GDCSim: A Tool for Analyzing Green Data Center Design and Resource Management Techniques

Sandeep K.S. Gupta*, Rose Robin Gilbert*, Ayan Banerjee*, Zahra Abbasi*, Tridib Mukherjee†[1], Georgios Varsamopoulos*
Impact Lab, School of Computing, Informatics and Decision Systems Engineering, ASU, Tempe, Arizona*
Xerox Research Center, Webster,NY†
{sandeep.gupta,roserobin.gilbert,abanerj3,zabbasi1,georgios.varsamopoulos} @asu.edu*,Tridib.Mukherjee@xerox.com†

***Abstract*—Energy consumption in data centers can be reduced by efficient design of the data centers and efficient management of computing resources and cooling units. A major obstacle in the analysis of data centers is the lack of a holistic simulator, where the impact of new computing resource (or cooling) management techniques can be tested with different designs (i.e., layouts and configurations) of data centers. To fill this gap, this paper proposes *Green Data Center Simulator (GDCSim)* for studying the energy efficiency of data centers under various data center geometries, workload characteristics, platform power management schemes, and scheduling algorithms. GDCSim is used to iteratively design green data centers. Further, it is validated against established CFD simulators. GDCSim is developed as a part of the BlueTool infrastructure project at Impact Lab.**

***Keywords*-Simulator; energy efficient; green, data center;**

I. Introduction

Green initiatives in data center design encompasses energy efficiency, thermal awareness and usage of renewable sources of energy. To achieve energy efficiency and thermal awareness, data center designers have mainly focused on two primary objectives: (a) *improving physical design* [1], which includes reducing heat recirculation and hence hot-spots (to reduce the cooling demand), providing efficient cooling mechanisms [2] (to reduce cooling energy consumption), and developing low power computing equipment [3], [4]; and (b) *developing energy-efficient computing resource management strategies*, such as workload scheduling algorithms [5]–[7], cooling management [8], and power management schemes [9].

Performance and efficiency analysis of green strategies in data centers is traditionally done by: 1) *energy efficiency analysis* of resource management algorithms, which include *workload scheduling* [10]–[12], active server set selection [10], [13], and *power management* of servers with dynamic voltage and frequency scaling [14], 2) *thermal analysis* of the data center under different physical configurations and resource management strategies [1], [15]–[17], and 3) *closed loop operation* of the resource management strategies with the physical behavior such as thermal behavior and air flow within the data center [6], [13], [18]. Of much interest is the close operation of the resource management algorithms and physical behavior in a data center through thermal interactions. For example, the extent of heat recirculated in a data center (i.e., a physical design attribute) may affect the energy efficiency of a workload scheduling algorithm [7] (i.e., the computing operation). Similarly, the workload placement may cause hot spots and thus demand a redesign of the data center layout [17]. It is thus imperative to consider such *cyber-physical inter-dependencies* for devising techniques for the two primary objectives for green data center design.

"***BlueTool***" is an NSF-funded computer research infrastructure project, with main objective of developing and provide the appropriate research infrastructure (both in hardware and software) to promote awareness on the environmental significance of data center operation around the world and means to design green data centers. To our knowledge, there is no publicly available integrated software tool that can be supplied both a physical data center description and a management technique description, and simulate them in one setting. In alignment with the goals of "BlueTool", this paper **proposes** the *Green Data Center Simulator (GDCSim)*, a simulation tool that unifies the simulation of management techniques with a simulation of the physical behavior of a data center.

Such a holistic simulation tool should have the following **key design features** as shown in Table I. The key features include: 1) *automated processing*, which enables operation without user intervention, 2) *online analysis capability*, which allows real time simulation of management decisions based on changes in the physical environment in the data center, 3) *iterative design analysis*, which enables design time testing and analysis of different data center configurations before deployment, 4) *thermal analysis capability*, which characterizes the thermal effects within the data center room at a given time, 5) *workload and power management*, which enables workload scheduling and controlling the power modes of servers in the data center, and 6) *consideration of cyber-physical interdependency*, which enables feedback of information on temperature and air flow patterns in the data center to the management algorithms and the closed loop operation of the servers and cooling units (CRAC) to achieve energy efficient operation. From Table I, it can be seen that considerable amount of research is devoted to online analysis of workload management algorithms which also consider the cyber-physical interdependencies within the data center. Further, Computational Fluid Dynamics (CFD) simulator tools to characterize the thermal effects and airflow patterns also exists in abundance. However a simulator, which enables holistic design of a data center with the above features is missing. GDCSim, proposed in this paper, fills this gap.

The principal **challenge**, in this regard, is the accurate yet computationally lightweight characterization of the cyber-physical inter-dependencies. Accurate characterization of the

GDCSim is developed as a part of the NSF CRI project #0855277 "BlueTool" (http://impact.asu.edu/BlueTool/). Work in this paper was also funded in part by NSF grant #0834797.

[1]The work was done when this author was with Arizona State University.

