Chapter 4
Simulation and Performance Analysis of Data Intensive and Workload Intensive Cloud Computing Data Centers

Dzmitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan

4.1 Introduction

Data centers are becoming increasingly popular for the provisioning of computing resources. The cost and operational expenses of data centers have skyrocketed with the increase in computing capacity [1]. Energy consumption is a growing concern for data center operators. It is becoming one of the main entries on a data center operational expenses (OPEX) bill [2, 3]. The Gartner Group estimates energy consumptions to account for up to 10% of the current OPEX, and this estimate is projected to rise to 50% in the next few years [4]. However, computing-based energy consumption is not the only power-related portion of the OPEX bill. High power consumption generates heat and requires an accompanying cooling system that costs in a range of $2–$5 million per year for classical data centers [5]. Failure to keep data center temperatures within operational ranges drastically decreases hardware reliability and may potentially violate the service level agreement (SLA) with the customers.

From the perspective of energy efficiency, a cloud computing data center can be defined as a pool of computing and communication resources organized in the way to transform the received power into computing or data transfer work to satisfy user demands. The first power saving solutions focused on making the data center hardware components power efficient. Technologies, such as dynamic voltage and frequency scaling (DVFS), and dynamic power management (DPM) [6], were

D. Kliazovich (✉) • P. Bouvry
University of Luxembourg, 6 rue Coudenhove Kalergi, Luxembourg
e-mail: Dzmitry.Kliazovich@uni.lu; pascal.bouvry@uni.lu.

S.U. Khan
North Dakota State University, Fargo, ND 58108-6050, USA
e-mail: samee.khan@ndsu.edu

C. Kachris et al. (eds.), *Optical Interconnects for Future Data Center Networks*, Optical Networks, DOI 10.1007/978-1-4614-4630-9_4,

extensively studied and widely deployed. Because the aforementioned techniques rely on power-down and power-off methodologies, the efficiency of these techniques is at best limited. In fact, an idle server may consume about two-thirds of its peak load [7].

Because the workload of a data center fluctuates on a weekly (and in some case on hourly) basis, it is a common practice to overprovision computing and communicational resources to accommodate the peak load. In fact, the average load accounts only for 30% of data center resources [8]. This allows putting the rest of the 70% of the resources into a sleep mode for most of the time. However, achieving the above requires central coordination and energy-aware workload scheduling techniques. Typical energy-aware scheduling solutions attempt to: (a) concentrate the workload in a minimum set of the computing resources and (b) maximize the amount of resource can be put into sleep mode [9].

Most of the current state-of-the-art research on energy efficiency has predominantly focused on the optimization of the processing elements. However, as recorded in earlier research, more than 30% of the total computing energy is consumed by the communication links, switching and aggregation elements. Similar to the case of processing components, energy consumption of the communication fabric can be reduced by scaling down the communication speeds and cutting operational frequency along with the input voltage for the transceivers and switching elements [10]. However, slowing the communicational fabric down should be performed carefully and based on the demands of user applications. Otherwise, such a procedure may result in a bottleneck, thereby limiting the overall system performance. A number of studies demonstrate that often a simple optimization of the data center architecture and energy-aware scheduling of the workloads may lead to significant energy savings. Reference [11] demonstrates energy savings of up to 75% that can be achieved by traffic management and workload consolidation techniques.

In this chapter, we survey power-saving techniques implemented at both component and system levels. In energy efficiency optimization we focus on both computing and communication fabrics. As the system level, energy-efficient network-aware scheduling solutions are presented. Finally a simulation environment, named GreenCloud, for advanced energy-aware studies of cloud computing data centers in realistic setups is presented. GreenCloud is developed as an extension of a packet-level network simulator ns-2 [12]. Unlike few existing cloud computing simulators such as CloudSim [13] or MDCSim [14], GreenCloud extracts, aggregates, and makes information about the energy consumed by computing and communication elements of the data center available in an unprecedented fashion. In particular, a special focus is devoted to accurately capture communication patterns of currently deployed and future data center architectures.

4.2 Simulating Energy-Efficient Data Centers

In this section, we present the main aspects of design of energy-efficient data centers, survey the most prominent architectures, and describe power-saving techniques implemented by individual data center components.

4.2.1 Energy Efficiency

Only a part of the energy consumed by the data center gets delivered to the computing servers directly. A major portion of the energy is utilized to maintain interconnection links and network equipment operations. The rest of the electricity is wasted in the power distribution system, dissipates as heat energy, and used up by air-conditioning systems. In light of the above discussion, we distinguish three energy consumption components: (a) computing energy, (b) communicational energy, and (c) the energy component related to the physical infrastructure of a data center.

The efficiency of a data center can be defined in terms of the performance delivered per watt, which may be quantified by the following two metrics: (a) Power Usage Effectiveness (PUE) and (b) Data Center Infrastructure Efficiency (DCiE) [15,16]. Both PUE and DCiE describe which portion of the totally consumed energy gets delivered to the computing servers.

4.2.2 Data Center Architectures

Three-tier trees of hosts and switches form the most widely used data center architecture [17]. It (see Fig. 4.1) consists of the core tier at the root of the tree, the aggregation tier that is responsible for routing, and the access tier that holds the pool of computing servers (or hosts). Earlier data centers used two-tier architectures with no aggregation tier. However, such data centers, depending on the type of switches used and per-host bandwidth requirements, could typically support not more than 5,000 hosts. Given the pool of servers in today's data centers that are of the order of 100,000 hosts [11] and the requirement to keep layer-2 switches in the access network, a three-tiered design becomes the most appropriate option.

