e-STAB: Energy-Efficient Scheduling for Cloud Computing Applications with Traffic Load Balancing

Dzmitry Kliazovich[1], Sisay T. Arzo[2], Fabrizio Granelli[2], Pascal Bouvry[1] and Samee Ullah Khan[3]

[1] University of Luxembourg
6 rue Coudenhove Kalergi, Luxembourg
dzmitry.kliazovich@uni.lu,
pascal.bouvry@uni.lu

[2] University of Trento
Via Sommarive 14, Trento, Italy
sisay.arzo@studenti.unitn.it,
granelli@disi.unitn.it

[3] North Dakota State University
Fargo, ND 58108-6050
samee.khan@ndsu.edu

Abstract— **Energy consumption accounts for a large percentage of the operational expenses in data centers that are used as backend computing infrastructure for cloud computing. Existing solutions for energy efficiency and job scheduling are focusing on job distribution between servers based on the computational demands, while the communication demands are ignored. This work emphases the role of communication fabric and presents a scheduling solution, named e-STAB, which takes into account traffic requirements of cloud applications providing energy efficient job allocation and traffic load balancing in data center networks. Effective distribution of network traffic improves quality of service of running cloud applications by reducing the communication-related delays and congestion-related packet losses. The validation results, obtained from the GreenCloud simulator, underline benefits and efficiency of the proposed scheduling methodology.**

Keywords—energy-efficient scheduling; cloud computing; load balancing; data centers.

I. Introduction

Cloud computing is entering our lives and dramatically changing the way people parse information. Cloud provides platforms enabling a large variety of terminal devices owned by individuals to operate. There are about 1.5 billion computers [1] and 6 billion mobile phones [2] in the world today. The next generation of user devices [3] offers not only constant readiness for operation, but also constant information consumption. In such an environment, computing, information storage, and communication becomes a utility. Cloud computing is an effective way to offer manageable and secure infrastructure with reduced cost of operations [4].

Cloud computing relies on the data centers as their primary backend computing infrastructure. Currently, over 500 thousand data centers are deployed worldwide [5]. The operation of large geographically distributed data centers requires a considerable amount of energy that accounts for a large slice of the total operational costs [6-7]. The Gartner group estimates that the energy consumption accounts for up to 10% of the current data center operational expenses (OPEX), and this estimate may rise to 50% in the next few years [8]. The cost of energy for running servers may already be greater than the cost of the hardware itself [9], [10]. In 2010, data centers consumed about 1.5% of the world’s electricity [11]. In terms of CO_2 emissions, it corresponds to more than 50 million metric tons annually.

Energy efficient computing has never been the primary goal of the IT industry. Since the 1980s, the only target for the IT industry has been to deliver more and faster computational power, which was normally achieved by packing more electronics into a smaller space, and running the packaging at a higher frequency. Higher power consumption generates heat and requires an accompanying cooling system that costs in the range of $2 to $5 million per year for classical data centers [8]. In most cases, the cooling systems require more power than the core IT equipment [12], [13].

In data centers, to ensure high levels of reliability, there is an overprovision of computing, storage, power distribution, and cooling infrastructures [27]. Therefore, the energy consumption is not proportionate to the workload processing. To measure this inefficiency, the Green Grid Consortium [14] developed two metrics, the Power Usage Effectiveness (PUE) and the Data Center Infrastructure Efficiency (DCIE) [15]. Both PUE and DCIE account for the proportion of power delivered to the IT equipment relative to the total power consumed by the data center facility. Currently, roughly 40% of the total energy is consumed by the IT equipment [16]. Other systems contributing to the data center energy consumptions are cooling and power distribution systems that account for approximately 45% and 15% of total energy consumption, respectively.

There are two main alternatives for making data center consume less energy: (a) shutting it down or (b) scaling down its performance. The former method, commonly referred to as Dynamic Power Management (DPM) results in most of the savings as the average workload often stays below 30% in cloud computing systems [17]. The latter corresponds to the Dynamic Voltage and Frequency Scaling (DVFS) technology that can adjust the hardware performance and power consumption to match the corresponding characteristics of the workload.

In this paper, we design a scheduling approach for a cloud computing system that optimizes the energy consumption of the data center IT equipment while providing load balancing of traffic flowing within the data center network. An effective distribution of network traffic improves Quality of Service (QoS) of running cloud applications by reducing communication-related delays and congestion-related packet losses. The validation results, obtained from the GreenCloud simulator [29], underline the benefits and efficiency of the proposed scheduling methodology. Specifically, the main contributions of the paper are the following:

- Design of a scheduler which optimizes energy efficiency and load balancing of network traffic in cloud computing data centers;
- Development of a formal model followed by the e-STAB scheduler for the selection of a proper server, a rack and a module for each incoming to the data center workload in real time.
- Comprehensive performance evaluation of the e-STAB scheduler and comparison with other consolidation-based data center schedulers.

