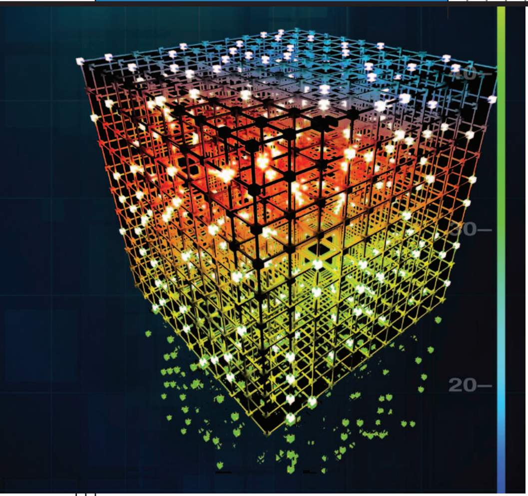

SimWare: A Holistic Warehouse-Scale Computer Simulator

Sungkap Yeo and Hsien-Hsin S. Lee, *Georgia Institute of Technology*

To optimize datacenter energy efficiency, SimWare analyzes the power consumption of servers, cooling units, and fans as well as the effects of heat recirculation and air supply timing. Experiments using SimWare show a high loss of cooling efficiency due to the nonuniform inlet air temperature distribution across servers.

Cloud computing has emerged as a cost-effective way to meet both enterprise and consumer demands. More and larger warehouse-scale computers are being built to support this paradigm shift, but constructing and operating such systems is expensive. In addition to the high infrastructure cost, a datacenter consumes considerable energy.

Much of this energy goes to cooling the servers. Legacy systems can use more than half of their total power for cooling, which is highly inefficient. State-of-the-art datacenters can operate at higher discharge temperatures, thereby using less than 10 percent of their total power for cooling. But the perceived energy savings can be deceptive.

Researchers have proposed various ways to measure datacenter efficiency. Power usage effectiveness[1]—the ratio of total facility power consumed to power delivered to computing equipment—is a common metric, but PUE ignores the nonnegligible impact of fan energy consumption.[2] Decreasing the power that the computing room air conditioners (CRACs) use leads to an increase in room temperature. This in turn causes the server fans to blow harder and consume more power, resulting in a misleadingly low PUE. In fact, the energy cost of the higher fan rotation speed can eventually overwhelm the energy savings from reducing CRAC power consumption. To address this problem, James Hamilton proposed a revised metric, total PUE, that factors fan power consumption out of the useful server power.[3] All in all, to accurately quantify datacenter energy efficiency, it is important to include the energy use of *all* components.[4]

Numerous software tools that simulate datacenters are available but exclude critical parameters. For example, CloudSim[5] and DCSim[6] do not consider the effects of increased fan power use and heat recirculation, and others largely ignore the air travel time from CRACs to servers.[4,7-9] To overcome these shortcomings, we developed SimWare, a warehouse-scale computer simulator with detailed temperature, power, and performance models for servers and CRACs that also models the impact of heat recirculation and air supply timing.

MOTIVATION

Many studies of datacenter energy efficiency[4-9] have not considered the impact of temperature on server power use. In general, a server operating at a higher temperature consumes more power. To explore this relationship, we ran the Linpack benchmark using a Xeon 5160 server and measured the total system power consumption, fan power consumption, fan speed, and processor die temperature at different inlet air temperatures.

As Figure 1a shows, as the inlet air temperature increased, total system power consumption increased because of the rise in fan power consumption. To understand why fans consume more power when the server operates at a higher temperature, assume that a processor die's temperature is 70°C. When the inlet air temperature is 10°C, the difference between them is 60°C. However, when the surrounding temperature is 40°C, the difference becomes 30°C. The latter situation requires twice as much air to cool as the former and so the fan must rotate twice as fast.

 Published by the IEEE Computer Society

Figure 1b shows that as the inlet air temperature increased, the fan speed steeply increased, but the processor die's temperature remained the same until the inlet air temperature reached 92°F (~33.3°C), just over the 89.6°F (32°C) emergency temperature of an A1-class server.[10]

Prior studies have ignored changes in fan power consumption, which accounts for 10 to 30 percent of total system power consumption.[11] Assuming constant fan power will result in overly optimistic results. Many proposed techniques for saving cooling energy leave more servers at a higher inlet air temperature than the baseline.[7-9] Although this saves significant energy in CRACs, the impact on server power must be carefully evaluated.

Previous studies have also disregarded the time it takes cool air to travel from a CRAC to a server. Without considering this factor, a datacenter could easily maximize power savings simply by setting a CRAC to raise the supply air temperature ($T_{\text{supply air}}$) until any server's inlet air temperature ($T_{\text{inlet air}}$) equals its emergency temperature ($\forall T_{\text{inlet air}} = T_{\text{emergency}}$). In reality, however, the time delay occasioned by the flow of cool supply air from the CRAC to the server to keep $T_{\text{inlet air}}$ from exceeding $T_{\text{emergency}}$ necessitates the inclusion of a temperature safety margin ($T_{\text{safety margin}}$) when raising $T_{\text{supply air}}$, which results in a loss of cooling efficiency.

