Minimizing Data Center Uninterruptable Power Supply Overload by Server Power Capping

Fawaz AL-Hazemi[1], *Senior, IEEE*, Josip Lorincz[2], *Senior, IEEE* and Alaelddin F.Y. Mohammed[3], *Member, IEEE*

***Abstract*— Recent analyses show an increase in data center (DC) service downtime due to the increase in power supply outages caused by overloaded uninterruptable power supplies (UPSs). As a solution to the UPS overload problem, this paper presents developed a power distributor manager (PDM) that caps the power consumption of the DC by enforcing a power restriction on the running servers based on a UPS power minimization cost function. The PDM proposed controls the power consumption per UPS in a DC by means of processor time-sharing and dynamic frequency scaling, which eliminates UPS overloading problems and thus reduces server outages.**

***Index Terms*—optimization, DC, DFS, power supply management, power capping, servers, virtual machines, UPS**

I. INTRODUCTION

Recently, Uptime Institute has released its ”Data center Survey Results (2018)” showing a significant increase in data center (DC) outages due to on-premise power failure [1]. The survey reported that on-premise power outages account for 36% of all failures, and compared with certain other causes of DC outages (e.g., 25% for network issues and 22% for IT/software issues), this type of failure dominates. An on-premise power outage is a consequence of the inability of uninterruptable power supplies (UPSs) as one of the main DC power supply elements to ensure a redundant power supply source. Examples of power outage impacts on DC operation and its implications for humanity can be found in different industry sectors, the most prominent being the airline industry [2]. Industries such as airlines rely on back-end DCs for their core business (airport operations and aviation), and power failures in such DCs result in service outages. Recently, Delta Airlines (August 2016) and British Airways (May 2017) experienced DC power supply outages due to UPS overload failures, suspending hundreds of flights and thousands of travelers from aviation operation [2].

In the cloud computing era, Internet DCs are growing rapidly in terms of server capacity. Frequently, these server capacity upgrades are not followed with corresponding improvements in power supply infrastructure, which is a consequence of the cost of investments in such infrastructure, ranging from tens to hundreds of millions of U.S. dollars [3]. Such reasoning is in line with the power oversubscription concept that has been recently proposed as a solution for DC capital expense reduction and is based on deferment of expensive power supply infrastructure upgrades to future years [4]. More specifically, the power oversubscription concept regards provisioning more servers on the existing power supply infrastructure of a DC. Such a concept is based on the assumption that all the servers do not have power demand peaks at the exact same time [4, 5].

Hence, the power oversubscription concept enables the hosting of many more servers than the nominal capacity of the DC power supply infrastructure can accommodate, without the need for immediate upgrades of the power supply infrastructure [3]. In addition to the power oversubscription concept, a novel energy-efficient scheme has been proposed for DCs based on dynamic on/off UPS switching during server consolidation [6]. The concept proposes to improve DC energy efficiency by continuously adapting UPS activity states according to a number of active servers and corresponding virtual machines (VMs).

Obviously, a drawback of the power oversubscription approach and UPS switching concept is the increased chance of UPS overload, which may lead to UPS failures and consequently to undesired server or even DC outages. Certain attempts that try to address this problem are based on the use of additional energy storage devices (ESDs) as backup energy enablers [7, 8]. Nevertheless, solutions based on ESDs do not consider ESDs overload, which can cause power supply outages. To ensure appropriate power supply levels in the case of utility grid outages, power supply capping should be done to dynamically control the power demand of DC equipment that must be within a UPS output power rating.

A considerable amount of existing research related to power capping in DCs has been conducted at different levels such as the server level [9, 10], rack enclosure level [11, 12], and DC level [13, 14]. However, at the UPS level, to the best of our knowledge, there is no existing work that proposes dynamic power capping of servers power demand according to the maximal power rating of the UPS used for the redundant power supply of the corresponding servers.