The scheduler proposed in this work is different from our previously presented DENS scheduler [28] in the following: a) the way network traffic is analyzed, b) possibility to load balance the traffic and c) metrics used for the selection of the data center servers, racks and modules.

The rest of the paper is structured as follows: Section II presents the design and operational strategy of the proposed scheduler outlining the motivation of different design choices. Section III presents the details of the simulation scenario and discusses the obtained evaluation results. Finally, Section IV concludes the paper with final remarks and objectives for future research on the topic.

II. Energy-Efficient Scheduling with Traffic Load Balancing

A. *Motivation*

Job scheduling is at the heart of the successful operation of power management in data centers. Most of the existing approaches for job scheduling in data centers focus exclusively on the job distribution between computing servers [18] targeting energy efficiency [19] or thermal awareness [20]. However, only a few approaches consider data center network characteristics [21-23].

As most of the energy savings come from DPM-like power management procedures [17], job schedulers follow workload consolidation policy by maximizing the load on the operational computing servers and increase the number of idle servers that can be put into the “sleep” mode. Such scheduling policy works well in systems that can be abstracted as a homogenous pool of computing servers. However, policies should be adapted to network topologies in real data centers. For example, considering the most widely used fat tree data center architecture [24] presented in Fig. 1, blind concentration scheduling may group all of the highly loaded computing servers on a few racks, which creates a bottleneck for network traffic at a rack or an aggregation switch.

At the rack level, all of the servers are usually connected using Gigabit Ethernet (GE) interfaces. Typically, racks hosts up to 48 servers and have only two links of 10GE connecting them to the aggregation network. This corresponds to the mismatch of $48GE\ /\ 20GE\ = 2.4$ between the incoming and the outgoing bandwidth capacities. Being implemented in a data center with cloud applications requiring communication, the scheduler should tradeoff workload concentration with the load balancing of network traffic.

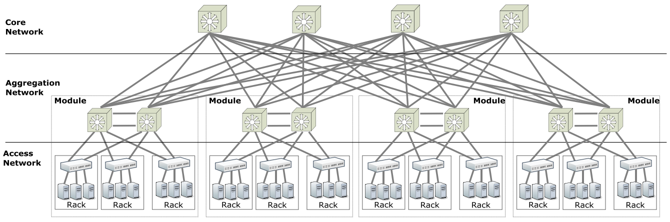

Figure 1. Three-tier data center architecture.

B. *e-STAB Scheduler*

The **e**nergy-efficient **s**cheduler for cloud computing applications with **tra**ffic load **b**alancing (e-STAB) is designed to optimize energy consumption of cloud computing data centers. The e-STAB scheduler treats communicational demands of the jobs equally important to that of the computing requirements. e-STAB is a scheduler aiming to: **(a)** balance communication flows produced by the jobs and **(b)** consolidate jobs on a minimum amount of the computing servers. As network traffic can be highly dynamic and often difficult to predict [31], the e-STAB scheduler analyses both the load on the network links and the occupancy of outgoing queues at the network switches. e-STAB allocates jobs favoring network resources that offer the most of the available bandwidth and penalizes resources for which the load approaches the available transmission capacity when the traffic queues growing in size. Queuing analysis aids in preventing a buildup of network congestion. Such techniques are already implemented in several transport-layer protocols [32] that estimate buffer occupancy of the network switches and can react before congestion related losses occur.

The e-STAB scheduling policy can be defined with the following two steps executed for every incoming cloud computing data center workload:

- **Step 1:** Select a group of servers S connected to the data center network with the highest available bandwidth, provided that at least one of the servers in S can accommodate the computational demands of the scheduled job. The available bandwidth is defined as an unused capacity of the link or a set of links connecting the group of servers S to the rest of the data center network.
- **Step 2:** Within the selected group of servers S, select a computing server with the smallest available computing capacity, but sufficient to satisfy the computational demands of the scheduled task.

