Hybrid Simulator for Cyber-Physical Energy Systems

Ayan Banerjee, Joydeep Banerjee, Georgios Varsamopoulos, Zahra Abbasi, and Sandeep K. S. Gupta
IMPACT Lab, Arizona State University, Tempe, Az
Email: {abanerj3,jbanerje,georgios.varsamopoulos,zabbasi1,sandeep.gupta}@asu.edu

***Abstract*—Simulating cyber-physical energy systems such as data centers requires the characterization of the energy interactions between the computing units and the physical environment. Such interactions involve discrete events such as changes in operating modes of cooling units, and also transient processes such as heat flow. An event-based simulator fails to capture continuous transient effects while a time-stepped simulation can ignore events occurring within the decision interval. This paper proposes an error-bound hybrid simulator that integrates discrete event-driven (ED) simulation and finite-horizon time based simulation (FHT) and simulates energy interactions in a cyber-physical energy system. We apply this simulator to a data center case and validate the simulation results by comparing them with simulations performed in previous literature. We also evaluate the accuracy of the simulator by comparing the test case results with realistic data obtained from a real data center deployment. The error bound of the simulator is a user input and influences the time interval of the ED and FHT modules.**

I. Introduction

Energy-aware and green computing initiatives necessitate management of the equipment according to the energy needs of the computation. Hence, design of computing infrastructure such as data centers should include an integrated analysis of computational processes, energy transfer, and thermal dynamics to reduce cost of operation (e.g., electricity cost), carbon footprint (e.g., by use of alternative energy source), and increase sustainability and lifetime [1], [2]. Such an integrated approach mandates the consideration of data centers as cyber-physical energy systems (CPESes), which exhibit both computational behavior (cyber processes) and significant energy exchange (physical processes) within themselves and with the environment [3]. This paper presents the design and deployment of a holistic simulator for CPESes and shows its application to aide the sustainable design of data centers.

Traditionally, simulation of data centers has relied on steady-state assumptions on the cyber performance, e.g., arrival rate, service rate, or throughput [4], [5], and steady-state evaluation of physical dynamics, e.g., thermal interaction [6]–[8]. However, several design parameters of CPESes exhibit a high degree of variability, for example in data centers input workload, thermal exchange between servers and cooling units, and energy supply from renewable sources. Recent research has shown considerable savings in energy and cost in CPESes when such variabilities are factored into the design and management phase [9].

Typically, steady-state assumptions on CPES processes enable event driven simulation (ED). For example, if an average utilization of a data center is assumed over a long period of time, then the server resource allocation following load balancing policy remains steady and does not change. Thus, only events such as infrequent changes in utilization levels have to be simulated. Transient assumptions however, need simulation at a very fine time granularity, i.e. finite horizon time-stepped simulation (FHT), to enable accurate evaluation. In a CPES, both steady-state properties and transient variabilities coexist. Employing event-based simulation on transient properties will lead to non-scalable computation time due to large number of events. On the other hand, FHT simulation may ignore events that occur within a discrete time interval. Thus, a CPES simulator should employ a hybrid simulation paradigm with integration of event-based execution for steady-state processes and time-stepped execution for transient processes into a cohesive simulation engine. Such hybrid operation of CPESes is emphasized by several researchers in different domains [10], [11]. Moreover, both ED and FHT involve discretization of time that will introduce error in evaluation of the continuous dynamics. Such errors can lead to wrong estimation of energy consumption or cyber-performance. Further, the simulator also has to provide a guaranteed error bound on its estimation of the steady-state and transient processes. This paper has the following contributions:

1. a formal description of CPES simulation and the hybrid simulation tool that captures the relevant and necessary cyber and physical events and processes,

2. evaluation of thermal aware server provisioning algorithms in data centers under transient assumptions using the simulator and validation of results using experiments in BlueCenter[1],

3. a theoretical analysis to bound the error of simulations.

II. Cyber-physical energy system model

The system model of cyber-physical energy systems (CPES) is shown in Fig. 1, which is detailed for the specific example of data centers in Fig. 2. The CPES has computing units (servers in data centers), which process requests to provide a service, and also has physical behavior such as electricity consumption and thermal interactions with cooling units through air flow. The interactions between the computing units and physical systems in a CPES are limited to energy transfer for operation of the computing units and heat transfer leading to thermal effects. The CPES takes workload as input and provides service with performance guarantees.