Firstly, the main contribution of this letter is, therefore, the introduction of a novel approach based on a power distributor manager (PDM) that caps the power consumption of a set of servers sharing the same power delivery path (i.e., UPS). The PDM solution proposed considers the resource demands of the servers and related power consumption and distributes the overall UPS power budget among servers and corresponding VMs. Secondly, a novelty of the proposed approach is that power distribution is modeled as a linear integer optimization problem solved separately per each UPS in a DC. The experimental results obtained show that by using the developed PDM solution, the UPS will not be overloaded at any moment, which eliminates UPS failures and consequently server disruption.

The rest of the paper is organized as follows: In Section 2, the power optimization model and design of the proposed PDM are presented. Section 3 describes an experimental setup used for practical evaluation of the proposed solution. The evaluation results obtained concerning the performance of the PDM are presented and discussed in Section 4. Finally, certain concluding remarks are given in Section 5.

II. POWER DISTRIBUTION MODEL

A. Optimization model for power capping

Although UPSs in DCs are deployed in a redundant configuration, increases in installed server capacities over time can compromise the initial UPS redundancy, and each piece of power supply equipment can become overloaded. More precisely, UPS overload can manifest during main power supply failure, when one piece of power supply equipment fails or when the UPS accepts the additional load, since the other UPSs are switched off in the process of dynamic on/off UPS switching [6].

To model such a UPS overloading problem in a DC, a set

KAIST, South Korea, (e-mail: fawaz@kaist.ac.kr[1], alaelddin@kaist.ac.kr[3])

FESB, University of Split, Croatia, (e-mail: josip.lorincz@fesb.hr[2])

TABLE I: Overview of variables, parameters and sets used in the analyses

	Description	Values used in analyses
P_{b_u}	Total rated power budget of *u-th* UPS	270 W
$PH_{s_u,max}$	Set of maximal power consumptions for servers supplied over *u-th* UPS	$P_{1u,max}$ & $P_{2u,max}$= 89 W $P_{3u,max}$: 111 W
$PH_{s_u,min}$	Set of minimal power consumptions for servers supplied over *u-th* UPS	$P_{1u,min}$ & $P_{2u,min}$= 79 W $P_{3u,min}$= 90 W
S_u	Set of servers powered by *u-th* UPS	M_u =3
V_u	Set of VMs hosted on servers powered by *u–th* UPS	N_u =11
U	Set of all UPSs in a DC	L =1
RVv_u	Set of VMs computing demands	RV_u = 0% – 100%
PU_{v_u,s_u}	Set of unit power demands when v_u-*th* VM exploits 1% of server s_u CPU load	$PU_{v_u,1}$& $PU_{v_u,2}$= 0.01 W $PU_{v_u,3}$= 0.21 W
X_{v_u,s_u}	Binary integer (controllable) variable	0, 1

of all UPSs $U=\{1, 2, \ldots, u, \ldots, L\}$ used for redundant power supply of DC servers is assumed (Fig. 1). Additionally, let $S_u= \{1, 2, \ldots, s_u, \ldots, M_u\}$ define a set of physical servers for which the *u-th* UPS ensures a redundant power supply. Since most DC services are realized as cl]\oud services via VMs, a set of all VMs that can be hosted on servers with power supply over the *u-th* UPS is defined as $V_u=\{1, 2, \ldots, v_u, \ldots, N_u\}$, where $N_u \geq M_u$. Each host server s_u has maximal CPU resource capacity (100% of CPU load), and each VM consumes Rv_u of this CPU resource capacity, which forms a set of VM demands (% of CPU time) for CPU resource capacities equal to: $RVv_u = \{R_{1,u}, \ldots, Rv_u, \ldots, R_{Nu}\}$. The unit power consumption of the v_u–*th* VM activated at the s_u –*th* server is assumed to be fixed and equal to P_{v_u,s_u}, and the set of unit power consumptions for the heterogeneous servers supplied by the *u-th* UPS is denoted: $PU_{v_u,s_u}=\{P_{1,1}, \ldots, P_{v_u,s_u}, \ldots, P_{N_u,M_u}\}$. The unit power consumption corresponds to the power consumed when the VM exploits 1% of the central processing unit (CPU) resources during VM activity. The set of minimal power consumptions for servers with heterogeneous hardware configurations and supplied over *u-th* UPS are defined $PH_{s_u,min} = \{P_{1u,min}, \ldots, Ps_{u,min}, \ldots, P_{Mu,min}\}$. Similarly, $PH_{s_u,max} = \{P_{1u,max}, \ldots, Ps_{u,max}, \ldots, P_{Mu,max}\}$ represents the set of maximal power consumptions for the servers supplied over the *u-th* UPS. The $P_{s_u,min}$ is the minimal static power that the s_u-*th* server consumes when the server is in an idle state (powered on and without any VM load), while $PH_{s_u,max}$ is the maximal declared power consumption of the s_u-*th* server (in the case of maximal CPU load). The coefficients, sets and variables used in the analyses with corresponding values are listed in Table I.

