Power Comparison of Cloud Data Center Architectures

Power Comparison of Cloud Data Center Architectures

Pietro Ruiu[⋆†], Andrea Bianco[†], Claudio Fiandrino[‡], Paolo Giaccone[†], Dzmitry Kliazovich[‡]
[⋆] Istituto Superiore Mario Boella (ISMB), Torino, Italy
[†] Dip. di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy
[‡] Computer Science and Communications Research Unit, University of Luxembourg, Luxembourg
E-mails: [⋆]pietro.ruiu@ismb.it, [†]{bianco,giaccone}@polito.it, [‡]{name.surname}@uni.lu

***Abstract*—Power consumption is a primary concern for cloud computing data centers. Being the network one of the non-negligible contributors to energy consumption in data centers, several architectures have been designed with the goal of improving network performance and energy-efficiency. In this paper, we provide a comparison study of data center architectures, covering both classical two- and three-tier design and state-of-art ones as Jupiter, recently disclosed by Google. Specifically, we analyze the combined effect on the overall system performance of different power consumption profiles for the IT equipment and of different resource allocation policies. Our experiments, performed in small and large scale scenarios, unveil the ability of network-aware allocation policies in loading the the data center in a energy-proportional manner and the robustness of classical two- and three-tier design under network-oblivious allocation strategies.**

I. Introduction

Cloud computing is nowadays the de facto approach for provisioning and consuming services on a pay-as-you-go basis. Data centers play a key role in cloud computing as they host virtually unlimited computational and storage capacities that companies and end users can access and exploit over the Internet [1]. The data center network (DCN) interconnects computing servers with the wide area network. Proper design of DCNs is fundamental for the performance of cloud applications. Since most of cloud applications follow a Software-as-a-Service (SaaS) model [2] communication processes, not computing, tend to become the bottleneck limiting overall performance [3].

Data centers require a tremendous amount of energy to operate. In 2012, data centers energy consumption accounted for 15% of the global ICT energy consumption. This figure is projected to rise of about 5 to 10% in 2017 [4]. Typically, data centers spend most of the energy for powering and cooling the IT equipment (75%). Power distribution and facility operations account for the remaining 25%. Since the efficiency of cooling systems is increasing [5], more research efforts should be put in making green the IT system, which is becoming the major contributor to energy consumption.

Ideally, the power consumption of the computing and networking devices should be linearly proportional to the load [6]. However, the power consumption of the equipment is not zero when devices are idle. Devices in idle mode consume power because of overheads that are independent of the load. Moreover, typical power consumption profiles from idle to peak power are not energy-proportional. The energy-proportionality phenomenon describes the relation between the variation of power consumption with the load increase. As data center facilities usually upgrade capacity over time, this results having computing and communication equipment with heterogeneous capabilities, including different energy consumption profiles [7]. Assessing energy-proportionality is very important as it provides useful insights on the overall efficiency of the IT equipment and to devise effective resource allocation policies. For example, allocating computing-intensive tasks over non-proportional equipment may result in a waste of energy. In the literature, several metrics have been proposed to assess energy-proportionality of the IT equipment [8].

The concept of energy proportionality can also be used as a baseline to compare the performance of data center architectures [9]. Several data center networks, also called architectures or topologies in the remainder of the paper, have been proposed in order to overcome costs, scalability, agility and reliability issues and to accommodate growing traffic demands [10]. Architectures can be either *switch-centric* or *server-centric*. Switch-centric architectures, including Al-Fares et al. proposal [11], PortLand [12], VL2 [13] and Jupiter [14], rely on switches to perform traffic forwarding and routing. In server-centric architectures, also the computing servers are demanded to perform network communication operations. BCube [15] and DCell [16] are examples of server-centric architectures.

In this paper we provide a comparison study of data center network architectures. Unlike existing studies [9], our analysis jointly takes into account different power models (energy-proportional, non proportional, and realistic) and the resource allocation scheme. Resource allocation is at the heart of cloud computing. Virtual Machines (VMs), jobs and tasks are assigned to physical computing servers according to specific allocation policies [17]. Proper resource assignment ensures both Quality of Service (QoS) and use Quality of Experience (QoE) for cloud applications and cost saving for operators.