Although 10 Gigabit Ethernet (GE) transceivers are commercially available, in a three-tiered architecture the computing servers (grouped in racks) are interconnected using 1 GE links. This is due to the fact that the 10 GE transceivers: (a) are too expensive and (b) probably offer more capacity than needed for connecting computing servers. In current data centers, rack connectivity is achieved with inexpensive Top-of-Rack (ToR) switches. A typical ToR switch shares two 10 GE uplinks with 48 GE links that interconnect computing servers within a rack.

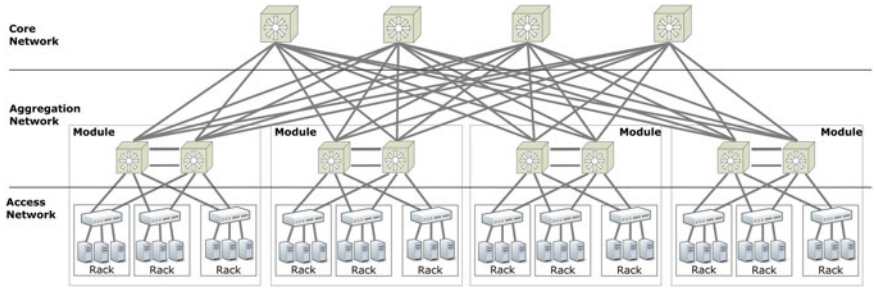

Fig. 4.1 Three-tier data center architecture

The difference between the downlink and the uplink capacities of a switch defines its oversubscription ratio, which in the aforementioned case is equal to $48/20 = 2.4 : 1$. Therefore, under full load, only 416 Mb/s will remain available to each of the individual servers out of their 1 GE links.

At the higher layers of hierarchy, the racks are arranged in modules (see Fig. 4.1) with a pair of aggregation switches servicing the module connectivity. Typical oversubscription ratios for these aggregation switches are around 1.5:1, which further reduces the available bandwidth for the individual computing servers to 277 Mbps.

The bandwidth between the core and aggregation networks is distributed using a multi-path routing technology, such as the equal cost multi-path (ECMP) routing [18]. The ECMP technique performs a per-flow load balancing, which differentiates the flows by computing a hash function on the incoming packet headers. For a three-tiered architecture, the maximum number of allowable ECMP paths bounds the total number of core switches to eight. Such a bound also limits the deliverable bandwidth to the aggregation switches. This limitation will be waved with the (commercial) availability of 100 GE links, standardized in June 2010 [19].

But how the data center architecture will look like in the future? The most promising trend in to follow a modular design. Traditional racks of servers will be replaced with standard shipping containers hosting 10 times as many servers as conventional data center in the same volume [20]. Each container is optimized for power consumption. It integrates a combined water and air cooling system and implements optimized networking solutions. These containers, being easy to ship, can become plug-and-play modules in future roof-less data center facilities [21]. Their current PUE is in the order of 1.2 [22] while the average PUE for the industry is between 1.8 and 2.0 [1] depending on the reporting source. Some skeptics addressing the problem of individual component failures and the overhead of shipping the whole container back to the manufacturer. This can be addressed by packing even more servers into self-contained container solutions requiring no operational maintenance [23]. Whenever an individual component fails the whole container can continue operation with only minor degradation in computing capacity. To make it a reality, each container as well as the data center itself

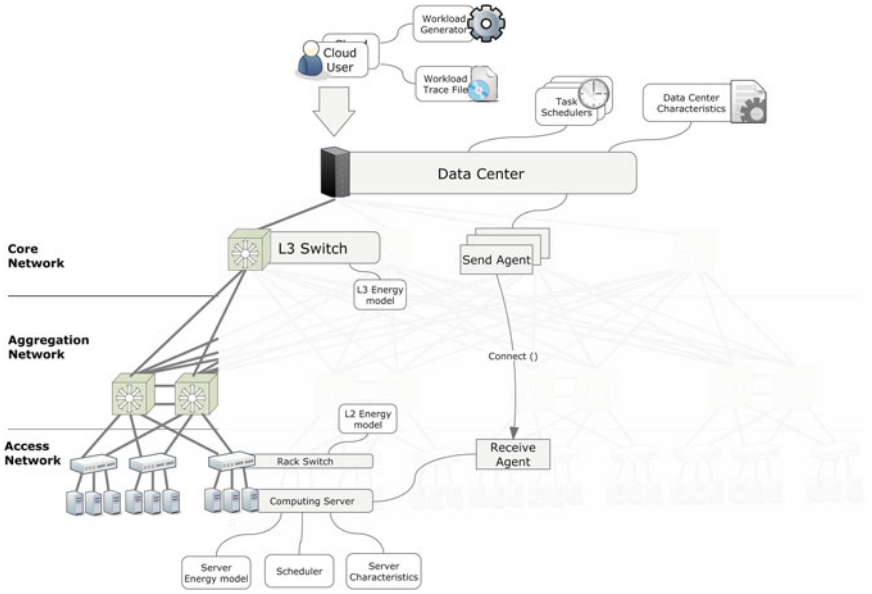

Fig. 4.2 Architecture of GreenCloud simulator

should follow a distributed design approach. But current data center architectures are completely hierarchical. This way, for example, a failure in the rack switch can disable all servers in the rack. A failure of the core or aggregation switches may degrade operation or even disable a large number of racks. Therefore, fat-tree architectures will be replaced with distributed approaches like DCell [24], BCube [25], FiConn [26], or DPillar [27] in future data centers.

