such, MDCSim can be used to evaluate the effects of different scheduling policies on power consumption as it has both performance and power models. However, it uses only service type workloads and it does not model the effects of using power saving policies such as DVFS.

B. Stochastic Queuing Simulator (SQS)

SQS [6] improves on MDCSim by providing more detailed estimates of datacenter service job response times. It models a datacenter as a pool of servers. Each server is modeled with a M/G/k queue, an improvement over MDCSim. The parameters of interarrival and service time distributions are estimated by analyzing traces of real applications, e.g. mail server of the university and web server [6]. Authors use Poisson splitting to separate the M/G/k queue into k separate M/G/1 queues for simplicity. They also argue that the servers in a datacenter generally operate in low utilization levels (~20%). Therefore, they assume that the average expected response time is equal to the average expected service time, i.e. there is no waiting in a queue.

Linear power model shown in Equation 1 is also used by SQS. However, in addition to this model, SQS also estimates the effects of DVFS on the overall power consumption by assuming that dynamic power is proportional to the cube of server frequency of operation.

SQS outputs average and a confidence interval for response times. It needs multiple runs with the same set of parameters to achieve the desired level of confidence over the outputs, leading to longer simulation times. Since the light utilization assumption may not always hold, SQS is not able capture the waiting time of the requests. Also, it only models service requests in its workload model. Therefore, it does not represent the effects of having different types of jobs in the datacenter. Finally, the default power model of SQS is not connected in any way to the performance model, which makes estimating energy efficiency more difficult.

C. Cloud Simulator (CloudSim)

The main goal of CloudSim [8] is to estimate per server and overall system performance and power metrics. In CloudSim datacenter model, the main abstraction unit is a virtual machine (VMs). Each VM serves a single request. VM-scheduler is responsible for allocating a server for each VM. If the specification of a VM includes a time constraint, the scheduler can move the VM to a different server or place VMs without tight time constraints into a wait queue to satisfy SLAs.

Servers are divided into disjoint clusters responsible for processing a specific type of request. CloudSim models the communication between different clusters and associates a static cost for each communication request using a latency matrix A where a_{ij} shows the delay that a message undergoes when transferring between entity i and j.

CloudSim has an extensible framework for power modeling. It supports different power consumption models as well as different power saving policies. The default model is a linear CPU utilization based function, such as the one shown in Equation 1, with no power saving policies. Any new power model or policy requires additional development.

The outputs of CloudSim can be categorized into two classes: cost and performance related. The former include power consumption cost and financial cost of renting computation power, whereas the latter include the number of SLA violations, number of VM migrations, and average response time of requests. This simulator can be used to test the impact of different power saving policies on both performance and cost. However, although it has a VM level model, CloudSim does not model conflicting requirements of different jobs running on the same server and the effects of these conflicts have on performance and energy costs.

D. Romonet – BCS Simulator

The main design goal of the BCS simulator is to analyze different datacenter layouts and estimate the cost of the system based on the chosen layout [9]. A datacenter layout consists of a set of elements to be used and the relative placement of these elements in the datacenter. Analysis is done using two modes: infrastructure and IT device analysis. Infrastructure analysis is used to estimate how energy efficient a system is at a very coarse grained level, assuming constant and static power costs. The energy efficiency metric used is *Datacenter Infrastructure Efficiency (DCIE)*:

$$DCIE = \frac{IT\ Equipment\ Power}{Total\ Facility\ Power} \quad (2)$$

The user defines device nodes to represent each class of elements e.g. servers, PDUs, etc. Different nodes are connected through flows where each flow is a part of the thermal or power chain. The infrastructure analysis mode outputs the total consumed energy, electrical loss and infrastructure maintenance cost over a specified time.

In IT device analysis mode, BCS-Sim adds the IT devices into the layout and simulates energy and cost of running them. The output of the simulation is a set of cost values reported for each device type. The IT device cost is further detailed into multiple categories such as hardware, maintenance, and energy cost. The simulator uses either the default workload traces it has or user can input a different trace to the simulation. It then applies load-to-power consumption transition per server and scales it to the overall datacenter to obtain the total power consumption. The basic time unit of the simulation is hours or days. If user wants to have results over a month, then single day results are scaled to a month.

