HOPE: Enabling Efficient Service Orchestration in Software-Defined Data Centers

Yang Hu¹, Chao Li², Longjun Liu³ and Tao Li¹

¹ Department of Electrical and Computer Engineering University of Florida, USA huyang.ece@ufl.edu taoli@ece.ufl.edu ² Department of Computer Science and Engineering Shanghai Jiao Tong University Shanghai, China *lichao*@cs.sjtu.edu.cn ³ Institute of Artificial Intelligence and Robotics Xi'an Jiaotong University, Xi'an, China *liulongjun@xjtu.edu.cn*

ABSTRACT

The functional scope of today's software-defined data centers (SDDC) has expanded to such an extent that servers face a growing amount of critical background operational tasks like load monitoring, logging, migration, and duplication, etc. These ancillary operations, which we refer to as management operations, often nibble the stringent data center power envelope and exert a tremendous amount of pressure on front-end user tasks. However, existing power capping, peak shaving, and time shifting mechanisms mainly focus on managing data center power demand at the "macro level" – they do not distinguish ancillary background services from user tasks, and therefore often incur significant performance degradation and energy overhead.

In this study we explore "micro-level" power management in SDDC: tuning a specific set of critical loads for the sake of overall system efficiency and performance. Specifically, we look at management operations that can often lead to resource contention and energy overhead in an IaaS SDDC. We assess the feasibility of this new power management paradigm by characterizing the resource and power impact of various management operations. We propose HOPE, a new system optimization framework for eliminating the potential efficiency bottleneck caused by the management operations in the SDDC. HOPE is implemented on a customized OpenStack cloud environment with heavily instrumented power infrastructure. We thoroughly validate HOPE models and optimization efficacy under various user workload scenarios. Our deployment experiences show that the proposed technique allows SDDC to reduce energy consumption by 19%, reduce management operation execution time by 25.4%, and in the meantime improve workload performance by 30%.

Keywords

Software-defined data center, power management, management workloads.

1. INTRODUCTION

The cloud data center today is a colossal system that consists

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Request permissions from Permissions@acm.org. ICS '16, June 01–03, 2016, Istanbul, Turkey. © 2016 ACM ISBN 978-1-4503-4361-9/16/05 ...\$15.00. DOI: http://dx.doi.org/10.1145/2925426.2926257 of a variety of hardware components and software applications [2, 27]. Their efficient operations rely on virtualization-based management frameworks which have evolved into a new concept as Software-Defined Data Center (SDDC) [35]. Today the SDDC expands virtualization from computing resources to storage and networking resources, offering significant improvement in scalability and manageability.

The fully virtualized computing environment of SDDC relies on many important background services that cannot be ignored. We refer to these services as Management Operations (MO) that mainly consist of system configurations (virtual machine allocation, virtual networking setting, virtual storage volume assignment, etc.) and ancillary service enhancement tasks (data replication, data deduplication, anti-virus scanning, load balancer, etc.). The execution of the management operations unavoidably introduces extra data center burden that we term as management workloads in this study. Without appropriate control, the management workloads can consume considerable computing resources, power budget, and energy in an SDDC, resulting in significant performance and cost overhead.

Taming management operations presents significant challenge for today's software-defined data centers. While there has been an initial study on characterizing management workloads in traditional virtualized data center, it mainly focuses on computing virtualization [3]. In fact, SDDC management operations brought by extra virtualization layers often manifest distinctive behaviors that are not correlated with typical user workloads. In addition, most of the SDDC management operations span across compute nodes, virtual network, and storage nodes. The performance and power impacts of the management operations on distributed heterogeneous SDDC components have not been studied thoroughly in prior works.

Although some management operations themselves are not resource-hungry, they can adversely affect a large amount of compute nodes. A management operation on the critical path, if happens to incur an unexpected delay, can have a cascade effect on the system, i.e., execution latency on massive related compute instances that have complex inter-dependency. In this case, the execution latency not only brings performance penalty but also results in significant energy overhead we termed as "tail energy". Our characterizations demonstrate that compared to the traditional data center, management workloads in SDDC could yield about 20% additional power demand, and increase the energy consumption of user workloads by almost 15%.

In this study we investigate the underlying root cause of efficiency bottleneck in a software-defined data center. Prior power management schemes largely focus on managing the power demand from the perspective of the entire system or facility, which we term as macro-level power management. Examples include server power demand reduction and power capping mechanisms that prevent power budget violation. These mechanisms do not distinguish user tasks from ancillary tasks, and therefore often incur unnecessary performance degradation of user workloads. For example, a rack-level power capping decision that intends to shave power surge caused by backup services could also throttle performance of a critical latency-sensitive user task.

To achieve the optimal energy efficiency with the slightest performance loss in an SDDC, applying micro-level in lieu of macro-level power management strategies is necessary, especially when data centers are becoming increasingly power-constrained[15–17, 19–21]. Here, the micro-level power management strategy refers to analyzing detailed workload interdependency and fine-tuning certain specific set/type of critical workloads for the sake of the overall system efficiency and performance.

This study explores the opportunity of applying micro-level power management strategies to management workloads in SDDC. There are two primary problems. First, what are the most critical management operations (MO) during runtime? Second, what technical approaches should we apply to minimize the operation overhead?

To understand management workload, we extensively investigate management operations that prevalently exist in computing, networking, storage, and orchestrating systems in an SDDC. We broadly classify typical SDDC management operations into four groups: compute instance management (either for fault-tolerance [43] or load balancing [3]), virtualized storage management [4], software-defined network rule processing behaviors [22], and intelligent data center log management/analysis (for system optimization and troubleshooting purpose [39]). We perform indepth characterization of their server-level power behaviors on our scaled-down OpenStack-based IaaS data center.

We propose Holistic Overhead Power Eliminator (HOPE), a lightweight system coordination strategy for minimizing energy and performance overhead brought by cross-system management operations. HOPE presents itself as a tiny system patch that lies in the data center middleware. It introduces two new functions in an SDDC. First, HOPE can perform global MO auditing through a novel structural dependency graph. The dependency graph enables data center to identify critical management operations based on the collected management workloads distribution information and physical server power statistics. Second, HOPE provides data center flexible tuning knobs to re-schedule the critical management operations in both proactive and reactive manners. This intelligent coordination can eliminate potential efficiency bottleneck in the SDDC.

We implement and deploy HOPE on our customized private cloud environment. We deploy OpenStack cloud management middleware and integrate HOPE as a service in OpenStack. To the best of our knowledge, this is the first work that characterizes and optimizes the power and energy overhead issues of the SDDC management workloads.

The contributions of this paper are as follows.

 We characterize the management workloads in softwaredefined data centers. Our investigation involves virtualized storage and networking subsystems, which lack thorough explorations in prior work.