SIMWARE OVERVIEW

Figure 2 provides an overview of SimWare, which supports different types of utilization traces as input and generates performance-, power-, and temperature-related statistics.

The simulator models power at both the datacenter and server levels. The datacenter-level model estimates

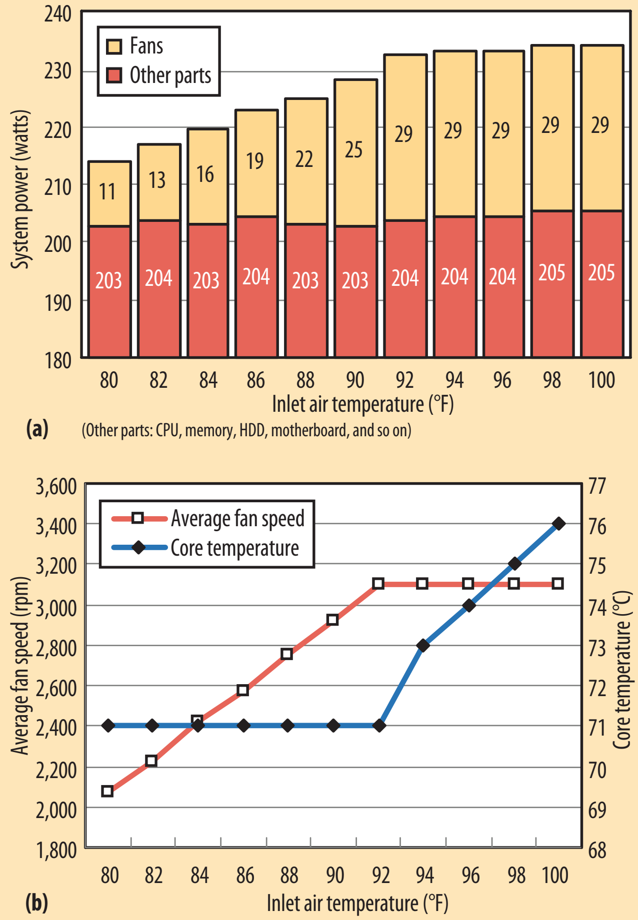

Figure 1. Inlet air temperature versus (a) total system power and fan power consumption and (b) fan speed and processor die temperature. The server consumed more power at a higher temperature primarily due to increased fan power use.

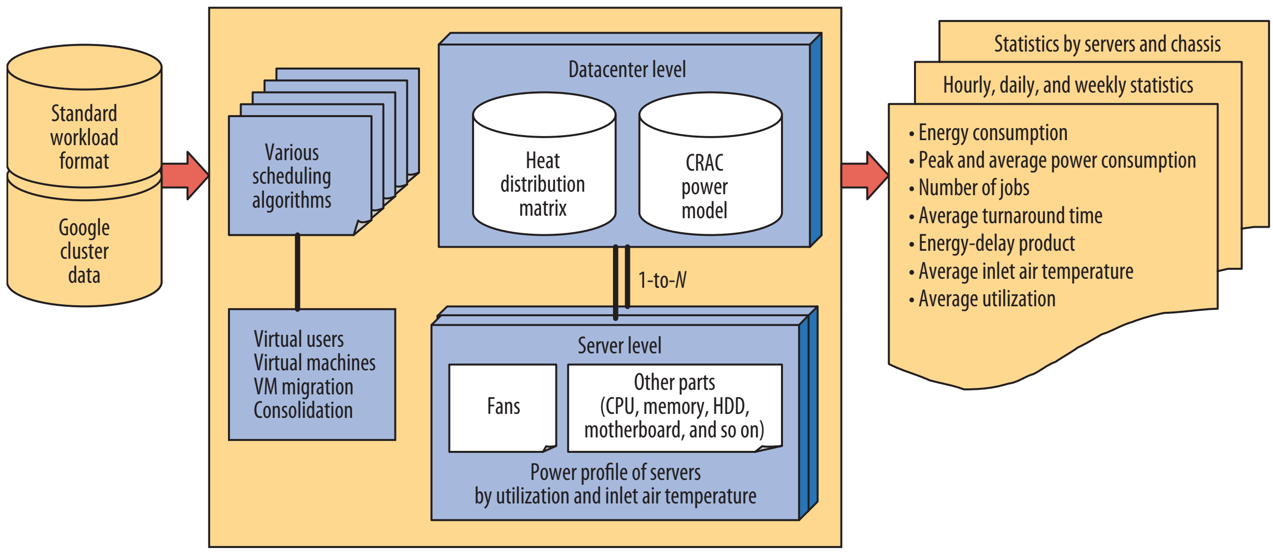

Figure 2. SimWare overview. The simulator models power at both the datacenter and server levels.