To achieve efficient power capping that minimizes UPS power outages in a DC, an integer linear optimization model has been proposed

$$\min\left(\sum_{v_u=1}^{N_u}\sum_{s_u=1}^{M_u} R_{v_u} P_{v_u,s_u} X_{v_u,s_u} + \sum_{s_u=1}^{M_u} P_{s_u,\min}\right) \quad (1)$$

s.t.

$$\sum_{v_u=1}^{N_u}\sum_{s_u=1}^{M_u} R_{v_u} P_{v_u,s_u} X_{v_u,s_u} + \sum_{s_u=1}^{M_u} P_{s_u,min} \leq P_{b_u} \quad \forall u \epsilon U \quad (2)$$

$$\sum_{v_u=1}^{N_u} R_{v_u} P_{v_u,s_u} X_{v_u,s_u} + P_{s_u,min} \leq P_{s_u,max} \quad \forall s_u \epsilon S_u \; \forall u \epsilon U \quad (3)$$

$$\sum_{s_u=1}^{M_u} X_{v_u,s_u} \leq 1 \quad \forall v_u \epsilon V_u, \forall u \epsilon U \quad (4)$$

$$X_{v_u,s_u} = \{0,1\} \quad \forall s_u \epsilon S_u, \forall v_u \epsilon V_u, \forall u \epsilon U \quad (5)$$

where the objective function (1) minimizes the overall power consumption of all the servers for which the *u-th* UPS is ensuring redundant power supply. To ensure a feasible solution of the optimization problem, the proposed optimization model is based on five constraints ((2)-(5)).

More specifically, constraint (2) limits the overall power consumed by the set of servers S_u to be less than the total power budget of the *u-th* UPS (P_{b_u}). The total power budget

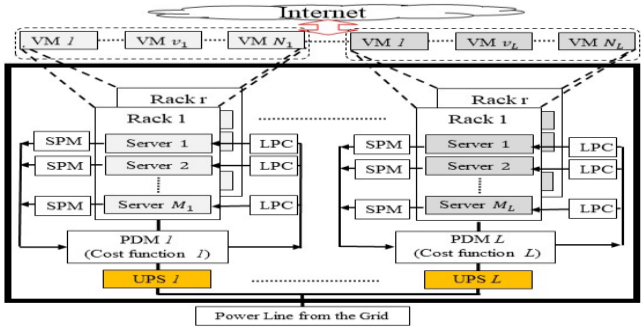


Fig. 1: Proposed power controller architecture for cloud DC

of the *u-th* UPS represents the maximal output rating of the UPS in terms of the real power expressed in Watts (Table I). Constraint (3) set the power allocation boundaries for each server in the DC to be lower than the maximum rated ($P_{s_u,max}$) power consumption that the s_u-*th* server could consume (Table I). Constraint (4) ensures that each active VM is hosted by only one s_u-*th* server. The last constraint (5) is a binary integer variable X_{v_u,s_u}limitation, which is equal to 1 if the v_u –*th* VM is hosted by the s_u-*th* server, and 0 otherwise. The optimization problem proposed is NP-hard, and the feasible solution is approximated for any combination of VM loads. Since the optimization model is executed separately per UPS *u* in a decentralized manner, the computational time is short due to the number of variables and constraints being limited to only one UPS.