The advantage of BCS-Sim is that it models both IT and non-IT elements in a datacenter, but at very coarse granularity. The simulator outputs energy efficiency and the power/capital cost of the devices in a system. However, it does not have a detailed enough server model to capture how performance changes with energy efficiency, which can be crucial for latency-bound requests.

E. Energy Efficiency Simulator (EEFSim)

The main goal of this simulator is to evaluate the power efficiency of a virtualized datacenter [10]. It is designed to evaluate different scheduling, consolidation and migration algorithms for VMs and show how these algorithms affect power efficiency. The simulator models a datacenter as a collection of identical servers, each capable of running multiple VMs. Applications are executed on physical machines with different VM properties to construct a power model as a function of utilization. These experiments are called calibration runs.

The platform is able to simulate HPC and transactional jobs generated by reproducing real traces. The jobs are represented by their CPU and memory requirements. However, power modeling uses only CPU as a parameter as shown in Equation 1. The results show that the average error for power consumption, compared to measurements, is 2.3% for 1300 seconds of simulation time when using a calibrated model. EEFSim uses VM level abstraction and models consolidation and migration of VMs, which are important techniques to improve energy efficiency. However, it is not able to estimate performance (e.g. response time). Also, it models only grid workloads. Simulation run time does not scale well with number of servers and becomes infeasible with more than 5000 servers.

F. GreenCloud

GreenCloud models a datacenter using a three layer architecture consisting of core, aggregate and access layers [5]. Servers are organized into racks with rack switches on top. The higher layer switches increase system availability by providing alternative paths and system capacity by increasing connectivity to host thousands of servers simultaneously. Advantage of GreenCloud is that it estimates networking infrastructure costs, including link and switch power consumption:

$$P_{switch} = P_{chassis} + n_{linecards} * P_{linecard} + \sum_{i=0}^{R} n_{ports,r} * P_r \quad (3)$$

where $P_{chassis}$ and $P_{linecard}$ are power consumed by switch hardware and active line card respectively, $n_{ports,r}$ is the number of ports at transmission rate *r,* and P_r is the power consumed by a port at rate *r.* This equation enables GreenCloud to use different power saving mechanisms for network elements, such as adjusting the transmission rate of a link or putting some idle line cards into sleep. Server power model is based on CPU frequency, thus enabling a study of DVFS policies:

$$P = P_{fixed} + P_f * f^3 \quad (4)$$

where P_{fixed} is the power that is independent of frequency f, and P_f is a frequency dependent power constant, to be found experimentally. Each server is assumed to have only a single processor.

Job creation is modeled with a Markovian process. A job leaves the system when both computing and communicational components complete. The communication component depends highly on the job size, which is also modeled with a Markovian process.

GreenCloud reports the total datacenter power consumption, as well as the power consumption of different IT device classes, such as servers and switches. It also analyzes some power saving schemes. However, the analysis does not show how these power saving mechanisms affect job performance. Additionally, it represents servers with a single processing core. Therefore, the user can not apply any scheduling policies within servers, nor model the effects of co-locating multiple different jobs within a single server.