Our contributions are as follows. We devise a common testing framework to investigate the performance of different data center architectures, with any combination of power consumption profiles in the devices (servers and switches), under two opposite allocation policies. The first policy is oblivious of the network state. The second policy is network- and power-aware, i.e., tries not only to minimize both the server consumption (by consolidating the VM workloads in few servers), but also network consumption (by spreading the traffic across the switches or concentrating it in few routing paths, depending on the specific power switch profile). We compare two-tier, three-tier and Jupiter networks under both allocation policies and for different power models (ideal and realistic).

II. Background and Motivation

The classic design of data center architectures interconnects servers and switches in two- or three-tier structures, being the three-tier structures the most widely adopted [10]. Three-tier topologies consist of three layers. The lowest level interconnects the computing servers to the network through access switches. The aggregation and core layers provide connectivity towards the data center gateway. Three-tier architectures such as the fat-tree proposed by Al-Fares et al. [11], PortLand [12], VL2 [13] and Jupiter [14] are switch-centric architectures. In these topologies, the network switches in upper layers are specialized and high power demanding devices. Moreover, computing servers do not participate in forwarding operations and experience the so called *bandwidth-oversubscription* problem [8]. Server-centric architectures like BCube [15] and DCell [16] do not experience bandwidth oversubscription, at the cost of having the servers actively engaged in routing and packet forwarding. Moreover, all the network switches are low power demanding and very cheap commodity switches.

Several studies have analyzed and compared data center architectures. In a very first study, Popa et al. [18] provided a cost comparison analysis of several data center architectures, including both switch-centric and server-centric designs. Shang et al. [9] analyze energy proportionality in data center architectures under different routing strategies, namely high-performance and energy-aware policies. Similarly to [9], in this work we also focus on the energy proportionality by considering multiple power consumption profiles for the networking equipment. Unlike previous studies, our analysis builds on two different resource allocation policies having considered the same energy-aware routing strategy.

The problem of resource allocation in cloud computing data centers consists of assigning to incoming VMs, jobs or tasks, a set of resource such as CPU cycles, RAM space, storage, bandwidth or their combination. Optimizing the allocation process can limit power consumption costs and provide consistent savings. For example, VMPlanner [19] shows that jointly optimizing VM placement and traffic flow routing not only helps in reducing power consumption, but also optimizes traffic distribution. HEROS [20] proposes a network- and energy-aware scheme for heterogeneous data centers, where the computing equipment can consists of many- and multi-core architectures, asymmetric cores, coprocessors, graphics processing units and solid-state drives.

To the best of our knowledge, a very little attention has been devoted to analyze the problem of resource allocation in different data center networks. The most similar study to our work is [21], where three-layer, fat-tree, BCube and DCell topologies have been considered for performance evaluation of a virtual machine placement policy that aims at addressing energy-efficiency and traffic engineering. However, this analysis lacks consideration of energy proportionality of the devices. In addition, our study performs a large-scale analysis as we compare the performance of data center architectures hosting thousands of servers. Moreover, we jointly investigate the effect on the overall system performance of realistic power consumption profiles and network-aware resource allocation policies. To illustrate, workload consolidation policies save energy by allocating and consolidating incoming VMs in a minimum number of servers. As a result, idle devices can be turned off or put in sleep mode. However, turning off energy proportional devices and consolidating the workload over highly non-proportional devices leads to a waste of energy.