TABLE I
CLASSIFICATION OF EXISTING SIMULATION TOOLS AND METHODS WITH RESPECT TO KEY FEATURES OF A HOLISTIC DATA CENTER SIMULATOR

Simulation Tools/Methods	Automated Processing	Online Analysis	Iterative Design	Thermal Analysis	Workload Management	Cyber-Physical Interdependency
Online thermal aware workload management [10]–[13], [19]–[21]	✗	✓	✗	✓	✓	✓
Dynamic CRAC control [18]	✗	✓	✗	✓	✗	✓
Offline thermal aware workload management [22]	✗	✗	✓	✓	✓	✓
SLA and power analysis [23], [24]	✓	✓	✗	✗	✗	✗
CFD Simulations [1], [15]–[17]	✗	✗	✓	✓	✗	✗
GDCSim	✓	✓	✓	✓	✓	✓

physical behavior of the data center involves CFD simulations, which are extremely slow and time consuming. On the other hand, resource management strategies, often use feedback from physical world to take online decisions on placement of workloads, cooling adjustment and power management of servers. Analyzing such algorithms requires online testing of a resource management strategy on different data center design, analyzing the effect of different types of workload on data center operation, and studying the impact of different cooling mechanisms on the energy efficiency of management strategies. Hence, the evaluation of physical behavior of a data center has to be fast. While the CFD component of the simulation needs to be minimized, several simplifying assumptions that have traditionally been used in on-paper analysis need to be discarded. Such assumptions include: (i) *steady state* assumptions of power consumption or cooling behavior, (ii) *linearity* of power consumption with respect to the utilization for servers [25], (iii) *convexity* of energy consumption characteristics [26], and (iv) *constant* cooling models [8]. Such assumptions greatly reduce the accuracy of the simulation and therefore the value of the tool.

The principal **contributions** of the paper are: (i) integrating physical and resource (computing and cooling) management aspects of a data center and capturing their inter-dependencies in an automated fashion; (ii) developing a simulation infrastructure to enable online analysis of different data center physical designs under various management algorithms before deployment; and (iii) enabling iterative analysis to improve the holistic design of a data center, by providing feedback on the performance of the data center under various workload characteristics. The final outcome of the paper is the *Green Data Center Simulator (GDCSim)*, which can be used by data center designers, operators and researchers for developing green data centers of the future.

GDCSim has the following components: **(i) BlueSim**, that takes a high level XML based specification as input and performs CFD simulations to output a heat recirculation profile [5] of the data center for fast thermal evaluation; **(ii) Resource management**, that makes informed decisions about workload and cooling management based on physical behavior of data center. **(iii) Simulator**, that captures the physical behavior of the data center in response to the management decisions and provides feedback to the resource management for awareness of the changes in data center physical behavior.

To model the cyber-physical inter-dependencies, the resource management should be aware of its physical impact. This necessitates the provision of feedback loops between the resource management and simulator. Such a feature would enable the resource management to take informed decisions to ensure energy efficiency.

Further, GDCSim is envisioned to be *modular* and *extensible*. Modularization ensures that the different components of GDCSim can be used independently. For example, the CFD simulator module, BlueSim, can be used solely for generating the thermal map of the data center and may not be used in conjunction with the resource management module. Extensibility ensures that new models and assumptions can be easily plugged into the simulator. Further, it should be generic to support both Internet and HPC workloads.

The functionality of GDCSim is demonstrated by two case studies with two different data center layout and workload types. In the first case study, a data center with twenty chassis was simulated for a transactional type workload. The average Energy Reuse Effectiveness (ERE) [27] was calculated to be 1.26 with a mean response time 0.0005 seconds. To demonstrate the iterative analysis for data center design, the same data center layout was simulated for same workload conditions and management schemes, but with different type of servers. The ERE was improved from 1.26 to 1.22 with no degradation of response time. A second case study involved a HPC data center with 50 chassis and the ERE for this data center was calculated to be 1.1 with no deadline violations.

The rest of the paper is organized as follows. Section II discusses the background and related works and introduces key terms and concepts in data center operation and design, Section III describes the architecture of the GDCSim tool, Section IV demonstrates the usage of the GDCSim tool with the help of two case studies, Section V validates the results obtained from the GDCSim tool against Flovent simulations (a widely accepted CFD simulator) and discusses the potentials of the tool, and finally Section VI concludes the paper.

II. BACKGROUND

A typical data center consists of a cold aisle/hot aisle configuration with raised floor, lowered ceilings and perforated tiles. The CRACs supply air through the raised floor via perforated tiles. Computational equipment such as servers, data storage and networking devices are mounted on racks. They receive cool air from the cold aisle, which in turn absorbs the heat generated by the equipment and exits into the hot

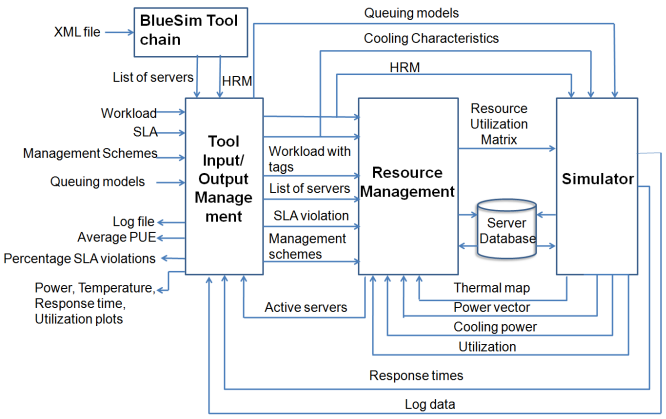

Fig. 1. GDCSim Tool Architecture.

aisle. The hot air is removed through the ceiling vents into the CRACs. Various models of CRAC operation are considered in the literature, which includes *constant cooling* and *dynamic cooling* [8] model.