B. System Design

The system design of the PDM proposed for a DC is presented in Fig. 1. The PDM is a central resource manager that collects power consumptions and distributes power allocations among DC servers based on the proposed optimization model. The PDM executes the developed optimization model ((1) – (5)) with the aim of avoiding any UPS power supply violations. As shown in Fig. 1, the PDM incorporates two software agents per server, which are a server power modular (SPM) and a local power controller (LPC). The first agent collects the power demands from each server by converting the VM resource demands into server power demands, while the second agent enforces the assigned power cap (threshold) to each server. Both components are located at the server-level end (Fig. 1), and different works have already reported the practical use of SPM and LPC [15]–[17].

The VM resource demands are the demands of the central processing unit (CPU) computing capability Rv_u (known in Xen hypervisor as the *CPU cap*), which represent a demand on the time portion of the CPU running time expressed in percentage (Table I). Each server gathers the resource demands of the VMs (or *cap demands* RVv_u) and passes the demands to the SPM located at the server-side manager (Fig. 1). The SPM converts the *cap demands* (with respect to the currently running CPU frequency) into *power demands* ($R_{v_u} P_{v_u,s_u}$) and reports the demands to the PDM. After the PDM collects all the *power demands* PU_{v_u,s_u} from the set of servers under control, the best available (optimum) power distribution among those servers will be generated (based on the optimization model (1)-(5)), and accordingly, each server has its *power assign* quantity. The *power assign* is received by the LPC located at the server-side (Fig. 1), which checks the *power assign* with the current hardware settings (both the running CPU frequency and the capping of the percentage of

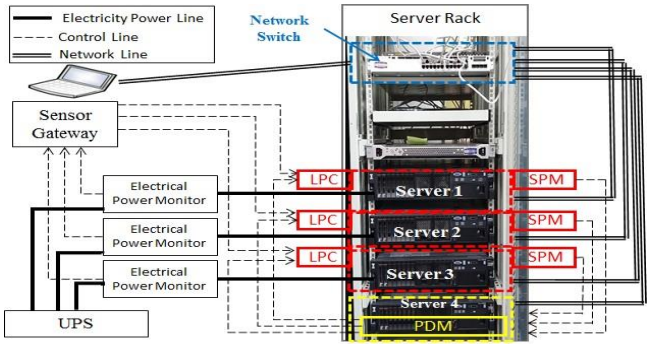


Fig. 2: Evaluation testbed with IBM System x3650 rack-mounted servers

CPU shared time portion). According to the LPC operation, the CPU frequency adjustment trough dynamic frequency scaling (DFS) is performed, and the capping on the percentage of the CPU shared time portion (or *CPU cap*) is tuned (*cap assign*). During the practical experiment, all these configuration setting changes (the *cap demand/assign* through CPU frequency and time portion scaling) and the power variations, i.e., the *power demand/assign* through the real power consumption measured were observed.

III. Experimental Evaluation Test Bed

In this letter, the performance of the proposed PDM was tested using the real evaluation testbed presented in Fig. 2. For testing purposes, a cluster of three rack-mounted IBM servers was used and, the details of each hardware configuration are listed in Table II. The objective of the evaluation was distribution of the power supply among servers according to the variations of VM loads while satisfying the available UPS power budget. For solving the optimization problem (1) – (5), the MATLAB optimization tool based on a greedy algorithm was used. The algorithm was executed on a hardware configuration of a separate server 4 (Table II), which was also placed in the rack (Fig. 2). Using the built-in Apache web server benchmarking tool, a deterministic workload was generated on each VM. Concurrently, the information of the instantaneous power consumption of each server was detected through the electric power monitor that intercepts the server electrical power supply cables (Fig. 2). This information was transferred to the PDM over the installed sensor gateway [18] (Fig. 2).