- We identify the "tail energy" issue in SDDC. We demonstrate that the energy/power overhead caused by critical management workloads can be significant and explore the opportunity to mitigate their impacts.
- We present HOPE, a novel system optimization framework for mitigating the energy overhead among cross-system management operations. We detail a graph-based dependency analysis tool used by HOPE to eliminate the efficiency bottleneck in SDDC.
- We implement HOPE on a heavily instrumented rack-scale cloud system and evaluate its efficacy in optimizing data center operation. We show that HOPE allows SDDC to reduce energy consumption by 19%, reduce management operation execution time by 25.4%, and in the meantime improve workload performance by 30%.

The rest of this paper is organized as follows. Section 2 summarizes the operational workloads in cloud data centers and characterizes their energy impact. Section 3 proposes operation workloads analysis methodology and our overhead mitigation techniques. Section 4 details of our prototype. Section 5 presents evaluation results. Section 6 discusses related work, and Section 7 concludes this paper.

2. UNDERSTANDING SDDC MANAGE-MENT SERVICES

We first introduce the backgrounds and our evaluation methodology. We then characterize representative management workloads in SDDC in terms of their efficiency impacts to motivate micro-level power management.

2.1 Backgrounds

Software-defined system enables flexible configuration and maintenance in data centers. However, it also introduces various management workloads that demand additional power and energy resources, as discussed below:

Ordinary VM management operations such as VM creation, VM snapshot revert, or VM live migrations often causes considerable traffics on networking and storage subsystems. VMware's work [32] states that the averaged automated live migration observed in various data centers occurs 6 times per day per VM, and 12% of VMs in a data center are involved in snapshot reverting operations. In addition, the energy impact of migration can offset over 12% of the energy saved through energyconscious workload packing [11].

Storage system management operations can exert considerable pressure on SDDC. For example, data deduplication is widely adopted by cloud data centers in backup and archive storage systems [4][1]. On ZFS file system [42] with inline deduplication, the average CPU usage could boost from 20% to 60% due to heavy computation and indexing. The inline deduplication causes the storage system to consume 10% more power on average [44].

Logging and analytic services are essential components of modern SDDC. They have been widely adopted by many cloud providers [33, 34, 38] for enabling real-time optimization and proactive troubleshooting. These services are normally hosted on a group of virtual machines within the SDDC and conduct compute-intensive log analysis tasks. According to [37], a log analysis VM with 2 vCPU and 4GB memory only supports log volumes of 3GB a day for about 10 users. It consumes about 10% of the computing resource of a server and 15% of the power consumption.



Figure 1. Power consumption caused by management workloads under traditional and SDDC configuration

In addition to the non-negligible energy consumption, the burstiness of operational workloads presents a more rigorous challenge to the SDDC power provisioning system. For example, Virtual Desktop Infrastructure (VDI) is a base approach to Desktop as a Service (DaaS), in which virtual desktops are hosted as virtual machines. The "boot storm" is a common scenario that all the virtual desktops are powered on within very short time when employees come to work. If other co-located tenant applications are executing power-hungry tasks in the meantime, the power delivery system is susceptible to this burst power demand.

2.2 Characterization Methodology

2.2.1 Evaluation Workloads

We investigate eight representative management operations in an SDDC as listed in Table 1. We use a taxonomy in accordance with prior work [32].

We choose typical management operations that are introduced by storage and networking virtualization techniques in SDDC, such as data deduplication, rule processing, virtual volume management, and log processing. In addition, we also select several traditional management operations such as basic VM creation, live migration, and snapshotting. They are essential in traditional data center management workflows, such as periodic snapshot/revert, VM patching, boot storm handling, automated load balancing, and after-hours maintenance. For these operations, we study their power/energy behaviors under both traditional and software-defined networking and storage configurations.

The basic management operations we selected exhibits different detachability and deferability. The detachability represents whether a management operation could be migrated to other physical servers. It reflects the spatial manageability of a management operation. For example, a VM creation operation is tightly coupled with user VM assignment. It cannot be executed on other machines while the log processing could be executed on any machine within SDDC. The deferability represents whether a management operation is mission-critical. It reflects the temporal manageability of certain management operation. For example, in the VDI boot storm the VM creation operations are non-deferrable. While the VM snapshotting for backup could be deferred to a less busy time window.

2.2.2 Experimental Setup

Our characterizations are conducted based on a scaled-down SDDC prototype, which consists of 6 compute nodes, 1 storage node with Fiber Channel storage pool, 1 network node, and 1

Operations	Resource Intensive	Detachable	Deferrable
Deduplication	CPU, I/O	Yes	Yes
Rule processing	CPU	Yes	No
Volume backup	CPU, Network	No	Yes
Volume migration	CPU, Network	No	Yes
Log processing	CPU	Yes	Yes
VM creation	CPU, Network	No	Yes
VM live migration	CPU, Network	No	Yes
VM snapshotting	CPU, Network	No	Yes
Table 1 List of basis management operations in SDDC			

Table 1. List of basic management operations in SDDC

cloud controller node. We employ HP DL380 G5 servers as physical nodes. Each of them uses two Xeon X5450 3.0GHz CPU and 32G buffered ECC RAM, 1Gbps Ethernet interface and 1TB SAS 15000RPM HDD. The storage server connects its 2TB storage pool (IBM DS4800) through Fiber Channel. The network node and cloud controller node all employ HP DL380 G5 servers with the same configuration as compute servers.

We deploy OpenStack Havana cloud service suites to implement a software defined data center [27]. The basic VM management operations could be obtained from computing service OpenStack Nova with KVM/libvirt. We deploy Open Stack Cinder block storage service as virtual volume support and VM live migration support. To enable inline deduplication we deploy ZFS filesystem on storage node and use ZVOL to provide volume service for Cinder. We deploy networking service OpenStack Neutron to enable software-defined networking functionality such as firewall rule processing and load balance as a service. Specifically, the NFV is based on Open vSwitch 1.2.0. We employ WTI VMR-8HD20-1 Outlet switched/metered PDUs to collect the power and energy readings of each server.

2.3 Energy Overheads Analysis

2.3.1 Traditional DC vs. SDDC

We first evaluate the power and energy behaviors of basic VM management operations by comparing power traces of VM live migration and snapshotting under SDDC environment and traditional virtualized data centers, as shown in Figure 1. In the traditional virtualization case, we use Linux Bridge as networking backend and do not use any storage virtualization. In the SDDC case, we use Open vSwitch as software-defined networking and use OpenStack Cinder volume with ZFS as storage virtualization. The test VM is configured as 2 vCPUs with 4GB RAM.

Figure 1(a) demonstrates the power consumption of source machine in the VM live migration scenario. It shows two power demand peaks A and B, which are introduced by live migration under traditional environment and SDDC environment, respectively. Notice that peak-B gains 29% more peak value, 65% more duration, and 53% more energy consumption than peak-A does.