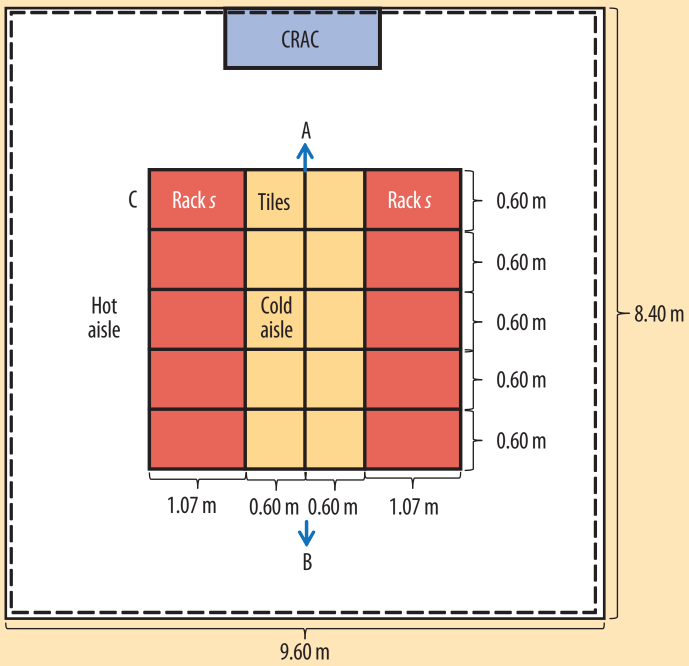

Figure 3. Layout of a raised-floor warehouse-scale computer.

heat flow using a heat distribution matrix (HDM)[12] and determines CRAC power consumption using the approach developed by Justin Moore and colleagues.[7] The server-level model estimates the power consumption of fans along with other components in terms of utilization and $T_{\text{inlet air}}$—that is, the thermal impact on server power.

Unlike other datacenter simulators, SimWare takes into account the air travel time from CRACs to servers. In addition, it can evaluate user-defined job-scheduling algorithms as well as virtual machine-related tasks.

Thermal impact on server power

In modeling the thermal impact on server power, we rely on the *law of convective heat transfer* and the *laws of fan affinity*.

The law of convective heat transfer states that heat transfer (in watts) is directly proportional to the amount of air and the temperature difference between the cooling object and surrounding air:

$$\text{Heat transfer (watts)} \propto \text{Temperature difference} \times \text{Amount of air} \qquad (1)$$

For simplicity, we assume that the density of air is constant at the temperature range of interest.

The laws of fan affinity define the relationship of the rotational speed, the amount of air, and the fan's power:

$$\text{Amount of air} \propto \text{Fan}_{\text{rpm}} \qquad (2)$$

$$\text{Fan}_{\text{power}} \propto \text{Fan}^3_{\text{rpm}} \qquad (3)$$

We first assume that a CPU's power consumption remains constant while the surrounding temperature increases from $T_{\text{inlet air}}$ to $T_{\text{inlet air}} + \alpha$. Meanwhile, we keep heat transfer constant. When the surrounding temperature changes from $T_{\text{inlet air}}$ to $T_{\text{inlet air}} + \alpha$, the initial temperature difference (ΔT) between the CPU and the surrounding air decreases to $\Delta T - \alpha$.

In Equation 1, when the temperature difference decreases by $(\Delta T - \alpha)/\Delta T$ times, the amount of air must increase by $\Delta T/(\Delta T - \alpha)$ times to maintain constant heat transfer. As Equation 2 indicates, to supply $\Delta T/(\Delta T - \alpha)$ times more air, the fan must rotate $\Delta T/(\Delta T - \alpha)$ times faster. As a result, the increased fan speed consumes $(\Delta T/(\Delta T - \alpha))^3$ times more power, according to Equation 3.

These laws make it possible to calculate a fan's power consumption relative to the CPU's power consumption and $T_{\text{inlet air}}$. Once the boundary conditions are defined, SimWare can calculate the exact power that a fan consumes.

Air travel time from CRACs

Several factors affect the time it takes air to flow from a CRAC to a server, including the datacenter layout, the proximity of the CRAC to the server, the air velocity discharged from the CRAC, and the plenum's height.

Using these physical parameters, we developed a thermodynamics-based scheme to estimate the air travel time. Our simulations revealed that a longer air travel time results in lower cooling efficiency. Therefore, to determine the lower bound of the air travel time's impact, SimWare estimates the fastest possible travel time.