In fat tree data centers (see Fig. 1), the servers are arranged into racks, forming a set of racks R. Subsequently, racks form the set of modules M. Therefore, to select the group of servers with the largest available bandwidth in Step 1, e-STAB must first find a module $m_i \epsilon M$ such that

$$Am(\mathrm{m_i}) = \max_{\forall \mathrm{m} \in \mathrm{M}}\big(Am(m)\big), \qquad (1)$$

where Am is the available bandwidth of a module m_i computed on a per-server basis. For a module $m_i \in M$ the available bandwidth can be computed as

$$Am_i = \frac{Cm_i - \lambda m_i}{Sm_i}, \qquad (2)$$

where Cm_i is the transmission capacity of a module i, calculated as a sum of maximum transmission speeds of all links connecting a module i to the network of core switches, λm_i is a currently effective transmission rate of the traffic, and Sm_i is a number of servers hosted in the module.

Equation (2) provides an instantaneous measure of the available bandwidth. However, as most of the transmissions are bursty, which either use full network link capacity for a short time or leave the link unutilized. Therefore, the available capacity must be calculated as an average over the time interval T:

$$Am_i(t) = \frac{1}{T}\int_t^{t+T} \left(\frac{Cm_i - \lambda m_i(t)}{Sm_i}\right) dt =$$

$$= \frac{1}{Sm_i}\left(Cm_i - \frac{1}{T}\int_t^{t+T} \lambda m_i(t)\, dt\right). \qquad (3)$$

Similar to the case of modules, for identifying a rack with the most of the available bandwidth, e-STAB must find a rack $r_j \in R$ such that

$$Ar(\mathrm{r_j}) = \max_{\forall \mathrm{r} \in \mathrm{R}}\big(Ar(r)\big), \qquad (4)$$

where Ar is the available bandwidth of a rack r_j computed on a per-server basis. For a module $r_j \in R$ the available bandwidth can be computed as

$$Ar_j(t) = \frac{\frac{1}{T}\int_t^{t+T}\big(Cr_j - \lambda r_j(t)\big)\, dt}{Sr_j} =$$

$$= \frac{1}{Sr_j}\left(Cr_j - \frac{1}{T}\int_t^{t+T} \lambda r_j(t)\, dt\right), \qquad (3)$$

where Cr_j is the transmission capacity of a rack j, calculated as a sum of maximum transmission speeds of all links connecting a rack j to the network of aggregation switches, λr_j is a currently effective transmission rate of the traffic, and Sr_j is a number of servers hosted in the rack.

One of the main goals of the e-STAB scheduler is to achieve load balanced network traffic and prevent network congestion. A helpful measure is the available bandwidth per compute node within the data center. However, such a measure does not capture the system dynamics, such as sudden increase in the transmission rate of the cloud applications.

To have a more precise measure of the network congestion, e-STAB scales the measures of the available bandwidth $Am_i(t)$ and $Ar_j(t)$ with the component related to the size of the bottleneck queue

$$Q(t) = 1 - \frac{1}{T}\int_t^{t+T}\left(e^{-\left(\frac{\rho\cdot(q(t)-1)}{Q_{max}}\right)^{\varphi}}\right) dt, \qquad (5)$$

where $q(t)$ is an instantaneous occupancy of the queue measured either in bytes or packets at the time t, Q_{max} is the maximum allowed size of the queue, while parameters ρ and φ control the shape of the function and are explained in the following paragraph.

Equation (5) is an integral version of the Weibull cumulative distribution presented in Fig. 2. It aims to favor the empty queues or queues with a minimum occupancy and penalize highly loaded queues that are on the threshold of buffer overflow (or on the threshold of losing packets). Parameter ρ controls the position of the falling edge of $Q(t)$ with the respect to the level of queue occupancy. Smaller values of ρ make e-STAB sensitive to even the smallest levels of congestion, while the larger values of ρ make e-STAB react only to the highly loaded queues and may lead to congestion-related losses. As confirmed by the experiments, presented in Section III of the paper, typical values for ρ must be selected to locate the falling edge of $Q(t)$ in between 0.3-0.5 of the maximum size of the queue. Parameter φ controls the shape of the falling slope of the $Q(t)$ function. It determines the speed of reaction to the growing congestion and should be chosen not to be sharp for a smooth system behavior.

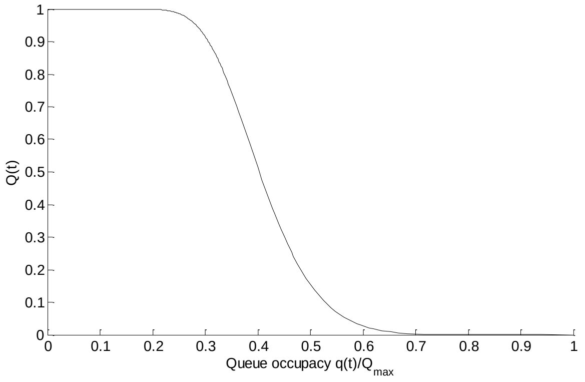

Figure 2. Queue-size related component of the STUB scheduler.