The evaluation scenario was based on VMs running a mathematic prime-counting function, which is a computationally intensive application fairly representing deterministic workload. For controlling the execution of the prime-counting function within the Apache web server of every VM, an external user device (laptop) connected with the server rack over a separate network connection was used (Fig. 2). Each primes function executed had a variable completion time that depended on the virtual CPU configuration (load and frequency) scheduled to the hosting VM (Table II). In this experiment, the prime-counting function calculation time was set at a 2 second maximum (for the lowest CPU load and frequency configuration). To obtain accurate and updated power distribution among the servers, the optimization model was executed in the PDM every 5 minutes. The practical experiment ran continuously for 3 hours, during which the activity of N_u VMs was changed through variations in percentage of VM load (from 0% to 100%). Experimental results were obtained through analyses with the parameter values presented in Table I. Different power budgets (Table I) of the heterogeneous server

TABLE II: Testbed configuration with heterogeneous servers

Server param.	Server 1	Server 2	Server 3	Server 4
CPU core no.	4	4	8	8
CPU fr. (GHz)	1.5~2.5	1.5~2.5	1.5~2.5	1.5~2.5
Memory (GB)	16	16	16	16
Storage (TB)	2	2	2	2
Linux oper. syst. (OS)	CentOS 7	CentOS 7	CentOS 7	CentOS 7
Hypervisor	Xen 4.1	Xen 4.1	Xen 4.1	N/A
No. of VMs	3	3	5	N/A
CPU core no./ VM no. ratio	1.33	1.33	1.6	N/A

configurations (Table II) were selected to demonstrate the power distributions and power capping performance on rack servers. The experimental outcomes concerning the UPS power capping are discussed in the next section.

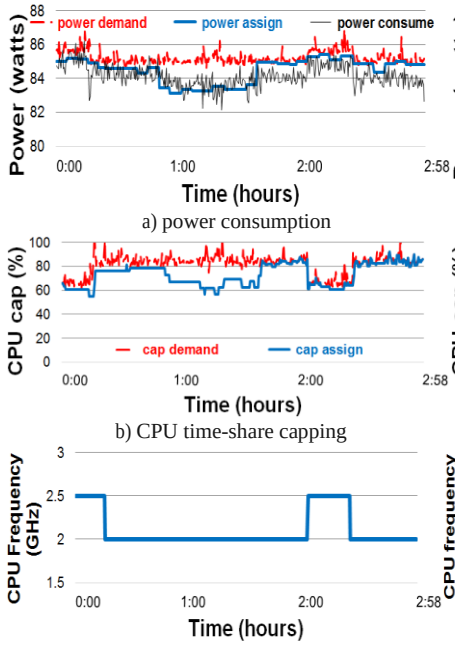
IV. Results and Discussion

The relationship between the *resource (cap) demand* of the VMs and the *power assign* to each server by the PDM was observed. First, the SPM of each server (Fig. 2) converted the *cap demands* of the hosted VMs (shown with dashed lines in Figs. 3b, 4b and 5b) into *power demands* (shown with dashed lines in Figs. 3a, 4a and 5a). The PDM over the LPC then assigned the available power to all three servers 1, 2 and 3 (Fig. 2), as shown with thick lines in Figs. 3a, 4a and 5a, respectively. The *power(s) assign(ed)* were distributed by the LPC of each server. The LPC tuned the shared time portion of the CPU time (or *cap assign*) and frequency among the VMs hosted by servers 1, 2 and 3, which is shown with thick lines in Figs. 3b, 4b and 5b, respectively. During the experiment, the proposed PDM solved the optimization problem in practically applicable computation time (order of a few milliseconds).

Despite the variable power budget of the cluster of servers (Table I), for every server, Figs. 3c, 4c and 5c show that the CPU DFS was infrequent throughout the entire experiment. Few CPU frequency changes were recorded for servers 1 and 2, as shown in Fig. 3c and Fig. 4c, respectively, while the CPU frequency of server 3 was unchanged throughout the entire experiment (Fig. 5c). This result is a consequence of the direct relationship between the CPU frequency scaling and CPU capacity (cap) assignment (% of CPU time-share capping), where a decrease in CPU capacity (Figs. 3c-5c) is followed with an increase in CPU frequency (Figs. 3b-5b) and vice versa. Both CPU configurations (CPU frequency scaling and capacity assignment) can control the power consumption of the server; however, changing the CPU frequency directly affects the power consumption of the server and impacts the performance of the hosted VMs. Therefore, controlling the server power consumption through the allocation of the CPU cap assignment is more preferable in terms of power consumption over CPU frequency scaling.