In a CPES, there are both discrete events and continuous processes. We consider θ as the time for which we observe the CPES. Events occur at discrete times $t_j \leq \theta$, $j \in J$, where J is the exhaustive set of events occurring during θ, and t_j

The work is partly supported by NSF grants CNS #1218505 & #0855277.

[1]BlueCenter is an experimental data center infrastructure developed by the IMPACT Lab (http://impact.asu.edu) at Arizona State University (ASU).

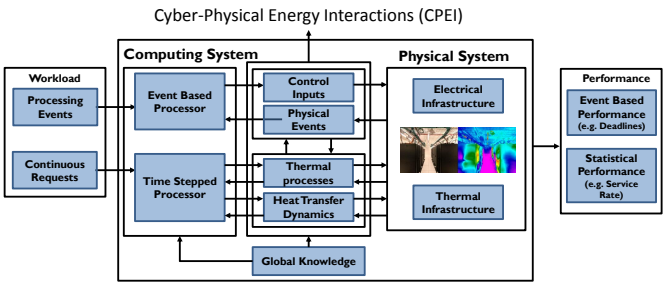

Fig. 1. Cyber-physical energy system model.

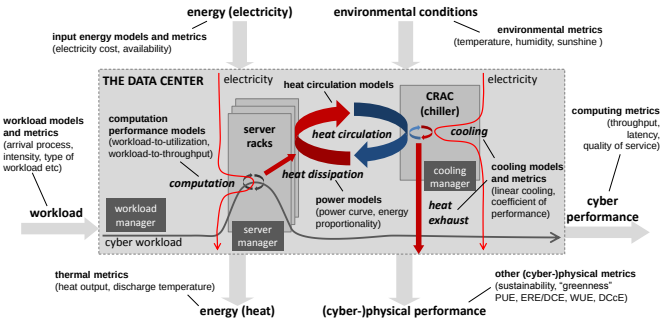

Fig. 2. Data center as a cyber-physical energy system

is the time when the *jth* event occurs. In the simulator, we assume discrete functions to determine the effects of events. Continuous time processes and their effects are represented by continuous functions. As a convention, we will denote discrete functions with capital letters and continuous functions with small letters. In a CPES, we consider n computing units. We assume that each computing unit processes some computational workload. Such workload can be of two types: a) event-based (w_e), which typically consists of long invariant processing requests that arrive in small numbers at a given time (e.g, high performance computing or HPC jobs), and b) time-stepped (w_f), which typically consist of short, highly varying processing requests that arrive in large numbers at a given time (transactional or Internet-like jobs). In a data center CPES, the continuous workload can be represented by an arrival rate, or number of requests per unit time, $w_f = f(t)$ for any $0 \leq t \leq \theta$, while the event-based workload can be represented by server requests at given times, $w_e = F(t_j)$ for any $j \in J$.

The computing units have two workload managers: a) event-based M_e, and b) time-stepped M_f. The simulator for CPES maintains the current state $CS(t)$, which is a vector consisting of state parameters of the computing units, physical systems and cyber-physical energy interactions. The workload manager takes the workload and the current state as input and outputs the utilization of each computing unit over time. The utilization for event-based workload is a discrete function where as the utilization for time-stepped workload is a continuous function as given in Equation 1.

$$M_e(w_e(t_j), CS(t_j)) = U(t_j), \text{ and }, \quad M_f(w_f(t), CS(t)) = u(t), \tag{1}$$

where $U(t_j)$ and $u(t)$ are n-dimensional vectors consisting of the discrete utilization U_i and continuous utilization u_i of a computing unit i at the event time t_j and at the time instant t, respectively. The overall utilization of the computing unit is given by Equation 2.

$$\mu(t) = u(t) + U(t_j), \text{ for } t_j < t \leq t_{j+1}. \tag{2}$$

The utilization of the computing units controls the power consumption of the servers:

$$p(t) = f_p(\mu(t)), \tag{3}$$

where f_p is a transformation function obtained by profiling the computing units. In data centers this power consumption is the key link that gives rise to energy interactions.