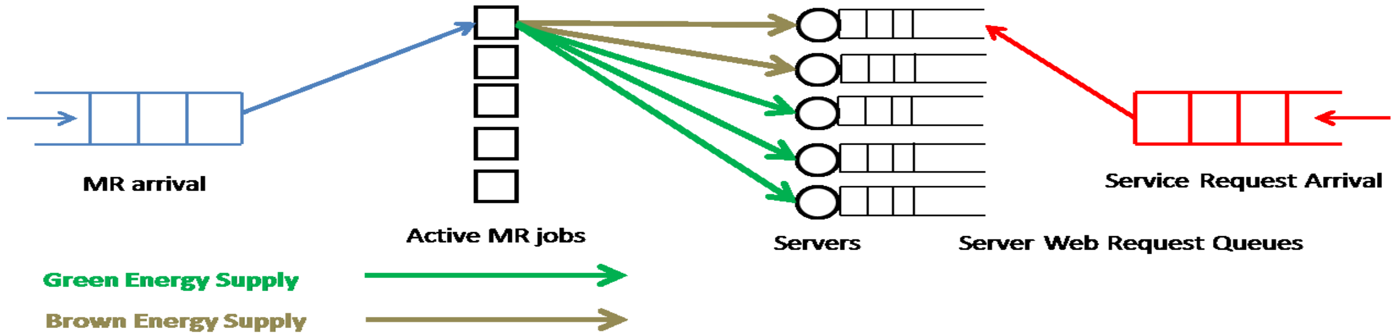


Figure 1: System model when running extra jobs with green energy

G. Green Datacenter Simulator (GDCSim)

GDCSim is able to study the energy efficiency and thermal properties of datacenters as a function of floorplan, power management, and scheduling policies [11]. Each server is represented with a process queue associated with each of its cores. The queuing model and scheduling algorithm are user specified. GDCSim supports two types of workloads: high performance computing (HPC) and Internet-transactional workloads (IDC). GDCSim specifies the format for the power model and requires the user to select and implement the model. It supports power management algorithms, including C-state management that controls sleep state transition of CPU, and P-state management, i.e. DVFS.

GDCSim's main contribution is thermal modeling. It uses BlueSim package with XML based layout specification and performs computational fluid dynamics (CFD) simulations to obtain a thermal profile of the given layout. The main unit in the thermal module is a chassis, and a layout consists of the relative placements of chassis and cooling units (CRACs). The output is the inlet and outlet temperatures of each chassis, which form the thermal map. The thermal model is important in terms of energy efficiency as it enables the control of cooling units, which can result in significant energy savings.

Despite the advantage of its thermal model, it implements only simple schedulers, such as Round-Robin and first-fit. It does not analyze the interference effects of different types of jobs. Finally, its thermal simulation is very slow as it relies on CFD, and as a result can be done only for smaller, steady-state test cases.

H. Green Energy Simulator - GENSim

GENSim [7] is a simulation platform developed to quantify how green energy integration (wind and solar) along with co-location of services and batch jobs in single servers can increase the overall energy efficiency. Examples of green energy integration methods include using all of green energy first with minimum possible brown energy, thus reducing the brown energy consumption; and using a contracted amount of brown energy to meet the nominal energy needs and then improving the overall energy efficiency by running extra workloads with green energy, thus increasing performance at no additional brown energy cost. Green energy prediction, instead of using only instantaneous availability, is shown to be more efficient when launching additional jobs with green energy.

GENSim is the only datacenter simulator capable of estimating the impact of co-locating both service and batch jobs on a single server. It uses RUBiS as a representative of service request workloads, and MapReduce to represent batch jobs [7]. Each MapReduce job consists of multiple tasks, running in parallel. A server has multiple slots reserved for MapReduce jobs/tasks.

Server performance and power consumption are estimated based on usage of CPU and memory resources. CPU utilization based power model shown in Equation 1 is used for servers. Total datacenter power is estimated by using aggregate server power scaled by the power usage efficiency ratio (PUE) as it accounts for the non-IT power consumption, e.g. power losses, cooling power, etc. [1].

Two levels of schedulers have been implemented: global and local. Global scheduler is responsible for selecting a server for an incoming job. When choosing a server, it aims to minimize the interference between jobs. If an incoming job is a service request, the global scheduler immediately assigns a server to it and places it to the local queue of the server (Figure 1). If it is a batch job, such as MapReduce, the global scheduler dispatches a number of tasks from the job and allocates a slot per task from several servers (Figure 1). The number of tasks dispatched depends on the throughput level required by the MapReduce job and the number of available slots in the system.

Local scheduler performs the scheduling decisions within a server. The main goal is to minimize response time violations of services. When a new service request arrives, it first estimates the response time based on the number of active tasks on the server. If the estimated response times are greater than the threshold, the local scheduler halts batch tasks one at a time until the estimated response times are met [7]. The simulation parameters are obtained via experiments on Intel Nehalem servers. A mixture of RUBiS and MapReduce workloads runs within Xen VMs to quantify the performance impact of allocating increasing number of cores to batch jobs. The system ensures that throughput hit to MapReduce is under 10% while keeping 90th%tile of service response times to less than 100ms [7]. Quality of Service (QoS) ratio for service requests is computed using measured 90th%tile response time over the target 90th%tile response time. Simulation results have an average error of 3%, 6% and 8% relative to measurements of average power consumption, RUBiS QoS ratio and average MapReduce job completion time respectively.