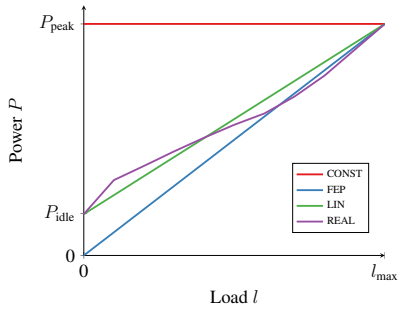


Fig. 1. Power consumption profiles for the IT equipment

III. Power Consumption in Data Center Architectures

We consider a data center in which S servers are connected through a generic packet network. The cloud controller receives the requests from the tenants and allocates the VMs on the basis of a set of resources (CPU, memory, storage) to be available in the server designed for allocation. The communication requirements are modeled for each VM as the amount of traffic requested for any pair of VMs. The cloud controller is assumed to be aware of the network state as well. Indeed, to satisfy the communication requirements, the network must guarantee sufficient bandwidth for all the paths traversed by the communication flows between the VMs.

A. Data Center Architectures

For the analysis, we take into account two- and three-tier architectures [10] and Jupiter [14], the architecture Google currently implements in its data centers. All these architectures are switch-centric and based on a Clos construction. Given any basic building block, Clos networks scale indefinitely. In particular, the two-tier architecture is typical for small data centers within the same POD and based on a classical 3 stage Clos switching network. Instead, the three-tier architecture is typical of large data center, spanning multiple PODs, and it is based on a classical 5 stage Clos network. Finally, the Jupiter architecture has been designed for massively large data centers and it is based on a multi-stage Clos topology, built from commodity switches. All building blocks of this architecture are heterogeneous. The smallest unit is composed by a set of switches called TOR and used for building the blocks of each layer. Inside blocks switches can be placed on two levels. The Aggregation blocks are splitted in sub-groups (called Middle Blocks), also composed by TOR. In our work we do not consider the recently proposed server-centric architectures, since they are not actually implemented in current facilities, mainly because of the high cost and complexity in cabling [14].

B. Power Consumption Models

Fig. 1 illustrates the profiles modeling the power consumption of IT equipment. The power profile of an ideal device does not consume any power under zero load and it increases linearly with the load, reaching P_{peak} under the maximum load

$l_{\max}$. We denote this profile as Full Energy-Proportional (FEP). Although being ideal and therefore not available in current devices, the FEP profile can be considered as a benchmark for comparing other profiles especially at low loads.

The constant power consumption profile (CONST) is completely insensitive to the load and the power spent remains always constant to P_{peak}. As a result, this profile performs weak especially for low levels of load.

The power consumption profile of a real device is typically described by a generic function where at the loads $l = 0$ and $l = l_{\max}$ correspond to P_{idle} and P_{peak} respectively. Fig. 1 denotes such a profile as REAL. To estimate P_{idle} and P_{peak} in real devices, we performed an analysis of real data. For the servers, we analyzed the performance metrics from a number of vendors and equipped with different CPU models[1] and computed the mean of peak and idle values over a sample with more than 500 servers. For switches, we computed the average values based on the datasheets of major vendors[2], with optical fiber interfaces and compatible with OpenFlow protocol. A sample of 30 switches was taken into cosideration for this analysis.

For simplicity, in the experiments we rely on the linearized power consumption profile (LIN), which is an approximation of the REAL profile. To better approximate the profile of real device, we also considered some piecewise-linear models that have been investigated in the literature. A lot of works has been conducted on energy consumption of servers, and it is easy to find public data on consumption profiles of different servers. For the realistic (REAL) profile of a server, we used the PowerEdge C6320 server equipped with Intel Xeon E5-2699 v3 2.30 GHz. On the other hand, it was not possible to find in literature a detailed consumption profile for the network switches. Thus, we considered the values provided in [22]. Table I summarizes the values used for the different profiles.

TABLE I. VALUES OF POWER CONSUMPTION FOR DIFFERENT PROFILES

	SWITCH		SERVER	
	Peak	Idle	Peak	Idle
FEP, LIN, CONST	300	200	750	544
REAL	300	254	750	121

C. VM Placement Policies

We consider a VM allocation scheme that allocates one VM at the time. Each VM may be associated to a set of other preexisting VMs, denoted as *destination VMs*, that have been already allocated in the data center and with which the new VM must communicate. A bandwidth request is associated for each destination VM. Note that a newly allocated VM may become destination for other future VMs, thus making our VM model general. Indeed, it captures different possible cases, being compatible with the scenario of isolated VMs (i.e. without any communication requirement) and also with the scenario of small or large clusters of VMs that communicates each other according to any communication graph.