The heat interactions within a CPES is fueled by the power consumption of the computing units. For example, in a data center, the air flow causes heat from one server to circulate within the room and affects the temperature rise at other servers. The heat map of a data center follows a set of linear differential equation (discussed in Section III-A2). The temperature rise map in a data center also is linearly dependent on the amount of heat flowing into a particular region. Based on the temperature rise, the cooling unit in a data center reacts by serving cold air into the data center. The rate of change of cold air temperature is also linearly related to the inlet temperature of the data center. Such transient processes thus affect the different physical parameters of the CPES. These physical parameters as well as the computational parameters form a CPES **state**. In case of data center CPES, the state consists of: a) server utilization, b) server power consumption, c) re-circulated heat matrix, where each entry is the fraction of heat from a computing unit to some other, d) inlet temperatures of the server, and e) outlet temperature of computer room air conditioning units (CRACs). A sample of this state at any time t is $CS(t)$, or the current state of the data center CPES.

Moreover, although the interaction are governed by linear continuous dynamics, the evolution of the physical parameters of the CPES can cause discrete events. For data centers, if the server inlet temperature rises above a pre-specified redline, then the processing speed of the server reduces i.e. it throttles. If the inlet temperature of the CRAC increases beyond a certain point, then it starts extracting more heat. These events can have effects on both the management algorithms and the physical dynamics. For example, a server throttling will change the job execution times, which will result in a change in the utilization vector $\mu(t)$. Again a change in CRAC power extraction will change the heat flow dynamics in the data center. Note that there exist a close coupling between decisions of the management algorithms and the effects in the physical domain through the current CPES state variables. Further, we notice that there are both discrete domain and continuous domain variables in a CPES. Thus, a CPES simulator should allow hybrid simulations in two major senses: a) it should allow simulations of management algorithms that are aware of physical dynamics in the data center, and b) it should allow both event-based and time-stepped simulations.

III. Overview of proposed simulator (SIM)

In an event-based simulation, the time is discretized into unequal intervals, where each interval starts at the start of

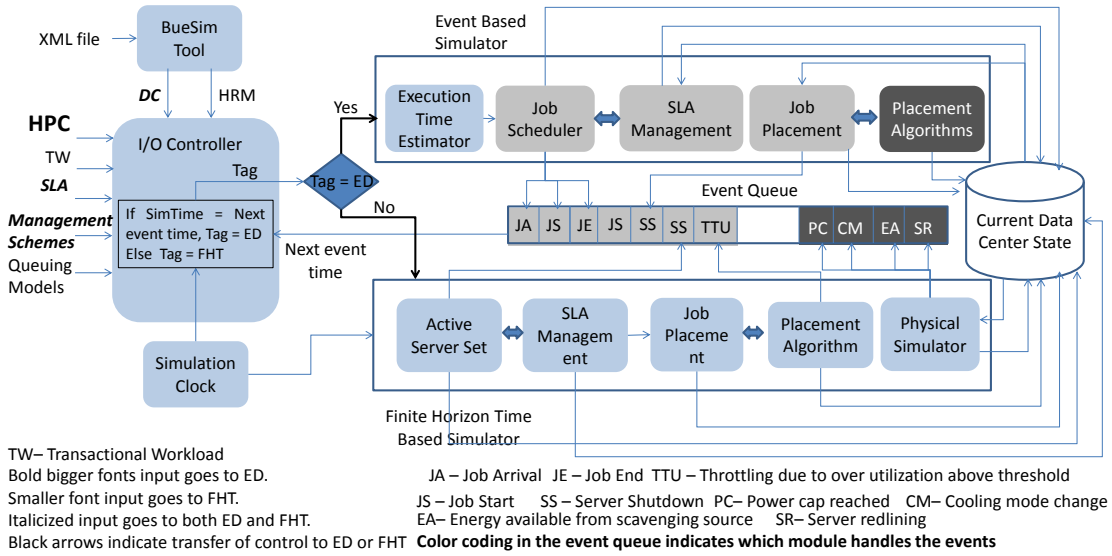

Fig. 3. The hybrid simulation engine for data centers.

an event. While in a time-stepped simulation the time is discretized into small equal intervals. The event-based representation of time expedites the simulation in case the events occur sparsely. The time-stepped representation of time is important for continuous evaluation of physical dynamics as well as highly varying computational workloads.

If a time-stepped representation is applied to both ED and FHT simulation then events that occur in between two time intervals may be ignored. However, if we apply event-based simulation to both ED and FHT simulation then we will have a large number of events to process at a given time which will slow down the simulation. Further, since we are discretizing the time into intervals the evaluation of the physical dynamics incurs errors. In the following sections we discuss our hybrid simulator and also show how we can bound the error in estimation of the physical dynamics.