	Attributes							
Platforms	*Resource Models*	*Workload Model*	*VM Support*	*Queuing Model*	*Power Model*	*Multi-tier*	*Open Source*	*Outputs*
BCS - Romonet	IT and non-IT elements	User supplied traces	X	N/A	User supplied power data	X	X	Power and cost of IT elements, energy efficiency of non-IT elements
EEFSim	CPU and memory	Real traces	✓	N/A	CPU-based linear	X	X	Power consumption
GreenCloud	CPU and switches	Synthetic workloads	X	N/A	CPU frequency based	X	✓	Power consumption of servers and network elements
SQS	CPU	Synthetic workloads	X	M/G/1	CPU-based linear	X	X	Average response time, power estimates
CloudSim	CPU, memory, disk	Synthetic workloads	✓	N/A	CPU-based linear	✓	✓	Power consumption, #SLA violations, #job migrations, average response time
MDCSim	CPU, disk, memory, network	Synthetic workloads	X	M/M/1	Utilization based linear	✓	X	Average response time, power consumption
GDCSim	CPU	User supplied	X	User defined	User defined, CPU-based	✓	X	Power consumption, average response time, utilization, thermal map
GENSim	CPU, memory	Synthetic workloads (based on measurements)	✓	M/M/k	CPU-based linear	X	X	Power consumption, quality of service, batch job completion time, green energy efficiency

Table 1: Summary of the simulation platforms introduced in Section II

III. Comparison between simulators

Table 1 summarizes the main attributes of simulation platforms described in the previous section and Table 2 lists their advantages/disadvantages. While all simulators can provide estimates of datacenter power consumption, they differ in how they model performance. The simulators that estimate only power consumption can easily draw misleading conclusions. For example, applying DVFS during peak load can decrease the power consumption; however it may also decrease performance to unacceptable levels, leading to costly SLA violations.

These simulators can test well-known methods to increase energy efficiency in a datacenter. For example, EEFSim and CloudSim model VM consolidation of same types of jobs. Energy savings are obtained by turning off unneeded servers after consolidation to save energy and thus reduce operating costs. However, they do not model the interaction between VMs running different types of jobs, and do not consider that powered off servers are not bringing in additional income while their capital cost has already been spent. GreenCloud and GDCSim can estimate the benefits of using DVFS. However, the former does not link power models with performance and the latter includes only an interface for power models, leaving the implementation details to the user. GreenCloud models DPM policies for network elements, such as turning off idle line cards. BCS-Sim and GDCSim can estimate changes in power consumption as a result of different datacenter layouts. The multi-tiered nature of MDCSim enables different scheduling algorithms to be tested. GENSim models different energy sources and implements scheduling policies based on energy type.

Most of the platforms model only single type of job, i.e. service requests, and thus neglect to capture the interaction between different jobs running on a single machine. Only GENSim studies the performance effects of different classes of jobs running on the same machine. It is also the only simulator which enables evaluation of green energy efficiency metrics. GDCSim and EEFSim model high performance and transactional workloads, where GreenCloud classifies the datacenter workload into computation-based and I/O-based workloads.

Only one simulator has a detailed thermal model: GDCSim. Most others either completely neglect the overhead of cooling, or use PUE to estimate it (e.g. GENSim and BCS-Sim). Since today's state-of-the-art datacenters run at PUEs of less than 1.2 [1], modeling detailed thermal interactions may not be as critical as it leads to long simulation times.

Lastly, only a few simulators, CloudSim and GreenCloud, are open source. However, the former does not let users define new workload models and the latter does not provide any performance related metrics, which is an impediment to evaluating the effectiveness of energy management policies.