To characterize the thermal behavior of a data center, an abstract heat recirculation model [28] has been used in our simulator. The air flow pattern inside a data center with all the physical objects fixed, is assumed to be in steady state [28]. The percentage of heat recirculated from one chassis to the other is termed as *cross interference coefficient*. If the power consumption of computational equipments is known, the thermal map of the data center can be quickly and accurately generated using the *heat recirculation matrix* (HRM).

The efficiency of a cooling unit is characterized by its Coefficient of Performance (COP). The COP of a cooling unit is the ratio of heat removed to the electric energy input. Hence the energy required to operate the cooling unit is given by:

$$\text{Cooling Energy} = \frac{\text{Heat removed}}{\text{COP}}$$

It is further important that the energy efficient resource management techniques respect the Service Level Agreements (SLAs). Typical SLA parameters used in GDCSim are response time and deadline violations, which are calculated using queuing models.

Data center energy efficiency is defined by various metrics such as Energy Reuse Effectiveness (ERE), Power Usage Effectiveness (PUE) [27], and the percentage of energy supplied by the power grid [29]. The ERE metric is given by:

$$\text{ERE} = \frac{(\text{Total Electric Energy} - \text{Energy Reused})}{\text{Energy consumed by IT equipment}}.$$

It can be seen that ERE gives credit to any attempt to reuse the waste heat released by the data center such as implementing heat activated cooling [30] to supplement the CRAC. To the best of our knowledge there is no previous work on a simulation tool that facilitates automated data center management by integrating physical as well as management aspects of a data center.

III. GDCSim Tool Architecture

In this section, we describe the basic architecture, functionalities, inputs and outputs of the tool as shown in Fig.1.

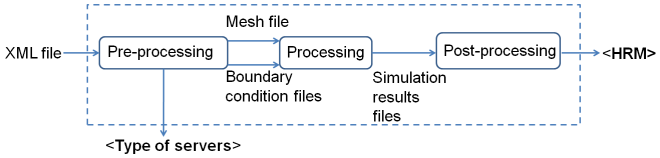

Fig. 2. The architecture of BlueSim tool chain.

The tool consists of four principal modules: 1) BlueSim, 2) Input/Output Management (IOM), 3) Resource Management (RM), and 4) Simulator, each of which is discussed in detail in the following sections. GDCSim maintains a server database with power curves and redline temperatures of different server models for various sleep (c-states), frequency (p-states) and throttling states (t-states).

A. BlueSim Tool

BlueSim is a simulation package, shown in Fig.2, which integrates various software for geometry generation, CFD simulations, and post processing. The primary objective of BlueSim is to generate an array of HRMs for different configurations of the data centers [5]. There are three main submodules in BlueSim:

1) Preprocessing module: The input to BlueSim is a high level XML based data center description in Computer Infrastructure Engineering LAnguage (CIELA). A sample data center specification using Ciela can be found at http://impact.asu.edu/Ciela.xml. It has various constructs that capture the generic layout of a data center: (i) equipment configuration, i.e., stacking of servers into chassis and chassis into racks, arrangement of these racks into rows, chassis power consumption, and air flow rate; (ii) physical data center layout, i.e., presence of raised floors, lowered ceilings, perforated tiles and vents. A parser parses the XML specification to create a geometry file which is then converted to a mesh file using GMSH [31], a standalone open source meshing software.

2) Processing: This module performs a series of offline CFD simulation of the specified data center to enable determination of HRM [5]. To maintain the usability of the tool chain, the Open Field Operation And Manipulation (OpenFOAM) C++ library was used to develop the data center CFD simulator.

3) Post Processing: Post processing to generate HRM is carried out by cross interference profiling [5]. An array of HRMs is generated for various active server set configurations (servers which are kept on for serving workload) and is sent to the IOM module. For our current simulation purpose we generate only a single HRM with all the servers active. The BlueSim generates an array of HRMs for different active server sets and sends it to the IOM module.

B. Tool Input/Output Management (IOM)

This module has two functions. First, it serves as a user interface for the entire tool chain. The user inputs are: job trace (λ), Service Level Agreements (SLAs), management schemes, and queuing model. Job trace defines the characteristics of the workload supplied to the data center. SLA requirements based on response time are currently implemented in this simulation tool. Depending on the response time feedback and

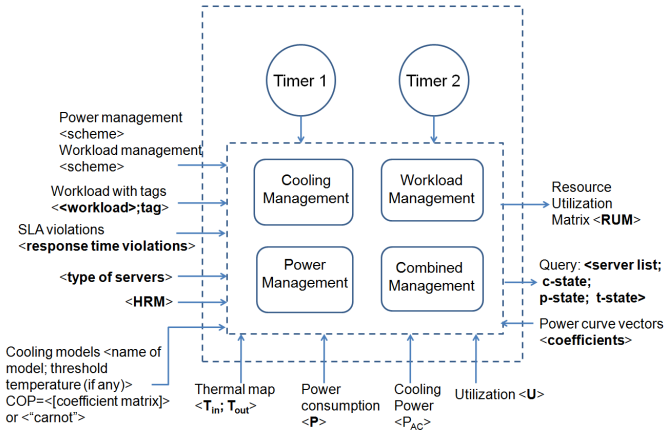

Fig. 3. Resource management module.

SLA requirements, any SLA violations are reported to the RM. Management schemes include the following: (i) power management schemes, (ii) workload management schemes, and (iii) cooling characteristics. Workload tags are attached to the job to distinguish between HPC and transactional types. The queuing model for response time computation is supplied to the simulator by the IOM. The second function of IOM is to store an array of HRMs for a data center for different active server sets. Based on the feedback from RM and simulator the IOM will report the appropriate HRM.