Figure 1(b) shows the compute node power consumption of VM snapshotting. The VM snapshotting consists of three stages, image creation, live snapshotting, and image uploading. Our measurement shows that the region-B (SDDC) costs 2.1X more time and consumes 1.53X more energy than region-A (traditional). The extra power consumption of management operations in the SDDC is mostly resulted from the additional processing overheads introduced by networking virtualization such as Open vSwitch. The software packet processing incurs high CPU utilization on the compute nodes. Therefore, the power issue of management workloads in a SDDC should be paid more attention than in traditional virtualized data centers.



Figure 2. The power and energy impacts of management workloads in data centers.



Figure 3. Task mapping of multiple management operations

2.3.2 Impact on User Workloads

We investigate how management workloads affect the power and energy consumption of physical servers running user workloads.

Our results show that management workloads could consume considerable amount of peak power budget. Figure 2 presents the power demand trace of servers deploying software testing workload, in which a compute node hosts three software testing VMs. The compute node periodically commits snapshots for testing VMs. If user applications are the only running workloads, the power demand is about 380W (Region A). However, when the three VMs begin to commit snapshot, the peak power reaches 420W (10% additional power demand). Therefore, without appropriate control, the management workload can easily exacerbate contention for power budget resource.

We further evaluate the energy consumption of the above management operations in a power under-provisioned data center. We deploy a lead-acid battery for pear power shaving purpose as described in recent work [10, 40][18] and measure the capacity of energy storage devices. We conduct this experiment on our 24V battery equipped SDDC testbed and set two power capping thresholds as 355W and 375W, respectively. We measure that the snapshotting workloads cost 1.98X more stored energy consumption than software testing workloads. Specifically, the management operations consume 112Ah stored energy while the user workloads consume 37.69Ah with a power cap value of 375W. For a more aggressive power cap value of 355W, the management workloads consume 192.98Ah while the user workloads consume 140.79Ah.

To quantify the power impact of management workloads, we use a new metrics call peak impact factor (*PIF*). It is defined as the ratio between peak power caused by MO alone and the idle power of the server. We measure the *PIF* value of various management operations on our platform and present the results in Table 2. We observe that the *PIF* of certain management operation is mainly determined by the number of concurrent executing operations with similar resource intensity. Although the absolute value of *PIF* depends on the underlying hardware, such trend still maintains. This is because the management operations normally lead to extreme resource utilizations.

2.3.3 Tail Energy

We observe that management workloads, especially in consolidated computing environment, can cause undesirable clus-

Onenations	PIF under different concurrent tasks (CT)		
Operations	1 CT	2 CT	3 CT
Deduplication	22.9% (volume)	37.1%	44.9%
Rule processing	29.7%	38%	54%
Volume backup	23.6%	37.6%	44%
Volume migration	18.9%	33.1%	35.7%
Log processing	33%	45%	51%
VM creation	20.6% (compute)	34.2%	38.3%
VM live migration	16.5% (source)	18.7%	22.4%
VM snapshotting	28.8% (compute)	37.1%	42.8%

Table 2. Measured PIF of different management operations



Figure 4. Power traces of related servers.

ter-wide execution delay and energy overhead. We refer to the additional energy overhead as "tail energy".

We demonstrate the tail energy issue by studying a real scenario that involves multiple management operations, as shown in Figure 3. In the figure, VM1 (compute instances hosted on compute node 2; image file hosted on a volume storage node) is committing a VM image snapshot to the cloud image storage node in the SDDC. The data deduplication service is running on the volume storage node as another management operation. In the meantime, VM2 (hosted on compute node 1) is migrating to compute node 2. Thus we have three management operations in this scenario. Figure 4 compares the power behaviors of different nodes with and without management workloads. We report the power consumption traces of the compute node and VM volume node using red curves. We then disable the deduplication management operation on volume node and show the re-produced power traces using blue curves. Along with the traces, we also provide the CPU utilization value and networking traffic of compute node 2 in Figure 5.

From Figures 4 and 5, we can see that the data deduplication service not only result in high power consumption on the volume node, but also degraded snapshotting speed due to CPU resource contention. It can also slow down the snapshotting process on volume node. The execution delay on the volume node can further affect the snapshotting progress on compute node 2 since the CPUs on the compute node 2 must wake up frequently to calculate the checksum for snapshot. This causes 16.5% additional energy consumption on the compute node 2. Similarly, the extended snapshotting process on compute node 2 further procrastinates the VM live migration processes and result in 91% additional energy consumption on compute node 1.

Once the deduplication process is disabled, the task execution time and energy consumption on both compute nodes can be saved, as shown in Figures 4(b) and 4(c).

Operations	Involved Nodes (sequentially)	PDF	Т
Deduplication	V	0.59	N/A
Rule processing	Networking	0.53	N/A
Volume backup	V→I	0.22	11
Volume migration	Vs→Vd	0.29	9.2
Log processing	С	0.66	N/A
VM creation	I→V→C	0.25	8.7
VM live migration	Cs→Cd	0.21	6
VM snapshotting	C→V→I	0.38	20

Table 3. Node inter-dependency and the value of performance degradation factor (*PDF*) and execution time T in a SDDC. (C: compute node, V: volume node, I: image node, s: source, d: destination).



Figure 5. Network traffic and CPU utilization of compute node 2. Traces of two scenarios are demonstrated sequentially.

Two factors lead to energy tails. First, consolidation of multiple management operations introduces intra-node interference and performance degradation on the initiating nodes. Second, the inter-node management operations (such as VM migration and snapshotting) may propagate the intra-node interference caused slowdown to the involved nodes that are influenced by the internode management operations. We term this phenomenon as internode dependency/correlation.

To solve the energy tail issue, we further investigate the inter-node correlation and resource interference in SDDC to provide preliminary knowledge. We summarize the initiating node and involving node pairs as shown in Table 3. Each correlation pair (connected by an arrow) represents a basic cloud operation (e.g. cloud image copy). The associated operational workloads of each correlation pair have similar PIFs and resource utilization characteristics. Based on the models that proposed in [12][10], we define an initiating node's *tail energy impact* to an involving node as: $T(1+d)^{n-1}$, where T is the execution time of management operation, d is the performance degradation factor (PDF) of the management operation, and n is the number of management operation between these two nodes. Based on intensive characterization, the performance degradation factors (PDF) and execution time T of management operations are shown in Table 3. For the management operations only residing on single node, T is not available. We will provide in-depth discussion in Section 3.

Summary: We have comprehensively investigated the power/energy characteristics and manageability of basic management operations in Section 2. We can observe the diversity of the management operations poses great challenge for designing effective power management scheme in modern SDDC. It is critical to apply a fine-grained micro-level power tuning strategy to improve efficiency.