IV. Case Study: Green Energy for Datacenters

In this study, the aim is to evaluate how green energy can best be leveraged within a datacenter. The variable nature of green energy is a big issue for datacenters as they require consistent energy supply to ensure workload QoS metrics are met. For this study, we assume that a datacenter has free local access to solar and wind power. We compare two different green energy integration methods. In Case1, green energy is combined with brown energy to process both service & batch jobs, thus reducing the overall amount of brown energy used, but keeping performance the same. In Case2, the baseline workload, consisting of service (e.g. Rubis) and batch jobs (e.g. MapReduce), is executed using a contracted amount of brown energy as is typical for most large scale datacenters such as Google. Whenever there is available green energy, extra batch jobs are scheduled, as shown in Figure 1. When green energy levels decline, the extra started tasks are terminated to keep brown power costs within contracted amount thus eliminating the need for expensive peak power. Alternatively, all the started batch jobs could have been finished even when there is not enough green energy, but at

	Advantages	Disadvantages
BCS	- Combination of IT and Non-IT elements - Detailed infrastructure model - Simulation over long time	- Does not have a detailed server model - Does not capture workload behavior - Does not have response time/throughput estimation
EEFSim	- VM level abstraction - Small power consumption error - VM migration model	- Does not model service requests - No performance output - Does not scale up with # servers
GreenCloud	- Network elements in a datacenter - Detailed power consumption (servers and network elements) - Focus on power saving policies - Different type of workloads	- Does not report performance statistics (response time, throughput) - A server with single core (in server scheduling is ignored)
SQS	- Statistical analysis to simplify the model - Fast simulation - Confidence level	- No power consumption output - Models only service type workloads - May not capture waiting time (uses light utilization assumption)
CloudSim	- VM level abstraction - Multi-tier model - Communication between tiers - Financial cost analysis	- Does not model different types of workloads and interference effects - Only service request constraints
MDCSim	- Multi-tier model - Communication between tiers - <10% simulation error	- Models only service requests - No throughput monitoring - Power model is fixed
GDCSim	- Thermal model - Connection between physical infrastructure and computing resources - Different type of workloads - Cooling aware policies can be used	- Does not have a model for the effects of different type of workloads on each other - Only response time is reported for performance statistics
GENSim	- Interference effects are modeled - Detailed workload model (service vs. batch jobs) - Scheduling in a server and among different servers - Green energy integration	- Does not have a thermal model - No support for power saving mechanisms (currently) - Does not have multi-tier model

Table 2: Advantages vs. Disadvantages of the simulators presented in Section II

the higher cost in terms of brown energy. We quantify benefits in terms of batch job performance (inversely proportional with job completion time) per brown energy consumed as follows:

$$En.Eff. = \frac{1}{Batch\ Job\ Completion\ Time * Brown\ Energy\ Consumed} \quad (5)$$

We also define a green energy efficiency metric to quantify how much green energy went to performing "useful" work relative to the total green energy available:

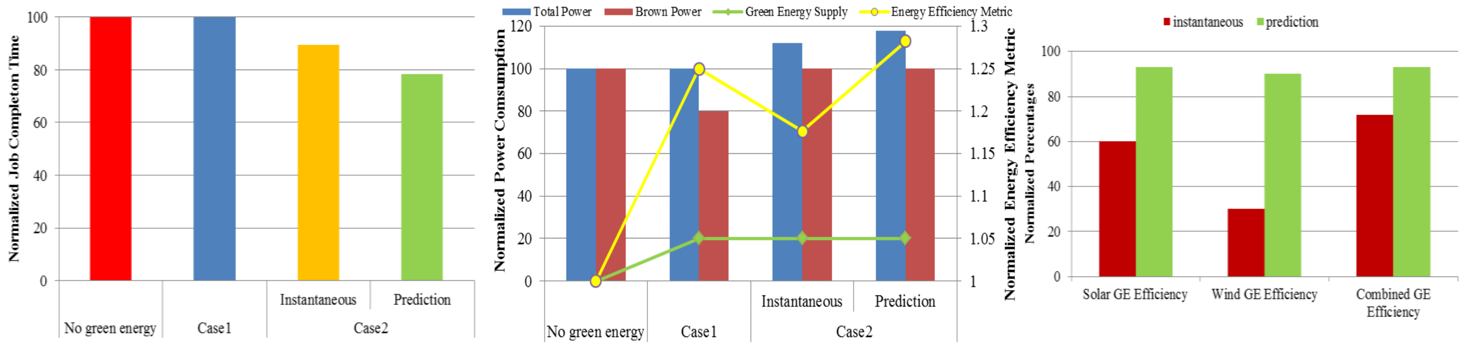
$$GE\ Efficiency = \frac{Green\ Energy\ doing\ useful\ work}{Total\ Green\ Energy\ supply} \quad (6)$$