C. Resource Management (RM)

RM module consists of algorithms for: (a) workload management, (b) power management, (c) cooling management, and (d) coordinated workload, power, and cooling management. The use of these algorithms are triggered by two types of events: HPC job arrival for HPC data centers and timeout events for Internet or transactional data centers (IDC). Transactional workloads are continuous in nature requiring decision making at different granularity of time:
(i) long-time decision making on active server set for peak load during coarsely granular long time epoch; and *(ii) short-time* decision making on percentage load distribution to the active servers based on the average load during the finely granular short time interval. To support such time based decision making, RM maintains two timers; the timeouts of these timers trigger the long-time and short-time decision making, respectively. A multi-tier resource management is required for two reasons: (i) Different resources in the system have different state transition delay. For instance changing the frequency of CPU takes about a few milliseconds, whereas transitioning into sleep states of a typical systems takes a few seconds. (ii) the variation of transactional workload such as Web traffic is very high and exhibits hourly/minute cyclic behavior, also sometimes the variation is unpredictable. Therefore, a multi-tier resource management is required to consider the cyclic behavior of workload at different time granularity and manages resources accordingly.

Apart from the timer events, the RM listens for HPC job arrival events from the IOM module. Provisioning and support of the algorithms, the decision making to choose among the algorithms, and the output from the RM module are described below.

1) Workload management: Workload management algorithm decides on "*when*" (i.e., scheduling) and "*where*" (i.e., placement) to serve the workload. For HPC workload, this involves scheduling and placement of specific jobs when HPC job arrival events occur; while for IDCs this involves distribution of requests (i.e., the short-time decision making) among servers. Three types of workload management algorithms are supported:
a. *Rank based*: This type of algorithm involves assigning ranks to the servers and placing (or distributing) workload based on the ranks. Many heuristic thermal-aware and energy-aware placement algorithms, e.g., FCFS [7] and EDF [8], fall under this category.
b. *Control based*: Control theory based RM can be used when an accurate model of a system in a certain interval can be made. The goal is to control the system in a closed loop way to get a desired response. In case of workload management, control based RM can be designed to closely track the performance parameters (e.g response time) of jobs and then control the workload arrival rate to a system to get a desired response time.
c. *Optimization based*: This type of algorithm requires an optimization problem to be solved, or the *best* solution from a set of feasible solutions to be selected. This approach might involve finding solutions for the workload placement problem, as in XInt [5], or for both scheduling and placement problem, as in SCINT [7], or for active server set selection, as in MiniMax [6], MIP [6].

2) Power management: Power management algorithms control the power mode of a system or different components inside a system. The idea is to save energy through transition to the lower power states of system/ components when the workload is low, or to achieve a certain power capping goal. The cost is incurred in terms of delay and power consumption due to transition from idle state to active and otherwise. Power management at the system level includes sleep state transition and dynamic server provisioning. Depending upon the workload management time granularity, the power manager can decide on either sleep state transitioning of a system, or switching a set of servers to on and off when workload varies. Power management of CPU includes c-state management that control sleep state transition of CPU , and P-state management (i.e., DVFS). C-states control the sleep states of CPU. In a specific power state, the DVFS of CPU is also of interest in which the idea is to scale the frequency of CPU to the offered workload in order to save energy. Control theory based approaches or optimization based approaches can be used for power management.

3) Cooling Management: Cooling management algorithm decides on the thermostat settings of the cooling units. This can be either static, i.e., constant pre-set thermostat setting, or dynamic, i.e., a schedule of thermostat setting depending on the events.

4) Combined management: A combined management involves integrated decision making of workload, power, and cooling management. Similar to the workload management there can be three different types of algorithms: (i) ranking

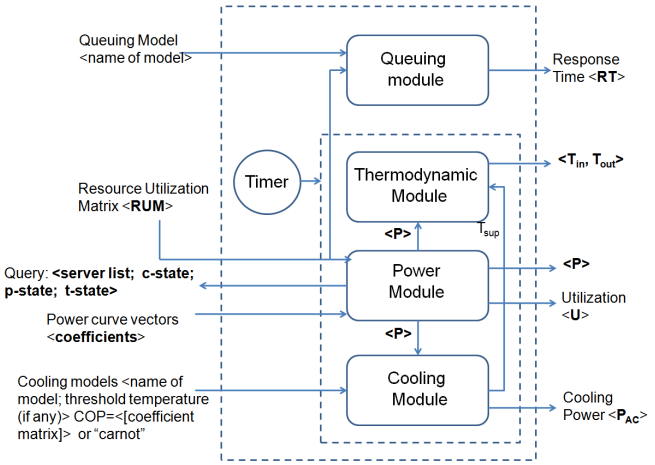

Fig. 4. Simulator module architecture.

based, (ii) control based, and (iii) optimization based. For the last two types, the control and optimization variables (respectively) increase for a combined algorithm. For the first type of algorithm, the ranking mechanism need to take into account the interplay between workload, power, and cooling (as in HTS [8]).

5) Choice of management algorithms: The required management scheme is chosen by the user through the IOM module. Any new management algorithm to be tested is defined directly at the RM module. The simulation tool also allows for optional use of (or new specification of) a management algorithm choice mechanism depending on: (i) the event occurred, (ii) the conformance to a time window to avoid reaching redline temperatures (especially the transient delay in cooling the data center if cooling management is employed), and (iii) SLA conformance [32].