In our simulations, a CRAC unit discharges 8 m^3/s of cool air into the plenum, indicated by the dotted line in Figure 3. Once this air has filled and pressurized the plenum, 0.6 m × 0.6 m tiles discharge the air into the room above. SimWare calculates how long this takes by dividing the plenum's volume by the discharge rate. For simplicity, the simulator assumes the most optimistic scenario, in which all the tiles discharge the same amount of air.

In reality, some of the cool air supplied to the room would bypass the servers and flow in the direction of A and B in Figure 3, but, again for simplicity, SimWare assumes that the supply air only fills up the volume above the tiles, or the cold aisle.

Calculated results indicate that it takes the cool air about six seconds to reach the servers at the bottom of the racks and seven seconds to reach the servers at the top. In real scenarios, tiles near the CRAC supply less cool air. Because cooling down certain servers such as C in Figure 3

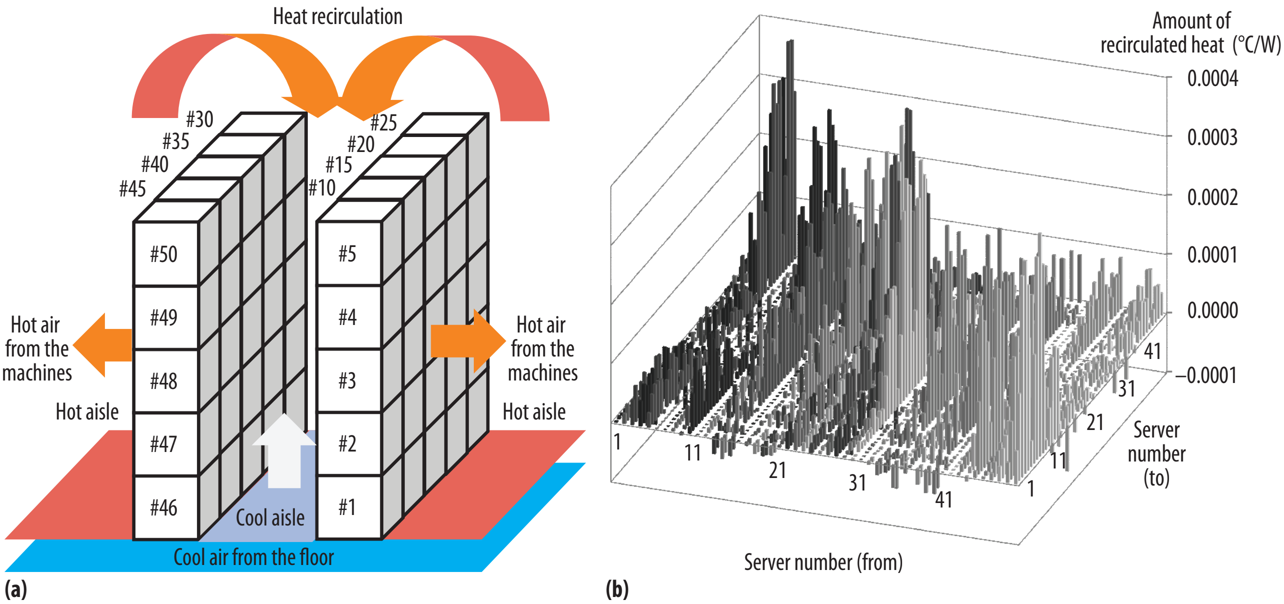

Figure 4. Generating a heat distribution matrix: (a) 3D view of the configuration of a simulated warehouse-scale computer and (b) its HDM.

is more difficult, the CRAC usually lowers the $T_{\text{supply air}}$ with a larger $T_{\text{safety margin}}$, thereby reducing cooling efficiency.

Heat distribution matrix

Building a datacenter simulator that integrates heat flow with temperature, power, and performance in one infrastructure was impractical because modeling recirculated heat as workload utilization changes requires a prohibitive amount of computation. SimWare mitigates this problem by employing heat flow as an HDM.[12]

Generating a datacenter's HDM requires a tool capable of complex computational fluid dynamics simulations such as Arizona State University's BlueTool (http://impact.asu.edu/BlueTool). Nevertheless, the concept is simple: an HDM converts the heat generated by a particular server into an increase in the temperature of all other servers.

For example, if a datacenter has 10 servers, the size of the HDM will be 10 × 10. The first row of the matrix represents how much one server's $T_{\text{inlet air}}$ is affected by the heat generated by all 10 servers. Matrix multiplication of the first row by the power consumption of all the servers will produce the first server's $T_{\text{inlet air}}$. In other words, each cell (i, j) of the HDM indicates the contribution of server j to the temperature increase of server i.

Figure 4 shows the configuration and HDM of a reference datacenter with a 50-blade chassis. In Figure 4b, server 50 has tall bars for servers 1-10, indicating that the heat generated by server 50 is more likely to recirculate to servers 1-10 than to the others.