We propose a two-tier hybrid simulator which has both event driven (ED) and finite horizon time-based (FHT) simulation. The FHT simulator uses short time cycles (slots) to manage: (i) transient behavior, (ii) interaction between the event-driven simulation and time-based simulation of cyber events, and (iii) events associated with the transactional workload. The ED simulator simulates: a) resource management modules that usually has long decision time interval such as server on/off management, b) low variability workloads such as HPC workloads, and c) events generated by the FHT.

Fig. 3 shows a simple example of how our simulator works for data centers. The example assumes that there is one-tier FHT and ED simulator engine, and that the simulator simulates job scheduling, server and cooling management. The input to the simulator is a data center configuration specified in XML. The specification is processed by BlueSim[2] to obtain transient heat models of data centers (discussed in Section III-A2). The XML specification and the transient heat model parameters along with the HPC or transactional workload, management schemes, and service level agreements are provided as input to the controller. As shown in Fig. 3, the event driven simulator acts as a controller to the time based simulator. Jobs are first fed to the job dispatcher in the ED. According to workload tags in the jobs similar to the one suggested in SIM [4] the transactional jobs are routed to the FHT, while the HPC jobs are routed to the ED. ED simulator processes events such as job arrival, job start and job end, and communicates with the workload manager, and placement modules to assign jobs to servers. FHT simulator processes the statistical information of the workload instead of every individual job. Therefore, the data center processes of these workload is simulated per-slot instead of per-request. Both ED and FHT can generate events and update $CS(t)$ of the data center.

Initially, the control is with the ED. The ED processes an event and gives the control back to the FHT. The FHT runs a fine grain time simulation until the simulation clock suggests that it is time for processing the next event in the event queue. The control is then transferred to the ED. In the following sections we discuss the models of computation and physical processes used by the simulator.

A. Transient models used by the simulator

The proposed simulator needs to consider both cyber and physical transients. Cyber transients include dynamic queuing models with mathematical expressions for utilization, service rate, and latency, workload generation models, and server throttling models. The physical transients include cooling models, heat recirculation models, and server power models.

1) Cyber aspects: An efficient simulator should rely on fast quantitative models to predict the transient and steady-state metrics of the computation's performance. Models of transient variation of performance is determined from experiments conducted in the BlueCenter. The second graph in Fig. 4 shows the variation of CPU utilization of IBM X Series 336 servers with different arrival rates which clearly shows

[2]BlueSim is a CFD simulator developed by IMPACT Lab [4]. It carries out a set of CFD simulations to obtain heat recirculation matrix that can be used to do fast simulations of transients.

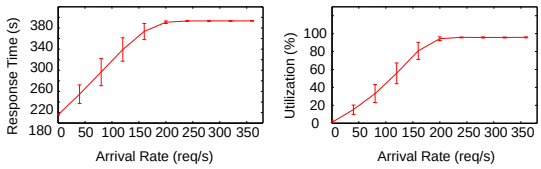

Fig. 4. Round trip time and utilization v.s. arrival rate, respectively.

a threshold utilization (150 req/s) after which the utilization becomes constant. This effect is due to rejection of jobs by the servers since the arrival rate is greater than the service rate. At this stage the servers are utilized at a constant rate and the round trip time also levels off. The saturation of utilization shown by the first graph in Fig. 4 is modeled by fitting a curve to this graph using the following function:

$$\text{performance} = f(\lambda, w, s, T) = \begin{cases} f_1(\lambda, w, s) \text{ if } T < T_{\text{threshold}}, \\ f_2(\lambda, w, s) \text{ if } T \geqslant T_{\text{threshold}}, \end{cases}$$

where λ denotes the workload intensity, w denotes the workload type , and s denotes the server type.

Particularly, the simulator should provide f_1 and f_2, to map the input workload of a server to its utilization level (denoted by u) and the delay of requests (denoted by d), e.g., $(u, d) = f_1(\lambda, w, s)$. This mapping is obtained by assuming a dynamic queuing model similar to the one used in large scale data networks [12].