4.2.3 Simulator Structure

In this section we introduce GreenCloud simulator which offers fine-grained simulation of modern cloud computing environments focusing on data center communications and energy efficiency. GreenCloud is an extension to the network simulator ns-2 [12]. It offers users a detailed fine-grained modeling of the energy consumed by the elements of the data center, such as servers, switches, and links. Moreover, GreenCloud offers a thorough investigation of workload distributions. Furthermore, a specific focus is devoted on the packet-level simulations of communications in the data center infrastructure, which provide the finest-grain control and is not present in any cloud computing simulation environment. Reference [28] provides more details on the GreenCloud simulator. Figure 4.2 presents the structure of the GreenCloud extension mapped onto the three-tier data center architecture.

4.2.4 Hardware Components and Energy Models

Computing servers are the staple of a data center that are responsible for task execution. In GreenCloud, the server components implement single core nodes that have a preset on a processing power limit in MIPS (million instructions per second) or FLOPS (floating point operations per second), associated size of the memory/storage resources, and contain different task scheduling mechanisms ranging from the simple round-robin to the sophisticated DVFS and DNS approaches.

The servers are arranged into racks with a ToR switch connecting it to the access part of the network. The power model followed by server components depends on CPU utilization. As reported in [2] and [7] an idle server consumes about two-thirds of its peak load consumption. This is due to the fact that servers must constantly manage memory modules, disks, I/O resources, and other peripherals. Moreover, the power consumption increases with the level of CPU load linearly. As a result, the aforementioned model allows implementation of power saving in a centralized scheduler that can provision consolidation of workloads in a minimum possible amount of the computing servers.

1. Another option for power management is dynamic voltage/frequency scaling (DVFS) [10], which introduces a trade-off between computing performance and the energy consumed by the server. The DVFS is based on the fact that switching power in a chip decreases proportionally to $V^2 \times f$, where V is the voltage and f is the switching frequency. Moreover, voltage reduction requires frequency downshift. This implies a cubic relationship from f in the CPU power consumption. Note that server components, such as bus, memory, and disks do not depend on the CPU frequency. Therefore, the power consumption of an average server (see Fig. 4.3) can be expressed as follows [29]:

$$P = P_{\text{fixed}} + P_f \times f^3, \tag{4.1}$$

where P_{fixed} accounts for the portion of the consumed power which does not scale with the operating frequency f, while P_f is a frequency-dependent CPU power consumption.

Network switches and links form the interconnection fabric that delivers workloads to any of the computing servers for execution in a timely manner. The interconnection of switches and servers requires different cabling solutions depending on the supported bandwidth, physical and quality characteristics of the link. The quality of signal transmission in a given cable determines a trade-off between the transmission rate and the link distance, which are the factors defining the cost and energy consumption of the transceivers.

The twisted pair is the most commonly used medium for Ethernet networks that allows organizing Gigabit Ethernet (GE) transmissions for up to 100 m with the transceiver power consumed of around 0.4 W or 10 GE links for up to 30 m with the transceiver power of 6 W. The twisted pair cabling is a low cost solution. However, for the organization of 10 GE links it is common to use optical multimode fibers.

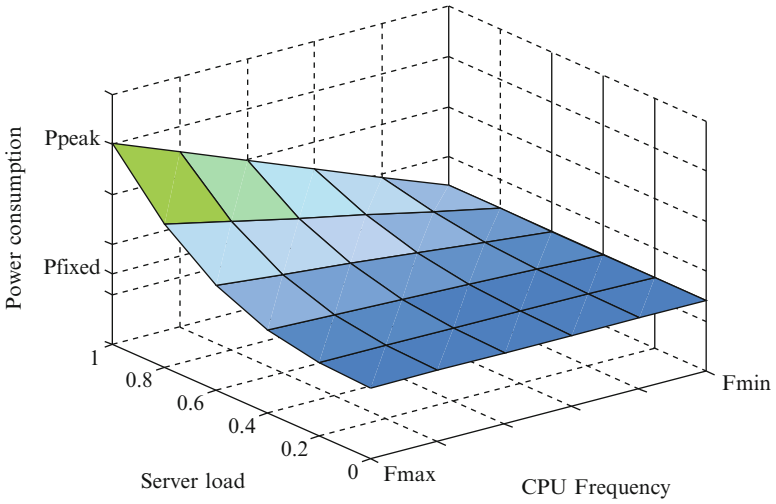

Fig. 4.3 Computing server power consumption

The multimode fibers allow transmissions for up to 300 m with the transceiver power of 1 W [30]. On the other hand the fact that multimode fibers cost almost 50 times of the twisted pair cost motivates the trend to limit the usage of 10 GE links to the core and aggregation networks as spending for the networking infrastructure may top 10%–20% of the overall data center budget [31].

The number of switches installed depends on the implemented data center architecture. However, as the computing servers are usually arranged into racks the most common switch in a data center is ToR switch. The ToR switch is typically placed at the top unit of the rack unit (1RU) to reduce the amount of cables and the heat produced. The ToR switches can support either gigabit (GE) or 10 gigabit (10 GE) speeds. However, taking into account that 10 GE switches are more expensive, current capacity limitation of aggregation and core networks gigabit rates are more common for racks.

Similar to the computing servers early power optimization proposals for interconnection network were based on DVS links [10]. The DVS introduced a control element at each port of the switch that depending on the traffic pattern and current levels of link utilization could downgrade the transmission rate. Due to the comparability requirements only few standard link transmission rates are allowed, such as for GE links 10 Mbps, 100 Mbps, and 1 Gbps are the only options.