6) RM output: The output from the RM module includes the following: *active server set*, i.e., set of servers which are not put into sleep state; *workload schedule*, i.e., start times of the jobs for HPC workload; *workload placement*, i.e., assignment of jobs for HPC workload and percentage distribution of requests for transactional workload; *power modes*, i.e., clock frequency of the server platforms in the active server set; and *cooling schedule*, i.e., the thermostat settings of cooling units. These outputs are compiled together to form a Resource Utilization Matrix (RUM) and sent to the simulator module for each time epoch/event. The structure of RUM is as follows: ⟨chassis list, utilization, c-state, p-state, t-state, workload tag, server type ⟩.

D. Simulator

The simulator consists of four sub modules:

1) Queuing Module: A queuing model is selected by the user at the IOM user interface. These models are used to calculate response times. SLA violations are determined by checking whether the response time exceeds a certain reference response time specified by the user.

2) Power Module: The Power module calculates the total power consumed by each server for a particular utilization level. The power module outputs a power consumption distribution vector $\mathbf{P}$= { P_1, P_2 … P_N} depending on the type of server, for a particular time epoch. The RUM supplied by the RM module contains the server model, c-state, p-state, t-state and utilization of each server for every time epoch. The power module queries the server database to retrieve the coefficient matrix of power curve for the particular model of server at a given state. Power consumption matrix is calculated based on these constraints and is supplied to Thermodynamic module and Cooling module.

The change in utilization of the servers, causes a time-delay before the power consumption reaches the new value. The time delay depends on the type of server being used. When the utilization of the server changes, the new power consumption value is stored in a queue for the respective delay period and is dispatched once the delay period is over.

3) Thermodynamic Module: The Thermodynamic module gives the thermal map of the data center. The inlet and outlet temperatures for the current time epoch, $\mathbf{T_{in}} = \{ T^1_{in}, T^2_{in} \ldots T^N_{in}\}$ and $\mathbf{T_{out}}$= { T^1_{out}, T^2_{out} … T^N_{out}} respectively, for each chassis are calculated by [5]:

$$\mathbf{T_{out}} = T_{sup} + (\mathbf{K} - \mathbf{A^T K})^{-1}\mathbf{P}, \ \mathbf{T_{in}} = T_{out} - \mathbf{K}^{-1}\mathbf{P}$$

Where T_{sup} is the CRAC supply temperature, K is the matrix of heat capacity of air through each chassis and A is Heat Recirculation Matrix. $\mathbf{T_{in}}$ and $\mathbf{T_{out}}$ together constitute the *thermal map* of the data center. This thermal map is then sent back to RM module as feed back.

4) Cooling Module: Two different cooling models are considered in the preliminary version of the tool chain. In *dynamic cooling model* we consider two basic modes of operation, namely; *high* and *low* modes. Based on the CRAC inlet temperature, the modes switch between high and low to extract P^{high} and P^{low} amount of heat, respectively. If the CRAC inlet temperature, T^{in}_{CRAC} crosses a higher threshold temperature T^{th}_{high}, the high mode is triggered and if T^{in}_{CRAC} crosses a lower threshold temperature T^{th}_{low}, a low mode is triggered. The threshold temperatures can be supplied by the user through the IOM module. Switching between modes incurs a time delay, during which the CRAC operates in its previous mode.

In *constant cooling model* the supply temperature T_{sup}, is constant. This means that the CRAC removes all the heat that is generated in the data center room. Another cooling behavior observed is *instantaneous cooling model*. In this model, the RM module adjusts the cooling load based on the total heat added by the IT equipment and the redline temperatures of the equipment. A different cooling behavior can be studied by replacing the current model with a user-defined cooling model. Moreover, renewable energy sources such as solar heat activated cooling units can be modeled in GDCSim. Such a system can be decomposed into thermodynamic, power, and COP models and included into the simulator.

The output of the simulator module is a compilation of the power model, thermodynamic model, queuing model and cooling model parameters into a log file and providing them as feedback to RM and IOM modules.

IV. Case Studies

In this section we present two case studies with HPC and IDC layouts supporting different server configurations, to demonstrate the functionality of the tool chain.

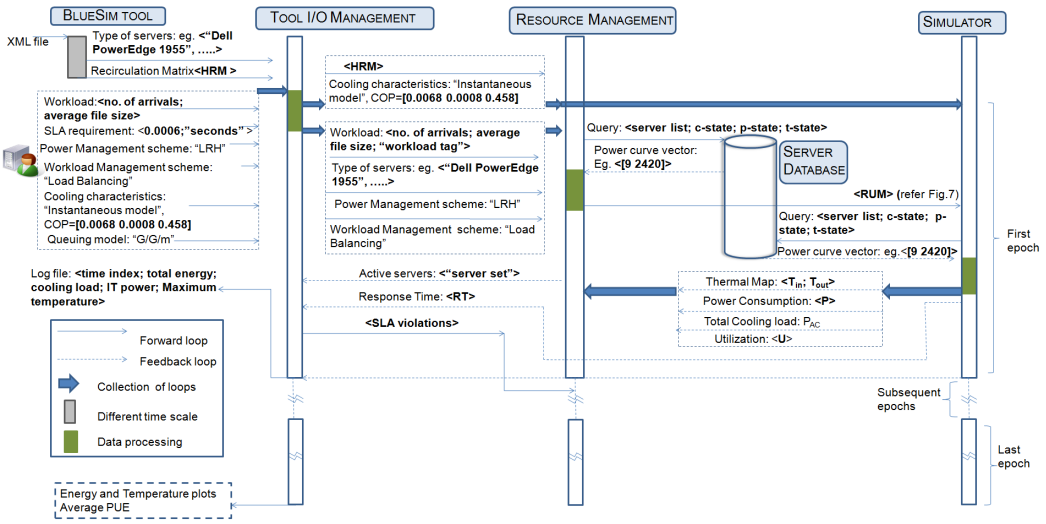

Fig. 5. Event sequence diagram for transactional workload on Internet data centers.