Based on the decision of data center management, it manages the workloads by assigning it to a server queue. It simulates the transient workload processing of the server by considering a G/G/c queue dynamics where rate of change of queue is denoted by $\dot{Q}_t$. The service rate for the server is currently considered uniform, however it varies based upon different states of the servers, such as throttling. Managing the arrival rate a_t, departure and service rate s_t into queues it computes the server utilization u_t. If there is excess workload than the queue capacity (q_{max}) of the active servers, it is rejected. We use the model in Equation 4 from [13].

$$\begin{array}{cc} \dot{Q}_t = \frac{(a_t - s_t) - (a_{t-1} - s_{t-1})}{t - (t-1)}, & a_t - s_t < q_{max} \\ \dot{Q}_t = 0, & a_t - s_t > q_{max} \text{ or } a_{t-1} - s_{t-1} > q_{max} \\ a_t - s_t = q_{max}, & a_t - s_t > q_{max} \\ a_t - s_t = 0, & a_t - s_t < 0 \end{array} \tag{4}$$

Throttling Model: Throttling of servers occur when the temperature rises above a redline value. We define a new parameter Red Line Temperature (RLT) which is the maximum outlet temperature for any of the server after which it throttles. The estimation of the redline temperature is modeled using physical models described in Section III-A2.

a) Workload (generation) models.: The simulator uses traces and synthetic workload models particularly to model Internet-type workload. The synthetic models should exhibit long-term and short term variations of the workload. The long term variations can be modeled using seasonal models such time series techniques e.g., SARIMA [14] while the short term variations can be done using previously proposed stochastic based models for different workload types, such as transactional web workload [15], and Web 2.0 workload [16].

We generate the scalable workload in every time slots using Surge method [17]. As the workloads are getting more diverse many researchers are focusing on considering transient workload and its effect of data center dynamics [18]. The transient workload has details of information arrival in every fine time slot (10ms).

2) Physical aspects: The physical aspects include:

a) Heat circulation model: The simulator uses the heat recirculation model proposed in [9] which predicts three aspects of thermal behavior: *division (spatial distribution)*: how the heat produced by each computing server is split and circulated into each other server or to each chiller; *temporal distribution*: how the heat portion of one server to another is distributed over time, which also captures the *hysteresis*: how long the heat takes to travel from one unit to another. A key assumption of this model is that the airflows in the data center are controlled and dominated by the fans such that the portion of air arriving everywhere over time is invariant to temperature. This ignores changes in the airflow due to pressure changes caused by the temperature such as hotter air rising faster (a likely cause for slight inaccuracy in our model). The heart of the model is a collection of $n \times n$ ***temporal contribution curves***, $c_{ij}(t)$, each one denoting how heat arrives to the air inlet of a *sink* (*receiving*) server j from the air outlet of a *source* server i. (For notational convenience, the chiller also is included in the ij enumeration). A c_{ij} curve conforms to Eq. 5.

$$c_{ij}(t) : (-\infty, 0] \rightarrow [0, 1], \text{ with } \int_{-\infty}^{0} c_{ij}(t)dt = 1. \tag{5}$$

$$\bar{T}_{ij}(t) = \int_{-\infty}^{0} c_{ij}(\tau) T_{i-}(t+\tau) d\tau. \tag{6}$$

$$T_{j+}(t) = \sum_{i=1}^{n} w_{ij} \bar{T}_{ij}(t) = \sum_{i=1}^{n} w_{ij} \int_{-\infty}^{0} c_{ij}(\tau) T_{i-}(t+\tau) d\tau, \tag{7}$$

A key concept of the model is the ***contributing temperature*** $\bar{T}_{ij}(t)$ of a source server i to a sink server j, which is the temperature rise as affected by the history of the source server i's outlet temperature $T_{i-}(t)$ through the spreading of c_{ij} (see Eq. 6). In addition to the c_{ij} curves, the model assumes $n \times n$ ***weighting factors*** w_{ij}, which define how each contributing temperature $\bar{T}_{ij}(t)$ factors into the actual temperature $T_{j+}(t)$ at the air inlet of server j. The weighting factors w_{ij} can be organized into a matrix W. The core hypothesis of the model is that the temperature at the air inlet of a server j is the convex weighted sum of the temperature contributions of all servers (and of the chiller) as they accumulate over time from the past (Eq. 7), where $\sum_{i=1}^{n} w_{ij} = 1, \forall j$, because of the thermal isolation assumption.

b) Cooling models: The cooling process has two aspects. One aspect is to describe the "supply" temperature of a chiller given the input temperature to it. Transient cooling models have been developed in our prior work [9], [19], [20]. These transient cooling models are used in the simulator. The RLT of the servers can be estimated using the heat flow dynamics within the data center. The heat flow in data center is represented using the models described in our prior work [9].