Figure 6. The architecture and workflow of HOPE

3. MANAGING OPERATIONAL OVER-HEADS IN SDDC

We develop holistic overheads power eliminator (HOPE), a software-based coordination framework tailored to the power/energy characteristics and manageability of SDDC management workloads (MO). HOPE employs two novel technical approaches to mitigate the management workloads power/energy issues and tail energy issues.

First, HOPE employs a graph-based correlation analysis mechanism to analyze the distribution of management operations and locate critical management operations. Second, HOPE exploits software-based critical management operation scheduling (CMOS) method to re-schedule critical management operations based on the guidance of graph-based analysis. We elaborate our system architecture and policy design in this section.

3.1 HOPE System Architecture

Figure 6 depicts the full system architecture and workflow of our proposed design. HOPE is deployed as a lightweight coordination framework in the SDDC middleware layer. It consists of three functionality modules, namely system auditing module, data analysis module, and committing modules. They process incoming management operations consequentially in our system.

3.2 System Auditing Module

The system auditing module (SA) is the information collecting unit of HOPE. It monitors the physical server power status and management operation distribution in the SDDC through various cloud APIs and hardware communication interfaces. For example, HOPE uses a switched intelligent PDU array to support flexible power budget allocation. The collected information is used in the decision-making process of the data analysis module.

The SA module supports two operating modes, proactive mode and reactive mode, as shown in Figure 6. The proactive mode is the default operating mode. It processes the routine management workloads initiated by the SDDC administrator which have a relative fixed agenda. The routine management workloads include after-hour maintenance, periodic snapshot/revert, boot storm handling, etc. In the proactive mode, a light-weight daemon named MO sensor periodically inspects the scheduled management operations on the MO agenda. It generates several "MO snapshots" by dividing the SDDC operating time into small time windows. The MO snapshot is the basic analysis object for the data analysis module. Each snapshot depicts the distribution of management operation in SDDC during this time window. The length of time window should be tuned according to the frequency of management operations and average management operation



Figure 7. The demonstration of graph based analytics

duration in the SDDC. The default time window length is 15 minutes.

The reactive mode is designed to handle the newly deployed management operations that cause power emergencies. Once the MO sensor receives a power alert generated by SDDC monitoring components, it interrupts the current running proactive mode and pre-empt the data analysis module to execute emergency MO analysis. It requests the updated SDDC status for building a new MO snapshot. The information includes SDDC management workload distribution, power consumption of physical hosts, power state of physical hosts, and intelligent PDU port status. The advantages of using event notification in the SA module instead of cyclic polling are two-fold: First, event notification pushes all the interested system events while polling may miss them, especially when a long loop period is adopted. Second, event notification significantly reduces the communication overhead.

3.3 Data Analysis Module

The Data Analysis (DA) module processes the management operation distribution snapshots sent by the SA module and provides critical management operation scheduling guidance for the committing module.

The core software component in the DA module is called MO analyzer. It can be activated by the SA module. During operation, the MO analyzer first employs a graph-based analytics method to inspect the physical server hot spots and locate the critical management operations. It then executes critical management operation scheduling (CMOS) based on the analysis and the manageability of critical MOs. Finally it calls commit module to apply the corresponding execution to eliminate the hot spots and tail energy issue.

3.3.1 Management Operation Correlation Network

The MO analyzer features a graph-based analysis method that can construct management operation correlation graph and identifies critical management operations.

Our graph-based analytics first builds management operation correlation network (MOCN) of a given MO snapshot to model the tail energy impacts among management operations. The MOCN is defined as a directed weighted graph G = (V, E), where the vertices V are physical machines in SDDC. A typical example of MOCN is shown in Figure 7. There is an edge e_k (v_i , v_k) between v_i and v_k if all the following conditions are met:

① At least one type of management operation MO_{ia} initiated by the physical machine v_i involves another physical machine v_k ;

2 There is another management operation MO_{ka} residing on the involved vertex v_k with the same resource intensity as MO_{ia} .



Figure 8. Critical MO scheduling flow chart.

We define the weight of the edge w_k as the energy overhead that MO_{ia} exerts on v_k , which is also termed as *MO tail energy impact factor (MO-TEF)*. Based on the discussion in section 2.3.3 and prior work [18][19], the weight w_k is represented as:

$$MO\text{-}TEF_{ia,k} = w_k = T_a * (1+d_a)^{n-1}$$

where d_a is the *PDF* of management operations MO_{ia} , and T_a is the energy impact duration of MO_{ia} , as listed in Table 3. The tail energy impact factor of vertex v_i (*vTEF*) is represented as a weighted sum of *MO-TEFs* of v_i . The weight represents the efficiency impacts of co-located management operations MO_{im} on MO_{ia} . Specifically, MO_{im} could be local management operations such as log processing, deduplication or management operations involving other vertices $v_i ... v_n$. The *vTEF* is defined as:

$$vTEF_i = \sum_{e_k \in E} (w_k * \prod_{m \notin MO_{ia}} (1+d_m)),$$

where E is the set of edges of v_s , and d_m is the *PDF* of colocated management operations MO_{im}. We validate the *MO-TEF* and *vTEF* models in the evaluation section.

3.3.2 Critical Management Operation Scheduling

Given the MOCN, MO analyzer can locate the critical management operations and assign the highest re-scheduling priority to them. Figure 8 demonstrate the decision-making flow chart for critical management operation scheduling.

MO analyzer first manages the hotspots in current MO snapshot. Here the physical server with the highest *vTEF* and *PIF* is treated as a hotspot. The MO analyzer first tries to find detachable MOs with highest *PIF* on the hotspot.

In our design, MO analyzer prefers migrating management operations on the hotspot than deferring them to minimize the interference of management operation scheduling agenda. The detachable MOs with the highest *PIF* will be migrated to physical nodes (in the same function group, i.e. compute nodes group, volume nodes group, or image nodes group) with the lowest *vTEF*. Note that the post-migration *vTEF* and *PIF* of the destination vertex must not be the highest in the MOCN. This situation implies that the current MO snapshot has evenly distributed management operations. In this case, the current decision-making period will be interrupted.

Benchmark	Description	
SoftwareTesting	Cloud based software testing benchmark	
GraphAnalytic	Machine learning and data mining benchmark	
DataAnalytic	Hadoop MapReduce framework	
OpenStack Rally	Cloud platform stressing benchmark	

Table 4. User workloads on our guest VMs

If MO analyzer fails to detect detachable MOs, it instead searches deferrable MOs on the hotspot. The deferrable management operations with highest MO-TEF will be deferred to the next MO snapshot control period. If none of the MO could be deferred on the selected vertex, this vertex will be tagged as unavailable. And MO analyzer re-starts the procedure from the vertex with second highest *vTEF*.