On the other hand, the power efficiency of DVS links is limited as only a portion (3%–15%) of the consumed power which scales linearly with the link rate. As demonstrated by the experiments in [32] the energy consumed by a switch and all its transceivers can be defined as:

$$P_{\text{switch}} = P_{\text{chassis}} + n_{\text{linecards}} \times P_{\text{linecard}} + \sum_{i=0}^{R} n_{\text{ports}.r} \times P_r, \tag{4.2}$$

where P_{chassis} is related to the power consumed by the switch hardware, P_{linecard} is the power consumed by any active network line card, P_r corresponds to the power consumed by a port (transceiver) running at the rate r. In Eq. (4.2), only the last component appears to be dependent on the link rate while other components, such as P_{chassis} and P_{linecard} remain fixed for all the duration of switch operation. Therefore, P_{chassis} and P_{linecard} can be avoided by turning the switch hardware off or putting it into sleep mode.

4.2.5 Jobs and Workloads

Workloads are the objects designed for universal modeling of various cloud user services. In grid computing the workloads are typically modeled a sequence of jobs that can be divided into a set of tasks. The tasks can be dependent requiring an output from other tasks to start execution or be independent. Moreover, due to the nature of grid computing applications (biological, financial, or climate modeling) the number of jobs available prevail the number of computing resources available. While the main goal is in minimization of the time required for the computing of all jobs which may take weeks or months the individual jobs do not have a strict completion deadline.

In cloud computing, incoming requests are typically generated for such applications like web browsing, instant messaging, or various content delivery applications. The jobs tend to be more independent, less computationally intensive, but have a strict completion deadline specified in SLA. To cover the vast majority of cloud computing applications, we define three types of jobs:

- *Computationally Intensive Workloads (CIWs)* model high-performance computing (HPC) applications aiming at solving advanced computational problems. CIWs load computing servers considerably, but require almost no data transfers in the interconnection network of the data center. The process of CIW energy-efficient scheduling should focus on the server power consumption footprint trying to group the workloads at the minimum set of servers as well as to route the traffic produced using a minimum set of routes. There is no danger of network congestion due to the low data transfer requirements, and putting the most of the switches into the sleep mode will ensure the lowest power of the data center network.
- *Data-Intensive Workloads (DIWs)*, on the contrary, produce almost no load at the computing servers, but require heavy data transfers. DIWs aim to model such applications like video file sharing where each simple user request turns into a video streaming process. As a result, the interconnection network and not the computing capacity becomes a bottleneck of the data center for DIWs. Ideally, there should be a continuous feedback from network switches to the central

workload scheduler. Based on such a feedback, the scheduler will distribute the workloads taking current congestion levels of the communication links. It will avoid sending workloads over congested links even if certain server's computing capacity will allow accommodating the workload. Such scheduling policy will balance the traffic in the data center network and reduce average time required for a task delivery from the core switches to the computing servers.

- *Balanced Workloads (BWs)* aim to model the applications having both computing and data transfer requirements. BWs load the computing severs and communication links proportionally. With this type of workloads the average load on the servers is proportional to the average load of the data center network. BWs can model such applications as geographic information systems which require both large graphical data transfers and heavy processing. Scheduling of BWs should account for both servers' load and the load of the interconnection network.

The execution of each workload object requires a successful completion of its two main components: (a) computing and (b) communicational. The computing component defines the amount of computing that has to be executed before a given deadline on a time scale. The deadline aims at introducing Quality of Service (QoS) constraints specified in SLA. The communicational component of the workload defines the amount and the size of data transfers that must be performed prior, during, and after the workload execution. It is composed of three parts: (a) the size of the workload, (b) the size of internal, and (c) the size of external to the data center communications. The size of the workload defines the number of bytes that after being divided into IP packets is required be transmitted from the core switches to the computing servers before a workload execution can be initiated. The size of external communications defines the amount of data required to be transmitted outside the data center network at the moment of task completion and corresponds to the task execution result. The size of internal to the data center communications defines the amount of data to be exchanged with another workload that can be executed at the same or a different server. This way the workload interdependencies are modeled. In fact, internal communication in the data center can account for as much as 70% of total data transmitted [11].

Figure 4.4 captures energy consumption measured in a DVFS- and DNS-enabled data center running different types of workloads. An efficient and effective methodology to optimize energy consumption of interdependent workloads is to analyze the workload communication requirements at the moment of scheduling and perform a coupled placement of these interdependent workloads—a co-scheduling approach. The co-scheduling approach will reduce the number of links/switches involved into communication patterns.

Figure 4.5 shows a typical distribution of energy consumption between data center components obtained via simulations.