A. Case 1: Transactional workload on Internet data centers

A data center with physical dimensions of 27.6' × 28' × 11.8' was considered. There are two rows of two 35U racks in each row, laid out in a typical hot-aisle/cold-aisle configuration. These racks consist of five 7U chassis each. Cold air is supplied at a flow rate of 5 m^3/s from a single CRAC.

The operation of the tool is illustrated in the event sequence diagram in Fig. 5. The XML based specification was submitted to the BlueSim tool which generated the HRM (Fig. 6) and extracted the type of servers from the specification. The type of servers is an array of strings that gives the model of servers used. The server models along with the following were input to the IOM: (a) a transactional workload in the following format: ⟨no. of arrivals;average file size⟩, (b) SLA requirement input consisting of a floating point number that represents the response time and a string value for the unit: 0.0006;"seconds", (c) LRH [7] power management scheme, (d) load balancing workload management scheme, (e) cooling characteristics input consisting of a string value "constant model" and a COP curve coefficient matrix [0.0068 0.0008 0.458] (The coefficient matrix format is as follows: $C_nT^n + C_{n-1}T^{n-1} + ...C_1T + C_0$, where T is the temperature. If Carnot cycle efficiency for CRAC is assumed, a string "Carnot" is entered in place of the coefficient matrix), and (f) a queuing model represented by string value "G/G/m". The IOM sent the HRM and cooling characteristics to RM and simulator modules.

On arrival of these inputs, the RM module sent a query to the server database as ⟨type of server;c-state;p-state;t-state⟩ and the database returned an array of power curve coefficient vectors, e.g.,⟨[9 2420]⟩. Workload scheduling was carried out and a RUM was generated and sent to simulator module. The simulator module accessed the power curve vectors from the data base and performed simulations to generate: (a) thermal map: ⟨$\mathbf{T_{in}}$; $\mathbf{T_{out}}$⟩, (b) IT power: ⟨**P**⟩ (c) cooling load: P_{AC}, and (d) utilization: ⟨**U**⟩. These were sent back to RM as feedback inputs.

Response time ⟨**RT**⟩ was supplied back to IOM which then calculated SLA violations and reported to RM module. RM module generated an active server set which was supplied to IOM.

The first time epoch was completed when the simulator generated a log file ⟨`time index; total energy; cooling load; IT power; Maximum temperature`⟩ which was sent back to IOM module. This whole process was carried out for all the time epochs till the workload had been completely simulated. Finally, the average ERE and time plots (Fig. 7) for power, average response time, utilization and temperature were generated by IOM from the log file.

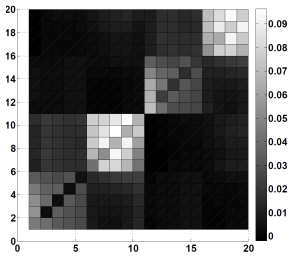

Fig. 6. Gray-scale-coded heat recirculation matrix.

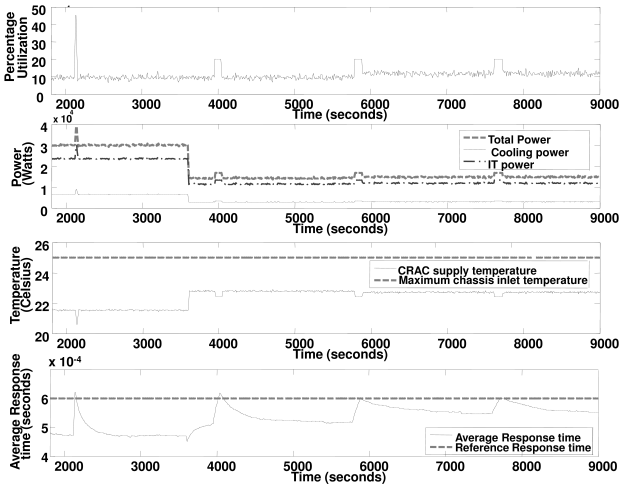

Fig. 7. Utilization, power, temperature and response time for case study 1.