SimWare uses the HDM to calculate each server's $T_{\text{inlet air}}$, which varies by server location because of heat recirculation. The HDM takes this effect into account by converting the impact of power consumption (in watts) to the temperature difference (in °C) between one server and the other servers. Unfortunately, an HDM does not model changes in convective flows as a consequence of variable fan speeds; it assumes that airflow patterns are temperature invariant, which could lead to temperature estimation errors in some datacenter geometries.

CRAC power consumption

Moore and colleagues[7] showed that the power CRACs require in a typical datacenter can be represented as a function of their $T_{\text{supply air}}$ and the amount of heat that must be removed:

$$\text{Power drawn from CRACs} = \frac{\text{Heat to remove (power drawn from servers)}}{0.0068T^2_{\text{supply air}} + 0.0008T_{\text{supply air}} + 0.458} \qquad (4)$$

SimWare uses this equation to calculate CRAC power consumption. When $T_{\text{supply air}}$ is 10°C, the denominator is about 1 and the CRACs consume the same amount of power as the servers. However, if the CRACs produce a higher $T_{\text{supply air}}$, the denominator increases, and they consume less power while removing the same amount of heat. When the CRACs increase $T_{\text{supply air}}$ to 20°C, they consume only one-third of the servers' power.

Input and putput

For input traces, SimWare currently supports Standard Workload Format (SWF) files and Google cluster data.

Several utilization traces in SWF collected from massively parallel processing systems and experimental datacenters are available at www.cs.huji.ac.il/labs/parallel/workload/swf.html. Based on ASCII, each line of an SWF file describes a submitted job and contains the job ID, the submitted time, the runtime, the number of allocated processors, the average CPU time used, and the dependency between jobs. Google cluster data (http://code.google.com/p/googleclusterdata) contains similar records collected from the company's warehouse-scale computers.

Once a simulation finishes, SimWare generates performance-, power-, energy-, and temperature-related data including job turnaround time (important for latency-sensitive Internet applications[13,14]), peak/average server and CRAC power consumption, energy use for a given time frame, and the current configuration's energy-delay product. SimWare also outputs the average room temperature, average temperature by server chassis, and utilization level.

Once a simulation finishes, SimWare generates performance-, power-, energy-, and temperature-related data.

Chassis and servers

SimWare currently simulates a warehouse-scale computer consisting of 500 blade servers in 50 chassis, with 10 servers per chassis. Each server has a 130-W TDP Xeon E7-2850 processor with 10 cores, resulting in a total of 5,000 cores. SimWare is not limited to this physical layout—it can simulate any size datacenter as long as an HDM can be generated for it.

Excluding the fans, each blade server consumes 260 W when fully loaded and consumes half of its peak power when idle.[15] The fan on the CPU heatsink must remove heat generated by the CPU at any time. Therefore, when the fan runs at its maximum speed, it should remove 130 W (the maximum CPU power) at $T_{emergency}$ (the highest operable temperature).

At this operating point, SimWare assumes that the fan consumes 15 W and runs at 3,000 rpm. Each server has two other fans with the same specification at the front and back panel. The rotational speed of these case fans is directly proportional to the server's power consumption and $T_{inlet\ air}$. SimWare also assumes that the fans cannot be turned off and that they run at 500 rpm when the server is idle.

The servers' $T_{emergency}$ is set to 30°C, which meets the A1-class server specification for datacenters.[10] SimWare assumes that the goal of fan control is to save fan power and keeps the die temperature lower than 70°C for reliability.

CRAC control policies

SimWare currently supports two CRAC control policies: *constant* and *dynamic*.

Constant control—supplying cool air at a constant temperature—is the most basic strategy. With this policy, $T_{supply\ air}$ is low enough to ensure that all servers stay below $T_{emergency}$ at any time. Because the cooling power is constant and set to the worst-case scenario, this policy wastes cooling power when the datacenter is underutilized.

To tackle this inefficiency, researchers have proposed dynamic control policies.[7,16] In SimWare, the CRAC first supplies the lowest possible $T_{supply\ air}$ and then gradually raises it at the rate of 0.01°C/sec until any server reaches a trigger temperature ($T_{trigger}$). When any server's $T_{inlet\ air} = T_{trigger}$, the CRAC starts to lower $T_{supply\ air}$ at the same rate.

In the ideal case, $T_{trigger}$ can be set as high as $T_{emergency}$—that is, the CRAC continues to raise $T_{supply\ air}$ until any server reaches $T_{emergency}$. However, due to the timing delay of the CRAC to lower $T_{inlet\ air}$, using $T_{trigger} = T_{emergency}$ as a condition will cause some servers to operate above $T_{emergency}$. A dynamic control policy thus requires a safety margin ($T_{trigger} = T_{emergency} - T_{safety\ margin}$), which leads to cooling inefficiency.