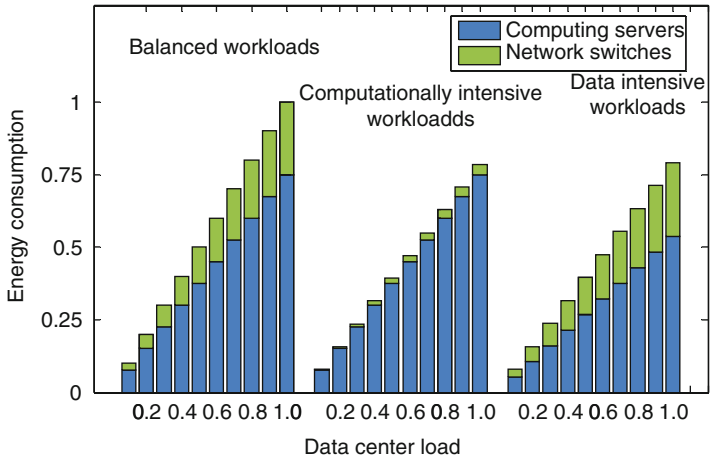

Fig. 4.4 Energy consumption for different types of workloads

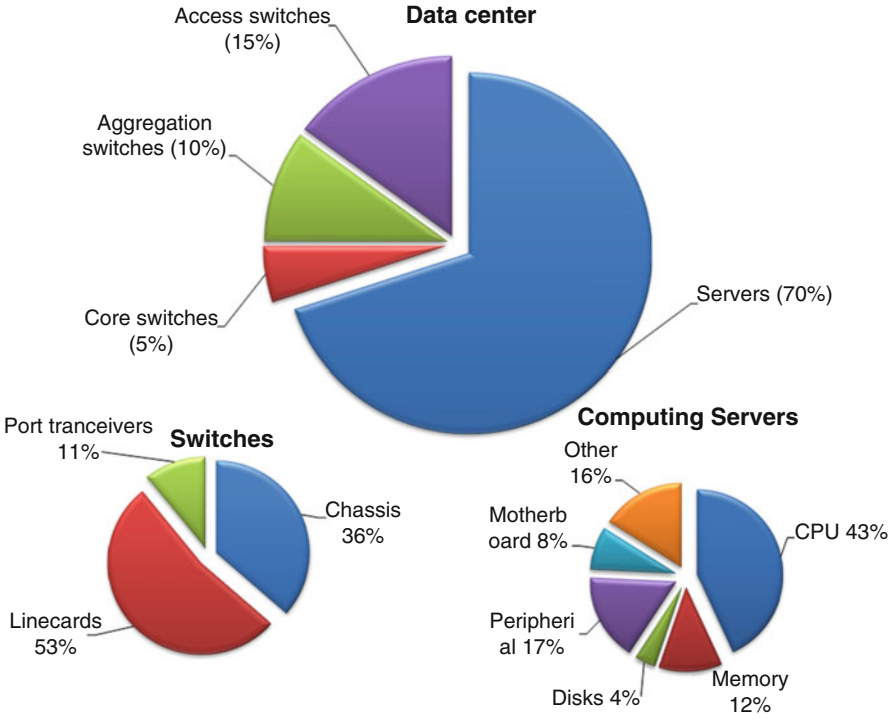

Fig. 4.5 Distribution of energy consumption in data center

4.3 Energy-Efficient Scheduling

4.3.1 Network Congestion

Utilizing a communication fabric in data centers entails the concept of running multiple types of traffic (LAN, SAN, or IPC) on a single Ethernet-based medium [33]. On one side, the Ethernet technology is cheap, easy to deploy, and relatively simple to manage, on the other side, the Ethernet hardware is less powerful and provisions for small buffering capacity. A typical buffer size in an Ethernet network is in the order of 100s of KB. However, a typical buffer size of an Internet router is in the order of 100s of MB [34]. Small buffers and the mix of high-bandwidth traffic are the main reasons for network congestion.

Any of the data center switches may become congested either in the uplink direction or in the downlink direction or both. In the downlink direction, the congestion occurs when individual ingress link capacities overcome individual egress link capacities. In the uplink direction, the mismatch in bandwidth is primarily due to the bandwidth oversubscription ratio, which occurs when the combined capacity of server ports overcomes a switch's aggregate uplink capacity.

Congestion (or hotspots) may severely affect the ability of a data center network to transport data. Currently, the Data Center Bridging Task Group (IEEE 802.1) [35] is specifying layer-2 solutions for congestion control, termed IEEE 802.1Qau specifications. The IEEE 802.1Qau specifications introduce a feedback loop between data center switches for signaling congestion. Such a feedback allows overloaded switches to hold off heavy senders from sending with the congestion notification signal. Such a technique may avoid congestion-related losses and keep the data center network utilization high. However, it does not address the root of the problem as it is much more efficient to assign data-intensive jobs to different computing servers in the way that jobs avoid sharing common communication paths. To benefit from such spatial separation in the three-tiered architecture (see Fig. 4.1), the jobs must be distributed among the computing servers in proportion to their communication requirements. Data-intensive jobs, like ones generated by video sharing applications, produce a constant bit-stream directed to the end-user as well as communicate with other jobs running in the data center. However, such a methodology contradicts the objectives of energy-efficient scheduling, which tries to concentrate all of the active workloads on a minimum set of servers and involve minimum number of communication resources. This trade-off between energy efficiency, data center network congestion, and performance of individual jobs is resolved using a unified scheduling metric presented in the subsequent section.

4.3.2 The DENS Methodology

The DENS methodology minimizes the total energy consumption of a data center by selecting the best-fit computing resources for job execution based on the load level

and communication potential of data center components. The communicational potential is defined as the amount of end-to-end bandwidth provided to individual servers or group of servers by the data center architecture. Contrary to traditional scheduling solutions [36] that model data centers as a homogeneous pool of computing servers, the DENS methodology develops a hierarchical model consistent with the state-of-the-art data center topologies. For a three-tier data center, DENS metric M is defined as a weighted combination of server-level f_s, rack-level f_r, and module-level f_m functions:

$$M = \alpha \times f_s + \beta \times f_r + \gamma \times f_m, \tag{4.3}$$

where α, β, and γ are weighted coefficients that define the impact of the corresponding components (servers, racks, and/or modules) on the metric behavior. Higher values of α favor the selection of highly loaded servers in lightly loaded racks. Higher values of β will prioritize computationally loaded racks with low network traffic activity. Higher values of γ favor selection of lightly loaded modules. The γ parameter is an important design variable for job consolidation in data centers. Taking into account that $\alpha + \beta + \gamma$ must equal unity, the values of $\alpha = 0.7$, $\beta = 0.2$, and $\gamma = 0.1$ are selected experimentally to provide a good balance in the evaluated three-tier data center topology. The details of the selection process are presented in [37].