The state-of-the-art method to integrate green energy in a datacenter is supply-following-load mechanism [12], which uses instantaneously available green energy (within 1min interval estimates). Conversely, we propose a predictive policy, which estimates near term green energy availability and uses these estimates to drive job scheduling decisions. For 30min prediction, relevant to scheduling additional MapReduce batch jobs which typically complete within this time frame [7], weather conditioned moving average for estimating solar power has mean error of 9.6%; and nearest-neighbor estimator for wind energy prediction has 21.2% mean error [7]. Instantaneous and predictive policies for green energy do not lead to different results for the situation where green energy is used first to power the datacenter, and any additional energy needs are supplemented with brown. However, in the second case, where green energy is used to run jobs which otherwise would not have been scheduled, we observe significant differences.

We simulate a datacenter using 6500 minutes of solar and wind power data. Solar data is gathered from a solar panel located at University of California San Diego and wind data is obtained from a wind farm installation located in Lake Benton, provided by National Renewable Energy Laboratory. We performed our simulations for solar only, wind only and combined renewable energy supplies. The simulations are done using GENSim [7] and data obtained by running RUBiS and MapReduce on Intel Nehalem servers within a datacenter container. The results of performance and power simulations have been validated on measurements as described in the previous section.

We use MapReduce job completion time to compare performance of the two study cases and the baseline (no green energy) in Figure 2a. We see that Case1 obtains the same performance as the baseline example, as it does not process any extra batch jobs. Both policies in Case2 result in better job performance compared to baseline because of extra jobs run. However, prediction leads to an additional 15% faster job completion time compared to instantaneous green energy usage, a key benefit.

Figure 2b shows results for energy efficiency and power consumption. Case 1 leads to 20% lower brown energy usage than the baseline as it uses all available green energy. Conversely, Case2 increases the total power consumption as it runs additional jobs with additional green energy while keeping the brown energy consumption constant. The increase in total power is less than 20% for instantaneous policy because green energy levels are not high enough to initiate more jobs. Thus, not all the green energy supply is used. Finally, we compare the energy efficiency with these three policies based on the metric defined in Equation 5. We see that predictive policy makes better use than both instantaneous policy and Case1 by 12% and 5% respectively.



a) MapReduce Job Completion Time

b) Total and Brown Power Consumption with Energy Efficiency Metric values

c) Green Energy Efficiency % for Case 2

Figure 2: Study of the effects of instantaneous vs. predictive green energy use in datacenters

Lastly, we study the effects of using instantaneous vs. predictive green energy usage on the green energy efficiency. Green energy usage efficiency is on average 2x higher with prediction as compared to the instantaneous case (Figure 2c). We also observe that as the variability of green energy increases (e.g. when using only wind), the difference between predictive and instantaneous green energy efficiency grows to as high as 3x, pointing to clear benefits of using green energy prediction.

V. CONCLUSION

Datacenter modeling and simulation are key to understanding performance and energy tradeoffs when designing management policies such as DVFS, workload consolidation and others. We survey the available datacenter simulators, their capabilities and analyze their limitations. Additionally, we provide a case study with GENSim, where we analyze different green energy integration methods in a datacenter environment. We define an energy efficiency metric, focusing both performance and brown energy consumption values, and compare the integration methods in terms of this metric to find the most energy efficient solution. The simulation results show that running extra workloads along with green energy prediction is a very effective solution, obtaining 30% better energy efficiency and 22% better job completion time, compared to no green energy use, with over 90% green energy efficiency.

VI. SHORT BIOGRAPHIES

Baris Aksanli is a PhD student in the Computer Science and Engineering Department at University of California, San Diego. He entered the PhD program in September, 2010. His advisor is Prof. Tajana Rosing and he is a member of System Energy Efficiency Lab (SEELab). He received his bachelor degrees in Computer Engineering and Mathematics from Bogazici University, Istanbul in June 2010. His current research interests are high performance computing, statistical data analysis and energy efficiency in large scale systems.