By utilizing the aforementioned information (Eq. (3) and eq. (5)), the available per-server bandwidth can be appropriately adjusted with the $Q(t)$ metric. This will allow e-STAB to select a module as:

$$Fm_i(t) = Am_i(t) \cdot Qm_i(t) =$$

$$= \frac{1}{T}\int_t^{t+T}\left(\frac{(Cm_i - \lambda m_i(t))\cdot e^{-\left(\frac{\rho\cdot qm_i(t)}{Qm_i.max}\right)^{\varphi}}}{Sm_i}\right) dt \qquad (6)$$

and a rack as

$$Fr_j(t) = Ar_j(t) \cdot Qr_j(t) =$$

$$= \frac{1}{T}\int_{t}^{t+T}\left(\frac{(Cr_{j}-\lambda r_{j}(t))\cdot e^{-(\frac{\rho\cdot qr_{j}(t)}{Qr_{j}.max})^{\varphi}}}{Sr_{j}}\right) dt \qquad (7)$$

where $Qm_i(t)$ and $Qr_j(t)$ are weights associated with occupancy levels of the queues, $qm_i(t)$ and $qr_j(t)$ are sizes of the queues at time t, and $Qm_i.max$ and $Qr_j.max$ are maximum allowed sizes of the queues at the module i and rack j, respectively.

Figure 3 presents the evolution of $Fm_i(t)$ and $Fr_i(t)$ with the respect to different values of the network traffic and buffer occupancy. The function is insensitive to the level of utilization of the network links for highly loaded queues, while for the lightly loaded queues, the links with the lighter load are preferred to the heavily utilized links.

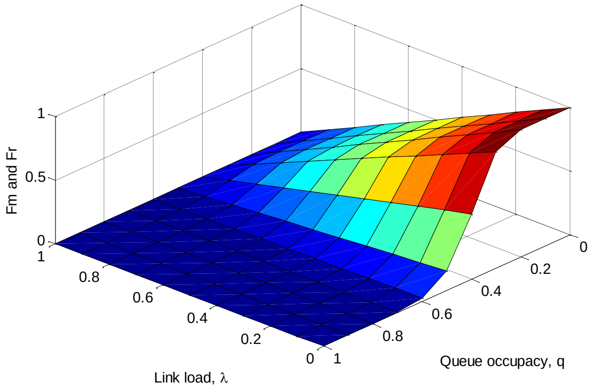

Figure 3. Selection of racks and modules by the STAB scheduler.

Having selected a proper module and a rack based on their traffic load and congestion state indicated by the queue occupancy, we must select a computing server for the job execution. To do so, we must analyze energy consumption profile of the servers.

According to the basic model, an idle server consumes around two-thirds of its peak load to keep memory, disks, and I/O resources running, while the rest of the power is consumed by the CPU and scaled linearly with the offered computing load [25]. More precise models [6] suggest non-linear power consumption models. In this work, we rely on a more detailed energy consumption model, which is based on the energy-consumption benchmarks for different severs from a number of manufactures [26]:

$$P(l) = P_{idle} + \frac{P_{peak}-P_{idle}}{2}(1 + l - e^{-\frac{l}{\tau}}), \qquad (8)$$

where $P(l)$ is the energy consumed by the server operating at the load level $l \in [0,1]$, P_{idle} and P_{peak} are a server consumptions at the minimum and the maximum load levels respectively, τ is a scaling coefficient, that is typically in the range of $[0.5, 0.8]$, and corresponds to the utilization level at which the server attains asymptotic power consumption.

In DVFS is used, then the power consumption of a server can be reduced proportionally to $V^2 \cdot f$, where V is a voltage and f is a frequency of the chip. Voltage reduction requires a frequency downshift, which implies a cubic relationship from f in the CPU power consumption. As a result, for a DVFS-enabled server Eq. (8) can be rewritten as follows:

$$P(l) = P_{idle} + \frac{P_{peak}-P_{idle}}{2}(1 + l^3 - e^{-(\frac{l}{\tau})^3}). \qquad (9)$$