The factor related to the choice of computing servers combines the server load $L_s(l)$ and its communication potential $Q_r(q)$ that corresponds to the fair share of the uplink resources on the ToR switch. This relationship is given as:

$$f_s(l,q) = L_s(l) \times \frac{Q_r(q)^{\phi}}{\delta_r}, \tag{4.4}$$

where $L_s(l)$ is a factor depending on the load of the individual servers l, $Q_r(q)$ defines the load at the rack uplink by analyzing the congestion level in the switch's outgoing queue q, δ_r is a bandwidth over provisioning factor at the rack switch, and ϕ is a coefficient defining the proportion between $L_s(l)$ and $Q_r(q)$ in the metric. Given that both $L_s(l)$ and $Q_r(q)$ must be within the range $[0,1]$ higher ϕ values will decrease the importance of the traffic-related component $Q_r(q)$. Similar to the case of computing servers, which was encapsulated in Eq. (4.4), the factors affecting racks and modules can be formulated as:

$$f_r(l,q) = L_r(l) \times \frac{Q_m(q)^{\phi}}{\delta_m} = \frac{Q_m(q)^{\phi}}{\delta_m} \times \frac{1}{n} \sum_{i=1}^{n} L_s(l), \tag{4.5}$$

$$f_m(l) = L_m(l) = \frac{1}{k} \sum_{j=0}^{k} L_r(l), \tag{4.6}$$

where $L_r(l)$ is a rack load obtained as a normalized sum of all individual server loads in the rack, $L_m(l)$ is a module load obtained as a normalized sum of all of

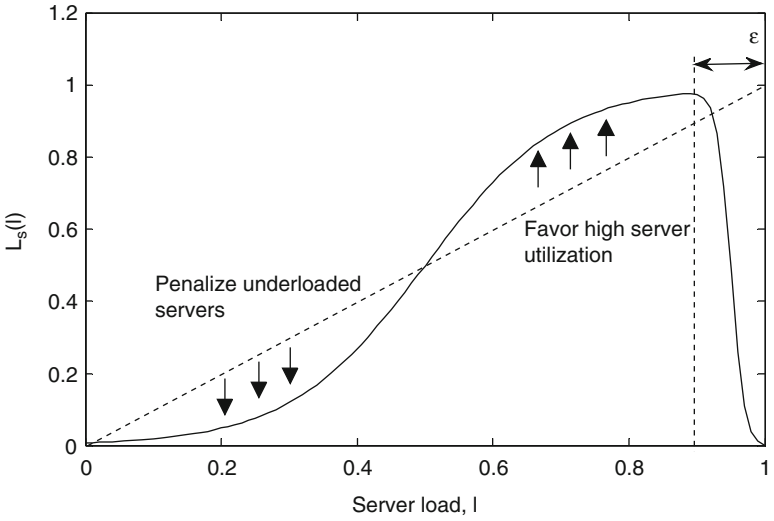

Fig. 4.6 Computing server selection by DENS metric

the rack loads in this module, n and k are the number of servers in a rack and the number of racks in a module respectively, $Q_{\mathrm{m}}(q)$ is proportional to the traffic load at the module ingress switches, and δ_{m} stands for the bandwidth over provisioning factor at the module switches. It should be noted that the module-level factor f_{m} includes only a load-related component l. This is due to the fact that all the modules are connected to the same core switches and share the same bandwidth using ECMP multi-path balancing technology.

The fact that an idle server consumes energy that is almost two-thirds of its peak consumption [7] suggests that an energy-efficient scheduler must consolidate data center jobs on a minimum possible set of computing servers. On the other hand, keeping servers constantly running at peak loads may decrease hardware reliability and consequently affect the job execution deadlines [38]. To address the aforementioned issues, we define the DENS load factor as a sum of two sigmoid functions:

$$L_{\mathrm{s}}(l) = \frac{1}{1+\mathrm{e}^{-10\left(l-\frac{1}{2}\right)}} - \frac{1}{1+\mathrm{e}^{-\frac{10}{\varepsilon}\left(l-\left(1-\frac{\varepsilon}{2}\right)\right)}}. \tag{4.7}$$

The first component of Eq. (4.7) defines the shape of the main sigmoid, while the second component is a penalizing function aimed at the convergence towards the maximum server load value (see Fig. 4.6). The parameter ε defines the size and the incline of this falling slope. The server load l is within the range $[0, 1]$. For the tasks having deterministic computing load the server load can be computed as the sum of computing loads of all of the running tasks. Alternatively, for the tasks with predefined completion deadline, the server load l can be expressed as the minimum amount of computational resource required from the server to complete all the tasks right-in-time.

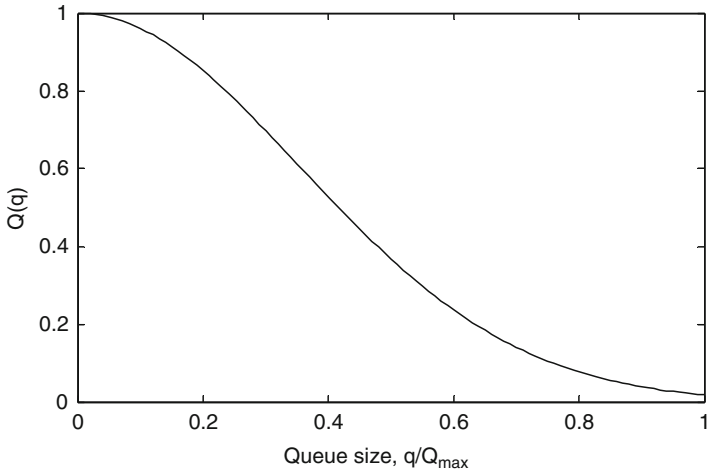

Fig. 4.7 Queue selection by DENS metric

Being assigned into racks, computing servers share the same ToR switch their uplink communication demands. However, defining a portion of this bandwidth used by a given server or a flow at the gigabit speeds during runtime is a computationally expensive task. To circumvent the aforementioned undesirable characteristic, both Eqs. (4.4) and (4.5) include a component that is dependent on the occupancy level of the outgoing queue $Q(q)$ at the switch and scales with the bandwidth over provisioning factor δ.