If all MOs in current MO snap could neither be migrated nor be deferred, MO analyzer will change the power states of hot spot to maximum performance. This is because the hot spot is the root cause of the global tail energy issue. Though increasing the performance of hot spot incurs extra power consumption, it helps shortening the tail energy duration and saving global energy consumption in SDDC. In emergency cases, the temporary power budget violation could be handled by energy storage devices

3.4 Committing Module

The committing module of HOPE receives the critical management operation scheduling decisions and executes corresponding operations through various cloud APIs and hardware management interfaces. It features a software-based MO handling agent and a hardware-based MO handling agent. The SW-based MO handling agent communicates with the SDDC management framework. It forwards the MO migration instructions to the corresponding control agents of computing service (e.g. OpenStack Nova), software-defined network service (e.g. OpenStack Neutron and Open vSwitch), and software-defined storage service (e.g. OpenStack Cinder and ZFS). The HW-based MO handling agent directly communicates with the physical servers in SDDC to change the power states. We elaborate our implementation in Section 4.

4. SYSTEM IMPLEMENTATION

We implement HOPE in 8KLOC of python and C on our lab's scaled-down IaaS data center. The IT hardware configurations are discussed in Section 2.2.2. In this section, we describe primary modules of the HOPE prototype.

We build a hierarchical monitoring framework that oversees system status and collects data from the power delivery architecture level to cloud middleware level. In our design, we deploy sub monitoring modules at different levels and collect data independently. All these sub-modules store data in MySQL database. They also expose RESTful API for access of resources from external services. The central monitor sends on-demand requests to these sub-modules to fetch the information. Various components in the power delivery architecture level use different communication protocols from Modbus to SNMP. We leverage OpenStack Kwapi [28] communication interface and design our own drivers to encapsulate communication APIs of the power system. We collect status data such as: peak power budget, daily energy used, battery voltage, battery charge/discharge current, iPDU port status, iPDU port readings, etc.



At the IT resource level monitoring we choose industrygrade monitoring software Zabbix [41] to collect cloud data. We monitor runtime performance information such as CPU utilization, network bandwidth, disk bandwidth and RAM of each server node. We also use Zabbix to sample the process information of management workloads on cloud control nodes, compute nodes and storage nodes.

At the cloud middleware level monitoring we leverage OpenStack Telemetry (Ceilometer) [26] metering service as our cloud monitoring tool. It is capable of collecting various metrics with the goal of not only metering but also monitoring and alarming. We use Ceilometer to capture the events of cloud components such as compute service, network service and storage service.

The design of HOPE is practical and scalable. The HOPE is scalable in accordance with the cloud controller node number in the cloud platform. Since OpenStack supports multiple cloud controllers that manage several sub-clusters in a cloud environment (e.g. 1 controller node administrates a sub-cluster of 200 compute nodes), we could deploy multiple HOPE controllers to manage the management operations within each sub-cluster.

5. EVALUATION

We comprehensively evaluate the efficiency of HOPE optimization. We first validate our graph-based analysis mechanism and critical management operation scheduling mechanism using various user workloads. After that, we deploy various workloads on guest VMs to reproduce an IaaS production environment. We choose *SoftwareTesting*, *GraphAnalytic* and *DataAnalytic* from CloudSuite [8] to mimic cloud user behaviors. We also deploy *OpenStack Rally* benchmark [29] to stress the cloud platform with concurrent management operations. Table 4 summarizes configuration details of these workloads used in our experiments.

5.1 Model Validation

We first validate two models that are discussed in Section 3.3: MO tail energy impact factor (*MO-TEF*) and tail energy impact factor of vertex (vTEF). They are the crux of designing effective management operation control. We compare our model prediction value with real measurements.

5.1.1 Validation of MO-TEF

In the *MO-TEF* test case, we construct tail energy scenarios manually. We examine the *MO-TEF* of management operation and the measured energy consumption on all the involved nodes. We vary both management operation consolidation degree and management operation distribution on physical servers. We evaluate different management operations and report the results in Figures 9 and 10. We report the relative model error (RME) as $\bar{\eta} =$



Figure 11. The evaluated management operations scenario (volume backup of VM1 from host[d] to host[f]; volume of VM5 migration from host[e] to host[d], offline deduplication on host[d], volume backup on host[e], log processing task on host[b], VM4 live migration from host[b] to host[a], VM2 snapshotting to host[d] and VM6 snapshotting to host[e].)



Figure 12. vTEF and induced energy of each host in the snapshot

Figure 13. Host power and energy before and after CMOS execution

500

350 g

300 250

image

volume2

|Md - Me'|/Me', where Md is the calculated MO-TEF, and Me real measured energy consumption on the image node.

In the first scenario, we initially commit three VM snapshotting operations from one compute node to three volume nodes (a, b, c), respectively. We deploy a volume backup MO on volume node a. It backs up a 100G volume to image node. The MO-TEF and energy consumption values are normalized for clear presentation. In this scenario, the duration T is 20 and the PDF of VM snapshotting is 0.38. The real measured energy values are shown in Figure 9. From Figure 9(a), we can see that the MO-TEF accurately reflects the induced extra energy consumption on the involved image node. The model error is within $\pm 5\%$. Apart from varying the MO consolidation on involved node, we further add interference MO (log processing) on compute node as the second scenario. The MO setups are shown in Figure 9(b). The PDF of log processing is 0.66. We can notice the model error is controlled within $\pm 10\%$.

We then change the VM snapshotting scenario to volume migration scenario and re-evaluate the accuracy of MO-TEF. Three VM volumes perform volume migration from one volume node to three volume nodes (a, b, c) respectively, while the volume node a still runs a 100G volume backup service. Similarly, we gradually consolidate migrations on volume node a. In this scenario, the duration T is 9.2 and PDF of volume migration is 0.29. We also incorporate data deduplication as interference MO on source volume node and report the evaluation results in Figure 10(b). The PDF of data deduplication is 0.59. We can notice the relative model errors of volume migration (within $\pm 5\%$) are better than VM snapshotting. This is because the VM snapshotting and log processing have more fluctuating performance and power characteristics than volume migration and data deduplication.

Summary: MO-TEF effectively indicates the severity of tail energy, which is brought by critical MO.

5.1.2 Validation of vTEF

We continue to validate the design of *vTEF*. In this test case, we construct a comprehensive SDDC scenario and demonstrate a 10-minute-long MO snapshot in Figure 11(a). The MO distribution details are described in the caption. Our goal is to demonstrate that vTEF is capable of locating the hotspots. We compare the vTEF of each node in this MO snapshot to the sum of induced energy of this node's MOCN neighbors. The rationale is that, if a node has the highest vTEF, the sum of its neighbor's induced energy in this snapshot should also be highest. For example, the value of vTEF of host[b] implies the sum of induced energy on host[a] and host[d]. To better understanding the MO correlation in this snapshot, we illustrate the MOCN structure in Figure 11(b).