To capture the effects of a generic VM allocation algorithm on the network topology, we consider two opposite VM allocation policies. The two schemes are different only in the way of chosing a server for VM placement:

- Random Server Selection (RSS) chooses at random one server to allocate the new VM.
- Min-Network Power (MNP) chooses the server with minimum *network* power cost for the VM to communicate with its already allocated destination VMs.

After a server is selected, both policies check the compatibility of the VM with the candidate server and the network. Only if the server has enough local resources (in terms of CPU, memory, storage) and the network is able to sustain the required communication traffic between the VM and all the destination VMs (already allocated in some servers), the VM is actually allocated to the server. Otherwise, a new server needs to be selected.

Algorithm 1 Network-aware VM allocation policies

```
1: procedure FIND-SERVER-FOR-VM(v)
2:     Ω ← list of all the destination servers of VM v
3:     B ← list of all bandwidth requests of VM v for all the destination servers in Ω
4:     π ←SORT-SERVER-RSS( ) or π ←SORT-SERVER-MNP(Ω,B)
5:     for i = 1 ... S do                    ▷ Loop on all the possible candidate servers
6:         s = π_i                                                  ▷ Pick next server
7:         if server s has enough local resources for VM then
8:             if server s has enough bandwidth towards all servers in Ω then
9:                 allocate VM on server s
10:                reserve the bandwidth from s to all servers in Ω
11:                return s                                         ▷ End of search
12:            end if
13:        end if
14:    end for
15:    return BLOCKING-EVENT▷ VM cannot be allocated due to lack of resources
16: end procedure

17: function SORT-SERVER-RSS( )
18:    return random permutation of S servers
19: end function

20: function SORT-SERVER-MNP(Ω, B)
21:    for s = 1 → S do                                  ▷ Search across all the servers
22:        δ_s ← 0                ▷ Init incremental power to reach candidate server s
23:        for any d ∈ Ω do                ▷ Consider all possible destinations for s
24:            P ← path at min power cost from s to d
25:            δ_s ← δ_s + additional network power due to B_d traffic on P path
26:        end for
27:    end for
28:    return permutation of S servers with increasing network power δ_s
29: end function
```

Algorithm 1 shows the details of the two policies. For the sake of clarity and simplicity, the code is different from the one implemented in our simulations, even if they are functionally equivalent. Our implementation has been designed to minimize the computational complexity and thus augments the scalability of the approach for large data centers. Referring to the pseudocode, both RSS and MNP policies receive as input the new VM v to allocate (ln. 1), with the set of destination VMs to communicate with and the corresponding bandwidth requests. Based on those, the algorithms evaluates the corresponding set of servers where the destination VMs were previously allocated as well as the required bandwidth requests (ln. 2-3).

Now, a sorted list of candidate servers is created (ln.4) according to one of the two possible policies. In RSS the candidate server is randomly chosen (ln. 18), whereas in MNP

[1] https://www.spec.org/power_ssj2008/results/power_ssj2008.html

[2] https://www.opennetworking.org/sdn-openflow-products?start=50l

it is chosen to minimize the potential increment of power consumption due to the newly allocated VM, based on the power profile of all the switches along the routing path. Indeed, for each possible candidate server (loop in ln. 21-27), MNP computes the incremental power if allocating the path from the candidate server to all the destination servers (ln. 23-26). Finally, a list of candidate servers is returned sorted in increasing network power.