Instead of relying on the absolute size of the queue, the occupancy level q is scaled with the total size of the queue $Q_{\max}$ within the range $[0, 1]$. The range corresponds to none and full buffer occupancy. By relying on buffer occupancy, the DENS metric reacts to the growing congestion in racks or modules rather than transmission rate variations. To satisfy the aforementioned behavior, $Q(q)$ is defined using inverse Weibull cumulative distribution function:

$$Q(q) = \mathrm{e}^{-\left(\frac{2q}{Q_{\max}}\right)^2}. \tag{4.8}$$

The obtained function, illustrated in Fig. 4.7, favors empty queues and penalizes fully loaded queues. Being scaled with the bandwidth over provisioning factor δ in Eqs. (4.4) and (4.5) it favors the symmetry in the combined uplink and downlink bandwidth capacities for switches when congestion level is low. However, as congestion grows and buffers overflow, the bandwidth mismatch becomes irrelevant and immeasurable.

Figure 4.8 presents the combined $f_s(l, q)$ as defined in Eq. (4.4). The obtained bell-shaped function favors selection of servers with the load level above average located in racks with the minimum or no congestion. Reference [37] provides more details about DENS metrics and its performance in different operation scenarios.

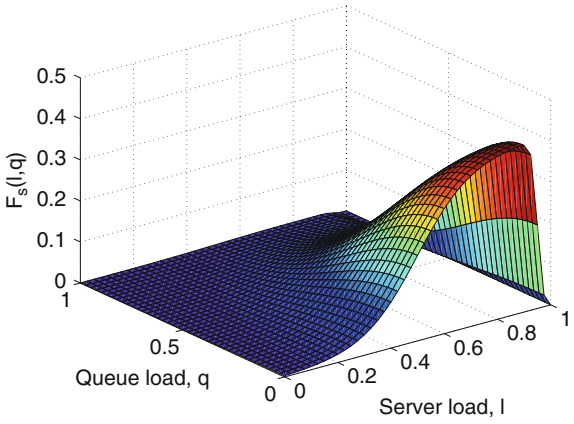

Fig. 4.8 Server selection by DENS metric according to its load and communicational potential

4.4 Conclusions

The cost and operating expenses of data centers are becoming a growing concern as cloud computing industry is booming. The challenge of energy efficiency allows maintaining the same data center performance while the level of energy consumption is reduced. This can not only significantly reduce costs of operating the IT equipment and cooling but also increase server density enlarging the capacity of existing data center facilities.

To understand the optimization space we surveyed energy consumption models of computing servers, network switches, and communication links. Thereafter, main techniques for energy efficiency, like DVFS or dynamic shut-down, are studied at both the component and system levels. It is demonstrated that approaches for centralized coordination and scheduling are required to achieve satisfactory optimization levels. Such coordination should combine traditional scheduling approaches with the awareness of the state of communication equipment and network traffic footprints. Furthermore, the characteristics of the incoming workloads must be taken into account. Currently, GreenCloud simulator and presented energy-aware scheduling approaches are being extended to cover scenarios which include geographically distributed data centers and renewable sources of energy.

References

1. Brown R, Chan P, Eto J, Jarvis S, Koomey J, Masanet E, Nordman B, Sartor D, Shehabi A, Stanley J, Tschudi B (2007) Report to congress on server and data center energy efficiency: Public law 109–431. Lawrence Berkeley National Laboratory. 1–130. Available at http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf
2. Fan X, Weber W-D, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In: ACM international symposium on computer architecture, San Diego, CA, June 2007