Knowing the energy consumption of a server, we can derive a metric to be used by the e-STAB scheduler for server selection as follows:

$$Fs_k(t) = \frac{1}{T}\int_{t}^{t+T}\left(\frac{1}{1+e^{-\frac{10}{\varepsilon}\left(l_k(t)-\frac{\varepsilon}{2}\right)}} - \frac{1}{2}\left(1-\frac{P_{idle}}{P_{peak}}\right)\right.$$
$$\left.\left(1 + l_k(t)^3 - e^{-\left(\frac{l_k(t)}{\tau}\right)^3}\right) dt, \qquad (10)$$

where $l_k(t)$ is an instantaneous load of server k at time t and T is an averaging interval. While the second summand under the integral in Eq. (10) is a reverse normalized version of Eq. (9), the first summand is a sigmoid designed to penalize selection of idle servers for job execution. We must note that whenever possible, it is beneficial to maximize the number of idle servers to decrease energy consumption. This is due to the fact that even an idle server consumes around two-thirds of their peak energy consumption [25]. The parameter ε corresponds to the CPU load of an idle server required to keep an operating system and virtual machines running. Figure 4 presents a chart for $Fs_k(t)$.

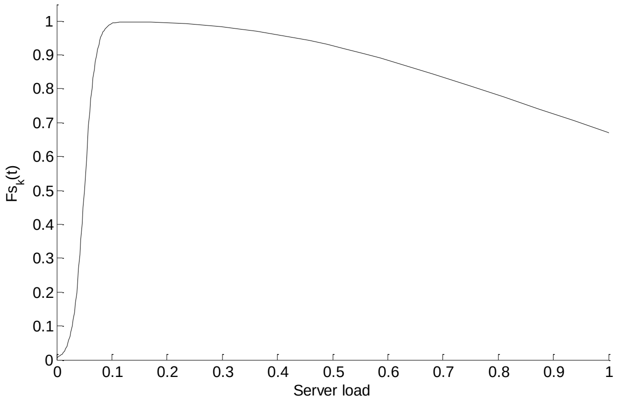

Figure 4. Selection of computing servers by the STUB scheduler.

Taking into account the metrics obtained for the selection of a module (Eq. (6)), a rack (Eq. (7)), and a server (Eq. (10)), we can define a joint metric calculated by the e-STAB scheduler for every incoming workload as follows:

$$G_{i,j,k}(t) = (Fm_i(t),\ Fr_j(t),\ Fs_k(t)). \qquad (11)$$

Then, for any two states, characterized by a unique combination of indexes i, j and k representing a module, a rack and a server, of the system $g_a = Fm_{\alpha(a)}, Fr_{\beta(a)}, Fs_{\gamma(a)}$ and $g_b = Fm_{\alpha(b)}, Fr_{\beta(b)}, Fs_{\gamma(b)}$, respectively, the following relation holds:

$$G(g_a, g_b) \equiv \begin{cases} \gamma(a), \text{if } \ Fm_a(t) > Fm_b(t) \\ \gamma(a), \ \text{if } Fm_a(t) = Fm_b(t) \cap Fr_a(t) > Fr_b(t) \\ \gamma(a), \ \text{if } Fm_a(t) = Fm_b(t) \cap Fr_a(t) = Fr_b(t) \cap, \\ \quad \cap Fs_a(t) > Fs_b(t) \\ \gamma(b), \ \text{otherwise} \end{cases} \tag{12}$$

where $\gamma(a)$ and $\gamma(b)$ are indexes of computing servers for the state a and state b of the system, respectively.

III. Performance Evaluation

Performance of the proposed e-STAB scheduling methodology was tested by implementing its functionalities in the GreenCloud simulator [29]. GreenCloud is a packet-level cloud computing simulator focusing on cloud communications and energy efficiency. It offers tools to monitor energy consumption in data center servers, switches and routers, while the power management solutions, such as DVFS and DPM [17], are implemented in both computing hardware and network equipment. The GreenCloud simulator is currently available for download under General Public License (GPL) agreement [30].

A. Simulation Scenario

The most widely used three tier data center topology [24] was selected for simulations. It comprised of 960 servers arranged into 20 racks interconnected by 4 core and 8 aggregation switches. The network links interconnecting the core and aggregation switches as well as the aggregation and access switches of 10 Gb/s. The bandwidth of the access links connecting computing servers to the top-of-rack switches is 1 Gb/s. The propagation delay of all links is set to 3.3 µs. Table I presents a summary of simulation setup parameters.