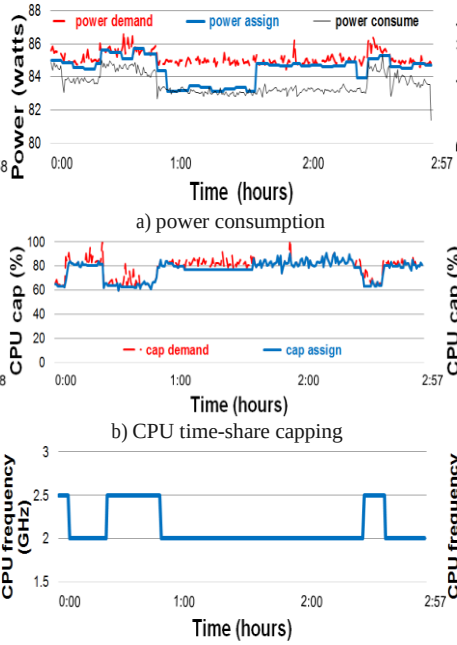
For example, although server 3 with 5 VMs has higher computing load (prime-counting function), the better CPU core number/VM number ratio (Table II) of server 3 ensures that only use of the CPU capacity (time-share) assignment (Fig. 5b) without CPU frequency scaling (Fig. 5c) can meet the optimization power restrictions. For the other two servers 1 and 2, ensuring the UPS power restrictions must be done through CPU capacity capping (Fig. 3b-4b) and CPU frequency scaling (Fig. 3c-4c), respectively, due to the less favourable CPU core number/VM number ratio (Table II).

Nevertheless, the results obtained show that the proposed



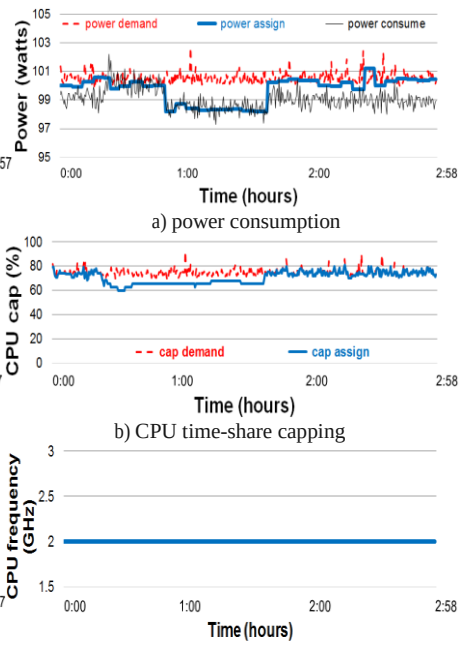
a) power consumption
b) CPU time-share capping
c) CPU frequency scaling

Fig. 3. Results for server 1



a) power consumption
b) CPU time-share capping
c) CPU frequency scaling

Fig. 4. Results for server 2



a) power consumption
b) CPU time-share capping
c) CPU frequency scaling

Fig. 5. Results for server 3

optimization model based on the PDM developed ensures appropriate power assignment during the whole experiment, with respect to the UPS overall power rating P_{b_u} (Figs. 3a-5a). Apparently, during analysis spikes in the *power consumed* that are slightly larger than the *power assign(ed)* were observed in certain periods (Figs. 3a- 5a). The variation was within the one Watt only. These spikes were caused by the LPCs running on the server side, where the LPC was a proportional-integral-derivative (PID) controller with a one-watt overshoot threshold. During PID overshoot on a server where the spikes were observed, other servers were consuming power below the assigned power (Figs. 3a – 5a), which made the overall power consumption of the set of servers within the UPS power budget. Hence, the PDM proposed ensures that the power consumed by the servers does not exceed the available UPS power budget and consequently the overall DC power limitations set by the utility company supplying DC with electric power.