EXPERIMENTAL EVALUATION

We used SimWare to run an SWF workload from 10 high-performance computing clusters in the Shared Hierarchical Academic Research Computing Network in Canada. The SHARCNET log contains nearly 1.2 million accounting jobs from December 2005 through January 2007. We omitted results from the other trace files due to their similarity.

The black graph in Figure 5a plots the simulated system's daily utilization. For days 0-50, the average utilization is less than 1 percent. For days 50-150, the workload is moderate, with an average utilization of 5.3 percent and a maximum utilization of 44.3 percent. The workload is heavy for the remaining days, with an average utilization of 71.3 percent.

Figure 5a also shows the simulated system's CRAC and server power consumption. Total power consumption generally tracks the utilization level except when the system is underutilized. Because SimWare assumes that servers consume half of their peak power when idle, the system is not energy-proportional.[15]

In addition, Figure 5a plots the normalized latency of submitted jobs. Because SHARCNET has more than 7,000 cores and the simulated system has 5,000 cores, normalized latency drastically increases when the latter is at high utilization.

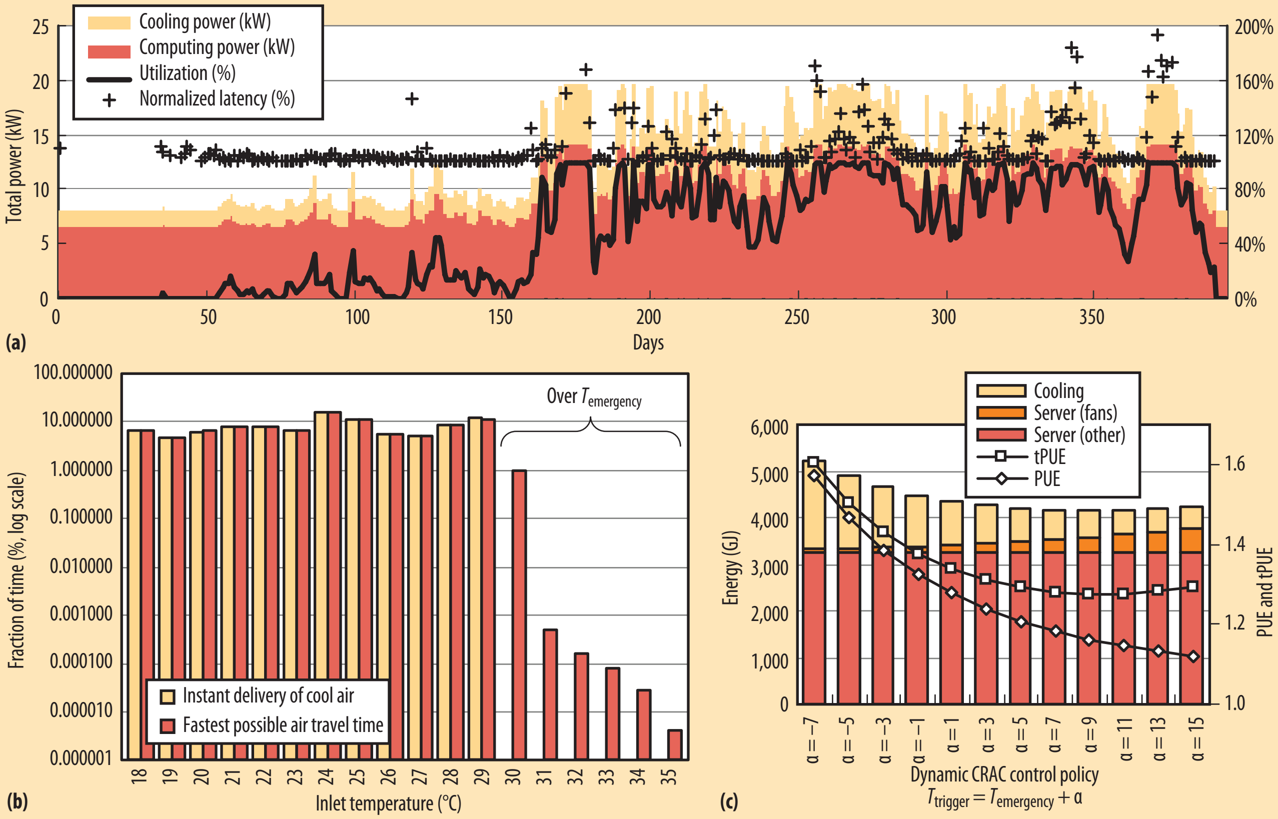

Figure 5. SimWare simulation of a Standard Workflow Format (SWF) workload from SHARCNET: (a) power consumption, utilization, and latency; (b) effect of air travel time; and (c) energy usage breakdown, power usage effectiveness (PUE), and total PUE (tPUE).

To evaluate the importance of air travel time, we ran simulations with two different configurations: one that assumes the cool air from the CRACs instantly lowers the servers' $T_{inlet\ air}$ (zero air travel time), and one that assumes the fastest possible air travel time. These two simulations shared all other parameters.