B. Error analysis of the simulation

Transient models are continuous time models. In order to simulate in a computer, the time has to be discretized. Such

discretization can lead to errors in estimating the values of the variables in the physical system models. In this section we determine a temporal sampling such that the error of simulation does not exceed a pre-specified bound ϵ. To this effect, we observe that most of the models considered in this work are linear time invariant transients. The only non-linearities existing in the data center transient is the conversion of arrival rate to utilization and then from utilization to power which is ignored in this error analysis. For example, if we consider the transient heating model, the error analysis can be done by considering the linear time invariant system:

$$\dot{x} = Ax + B, \tag{8}$$

where x is the $n + 2n^2$ dimensional vector obtained from the transient cooling model. The vector consists of outlet temperatures of all the chassis (n in number), the temperature contributions of a chassis j to the inlet of a chassis i, which is n^2 in number, and n^2 other intermediate variables used to express the transient heating effects as an LTI system. We want to ensure that the error due to temporal discretization is less than ϵ. Equation 9 expresses this in mathematical terms.

$$||x(t+\tau) - x(t)||_\infty < \epsilon, \forall \tau \in [0, h_t], \tag{9}$$

where $||x(t+\tau)-x(t)||_\infty$ denotes the infinity norm which means the maximum of the elements in the vector $x(t+\tau)-x(t)$. Here h_t is the time interval of the FHT.

If we consider the maximum time variance of the vector $x(t)$ and express it in terms of h_t, then we can get an expression for h_t as shown below in terms of ϵ.

$$\begin{aligned} ||\dot{x(t)}||_\infty &= ||Ax(t) + B||_\infty, \\ \frac{||x(t+h_t) - x(t)||_\infty}{h_t} &\leq ||A||_\infty ||x(t)||_\infty + ||B||_\infty, \\ ||x(t+h_t) - x(t)||_\infty &\leq h_t(||A||_\infty ||x(t)||_\infty + ||B||_\infty). \end{aligned} \tag{10}$$

Thus, to satisfy Equation 9, the right hand side of the Equation 10 can be less than ϵ. Thus, we have

$$h_t(||A||_\infty ||x(t)||_\infty + ||B||_\infty) \leq \epsilon, \text{ i.e., } h_t \leq \frac{\epsilon}{||A||_\infty ||x(t)||_\infty + ||B||_\infty}. \tag{11}$$

The value of h_t depends on the maximum value of the elements in $x(t)$. This maximum value can be arbitrary unless we define the type of errors we cannot tolerate. Equation 11 can be used in two ways: a) given an error of ϵ °C on the outlet temperature of the servers an appropriate h_t can be determined such that the simulation error does not go beyond ϵ, b) given a h_t, FHT time slot, the error in simulation can be determined.

IV. Validation and evaluation

The hybrid simulator takes a mixture of SURGE workload trace for a total of 30 percent of total data center utilized and HPC workload trace from ASU HPC data center. The simulation was run for a 24 chassis data center with 5 servers in each chassis, service rate of 200 request per second, queue capacity of 30 requests and RLT of 35 °C. The simulation was run for 1 hour with each iteration having time intervals of 100 ms. Two epochs of 30 minutes each were considered for workload management with chassis 14 through 19, 22 and 24 on in the first epoch and 16 and 17 in the second epoch. The CRAC switches cooling mode when the inlet temperature exceeds 28 °C. The hybrid simulator is compared

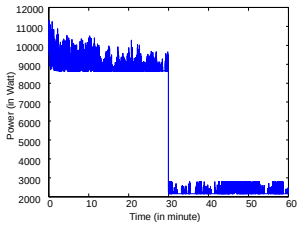

Fig. 5. Total power consumed in the data center using transient simulator for 1 hour period.

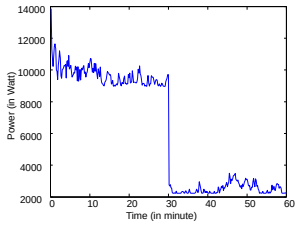

Fig. 6. Total power consumed in the data center using GDCSim simulator for 1 hour period.