For both policies, the main loop (ln. 5-14) considers each candidate server sequentially, and checks whether the server has enough local resources (ln. 7) and whether the network provides enough bandwidth (ln. 8) to satisfy bandwidth requests from the VM to its destination VMs/servers. If both conditions are met, the candidate server is selected, otherwise the next candidate is considered. In the case the search is not successful, the VM experiences a blocking event (ln. 15) since either not enough resources are available in the servers or no enough bandwidth is available in the network to satisfy its communication demand.

When comparing the two approaches, RSS is expected to spread the VMs across all the servers in the data center, thus distributing the network traffic evenly. This policy can be considered as the worst-case in terms of the overall performance (blocking probability and power consumption), since it is not able to exploit the locality of the traffic among VMs. Instead, MNP consolidates as much as possible the VMs in the available servers (since the network cost is null in this case) and only when it is necessary to allocate the VMs into different servers, it considers the path with the minimum power cost; such approach exploits locality of the traffic among VMs. This behavior corresponds to either distribute the traffic on the network or to consolidate it in the minimum number of network paths, depending on the specific power consumption models considered for the network devices. As summary, by construction MNP consolidates VM workloads in the minimum number of servers, chosen in order to optimally distribute or consolidate the traffic load on the network.

IV. Comparison of Data Center Architectures

A. Simulation Scenario

We developed an ah-hoc event-driven simulator in C++, which models the whole data center, in terms of servers, interconnection network and arrival and allocation of VMs. The load on server s is characterized by three values: $\rho_s^{CPU} \in [0,1]$ for the CPU, $\rho_s^{RAM} \in [0,1]$ for the internal volatile memory and $\rho^{HD} \in [0,1]$ for the non-volatile memory. All these values have been normalized to the maximum capability available at the server. We assume heterogenous resources across all the server, thus we can directly sum all the normalized load to get the *overall average data center load*, defined as follows:

$$\rho_{\text{tot}} = \max \frac{1}{S} \left\{ \sum_{s=1}^{S} \rho_s^{CPU}, \sum_{s=1}^{S} \rho_s^{RAM}, \sum_{s=1}^{S} \rho_s^{HD} \right\},$$

i.e., the maximum average load across the three kind of resources.

Whenever a VM is generated, it is associated with a random triple describing the CPU, RAM and storage requirements, and with a destination VM, chosen at random among the already allocated VMs, with which the VM exchanges traffic. The traffic between the two VMs is assumed to be bidirectional and is associated a random value. If we define the degree of a VM as the total number of VMs with whom it is communicating, this VM generation model allows to obtain VMs with random degree, with an average close to one. The choice of the destination VM is affecting the degree distribution. We adopted the approach shown in Algorithm 2 to generate the traffic flows in a set of V VMs, given an "attachment" probability $p \in (0,1]$. We use geometric trials to find the destination VM to which the new VM is connected. This allows to distribute the communications among all the VMs fairly. Actually, the value of p gives the level of variance on the VMs. When p is close to 1, the maximum degree of the VMs is also close to 1. Whereas, when p approaches 0, for enough large V, the maximum degree becomes much larger than 1.

Algorithm 2 VM random generation

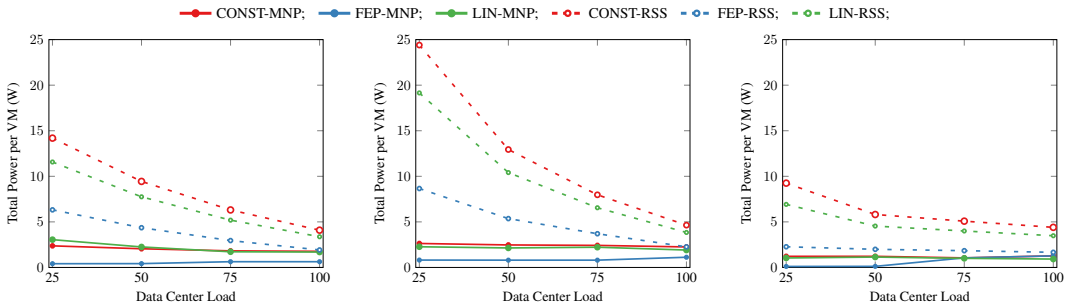
```
1: procedure GENERATE VM TRAFFIC FLOWS(V,p)
2:     for v = 2 ... V do
3:         d ← 1                                  ▷ Initial candidate for destination VM
4:         for k = v − 1 ... 1 do                 ▷ Try to connect to starting from closer VM
5:             if rand() < p then                 ▷ Bernoulli trials
6:                 d ← k                          ▷ Found d as destination VM of v
7:                 break
8:             end if
9:         end for
10:        Connect VM v with d
11:    end for
12: end procedure
```