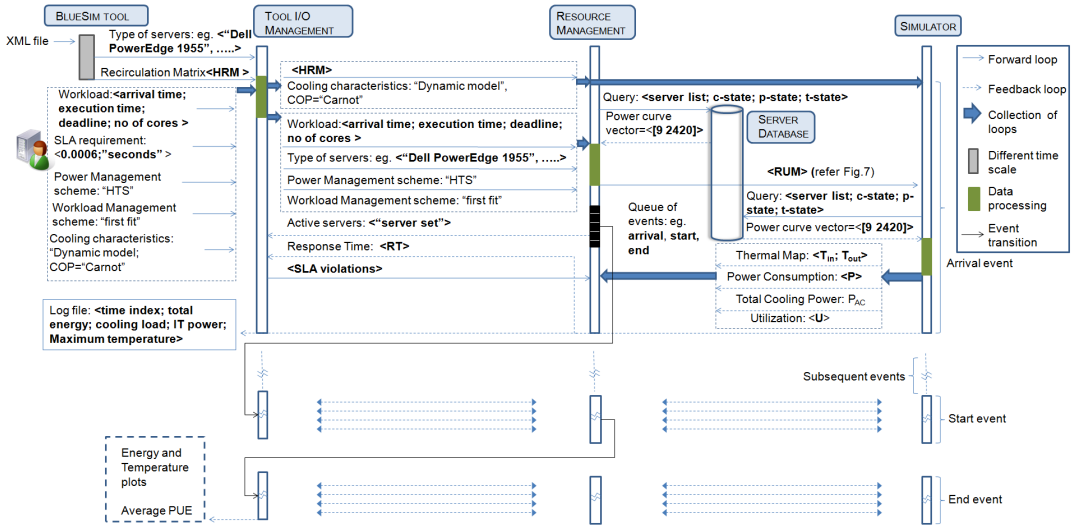

Fig. 8. Event sequence diagram for Event based scheduling in HPC data center.

Further, the tool chain was used to iteratively design a data center and decide the best possible management strategy that ensures better energy efficiency and reduced SLA violations, for given workload conditions. Two data centers, with the same layout as in case study 1, but different server power curves, were used for this process. Each data center configuration was simulated for two different management schemes and the response time and ERE were reported as shown: For data center configuration 1, using power management scheme LRH, the ERE was reported to be 1.26 and for MIP scheme, the ERE was 1.24. In data center configuration 2, an ERE of 1.23 was observed for LRH scheme while for MIP scheme the ERE was 1.22. The workload distribution for all the cases were using load balancing scheme and the mean response time observed was 0.0005 seconds. This shows that management scheme MIP gives the best energy efficiency on data center configuration 2 for the given workload conditions.

B. Case 2: Event based scheduling in HPC data center

A second case study was carried out on a data center with physical dimensions 27.6' × 28' × 11.8'. There are two rows of five industry standard 42U racks in each row, laid out in hot/aisle cold aisle configuration. These racks consists of five 7U chassis each. The CRAC supplies air at a flow rate of 16 m^3/s through the perforated tiles on a raised floor. XML based specification was created and supplied to the BlueSim. The following management schemes were used: (a) power management scheme: HTS scheme [6], (b) workload management scheme: first fit, and (c) cooling characteristics: dynamic cooling model.

The event sequence diagram for this case study is shown in the Fig. 8 A HPC type workload was used for this case study. The input format for such workloads is ⟨`arrival time; start time; deadline; no. of cores required`⟩. The event sequence diagram for this case study is as shown in Fig. 9. A discrete event simulation was carried out, with the events being workload arrival event, start job processing event and end job processing event. An event queue was maintained by the RM module and job scheduling was done at every event. The rest of the simulation was carried out similar to the previous case study and the ERE was reported to be 1.1. Fig. 9 shows the power, temperature and utilization plots.

V. Validation and Discussions

The BlueSim tool was validated by comparing it with the Flovent CFD simulator. The HRM for a particular data center was generated using BlueSim, which was then used to generate the thermal map of the data center for a particular time instant. This scenario was replicated in Flovent and simulations were carried out to generate the thermal map. An average error of 7.36 % was observed.

The simulation results from the case studies demonstrate certain salient features of the tool chain. The modularity of the tool chain allows the BlueSim tool to be used as a standalone data center CFD package. The simulator module can be used in conjunction with any resource management algorithm to estimate data center performance. The generic features of a data center layout is captured by Ciela which provides an easy and intuitive way of data center physical modeling. Moreover, the tool facilitates testing of online resource management algorithms on different data center designs and workload types. As seen from the case studies, thermal aware workload management schemes avoid redlining of servers while improving the efficiency. Further, the tool captures the transient behavior of the data center which results in improved accuracy over steady-state analysis.

Cyber-physical interdependencies are captured using feedback loops. These feedback loops after each time epoch allow the resource management schemes to take informed decisions for the next time epoch. GDCSim provides a scalable platform that can be used to simulate different data center layouts with different number and type of racks, servers and cooling units as shown by the two data center layouts used in the case studies. This feature enables iterative design analysis which helps in design time testing and analysis of different data center configurations before deployment. Extensibility of the

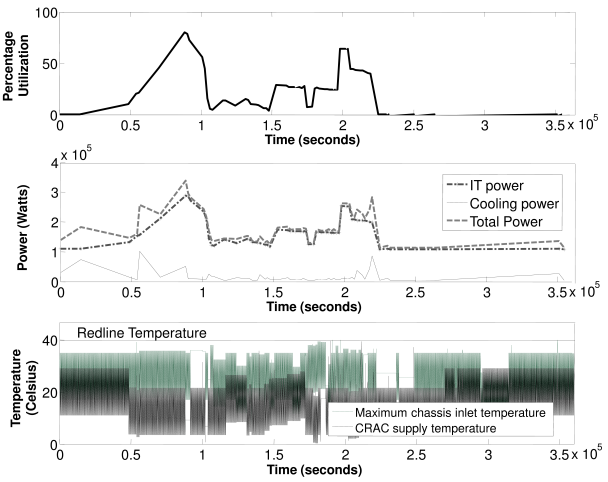

Fig. 9. Utilization, power and temperature plots for case study 2.

tool allows users to add new power consumption, cooling and resource management models. Finally, different modules of the tool chain are interfaced together to enable automation.

The accuracy of the overall system depends on the accuracy of individual models. The models used in resource management have been verified separately in our previous work [5]–[8] . The experimental validation and generation of new models to leverage the functionality of the tool will be left for future work.