Address: Department of Computer Science & Engineering, UC San Diego 9500 Gilman Drive, La Jolla, 92093.
Email: baksanli@ucsd.edu
Tel: (858) 534-9892

Jagannathan Venkatesh is a PhD student in the Computer Science and Engineering Department at the University of California, San Diego. He entered the PhD program in September, 2010 under Prof. Tajana Rosing and as a member of the System Energy Efficiency Lab (SEELab). His research interests are renewable energy integration in computing and commercial domains, context-aware computing in mobile platforms, and high-performance parallel computing. He received his bachelor's degree in Computer Science and Electrical engineering from the University of Virginia in May 2008, advised by Harry Powell.

Address: Department of Computer Science & Engineering, UC San Diego 9500 Gilman Drive, La Jolla, 92093.
Email: jvenkate@ucsd.edu
Tel: (858) 822-5313

Tajana Simunic Rosing is currently an Associate Professor in Computer Science Department at UCSD, a director of System Energy Efficiency Lab at UCSD and a head of the Large Scale Systems thrust that is a part of the MuSyC center. Her research interests are energy efficient computing, embedded and wireless systems. Prior to this she was a full time researcher at HP Labs. Concurrently at Stanford University she has been involved with leading research of a number of graduate students and has taught graduate level classes. She finished her PhD in 2001 at Stanford, along with finishing her Masters in Engineering Management. Her PhD topic was Dynamic Management of Power Consumption. Prior to pursuing the PhD, she worked as a Senior Design Engineer at Altera Corporation. She obtained the MS in EE from University of Arizona. She has served at a number of Technical Paper Committees including DAC, DATE and IPSN conferences, and is currently an Associate Editor of IEEE Transactions on Mobile Computing. In the past she has been an Associate Editor of IEEE Transactions on Circuits and Systems.

Address: Department of Computer Science & Engineering, UC San Diego 9500 Gilman Drive, La Jolla, 92093.
Email: tajana@ucsd.edu
Tel: (858) 534-4868

VII. BIBLIOGRAPHY

[1] L. A. Barosso and U. Holzle, The data center as a computer: An Introduction to the Design of Warehouse-Scale Machines, 2009.

[2] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, 2011.

[3] J. Mankoff, R. Kravets and E. Blevis, "Some computer science issues in creating a new sustainable world," *IEEE Computer,* vol. 41, no. 8, 2008.

[4] S. Lim, B. Sharma, E. Kim and C. Das, "MDCSim: A multi-tier data center simulation platform," in *Cluster Computing and Workshops (CLUSTER)*, 2009.

[5] D. Kliazovich, P. Bouvry, Y. Audzevich and S. Khan, "GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference* , 2010.

[6] D. Meisner and T. Wenisch, "Stochastic queuing simulation for data center workloads," in *Workshop on Exascale Evaluation and Research Techniques (EXERT)*, 2010.

[7] B. Aksanli, J. Venkatesh, L. Zhang and T. Rosing, "Utilizing green energy prediction to schedule mixed batch and service jobs in data centers," in *Workshop on Power-Aware Computing and Systems (HotPower)*, 2011.

[8] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience,* 2010.

[9] "DCGS Inc. THe Romonet - BCS Data Center Simulator," [Online]. Available: https://dcsg.bcs.org/welcome-dcsg-simulator.

[10] F. Julia, J. Roldan, R. Nou, O. Fito, A. Vaque, I. Goiri and J. Li, "EEFSim: Energy efficency simulator," Research Report number: UPC-DAC-RR-CAP-2010-15, 2010.

[11] S. Gupta, R. Gilbert, A. Banarjee, Z. Abbasi, T. Mukherjee and G. Varsamapoulos, "GDCSim: A tool for analyzing Green Data Center design and resource management techniques," in *Green Computing Conference and Workshops (IGCC)*, 2011.

[12] A. Krioukov, C. Goebel, S. Alspaugh, Y. Chen, D. Culler and R. Katz, "Integrating Renewable Energy Using Data Analytics Systems: Challenges and Opportunities," *IEE Data Engineering Bulletin,* 2011.