We measure the average power in current MO snapshot, the sum of its neighbors' induced energy, and vTEF of each physical node. The results are shown in Figure 12. For better demonstration we report the normalized value of *vTEF* and induced energy. It shows that the *vTEF* effectively reflects the energy impacts of a given physical host.

Note that the induced energy and vTEF of compute node 1 and image node is 0. This is because they do not have direct edge in the MOCN. Though volume node 1 has higher out-degree in the MOCN, the average weights of its edges are lower than compute node 2's. Moreover, as a local interference MO, log processing has higher PDF (0.66) than data deduplication (0.59). The host[b] (compute node 2) is thus identified as a hotspot in this snapshot.

Summary: vTEF effectively indicates the hotspots in a given management operation snapshot.



Figure 14. A 1-hour long full system power trace collected from HOPE power monitor

5.2 Effectiveness of Software Based Handling

We first examine the effectiveness of hotspot/critical MO elimination based on the management operation snapshot shown in Figure 11(a). To isolate the impact of critical management operation scheduling, we deactivate the power state tuning function in data analysis module to fully represent the effectiveness of the CMOS mechanism.

Note that the compute node 2 has been identified as a hotspot in the prior evaluation. HOPE takes further step to locate the critical management operation on this node and re-schedule it. We compare the measured energy consumption during the MO snapshot and average power of each host in Figure 13. The average host power drops from 367W to 353W. The energy consumption of whole cluster drops from 0.37kWh to 0.34kWh. We further explore the CMOS execution and find that the MO1 (Log processing) on this host is located as critical MO since it is detachable. HOPE then re-schedules it to host[c]. Note that although the vTEF of host[a] is 0, HOPE does not choose it as the MO migration destination node. This is because HOPE detects that there are isolated MOs residing on this node and they cause high PIF on this node. Consequently, HOPE migrates the critical MO to compute node 3. We can observe the power consumption of host[b] drops from 430W to 375W, and the power consumption of host[c] increases from 355W to 372W.

Furthermore, both the power and energy of the image node decrease. Though the host[f].MO2 may be delayed due to the performance degradation of host[c], the execution time of host[f].MO3 in fact has been boosted due to the performance improvement of host[d] that was caused by the boosted host[b].MO3 (VM2 snapshotting). In addition, detaching host[b].MO1 improves the performance of host[d], and also benefits host[f] due to the correlation of host[d].MO2.

Summary: Our software based critical management operation scheduling mechanism can effectively eliminate the energy tails and hotspots in the SDDC, thereby reducing the energy and power consumption of an SDDC.

5.3 Full System Evaluation

We evaluate the HOPE in a typical SDDC production environment. We deploy various workloads on guest VM to reproduce an IaaS production environment, as listed in Table 4. We also deploy *OpenStack Rally* benchmark [29] to stress the cloud platform with concurrent management operations. Specifically, OpenStack Rally automates the benchmarking and functional testing for OpenStack based cloud platform. It generates cloud management operations such as VM creation, VM snapshotting, VM live migration, and etc. in the user defined pattern. We host workloads in CloudSuite on virtual machine instances with the same configuration in Section 2.



Figure 15. Normalized performance improvement when running different user workloads.

5.3.1 Full System Power Trace Analysis

We start by analyzing the effectiveness of HOPE using real traces obtained from our 8-node prototype SDDC as described in Section 2. The results are shown in Figure 14. In this experiment we deploy three VMs on each node. The VMs repeatedly run *SoftwareTesting*, *GraphAnalytic* and *DataAnalytic*, respectively. We also deploy OpenStack Rally to randomly generate management operations among the SDDC physical servers. We mark two representative regions (A, B) that reflect the typical management operation behavior.

First we observe a power demand drop in Region A. This represents a typical critical management operation deference initiated by HOPE. By zooming in to see the power trace of volume node 1, we can notice a peak power surge of up to 320W, as pointed by arrow A. Event log shows that two volume creation operations are committed on volume node 1, which makes it the hotspot in current MO snapshot. As a result the deduplication operation is identified as critical MO and is deferred to re-execute at night. The CMOS handling frees up the system resource and accelerates the rest of the management workloads. This also reduces global power consumption from 2745W to 2652W.

Region B represents a typical critical MO migration event. By viewing the zoomed power trace of compute node 2 and 3, we can notice a nearly 50W power boost on compute node 3 caused by two VM snapshotting operations. As a result, the compute 3 exhibits the highest *PIF* in the current MO snapshot and makes it the hotspot. HOPE is able to identify the log processing task on compute 3 as critical MO and migrates it to the cold spot compute node 2.

5.3.2 Performance Benefits of HOPE

We examine the effectiveness of power/energy management of HOPE by comparing it to a baseline SDDC system that does not adopt management operation handling mechanism. In this experiment we deploy 2 VMs on each compute node. All VMs host the same user workloads (*SoftwareTesting, GraphAnalytic*, and *DataAnalytic*) at a time. Figure 15 shows the results.



Figure 16. State-of-charge of battery after 1-hour operation

Our results demonstrate that HOPE significantly reduces the execution time of management operations in SDDC. *Software-Testing* gains the best management operation execution time saving (25.4%) due to its relatively flat power consumption profile.

Correspondingly, HOPE reduces the energy consumption of management operations (up to 33%) by eliminating the tail energy. We refer to the saving of energy consumption of MO as MO energy efficiency. Note that the improvement of MO energy efficiency is higher than the improvement of MO execution time. This is because the average power consumption of physical nodes decreases.

The elimination of tail energy frees up precious system resource to run user workloads. Our results show that it can increase the throughput of user workloads by up to 30%. In our experiment, the throughput of ST is calculated by the reported progress. The throughput of GA and DA could be calculated by the processed data size. The average system energy efficiency is improved by 19%.

5.3.3 Impact on Power Constrained Data Centers

We evaluate HOPE under a power-constrained SDDC scenario to further illustrate the benefits of HOPE. In recent years, many power-constrained data centers start to tap into additional energy sources and energy storage devices to improve data center sustainability and efficiency.

In this work we leverage an on-site power generation set for shaving the peak power of a SDDC. Our power generation set includes an OutbackPower FLEXMax 80 charge controller, an OutbackPower GVFX3524 inverter, and 200Ah 24V lead-acid battery system. We also deploy battery array monitor FLEXnet DC to monitor parameters of battery voltage, current and state of charge for better managing battery life. The battery system connects one of the dual supplies of SDDC servers through inverter. This promises the on time peak shaving. We set three power capping thresholds as 380W, 400W, and 420W. Power spikes over the threshold will be shaved by battery system. We measure the state-of-charge of battery system after 1-hour running of three user workloads. The results are shown in Figure 16. On average, HOPE provides nearly 20% capacity saving of energy consumption.