VI. Conclusion

In this paper, we developed a simulation tool that captures the inter-dependencies between online resource management schemes and physical behavior of data centers for green data center design. We validated the tool using Flovent, which is a widely used CFD simulation software. We demonstrated the functionality of the tool using two case studies, one each for HPC data centers and Internet data centers with different layouts. The iterative design procedure of a data center was shown by simulating two different configurations of data centers and analyzing their energy efficiency under different online management schemes. Once operational, GDCSim will be publicly available through the BlueTool website, `http://impact.asu.edu/BlueTool/`.

References

[1] U. S. et al., "Cfd-based operational thermal efficiency improvement of a production data center," in *Proceedings of the First USENIX conference on Sustainable information technology*, ser. SustainIT'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 6–6.

[2] A. D. et al., http://www.futurefacilities.com, cooling Path Management for the Mission Critical Facility (MCF): A Case Study.

[3] H. S. et al., "Ultra-low power digital subthreshold logic circuits," in *Proceedings of the 1999 international symposium on Low power electronics and design*. New York, NY, USA: ACM, pp. 94–96.

[4] K. S. et al., "50% active-power saving without speed degradation using standby power reduction(spr) circuit," in *Solid-State Circuits Conference. Digest of Technical Papers. 42nd ISSCC, 1995 IEEE International*, pp. 318 –319.

[5] Q. T. et al., "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, pp. 1458–1472, 2008.

[6] Z. A. et al., "Thermal aware server provisioning and workload distribution for internet data centers," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, pp. 130–141.

[7] T. M. et al., "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol. 53, no. 17, pp. 2888 – 2904, 2009.

[8] A. B. et al., "Integrating cooling awareness with thermal aware workload placement for hpc data centers," *Elsevier Comnets Special Issue in Sustainable Computing (SUSCOM) 2011*.

[9] L. M. et al., "Autonomic power management schemes for internet servers and data centers," in *In IEEE Global Telecommunications Conference (GLOBECOMM*, 2005.

[10] E. P. et al., "Temperature-aware dynamic resource provisioning in a power-optimized datacenter," in *Design, Automation, and Test in Europe*, 2010, pp. 124–129.

[11] J. M. et al., "Weatherman: Automated, online, and predictive thermal mapping and management for data centers," in *In International Conference on Autonomic Computing*, 2006.

[12] L. E. R. et al., "C-oracle: Predictive thermal management for data centers." in *HPCA*. IEEE Computer Society, pp. 111–122.

[13] Y. C. et al., "Integrated management of application performance, power and cooling in data centers," in *NOMS*, 2010, pp. 615–622.

[14] A. G. et al., "Optimal power allocation in server farms," in *SIGMETRICS/Performance*. ACM, 2009, pp. 157–168.

[15] C. D. P. et al., "Ipack2001-15622 computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications."

[16] C. P. et al., "Thermal considerations in cooling large scale high compute density data centers," in *In ITherm 2002 - Eighth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pp. 767–776.

[17] L. M. et al., Applied Math Modeling Inc., Concord, NH, January,2011, using CFD for Data Center Design and Analysis.

[18] C. E. B. et al., "Dynamic thermal management of air cooled data centers," 2006.

[19] T. H. et al., "Mercury and freon: temperature emulation and management for server systems," in *Architectural Support for Programming Languages and Operating Systems*, 2006, pp. 106–116.

[20] F. A. et al., "Joint optimization of idle and cooling power in data centers while maintaining response time," *SIGPLAN Not.*, vol. 45, pp. 243–256, March 2010.

[21] E. P. et al., "Minimizing data center cooling and server power costs," in *International Symposium on Low Power Electronics and Design*, 2009, pp. 145–150.

[22] J. C. et al., "A cfd-based tool for studying temperature in rack-mounted servers," *IEEE Transactions on Computers*, vol. 57, pp. 1129–1142, 2008.

[23] S.-H. L. et al., "Mdcsim: A multi-tier data center simulation, platform," in *Cluster Computing and Workshops. IEEE International Conference on*, 31 2009-sept. 4, pp. 1 –9.

[24] D. M. et al., "Towards a scalable data center-level evaluation methodology," in *Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on*, pp. 121 –122.

[25] G. V. et al., "Trends and effects of energy proportionality on server provisioning in data centers," in *International Conference on High performance Computing Conference*, Dec. 2010.

[26] M. L. et al., "Dynamic right-sizing for power-proportional data centers," in *Proc. IEEE INFOCOM*, Shanghai, China, 10-15 Apr 2011. [Online]. Available: http://netlab.caltech.edu/lachlan/abstract/RightSizing.pdf

[27] http://www.thegreengrid.org.

[28] Q. T. et al., "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters," in *Intelligent Sensing and Information Processing, 2006. Fourth International Conference on*, pp. 203 –208.

[29] S. K. S. G. et al., "Research directions in energy-sustainable cyber-physical systems," *Elsevier Comnets Special Issue in Sustainable Computing (SUSCOM) 2011 Invited paper (Accepted for publication)*.

[30] A. H. et al., "A sustainable data center with heat-activated cooling," in *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm),12th IEEE Intersociety Conference on*, 2010, pp. 1 –7.

[31] C. G. et al., "Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, 2009.

[32] T. M. et al., "Model-driven coordinated management of data centers," *Computer Networks*, vol. 54, no. 16, pp. 2869 – 2886, 2010, managing Emerging Computing Environments.