Figure 5b shows the distribution of $T_{inlet\ air}$ for all servers, with the bars representing the fraction of time that servers spend at a given $T_{inlet\ air}$. In the case of instant delivery of cool air, all the servers operate under $T_{emergency}$ (30°C). However, with nonzero air travel time, servers experience $T_{inlet\ air}$ over $T_{emergency}$, up to 35°C. Therefore, to ensure that $\forall T_{inlet\ air} < T_{emergency}$ at any time, a dynamic CRAC control scheme must secure a safety margin.

Even with the most optimistic air travel time, when $T_{trigger} = T_{emergency}$, one of the servers is above $T_{emergency}$ more than 49 percent of the time. However, if $T_{trigger} = T_{emergency} - 1$, all servers operate below $T_{emergency}$ 99.99 percent of the time. To make it 100 percent, $T_{trigger}$ must be as low as $T_{trigger} = T_{emergency} - 7$. Moreover, when $T_{trigger} = T_{emergency} - 7$, the average $T_{supply\ air}$ is 14.7°C, close to the typical outlet air temperature of CRACs.[8,17,18]

Figure 5c illustrates how much energy this safety margin costs. The bars represent server and cooling energy consumption for a given CRAC control policy. Each policy has the same algorithm but uses a different $T_{trigger}$ value. For example, the leftmost bar indicates that total energy consumption is slightly more than 5,000 gigajoules when $T_{trigger} = T_{emergency} - 7$.

Comparing the bars for $\alpha = -1$ and $\alpha = -7$, cooling energy consumption increases from 1,100 GJ to 1,900 GJ. The safety margin in this case thus costs about 800 GJ (≈73 percent). Cooling energy decreases with a larger α or higher room temperature, but server fans now consume more energy, negating the savings. Consequently, total energy consumption saturates at $\alpha = 9$. Although $\alpha > 9$ does not result in any energy savings, it can lead to a lower PUE. From $\alpha = 11$ to $\alpha = 15$, servers consume more energy while cooling units consume less. Consequently, PUE monotonically decreases regardless of total energy consumption. On the contrary, because tPUE[3] factors fan power out of the useful server power, a smaller tPUE guarantees maximum energy efficiency.

In general, the heat recirculation effect and the air travel time from CRACs to servers result in two types of inequality among servers. First, because hot air tends to circulate upward, top-rack servers typically have a higher $T_{\text{inlet air}}$ than lower-rack servers. In our simulation, the difference between the highest and lowest $T_{\text{inlet air}}$ among servers was 8.1°C. In other words, most servers are overcooled because the CRACs must lower the $T_{\text{supply air}}$ enough to ensure that $T_{\text{inlet air}} < T_{\text{emergency}}$ for every server. Second, due to air travel time, some servers require more time to cool down depending on their location. Because CRACs must set a safety margin based on the worst-case scenario, these two types of inequality among servers reduce the cooling system's efficiency.

To address this problem, we suggest implementing heterogeneous cooling capacities among datacenter servers. If top-rack servers have better cooling capacities and a higher $T_{\text{emergency}}$, the CRACs can safely discharge air at a higher temperature by using aggressive dynamic control policies.

We adopted this approach in the simulated sysyem. We selected 11 blade chassis with the highest average $T_{\text{inlet air}}$ and changed their $T_{\text{emergency}}$ from 30°C to 35°C. Using a dynamic CRAC control policy of $T_{\text{trigger}} = T_{\text{emergency}} - 2$ saved 37 percent more cooling energy than the baseline policy of $T_{\text{trigger}} = T_{\text{emergency}} - 7$ without compromising thermal guidelines.

By modeling all cooling units, computing nodes, and heat recirculation for a warehouse-scale computer, SimWare provides a holistic simulation-based infrastructure to help datacenter designers evaluate energy-saving policies more accurately and effectively. Using real utilization traces, it can also reveal new insights about cooling efficiency and enables further energy optimization opportunities that exploit the inequality in inlet temperatures and air traveling paths. Finally, in conjunction with tools that generate heat distribution matrices, SimWare can assess mechanical design options such as server placement, airflow management, and CRAC control strategies.