V. CONCLUSION

In this letter, the problem of UPS overload in DCs was analyzed. To address this problem, a PDM that assigns power caps to each server in a DC according to the available power budget of UPSs is proposed. The power assignment is based on a linear optimization model that considers UPS power budgets and resource demands of each VM installed on servers. The efficiency of the PDM developed was evaluated using real testbed comprising servers with a variable number of active VMs and corresponding loads. The results of the evaluation show that the proposed power capping solution assigns the appropriate power levels to the corresponding servers based on the VM capacity demands through dynamic adjustment of CPU time-sharing and DFS. By implementing the proposed PDM, efficient power capping that minimizes UPS overloading and consequently server or even DC failures can be achieved. In future work, we will focus on further improvement of the PDM functionality through the development of new heuristic optimization algorithms which can be applicable to DCs with a container-based architecture having more servers sharing power supply over one UPS.

REFERENCES

[1] A. Lawrence, “Uptime Institute data shows outages are common, costly, and preventable,” June 2018.

[2] K. Heslin, “Here is why major airline outages keep happening, and what you can learn from them,” *IT portal*, accessed June 23, 2017

[3] W. Zheng, “Power Capping with Optimized Computing Performance in DCs,” The Ohio Stet University, doctoral dissertation, pp. 1-155, 2016

[4] LA Barroso, U. Hölzle, “The datacenter as a computer: An introduction to the design of warehouse-scale machines,” *Synthesis lectures on computer architecture*, Vol. 8, No. 3, pp. 1-154, 2013

[5] A. Greenberg, J. Hamilton, D.A. Maltz, and P. Patel, “The cost of a cloud: research problems in data center networks,” *ACM SIGCOMM computer communication review*, Vol.39 No. 1, pp. 68-73,2009.

[6] A.H. Fawaz, Y. Peng, C.H. Youn, J. Lorincz, C. Li, G. Song and R. Boutaba, “Dynamic allocation of power delivery paths in consolidated data centers based on adaptive UPS switching,” *Computer Networks*, Vol. 144, pp. 254-270, 2018.

[7] H. Zhou, J. Yao, H. Guan and X. Liu, "Comprehensive understanding of operation cost reduction using energy storage for IDCs," *IEEE Conference on Computer Communications (INFOCOM)*, pp. 2623-2631, 2015.

[8] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy, “Energy storage in datacenters: what, where and how much?,” *ACM SIGMETRICS Perf. Eval. Review*, Vol. 40, No. 1, pp. 187-198, 2012

[9] C. Lefurgy, , X. Wang and M. Ware, “Power capping: a prelude to power shifting,” *Cluster Computing*, Vol. 11, No. 2, pp.183-195, 2008

[10] M. Chen, X. Wang and X. Li. “Coordinating processor and main memory for efficientserver power control,” *In Proceedings of the international conference on Supercomputing*, pp. 130-140, 2011.

[11] P. Ranganathan, P. Leech, D. Irwin and J. Chase, “Ensemble-level power management for dense blade servers,” *In ACM SIGARCH Computer Architecture News*, Vol. 34, No. 2, pp. 66-77, 2006

[12] X. Wang, and M. Chen, “Cluster-level feedback power control for performance optimization,” *In 2008 IEEE 14th International Symposium on High Performance Computer Architecture*, pp. 101-110, 2008.

[13] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang and X. Zhu, “No power struggles: Coordinated multi-level power management for the data center,” *ACM SIGOPS Operating SysT. Rev.*, Vol. 42, No. 2, pp. 48-59, 2008

[14] X. Wang, M. Chen, C. Lefurgy and T. Keller, “SHIP: Scalable hierarchical power control for large-scale data centers,” *In 2009 18th Internat. Conf. on Parallel Archit. and Compilation Tech.*, pp. 91-100, 2009.

[15] Y. Wang and X. Wang, “Performance-controlled server consolidation for virtualized data centers with multi-tier applications,” *Sustainable Computing: Informatics and Systems*, Vol. 4, No. 1, pp. 52-65, 2014.

[16] X. Shi, C. Briere, S. Djouadi, Y. Wang and Y. Feng, “Power-aware performance management of virtualized enterprise servers via robust adaptive control,” *Cluster Computing*, Vol. 18, No. 1, pp. 419-433, 2015.

[17] C. Li, R. Wang, D. Qian and T. Li, “Managing server clusters on renewable energy mix,” *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 11, No. 1, p.:1 Apr. 2016

[18] “Yoctopuce,” http://www.yoctopuce.com/, accessed March 25, 2018.