TABLE II. Data Center Simulation Scenarios

Parameter		Two-Tier	Three-Tier	Jupiter
Small data center scenario	Num. Servers	180	180	192
	Num. Switches	28	27	44
Large data center scenario	Num. Servers	4600	4800	4608
	Num. Switches	530	504	464

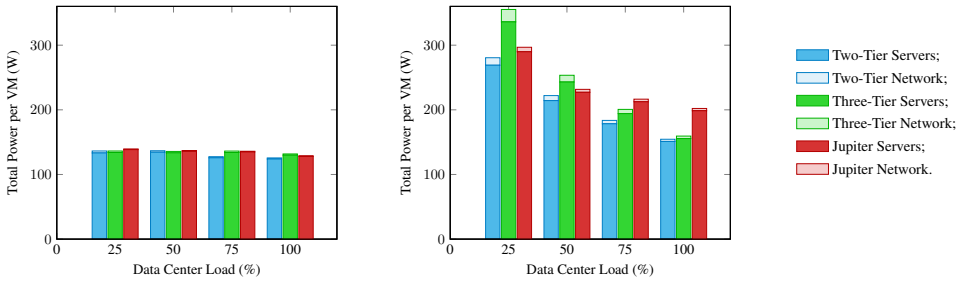
In the experiments, the VMs can not migrate and for simplicity we do not consider the case in which a VM finishes and leaves the server. Thus the overall load of the data center increases with the number of allocated VMs. In the case of a blocking event during the allocation of a new VM, the algorithm keeps generating new VMs until it reaches a given maximum number of VMs. The total number of VMs generated in each simulation run is set equal to $50\,000$. This simulation approach has two advantages. First, it allows to test the data center allocation for different values of load with just one simulation run; runs are only repeated to get acceptable confidence intervals for each level of load. Second, the data center keeps receiving requests until it completely saturates either in terms of server or network resources. This provides a kind of worst-case load scenario.

The network topology interconnecting servers is modeled with a directed graph, in which each edge is associated with a capacity measured in Gbps. The communication between VMs is simulated at the flow level, thus by allocating the requested bandwidth on the path connecting the two VMs. Notably, the simulation of the traffic at the flow level allows to investigate also large data center networks. For performance evaluation we considered two main scenarios, whose details are provided in Table II. All the three architectures have been scaled down to fit a given number of servers, following the original structure of each interconnection network. Note that



(a) Two-tier (b) Three-tier (c) Jupiter

Fig. 2. Comparison of network power per VM for small data center scenario



(a) Analysis of MNP policy (b) Analysis of RSS policy

Fig. 3. Comparison of network and server power consumption with LIN power consumption profile

the actual number of servers has been chosen to be compatible with the number of ports of the ToR switches. We devised a small data center scenario, with around 200 servers, to better assess and analyze the joint effect of the allocation policies and power consumption profiles. To validate the results on larger data centers, we considered also a large data center scenario, with around 5000 servers.

To be able to compare fairly the performance for the different architectures, we used the average power per VM metric. This is an interesting way to compare different data center architectures, since for the cloud operator it is directly related to the operational cost of each VM. In addition, it ensures our analysis to be completely independent of the number of devices each architecture actually hosts. In our results, we will consider the *total power*, obtained by summing the contribution of the servers and of the network devices, and the *network power*, obtained by considering only the contribution of the network devices.