TABLE I. Simulation Setup Parameters

	Parameter	Value
Topology	Core switches	4
	Aggregation switches	8
	Access switches	20
	Servers	960
	Access links	1 Gb/s, 3.3 µs
	Aggregation links	10 Gb/s, 3.3 µs
	Core links	10 Gb/s, 3.3 µs
Data center	Average load of data center	0.3
	Simulation time	60 minutes
	Distribution of job arrival	Exponential
	Power management in servers	DVFS/DPM
	Power management in switches	DPM

TABLE II. Power Profiles of Data Center Hardware

Hardware	Power Consumption (W)				
	Chassis	*Line card*	*Port*	*Peak*	*Idle*
Core switches	1558	1212	27		
Aggregation switches	1558	1212	27		
Access switches	146	-	0.42		
Servers				301	200

The process of job arrival is exponentially distributed to mimic realistic arrival of the customers to the system. The e-STAB scheduler makes a resource allocation decision for every incoming task. Then, as soon the server for task execution is known the task description is fragmented into IP packets and sent over the data center network. The size of the task description is chosen to fit into three IP packets of 1500 bytes each.

As soon as the task description is received at the server, the task execution is started. Upon completion, tasks generate an output destined to the end user, which is sent over the data center network up to the core switches. The path segment which corresponds to the wide-area network connection between the data center and the end user is modeled and left of scope of this paper.

Task outputs are transmitted using Transmission Control Protocol (TCP). The TCP can control sending rate constantly attempting to match network path bandwidth (flow control) and resolve any congestion- or link-related data losses occurred in the network (error control). The competition among different data flows in topology multiplexing points, like a rack switch or an aggregation switch, can be resolved and congestion-related packet drops can be recovered with retransmissions.

Each simulation runs for 60 minutes, during which around 30% of data center resources are utilized on average. All servers and switches which remain idle became a subject to enabling the sleep mode, or DPM. In addition, DVFS power management is used in servers to improve power consumption.

B. Results

The performance of e-STAB scheduler is compared against Green scheduler. The Green scheduler aims to consolidate all the incoming workloads in a minimum amount of computing servers based on the jobs' computing requirements. This comparison aims to underline the advantages of considering traffic patterns of cloud applications and taking them into account, while allocating data center resources for their execution, implemented in e-STAB scheduler. The performance of both schedulers is compared in terms of the produced network load, time of job execution and impact on the energy consumption of the system.

Figure 5 presents a distribution of server load for both schedulers. The Green scheduler consolidates execution of all the arriving workloads on servers numbered from 0 to 300. Having performed no analysis of the network traffic, the Green scheduler often selects multiple servers from the same rack. The e-STAB scheduler uses the same number of servers for serving computing demands in total. However, they are selected in bulks which distributed evenly through the whole pool of data center servers. The process of such selection is guided by the measure of per-server available bandwidth taken into account by the e-STAB scheduler.

Figure 6 presents the distribution of the traffic load measured in the uplink at different racks. As expected, the consolidation policy of the Green scheduler cannot achieve a balance of the traffic in data center network. The racks number 1-7 are overloaded, while the rest of the racks, numbered 8 through 20, stay with almost no traffic load. On the contrary, the e-STAB scheduler can balance the jobs according to their traffic demands. It leads to almost equal distribution of traffic load among the racks in the data center.

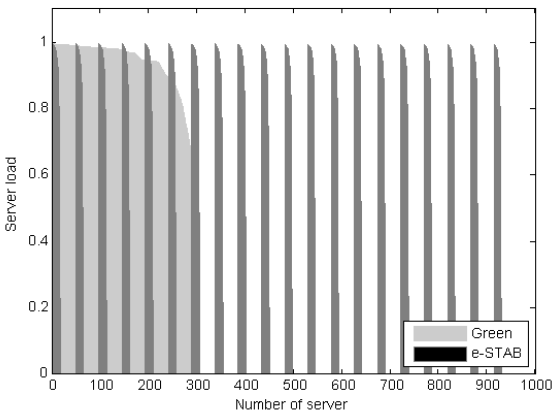

Figure 5. Distribution of the workload processing among servers.

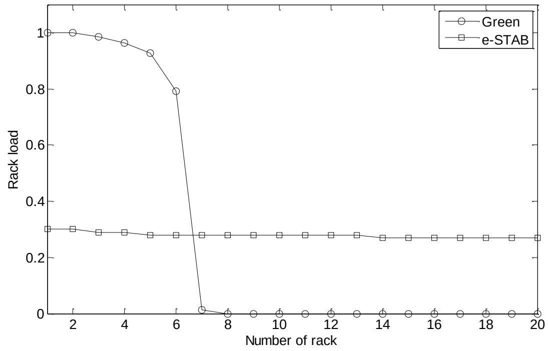

Figure 6. Distribution of the traffic load among racks.