5.3.4 Overhead Analysis

Finally we discuss the operation overheads of HOPE. The overhead of re-scheduling a detachable management operation is a one-time virtual machine live migration. The overhead of deferring a management operation is a pausing operation of a running VM or service. The impact of deference to the performance of management operations is negligible since most of the deferrable management operations are not mission-critical. In fact, all these re-scheduling operations are designed for mitigating the tail energy and eliminating the hot spots in system. The overall performance is improved and the overall energy consumption is reduced in the real system experiments, as shown in Figure 15.

Though the fast generated logs may affect the scalability and maintainability of a cloud in the presence of relational database such as MySQL[23]. Leading cloud platform such as OpenStack now supports NoSQL databases such as MongoDB[7] to provide the horizontal scalability and high performance in DBMS. Considering modern SDDC increasingly adopts real-time log analytics to provide troubleshooting, dynamic provisioning and high performance, HOPE can intuitively leverage the log management infrastructure in current SDDC without incurring additional data management issues.

Summary: HOPE can significantly benefit the energy efficiency, user workloads throughput, and lifetime of energy storage devices of a SDDC, while only introduce negligible overheads.

6. RELATED WORK

Data center management workload: VMware [32] characterize five typical management workloads present in virtualized data center: snapshot/revert, VM patching, boot storm, after-hours maintenance and automated live migration. They study the resource usage of each scenario and provided the summarized implications for computer architectures. In this work we propose fine-grained management workload operations in cloud data centers from the power perspective. We also propose graph-based analysis method for managing workload in cloud data centers and mitigation approach.

Data center power management: Prior works make attempts to management power and energy as a resource of data center infrastructure. Power Container [31] implements a finegrained power management mechanism for application requests. vPower [40] virtualizes the entire power hierarchy in a softwarebased approach, including both power source end and power demand end. VirtualPower [24] coordinates the power demands of co-located virtual machines on a virtualized server.

Overhead and interference analysis: Several prior works proposed interference analysis in data centers, while they only consider the interference among workloads or VMs. We study the power impact of consolidated management workloads in cloud data centers. These prior contributions would be well complementary to our work. DeepDive [25] uses mathematical models and clustering techniques to detect interference in cloud data centers. DejaVu [36] employs VM clone technique to run it in a black box to detect interference. DejaVu also handles new applications and allocates resource according to demands. Paragon [5] estimates the impact of interference on performance and uses that information to assign resources to incoming workloads. Quasar [6] is a cluster management system that uses classification technique to analyze the unknown workloads and makes decisions on allocating and assigning resources.

Unconventional power provisioning: Power routing [30] proposes shuffled power distribution topologies to reduce the opportunities of coordinated power spikes, thus saving the performance throttling and capital expenditure of power delivery system. There has been recent work integrating additional green energy and onsite stored energy into data centers [9, 13, 14]. Our study is orthogonal to these hardware/facility aware power management schemes and can help improve the overall efficiency on these data centers.

7. CONCLUSION

Large-scale data centers have complex ancillary management workloads with a substantial amount of power budget and resources dedicated to them. In this paper we investigate a microlevel power management that fine tunes the power behavior of ancillary management workload sets and user's computing tasks. Our technique employs management workload clustering techniques to analyze correlation between management operations. It uses software based management workload scheduling to balance the power demand caused by consolidated management workload. We evaluate our design under various deployment scenarios of user workloads. We also evaluate our design under powerconstrained scenario. Our deployment experiences show that the proposed technique allows SDDC to reduce energy consumption by 19%, reduce management operation execution time by 25.4%, and in the meantime improve workload performance by 30%.

8. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This work is supported in part by NSF grants 1527535, 1423090, 1320100, 1117261, 0937869, 0916384, 0845721 (CAREER), 0834288, 0811611, 0720476, by SRC grants 2008-HJ-1798, 2007-RJ-1651G, by Microsoft Research Trustworthy Computing, Safe and Scalable Multi-core Computing Awards, and by three IBM Faculty Awards. Yang Hu is also supported by University of Florida Alumni Fellowship. Chao Li is also supported, in part, by NSFC 61502302 and the National Basic Research Program of China (973 Program, No. 2015CB352403).

9. REFERENCES

- Alvarez, C. 2011. NetApp deduplication for FAS and V-Series deployment and implementation guide. *Technical ReportTR-3505.* January (2011).
- [2] Amazon EC2: http://aws.amazon.com/ec2/.
- [3] Arzuaga, E. and Kaeli, D.R. 2010. Quantifying Load Imbalance on Virtualized Enterprise Servers. Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering (New York, NY, USA, 2010), 235–242.
- [4] Cidon, A., Rumble, S.M., Stutsman, R., Katti, S., Ousterhout, J. and Rosenblum, M. 2013. Copysets: Reducing the Frequency of Data Loss in Cloud Storage. *Proceedings of the* 2013 USENIX Conference on Annual Technical Conference (Berkeley, CA, USA, 2013), 37–48.
- [5] Delimitrou, C. and Kozyrakis, C. 2013. Paragon: QoS-aware Scheduling for Heterogeneous Datacenters. Proceedings of the eighteenth international conference on Architectural support for programming languages and operating systems -ASPLOS '13. (2013), 77–88.
- [6] Delimitrou, C. and Kozyrakis, C. 2014. Quasar: Resourceefficient and QoS-aware Cluster Management. Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (New York, NY, USA, 2014), 127–144.
- [7] Dirolf, K.C.& M. 2011. MongoDB: The Definitive Guide.
- [8] Ferdman, M., Adileh, A., Kocberber, O., Volos, S., Alisafaee, M., Jevdjic, D., Kaynak, C., Popescu, A.D., Ailamaki, A. and Falsafi, B. 2012. Clearing the clouds: a study of emerging scale-out workloads on modern hardware. *Proceedings of the seventeenth international conference on Architectural*

Support for Programming Languages and Operating Systems (New York, NY, USA, 2012), 37–48.