B. Experimental Results

For evaluation purposes we have considered as power consumption profiles CONST, FEP and LIN, being the REAL profile similar to the LIN one. Fig. 2 compares the per-VM power consumption for the considered 3 data center networks and the two allocation policies (MNP/RSS). This analysis is performed for small data centers. The MNP policy outperforms the RSS policy for low and medium loads of the data center (25-75%) in all the three networks. Interestingly, considering the FEP power consumption profiles for the equipment and loads that saturate the data center capacity, the gain using an optimized policy such as MNP is minimum. It should also be noted that MNP policy ensures loading the data center in a energy-proportional manner, as the power spent per VM remains almost constant. On the other hand, the RSS policy

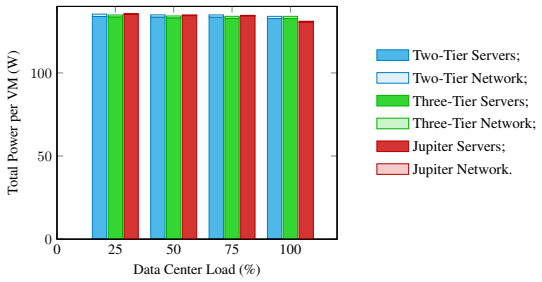


Fig. 4. Comparison of power consumption per VM for large data center scenario with MNP policy

at low loads of the data center requires more power per VM because the workload is not consolidated at the servers. As a result more devices need to be powered on. The two- and three-tier architectures in Fig. 2(a) and Fig. 2(b) provides similar performance while adopting the MNP policy. However, with the RSS policy, the two-tier architecture achieves better results. To illustrate this behavior, the power per VM spent by the two- and three-tier architectures is 11.57 W and 19.15 W respectively, for a data center load equal to 25% and taking into account the LIN power consumption profile. This is because the three-tier design relies on high power consuming devices in aggregation and core layers. Jupiter achieves better performance if compared with the two- and three-tier design (see Fig. 2(c)). Although Jupiter has more switches, the power per VM that is attributed to the network is lower for medium-low loads of the data center and becomes comparable with the other architectures only for high loads. Considering that Jupiter exploits low power demanding switches, the reason behind such performance is due to the higher number of VMs it can support.

Fig. 3 compares the performance of the architectures by

comparing the amount of power spent to operate the servers and the switches separately. For this analysis, we consider only the LIN profile for the IT equipment, being such a profile the most similar to a realistic one (see Fig. 1). As expected, the MNP policy provides higher energy savings, if compared with the RSS policy. In both policies, the computing servers account the most for power consumption. This is especially evident for the MNP policy in Fig. 3(a), where the network consumption accounts for only 1 to 2% of the overall power consumption. This is because through consolidation, the servers powered on become fully loaded and they contribute the most to the per VM power consumption (see Table I). For the MNP policy, all architectures provides similar performance, with the two-tier architecture being slightly superior only for high loads of the data center. This is not true for the RSS policy. Surprisingly, Fig. 3(b) shows that the three-tier topology achieves high gains being the most power hungry architecture at low loads and is efficient at high loads, where the difference with the performance of the two-tier architecture are minimal. The reason is two-fold. On one hand, the three-tier architecture requires more servers in different racks to be on. On the other hand, the three-tier architecture hosts high power consuming devices in the upper layers. As a result, their contribution to the power consumption at low loads has a high influence on the cost per VM. Indeed, the network accounts for 5.3% of the overall power consumption for a data center load of 25% and this value drops to 2.4% at 100% of load. For the two-tier architecture and Jupiter, this behaviour is less evident.

Having analyzed the performance in a controlled environment, we validate the performance of the MNP policy for large-scale data centers having considered the LIN power consumption profiles for the IT equipment. Indeed, such a policy achieves higher energy savings than the RSS policy, being of comparable complexity. Fig. 4 illustrates the results obtained. As expected, the total power spent per VM is in the same range of values of the results obtained for the small data center scenario, see Fig. 3(a). The graph confirms the ability the MNP policy in loading the data center in a energy-proportional manner. However, in the large scale data center scenario, the two tier architecture does not achieve anymore better performance for medium-high loads of the data center.