Table III presents an analysis of energy consumption of the data center hardware under operating different power management policies for both of the schedulers. With no power management enabled, in both cases the data center consumes around 260 kWh. The major part of this energy (around 80%) is spent to power computing servers.

TABLE III. ENERGY CONSUMPTION

Scheduler	**Power Management**		**Energy Consumed (kW·h)**				
	Servers	***Switches***	***Servers***	***Core switches***	***Aggr. switches***	***Access switches***	***Total***
Green	No	No	221	11.9	23.9	3.8	260.8
	DVFS/ DPM	No	85.9	11.9	23.9	3.8	125.7
	DVFS/ DPM	DPM	85.9	6.0	11.9	1.4	105.2
e-STAB	No	No	221	11.9	23.9	3.8	260.8
	DVFS/ DPM	No	85.9	11.9	23.9	3.8	125.7
	DVFS/ DPM	DPM	85.9	11.9	23.9	3.8	125.7

Enabling power management in computing servers with both DVFS and DPM schemes leads to the significant reductions of approximately 52% and brings the total consumed power down to 125.7 kWh for both e-STAB and Green schedulers. With power management enabled in both servers and switches, the Green scheduler shows even further reduction of approximately 16% or 20.5 kWh. However, enabling DPM power management in switches is not a common practice. Typically, data center switches operate continuously, especially in the core and aggregation networks, to provide stable connectivity. Therefore, in the most common setup the servers run DVFS/DPM, while the switches have no power management enabled. In such a setup, both schedulers lead to the same system-level power consumption of 125.7 kWh. This fact confirms that e-STAB provides energy-efficient scheduling with traffic load balancing at the no expense in energy consumption.

Figure 7 presents a record of task execution delay measured as a difference in time between when all computing work is finished at the server and the moment the task's output is successfully directed out of the data center by the appropriate core switch. With the Green scheduler task completion delay experiences high variation and has an average of 78 milliseconds, most of which corresponds to the time data packets wait in heavily loaded transmission queues. On the contrary, with e-STAB the delay is well bounded and stays at the minimum with an average of 50 milliseconds most of which corresponds to the propagation delay of the network links.

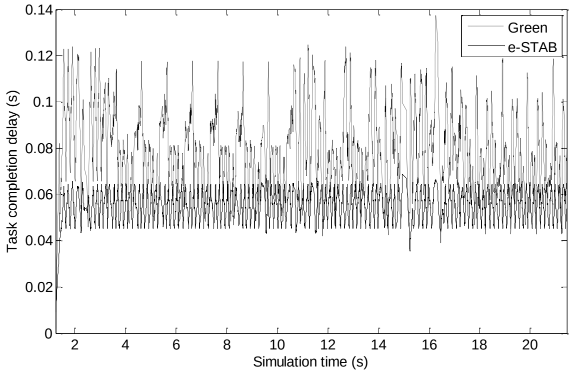

Figure 7. Tasks completion delay.

IV. CONCLUSIONS

In this paper we emphasize the role of communication fabric in modern cloud computing data centers and present a scheduling methodology e-STAB that while optimizing system energy consumption provides traffic load balancing in data center networks. Distribution of network flows produced by the running user applications helps to avoid congestion hotspots and packet losses due to the overflow in switches' buffers. As a result, e-STAB improves quality of service of running cloud applications by reducing the communication-related delays and congestion-related packet losses.

The validation results, obtained from the GreenCloud simulator for the three-tier data center architecture, underline benefits and efficiency of the proposed scheduling methodology and confirm that improvements come at no increase in the energy consumption for commonly power management schemes in data centers.

Future work will focus on the e-STAB implementation in realistic testbeds and testing.

Acknowledgment

The authors would like to acknowledge the funding from National Research Fund, Luxembourg in the framework of ECO-CLOUD project (C12/IS/3977641) and Marie Curie Actions of the European Commission (FP7-COFUND). Samee U. Khan’s work was partly supported by the Young International Scientist Fellowship of the Chinese Academy of Sciences, (Grant No. 2011Y2GA01).

References

[1] Internet World Statistics, available at http://www.internetworldstats.com.

[2] “Forecast: Mobile Data Traffic and Revenue, Worldwide, 2010-2015,” Market report, Gartner Inc., 2011.

[3] Google Glass project, available at https://plus.google.com/ 111626127367496192147.

[4] A. Weiss, “Computing in the clouds,” netWorker, vol. 11, no. 4, pp. 16–25, 2007.

[5] “State of the Data Center 2011,” Emerson Network Power, 2011.