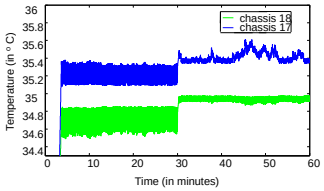

Fig. 7. Inlet temperature variation using transient simulator for chassis 17 and 18 for 1 hour period.

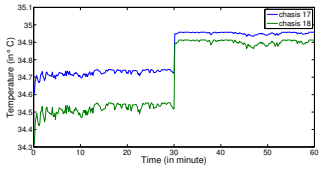

Fig. 8. Inlet temperature variation using GDCSim simulator for chassis 17 and 18 for 1 hour period.

with GDCSim developed by the authors, was run for same input data. The two simulations were compared with respect to the total power consumed by the data center (Figs. 5 and 6), and inlet temperatures (Figs. 7 and 8) for two chassis (chassis 17 and 18). We see that although the average trend is same in both power and temperature key events are missed in GDCSim simulator. Several short CRAC switching events are missed while computing total power hence GDCSim underestimates the power by 7.3%. From the temperature plots we see that due to high variability of temperature there are several places where chassis 17 crosses RLT and is not detected by GDCSim.

The same utilization schedule was run in BlueCenter and we recorded the power consumption. Fig. 9 (a) and (b) compare the power consumption from BlueCenter and the simulator averaged over 3 mins. There is a near match between the two curves from experiment and simulation plots.

A. Error evaluation

Physical events such as, server redlining, or power consumption reaching a power cap, or cooling mode change are based on thresholds of the parameters in $x(t)$. For example, if we want to capture all server redlining events then we should have $||x(t)||_\infty = RLT = 35°C$, $||A||_\infty = 1$ or maximum value of heat recirculation, and $||B||_\infty = \frac{p_{max}}{C_p}$ where p_{max} is the maximum power of any server and C_p is the specific heat, then for our chosen simulation time granularity $h_t = 100ms$ we get an ϵ of $3.7°C$. This means that the outlet temperature predictions have an error of $3.7/(T_{max} - T_{min}) = 9.25\%$, where $T_{max} = 45°C$ the maximum temperature attained by any server and $T_{min} = 5°C$ the minimum temperature. This error in temperature can be further reduced if we reduce the time step. But the simulation time increases proportionately.

V. Related Work

Authors in [21] develop a simulation tool to evaluate the data center power efficiency through simulating the cyber

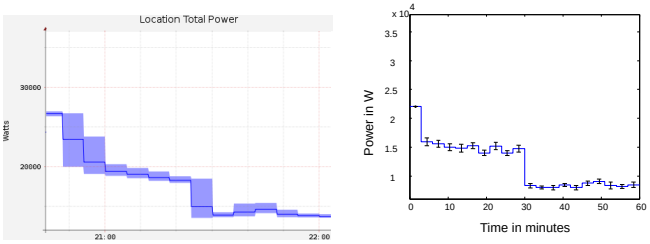

Fig. 9. a) Power consumption of servers in BlueCenter, and b) Power consumption from simulator with same utilization.

aspects of data centers. The tool uses steady-state queuing models to simulate a data center servicing multi-tier web applications. BigHouse further improves the MDCSim models though introducing stochastic queuing system which provides a stochastic discrete time simulation of generalized queuing models to simulate the performance of the data center applications in more detail than that of MDCSim [15]. CloudSim, is capable of simulating the per-server power and performance metrics, where the main abstraction unit is a VM [22]. The above tools ignore simulating the physical aspect of data centers and the potential interaction between cyber and physical aspects. They rely on the steady-state models to estimate the power consumption and performance of servers, lacking simulating the transient processes that can have long-time effects on data center (e.g., redlining) and an error analysis to specify the confidence intervals of the results. GDCSim, used to compare with the proposed simulator, is a cyber physical simulator, which accounts for both batch and transactional jobs, however it relies on steady-state models to estimate the metrics related to cyber and physical aspect of data centers. There are also some simulation tools that provide estimation of data center physical aspects such as Mercury [6], but ignore interaction between physical and cyber aspects.

VI. Conclusions

This paper proposes an error bounded hybrid simulator that can process both transient and steady-state behavior in a cyber-physical energy system. The simulator has an event driven module which considers computational and physical events and a finite horizon time driven module which considers transient computation and physical processes. We have used it for data center systems and validated its operation through experiments in a real deployments, BlueCenter. We also propose an analysis on the error bounds of the simulator. The error bound can be varied by adjusting the step size of the FHT. With a 100 ms time step we obtained an upper error bound of around 9% of the average temperature rise.