3. Raghavendra R, Ranganathan P, Talwar V, Wang Z, Zhu X (2008) No "Power" struggles: coordinated multi-level power management for the data center. In: SIGOPS Oper. Syst. Rev. 42(2): 48–59
4. Gartner Group. Available at: http://www.gartner.com/, Accessed Aug 2012
5. Moore J, Chase J, Ranganathan P, Sharma R (2005) Making scheduling "Cool": temperature-aware workload placement in data centers. In: USENIX annual technical conference (ATEC '05). USENIX Association, Berkeley, CA, USA, pp 5–5
6. Horvath T, Abdelzaher T, Skadron K, Liu X (2007) Dynamic voltage scaling in multitier web servers with end-to-end delay control. IEEE Trans Comp 56(4):444–458
7. Chen G, He W, Liu J, Nath S, Rigas L, Xiao L, Zhao F (2008) Energy-aware server provisioning and load dispatching for connection-intensive internet services. In: The 5th USENIX symposium on networked systems design and implementation, Berkeley, CA, USA
8. Liu J, Zhao F, Liu X, He W (2009) 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, QC, Canada, June 2009
9. Li B, Li J, Huai J, Wo T, Li Q, Zhong L (2009) EnaCloud: An energy-saving application live placement approach for cloud computing environments. In: IEEE international conference on cloud computing, Bangalore, India
10. Shang L, Peh L-S, Jha NK (2003) Dynamic voltage scaling with links for power optimization of interconnection networks. In: Proceedings of the 9th international symposium on high-performance computer architecture (HPCA '03). IEEE Computer Society, Washington, DC, USA, pp 91–102.
11. Mahadevan P, Sharma P, Banerjee S, Ranganathan P (2009) Energy aware network operations. In: Proceedings of the 28th IEEE international conference on Computer Communications Workshops (INFOCOM'09). IEEE Press, Piscataway, NJ, USA, pp 25–30.
12. The Network Simulator ns-2. Available at: http://www.isi.edu/nsnam/ns/, Accessed Aug 2012
13. Buyya R, Ranjan R, Calheiros RN (2009) Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities. In: Proceedings of the 7th high performance computing and simulation conference, Leipzig, Germany
14. Lim S-H, Sharma B, Nam G, Kim EK, Das CR (2009) MDCSim: a multi-tier data center simulation, platform. In: IEEE international conference on cluster computing and workshops (CLUSTER). pp 1–9
15. Rawson A, Pfleuger J, Cader T (2008) Green grid data center power efficiency metrics: PUE and DCIE. The Green Grid White Paper #6
16. Wang L, Khan SU (2011) Review of performance metrics for green data centers: a taxonomy study. The Journal of Supercomputing. Springer US, pp 1–18
17. Cisco Data Center Infrastructure 2.5 Design Guide (2010) Cisco press, March 2010
18. Thaler D, Hopps C (2000) Multipath issues in unicast and multicast nexthop selection. Internet Engineering Task Force. Request for Comments 2991, November 2000. Available at http://tools.ietf.org/html/rfc2991
19. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment 4: Media Access Control Parameters, Physical Layers and Management Parameters for 40 Gb/s and 100 Gb/s Operation," IEEE Std 802.3ba-2010 (2010) (Amendment to IEEE Standard 802.3-2008), pp 1–457
20. Christesen S Data center containers. Available at http://www.datacentermap.com/blog/datacenter-container-55.html., Accessed Aug 2012
21. Katz RH (2009) Tech Titans building boom. IEEE Spectrum 46(2):40–54
22. Worthen B (2011) Data centers boom. Wall Street Journal. Available at http://online.wsj.com/article/SB10001424052748704336504576259180354987332.html
23. Next generation data center infrastructure. CGI White Paper, 2010
24. Guo C, Wu H, Tan K, Shiy L, Zhang Y, Luz S (2008) DCell: a scalable and fault-tolerant network structure for data centers. In: ACM SIGCOMM, Seattle, Washington, DC, USA

25. Guo C, Lu G, Li D, Wu H, Zhang X, Shi1 Y, Tian C, Zhang1 Y, Lu S (2009) BCube: a high performance, server-centric network architecture for modular data centers. In: ACM SIGCOMM, Barcelona, Spain, 2009
26. Li D, Guo C, Wu H, Tan K, Zhang Y, Lu S (2009) FiConn: using backup port for server interconnection in data centers. In: IEEE INFOCOM 2009, pp 2276–2285
27. Liao Y, Yin D, Gao L (2010) DPillar: scalable dual-port server interconnection for data center networks. In: 2010 Proceedings of 19th International Conference on computer communications and networks (ICCCN), pp 1–6
28. Kliazovich D, Bouvry P, Khan SU (2010) GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. The Journal of Supercomputing. pp 1–21
29. Chen Y, Das A, Qin W, Sivasubramaniam A, Wang Q, Gautam N (2005) Managing server energy and operational costs in hosting centers. In: Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems. ACM, New York, pp 303–314
30. Farrington N, Rubow E, Vahdat A (2009) Data center switch architecture in the age of merchant silicon. In Proceedings of the 17th IEEE symposium on high performance interconnects (HOTI '09). IEEE Computer Society, Washington, DC, USA, pp 93–102
31. Greenberg A, Lahiri P, Maltz DA, Patel P, Sengupta S (2008) Towards a next generation data center architecture: scalability and commoditization. In: Proceedings of the ACM workshop on programmable routers for extensible services of tomorrow, Seattle, WA, USA
32. Mahadevan P, Sharma P, Banerjee S, Ranganathan P (2009) A power benchmarking framework for network devices. In: Proceedings of the 8th international IFIP-TC 6 networking conference, Aachen, Germany 2009
33. Garrison S, Oliva V, Lee G, Hays R (2008) Data center bridging, Ethernet Alliance. Available at http://www.ethernetalliance.org/wp-content/uploads/2011/10/Data-Center-Bridging1.pdf
34. AlizadehM, Atikoglu B, Kabbani A, Lakshmikantha A, Pan R, Prabhakar B, Seaman M (2008) Data center transport mechanisms: Congestion control theory and IEEE standardization. In: Annual Allerton conference on communication, control, and computing, pp 1270–1277.
35. IEEE 802.1 Data Center Bridging Task Group. Available at: http://www.ieee802.org/1/pages/dcbridges.html, Accessed Aug 2012
36. Song Y, Wang H, Li Y, Feng B, Sun Y (2009) Multi-tiered on-demand resource scheduling for VM-based data center. In: IEEE/ACM international symposium on cluster computing and the grid (CCGRID), pp 148–155
37. Kliazovich D, Bouvry P, Khan SU (2011) DENS: Data center energy-efficient network-aware scheduling. Cluster Computing, Springer US, pp 1–11.
38. Kopparapu C (2002) Load balancing servers, firewalls, and caches. Wiley, New York