[6] X. Fan, W.-D. Weber, and L. A. Barroso, “Power Provisioning for a Warehouse-sized Computer,” In Proceedings of the ACM International Symposium on Computer Architecture, San Diego, CA, June 2007.

[7] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, “No “Power” Struggles: Coordinated Multi-level Power Management for the Data Center,” APLOS 2008.

[8] Gartner Group, available at: http://www.gartner.com/

[9] A. Vasan and A. Sivasubramaniam “Worth their watts?-an empirical study of datacenter servers,” IEEE 16th International Symposium on High Performance Computer Architecture (HPCA), 2010.

[10] IDC, “Worldwide Server Power and Cooling Expense 2006-2010,” Market Analysis, 2006.

[11] J. G. Koomey, “Growth in Data center electricity use 2005 to 2010,” Analytics Press, 2011.

[12] “Reducing Data Center Cost with an Air Economizer,” Intel IT@Intel Brief, August 2008.

[13] N. Rasmussen, “Calculating Total Cooling Requirements for Data Centers,” White Paper #25, American Power Conversion, 2007.

[14] The Green Grid Consortium, available at http://www.thegreengrid.org/.

[15] “Green Grid Data Center Power Efficiency Metrics: PUE and DCIE,” White paper #6, The Grid Grid, 2008.

[16] R. Brown et al. “Report to congress on server and data center energy efficiency: public law 109-431,” Lawrence Berkeley National Laboratory, Berkeley, 2008.

[17] J. Liu, F. Zhao, X. Liu, and W. He, “Challenges Towards Elastic Power Management in Internet Data Centers”, in Proceedings of the 2nd International Workshop on Cyber-Physical Systems (WCPS 2009), in conjunction with ICDCS 2009., Montreal, Quebec, Canada, June 2009.

[18] Ying Song, Hui Wang, Yaqiong Li, Binquan Feng, and Yuzhong Sun, “Multi-Tiered On-Demand Resource Scheduling for VM-Based Data Center,” IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID), pp. 148 – 155, May 2009.

[19] A. Beloglazov, and R. Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers,” IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 826 – 831, May 2010.

[20] Qin 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 Transactions on Parallel and Distributed Systems, vol.19, no. 11, pp. 1458 – 1472, November 2008.

[21] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic Flow Scheduling for Data Center Networks,” in Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI '10), San Jose, CA, April 2010.

[22] A. Stage and T. Setzer, “Network-aware migration control and scheduling of differentiated virtual machine workloads,” in Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, International Conference on Software Engineering. IEEE Computer Society, Washington, DC, May 2009.

[23] Xiaoqiao Meng, V. Pappas, and Li Zhang, “Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement,” IEEE INFOCOM, San Diego, California, March 2010.

[24] “Cisco Data Center Infrastructure 2.5 Design Guide,” Cisco press, March 2010.

[25] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, “Energy-aware server provisioning and load dispatching for connection-intensive internet services,” the 5th USENIX Symposium on Networked Systems Design and Implementation, Berkeley, CA, USA, 2008.

[26] Server Power and Performance characteristics, available on: http://www.spec.org/power_ssj2008/

[27] G. L. Valentini, W. Lassonde, S. U. Khan, N. Min-Allah, S. A. Madani, J. Li, L. Zhang, L. Wang, N. Ghani, J. Kolodziej, H. Li, A. Y. Zomaya, C.-Z. Xu, P. Balaji, A. Vishnu, F. Pinel, J. E. Pecero, D. Kliazovich, and P. Bouvry, "An Overview of Energy Efficiency Techniques in Cluster Computing Systems," Cluster Computing, vol. 16, no. 1, pp. 3-15, 2013.

[28] D. Kliazovich, P. Bouvry, and Samee U. Khan, “DENS: Data Center Energy-Efficient Network-Aware Scheduling,” Cluster Computing, special issue on Green Networks, pp. 1-11, 2011.

[29] D. Kliazovich, P. Bouvry, and S. U. Khan, “GreenCloud: a packet-level simulator of energy-aware cloud computing data centers,” Journal of Supercomputing, special issue on Green Networks, pp. 1 - 21, 2010.

[30] Greencloud - The green cloud simulator, available at http://greencloud.gforge.uni.lu/.

[31] Aimin Sang and San-qi Li, “A predictability analysis of network traffic,” Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 1, pp.342-351, 2000.

[32] C. Barakat, E. Altman and W. Dabbous, “On TCP performance in a heterogeneous network: a survey,” IEEE Communications Magazine, vol. 38, no. 1, pp. 40-46, Jan 2000.