- [9] Goiri, Í, Katsak, W., Le, K., Nguyen, T.D. and Bianchini, R. 2013. Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy. *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2013), 51–64.
- [10] Govindan, S., Liu, J., Kansal, A. and Sivasubramaniam, A. 2011. Cuanta: Quantifying Effects of Shared On-chip Resource Interference for Consolidated Virtual Machines. *Proceedings of the 2Nd ACM Symposium on Cloud Computing* (New York, NY, USA, 2011), 22:1–22:14.
- [11] Hossain, M.M., Huang, J.-C. and Lee, H.-H.S. 2012. Migration Energy-Aware Workload Consolidation in Enterprise Clouds. 2012 IEEE 4th International Conference on Cloud Computing Technology and Science. (2012), 405– 410.
- [12] Koh, Y., Knauerhase, R., Brett, P., Bowman, M., Wen, Z. and Pu, C. 2007. An Analysis of Performance Interference Effects in Virtual Environments. 2007 IEEE International Symposium on Performance Analysis of Systems & Software. (2007), 200–209.
- [13] Li, C., Hu, Y., Liu, L., Gu, J., Song, M., Liang, X., Yuan, J. and Li, T. 2015. Towards sustainable in-situ server systems in the big data era. *Proceedings of the 42nd Annual International Symposium on Computer Architecture -ISCA '15*. (2015), 14–26.
- [14] Li, C., Hu, Y., Zhou, R., Liu, M., Liu, L., Yuan, J. and Li, T. 2013. Enabling datacenter servers to scale out economically and sustainably. *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture - MICRO-46*. (2013), 322–333.
- [15] Li, C., Qouneh, A. and Li, T. 2012. iSwitch: Coordinating and optimizing renewable energy powered server clusters. *Proceedings - International Symposium on Computer Architecture* (2012), 512–523.
- [16] Li, C., Wang, R., Hu, Y., Zhou, R., Liu, M., Liu, L.J., Yuan, J.L., Li, T. and Qian, D.P. 2014. Towards automated provisioning and emergency handling in renewable energy powered datacenters. *Journal of Computer Science and Technology*, 29, 4 (2014), 618–630.
- [17] Li, C., Wang, R., Li, T., Qian, D. and Yuan, J. 2014. Managing Green Datacenters Powered by Hybrid Renewable Energy Systems. *Proc. of 11th Int. Conf. on Automatic Computing*. (2014), 261–272.
- [18] Liu, L., Li, C., Sun, H., Hu, Y., Gu, J., Li, T., Xin, J. and Zheng, N. 2015. HEB : Deploying and Managing Hybrid Energy Buffers for Improving Datacenter Efficiency and Economy. *Proceedings of the 42nd Annual International Symposium on Computer Architecture - ISCA '15* (2015), 463–475.
- [19] Liu, L., Li, C., Sun, H., Hu, Y., Xin, J., Zheng, N. and Li, T. 2015. Leveraging heterogeneous power for improving datacenter efficiency and resiliency. *IEEE Computer Architecture Letters.* 14, 1 (2015), 41–45.
- [20] Liu, L., Sun, H., Li, C., Hu, Y., Xin, J., Zheng, N. and Li, T. 2016. RE-UPS: an adaptive distributed energy storage system for dynamically managing solar energy in green

datacenters. *The Journal of Supercomputing*. 72, 1 (2016), 295–316.

- [21] Liu, L., Sun, H., Li, C., Hu, Y., Zheng, N. and Li, T. 2016. Towards an Adaptive Multi-Power-Source Datacenter. *Proceedings of the 30th ACM on International Conference* on Supercomputing - ICS '16 (2016).
- [22] Moshref, M., Yu, M., Sharma, A. and Govindan, R. 2013. Scalable Rule Management for Data Centers. NSDI'13 Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation. (2013), 157–170.
- [23] Mysql 2011. MySQL Workbench.
- [24] Nathuji, R. and Schwan, K. 2007. VirtualPower: coordinated power management in virtualized enterprise systems. *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles - SOSP '07*. (2007), 265.
- [25] Novakovic, D., Vasic, N. and Novakovic, S. 2013. Deepdive: Transparently identifying and managing performance interference in virtualized environments. USENIX ATC'13 Proceedings of the 2013 USENIX conference on Annual Technical Conference. (2013), 219–230.
- [26] OpenStack Ceilometer: https://wiki.openstack.org/wiki/Ceilometer.
- [27] OpenStack Cloud Software: www.openstack.org.
- [28] OpenStack Kwapi: https://github.com/openstack/kwapi.
- [29] OpenStack Rally: https://github.com/openstack/rally.
- [30] Pelley, S., Meisner, D., Zandevakili, P., Wenisch, T.F. and Underwood, J. 2010. Power Routing: Dynamic Power Provisioning in the Data Center. *Proceedings of the Fifteenth Edition of ASPLOS on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2010), 231–242.
- [31]Shen, K., Shriraman, A., Dwarkadas, S., Zhang, X. and Chen, Z. 2013. Power Containers : An OS Facility for Fine-Grained Power and Energy Management on Multicore Servers. Proceedings of the 16th international conference on Architectural Support for Programming Languages and Operating Systems. (2013), 65–76.
- [32] Soundararajan, V. and Anderson, J.M. 2010. The Impact of Management Operations on the Virtualized Datacenter. *Proceedings of the 37th Annual International Symposium on Computer Architecture* (New York, NY, USA, 2010), 326– 337.

- [33] Splunk: http://www.splunk.com/product.
- [34] SumoLogic: http://www.sumologic.com/.
- [35] The Software-Defined Data Center: 2015. http://www.vmware.com/software-defineddatacenter/index.html.
- [36] Vasić, N., Novaković, D., Miučin, S., Kostić, D. and Bianchini, R. 2012. DejaVu: accelerating resource allocation in virtualized environments. ACM SIGARCH Computer Architecture News. 40, 1 (2012), 423.
- [37] vRealize Log Insight 3.0 Documentation Center: http://pubs.vmware.com/log-insight-30/index.jsp?topic=%2Fcom.vmware.log-insight.gettingstarted.doc%2FGUID-4E3853AA-EFBF-4004-B182-DF5F6DC3826F.html.
- [38] vRealize Operations IT Operations Management: VMware: http://www.vmware.com/ap/products/vrealize-operations. Accessed: 2015-12-13.
- [39] Wang, C., Rayan, I.A., Eisenhauer, G., Schwan, K., Talwar, V., Wolf, M. and Huneycutt, C. 2012. VScope: Middleware for Troubleshooting Time-sensitive Data Center Applications. *Proceedings of the 13th International Middleware Conference* (New York, NY, USA, 2012), 121–141.
- [40] Wang, D., Ren, C. and Sivasubramaniam, A. 2013. Virtualizing Power Distribution in Datacenters. *Proceedings* of the 40th Annual International Symposium on Computer Architecture (New York, NY, USA, 2013), 595–606.
- [41] Zabbix: The Ultimate Enterprise-class Monitoring Platform: http://www.zabbix.com/.
- [42] ZFS Filesystem: http://zfsonlinux.org/.
- [43] Zhang, I., Denniston, T., Baskakov, Y. and Garthwaite, A. 2013. Optimizing VM Checkpointing for Restore Performance in VMware ESXi. *Proceedings of the 2013* USENIX Conference on Annual Technical Conference (Berkeley, CA, USA, 2013), 1–12.
- [44] Zhou, R., Liu, M. and Li, T. 2013. Characterizing the efficiency of data deduplication for big data storage management. Workload Characterization (IISWC), 2013 IEEE International Symposium on.