V. Conclusions

In this paper we compared the performance of different data center architectures, including two- and three-tier topologies and Jupiter, that is the architecture Google currently implements in its data centers. For comparison, we analyze jointly the impact on the overall energy consumption given by two opposite resource allocation policies (RSS which is oblivious of the network state, and MNP which is network-aware) and under different power consumption profiles for the devices.

The results of this preliminary analysis reveal that optimized resource allocation strategies such as MNP load the data center in a energy-proportional manner and this is independent of the the specific network topology. On the other hand, for network-unaware policies, the two and three-tier architectures provide significant benefits than Jupiter in terms of the power spent per VM.

Acknowledgment

Dzmitry Kliazovich and Claudio Fiandrino would like to acknowledge the funding from National Research Fund, Luxembourg in the framework of ECO-CLOUD and iShOP projects.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[2] "Cisco Global Cloud Index: Forecast and Methodology, 2013-2018," 2014, White Paper.

[3] D. Kliazovich, J. Pecero, A. Tchernykh, P. Bouvry, S. Khan, and A. Zomaya, "CA-DAG: Modeling communication-aware applications for scheduling in cloud computing," *Journal of Grid Computing*, pp. 1–17, 2015.

[4] P. Corcoran and A. Andrae, "Emerging trends in electricity consumption for consumer ICT," 2013, White Paper. [Online]. Available: http://hdl.handle.net/10379/3563

[5] U.S. Department of Energy, "Improving data center efficiency with rack or row cooling devices," 2012, White Paper.

[6] L. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[7] Q. Zhang and R. Boutaba, "Dynamic workload management in heterogeneous cloud computing environments," in *IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–7.

[8] C. Fiandrino, D. Kliazovich, P. Bouvry, and A. Zomaya, "Performance and energy efficiency metrics for communication systems of cloud computing data centers," *IEEE Transactions on Cloud Computing*, 2015.

[9] Y. Shang, D. Li, and M. Xu, "A comparison study of energy proportionality of data center network architectures," in *IEEE ICDCSW*, 2012.

[10] A. Hammadi and L. Mhamdi, "A survey on architectures and energy efficiency in data center networks," *Computer Communications*, vol. 40, pp. 1–21, 2014.

[11] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM*, 2008.

[12] N. Mysore *et al.*, "PortLand: a scalable fault-tolerant layer 2 data center network fabric," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, 2009, pp. 39–50.

[13] A. Greenberg *et al.*, "VL2: a scalable and flexible data center network," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, 2009, pp. 51–62.

[14] A. Singh *et al.*, "Jupiter rising: A decade of Clos topologies and centralized control in Google datacenter network," in *ACM SIGCOMM*, 2015.

[15] C. Guo *et al.*, "BCube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.

[16] C. Guo *et al.*, "DCell: a scalable and fault-tolerant network structure for data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, 2008, pp. 75–86.

[17] M. Guzek, P. Bouvry, and E.-G. Talbi, "A survey of evolutionary computation for resource management of processing in cloud computing," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 53–67, May 2015.

[18] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, and I. Stoica, "A cost comparison of datacenter network architectures," in *CoNEXT*. ACM, 2010.

[19] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Computer Networks*, vol. 57, no. 1, pp. 179 – 196, 2013.

[20] M. Guzek, D. Kliazovich, and P. Bouvry, "HEROS: Energy-efficient load balancing for heterogeneous data centers," in *IEEE CLOUD*, 2015.

[21] D. Belabed, S. Secci, G. Pujolle, and D. Medhi, "Striking a balance between traffic engineering and energy efficiency in virtual machine placement," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 202–216, June 2015.

[22] L. Wang, F. Zhang, J. Arjona Aroca, A. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "GreenDCN: A general framework for achieving energy efficiency in data center networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 1, pp. 4–15, January 2014.