References

[1] S. K. S. Gupta, T. Mukherjee, G. Varsamopoulos, and A. Banerjee, "Research directions in energy-sustainable cyber-physical systems," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 57 – 74, 2011.

[2] S. K. Gupta and G. Varsamopoulos, "Towards formal framework for designing energy-cps," *National Workshop on Research Directions for Future Cyber-Physical Energy Systems*, 2009.

[3] A. Banerjee, K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security and sustainability of mission-critical cyber physical systems," *IEEE Proceeding on cyber-physical systems*, 2011.

[4] S. K. S. Gupta, R. R. Gilbert, A. Banerjee, Z. Abbasi, T. Mukherjee, and G. Varsamopoulos, "GDCSim - an integrated tool chain for analyzing green data center physical design and resource management techniques," in *International Green Computing Conference (IGCC)*, 2011.

[5] J. S. Ahn and P. B. Danzig, "Packet network simulation: speedup and accuracy versus timing granularity," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 5, pp. 743–757, 1996.

[6] T. Heath, A. P. Centeno, P. George, L. Ramos, and Y. Jaluria, "Mercury and Freon: temperature emulation and management for server systems," in *ASPLOS-XII*. New York, NY, USA: ACM Press, 2006, pp. 106–116.

[7] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "Tacoma: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Trans. Archit. Code Optim.*, vol. 9, no. 2, pp. 11:1–11:37, jun 2012.

[8] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol. 53, no. 17, pp. 2888 – 2904, dec 2009.

[9] M. Jonas, R. R. Gilbert, J. Ferguson, G. Varsamopoulos, and S. K. S. Gupta, "A transient model for data center thermal prediction," in *International Green Computing Conference (IGCC)*, jun 2012.

[10] A. Banerjee and S. K. S. Gupta, "Spatio-temporal hybrid automata for safe cyber-physical systems: A medical case study," *Intl' Conf' on Cyber-Physical Systems*, April 2013.

[11] A. Banerjee and S. K. Gupta, "Tackling new frontiers in modeling and analysis of cyber-physical systems," *1 st workshop on Cyber-Physical Systems Education*, 2013.

[12] Y. Guo, W. Gong, and D. Towsley, "Time-stepped hybrid simulation (tshs) for large scale networks," in *INFOCOM 2000.*, vol. 2. IEEE, 2000, pp. 441–450.

[13] S. Bohacek, J. P. Hespanha, J. Lee, and K. Obraczka, "A hybrid systems modeling framework for fast and accurate simulation of data communication networks," in *ACM SIGMETRICS*, New York, NY, USA, 2003, pp. 58–69.

[14] V. D. Van Giang Tran and S. Bacha, "Hourly server workload forecasting up to 168 hours ahead using seasonal arima model," in *Industrial Technology (ICIT), 2012 IEEE International Conference on*, march 2012, pp. 1127 –1131.

[15] D. Meisner, J. Wu, and T. F. Wenisch, "Bighouse: A simulation infrastructure for data center systems," in *Performance Analysis of Systems and Software (ISPASS), 2012 IEEE International Symposium on*, apr 2012, pp. 35 –45.

[16] A. Abhari and M. Soraya, "Workload generation for youtube," *Multimedia Tools and Applications*, vol. 46, pp. 91–118, 2010.

[17] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," *SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 1, pp. 151–160, 1998.

[18] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. of the 3nd ACM Symposium on Cloud Computing, SOCC*, vol. 12, 2012.

[19] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta, "Integrating cooling awareness with thermal aware workload placement for hpc data centers," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 2, pp. 134 – 150, 2011.

[20] G. Varsamopoulos, A. Banerjee, and S. K. S. Gupta, "Energy efficiency of thermal-aware job scheduling algorithms under various cooling models," in *International Conference on Contemporary Computing IC^3*, Noida, India, aug 2009, pp. 568–580.

[21] S.-H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "Mdcsim: A multi-tier data center simulation platform," in *CLUSTER.*, 31 2009-sept. 4 2009, pp. 1 –9.

[22] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," *CoRR*, vol. abs/0903.2525.