To promote our holistic simulation methodology and enable more green datacenter design activity, we have made SimWare freely available to researchers at http://arch.ece.gatech.edu/simware.html. C

References

1. C. Belady et al., eds., *The Green Grid Data Center Power Efficiency Metrics: Power Usage Effectiveness and DCiE*, white paper no. 6, The Green Grid, 2007.
2. R. Ayoub, R. Nath, and T. Rosing, "JETC: Joint Energy Thermal and Cooling Management for Memory and CPU Subsystems in Servers," *Proc. 18th Int'l Symp. High-Performance Computer Architecture* (HPCA 12), IEEE, 2012, pp. 299-310.
3. J. Hamilton, "Where Does the Power Go in High-Scale Data Centers?", keynote address, Usenix Ann. Technical Conf. (Usenix 09), Usenix, 2009; www.usenix.org/conference/usenix-09/keynote-address-where-does-power-go-high-scale-data-centers.
4. S. Pelley et al., "Understanding and Abstracting Total Data Center Power," *Proc. Workshop Energy-Efficient Design* (WEED 09), ACM, 2009; http://web.eecs.umich.edu/~spelley/Publications/weed09.pdf.
5. R.N. Calheiros et al., "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software—Practice & Experience*, Jan. 2011, pp. 23-50.
6. M. Tighe et al., "DCSim: A Data Centre Simulation Tool for Evaluating Dynamic Virtualized Resource Management," to appear in *Proc. 6th Int'l DMTF Academic Alliance Workshop Systems and Virtualization Management* (SVM 12), IEEE, 2012; http://digs.csd.uwo.ca/digs/administrator/components/com_jresearch/files/publications/tighe2012.pdf.
7. J. Moore et al., "Making Scheduling 'Cool': Temperature-Aware Resource Assignment in Data Centers," *Proc. Usenix Ann. Technical Conf.* (Usenix 05), Usenix, 2005, pp. 61-75.
8. R.K. Sharma et al., "Balance of Power: Dynamic Thermal Management for Internet Data Centers," *IEEE Internet Computing*, Jan./Feb. 2005, pp. 42-49.
9. Q. Tang, S.K.S. Gupta, and G. Varsamopoulos, "Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach," *IEEE Trans. Parallel and Distributed Systems*, Nov. 2008, pp. 1458-1472.
10. ASHRAE Technical Committee 9.9, *Thermal Guidelines for Data Processing Environments—Expanded Data Center Classes and Usage Guidance*, white paper, Am. Soc. Heating, Refrigerating and Air-Conditioning Engineers, 2011; www.eni.com/green-data-center/it_IT/static/pdf/ASHRAE_1.pdf.
11. D. Meisner, B.T. Gold, and T.F. Wenisch, "PowerNap: Eliminating Server Idle Power," *Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 09), ACM, 2009, pp. 205-216.
12. Q. Tang et al., "Sensor-Based Fast Thermal Evaluation Model for Energy-Efficient High-Performance Datacenters," *Proc. 4th Int'l Conf. Intelligent Sensing and Information Processing* (ICISIP 06), IEEE, 2006, pp. 203-208.
13. J. Mars et al., "Bubble-Up: Increasing Utilization in Modern Warehouse Scale Computers via Sensible Co-locations," *Proc. 44th Ann. IEEE/ACM Int'l Symp. Microarchitecture* (MICRO 11), IEEE, 2011, ppp. 248-259.
14. D. Meisner et al., "Power Management of Online Data-Intensive Services," *Proc. 38th Ann. Int'l Symp. Computer Architecture* (ISCA 11), ACM, 2011, pp. 319-330.
15. L.A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *Computer*, Dec. 2007, pp. 33-37.
16. C.E. Bash, C.D. Patel, and R.K. Sharma, "Dynamic Thermal Management of Air Cooled Data Centers," *Proc. 10th Intersociety Conf. Thermal and Thermomechanical Phenomena in Electronic Systems* (ITherm 06), IEEE, 2006, pp. 445-452.
17. F. Ahmad and T.N. Vijaykumar, "Joint Optimization of Idle and Cooling Power in Data Centers While Maintaining Response Time," *Proc. 15th Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 10), ACM, 2010, pp. 243-256.
18. C.D. Patel et al., "Thermal Considerations in Cooling Large Scale High Compute Density Data Centers," *Proc. 8th Intersociety Conf. Thermal and Thermomechanical Phenomena in Electronic Systems* (ITherm 02), IEEE, 2002, pp. 767-776.

Acknowledgments

The authors thank Tridib Mukherjee and Rose Robin Gilbert at Arizona State University for technical discussions about heat distribution matrices. This research was supported in part by NSF grant CNS-0644096.

Sungkap Yeo *is a PhD student in the School of Electrical and Computer Engineering at Georgia Institute of Technology. His research interests include power management for datacenters and microarchitectural support for the future cloud computing environment. Contact him at sungkap@gatech.edu.*

Hsien-Hsin S. Lee *is an associate professor in the School of Electrical and Computer Engineering at Georgia Institute of Technology. His current research investigates the interaction and design synergy among data-centric applications, computer architecture, physical design, and emerging technologies such as 3D integrated circuits and novel nonvolatile memory. Lee has a PhD in computer science and engineering from the University of Michigan at Ann Arbor. He is a senior member of IEEE and ACM. Contact him at leehs@gatech.edu.*

Selected CS articles and columns are available for free at http://ComputingNow.computer.org.