GreenPOD: Leveraging Queuing Networks for Reducing Energy Consumption in Data Centers

Fatoumata Balde
Université du Québec À Montréal,
Université Cheikh Anta Diop de Dakar
Email: balde.fatoumata@courrier.uqam.ca

Halima Elbiaze
Université du Québec À Montréal
Email: elbiaze.halima@uqam.ca

Bamba Gueye
Université Cheikh Anta Diop de Dakar
Email: bamba.gueye@ucad.edu.sn

Abstract—**Reducing the energy consumed in Clouds is becoming an increasingly challenging research direction due to the exponential growth of services hosted there and delivered. The big concern of data center providers is to find a trade-off between the offered resources and the current traffic load. GreenPOD aims to reduce the consumed energy as well as to satisfy predefined response time for fat-tree data center networks by using queuing theory based on mathematical model. GreenPOD considers different activation thresholds based on request queuing to switch ON/OFF different POD which is formed by a certain group of servers when needed. Our results show that GreenPOD outperforms previous energy-aware queuing theoretical model. In respect of a moderately demanded system (*resp.* highly demanded system), GreenPOD saves up to 61.72% (*resp.* 47.75%).**

Keywords*—*Energy consumption, Waiting time, Queuing theory, Dynamic management.

I. INTRODUCTION

The growing demand for Internet and Cloud services have dramatically increased the number of networking devices and servers operating in large data centers. As consequence, we note the soaring energy consumption of data centers. Nowadays, the consumed energy by data centers has received a lot of attention [1] [2] [3]. However, the peak load level of data center is about 60% of [4]. In contrast, when server's utilization rate are low, they still consume 65% of their maximum power consumption [1]. Therefore, former energy-aware data center works proposed approaches that turn off unused servers during low hours of workload [5], [6] or dynamically enabled/disabled one group of backup servers [7]. By enabling the possibility to dynamically activate only a subset of servers, it is mandatory to take into account the routing in order to avoid traffic congestion within the network [8].

Indeed, the performance of an energy-aware resource allocation is not only relevant to its dynamic management of networking devices and servers [9] [10] but also on the used topology and routing algorithms within the data center when a large number of servers are involved. Fortunately, fat-tree topology provides a full bisection bandwidth and offers a rich connectivity between nodes that ensure packets to be received by the destination [11] [8].

Our goal is to evaluate the power consumption within data centers from core routers to servers and saved energy when unused servers and switches are turned off. In contrast to previous studies [9] [10], *GreenPOD* considers a fat-tree topology [11] and different activation thresholds with respect to networking devices and servers in order to reduce energy consumption. Indeed, we develop a queuing model which depicts at each level of the fat-tree topology the number of servers within a given POD that should be activated in correlation with a fixed activation threshold. The decision to switch ON/OFF networking devices is based on the current load and the waiting time queue length. Furthermore, it takes into account the response time according to applications Service Level Agreement. GreenPOD reduces consequently the consumed energy under different traffic load by maintaining a good response time.

The rest of the paper is organized as follows. In Section II, related work on energy-efficient data centers is discussed. Section III describes and evaluates analytically GreenPOD energy consumption model. Then, we evaluate in Section IV the GreenPOD architecture under different network settings. Finally, Section V concludes the paper.

II. RELATED WORK

Several authors proposed new architectures for data centers whose deployment is less costly and offers more resilience [8] - [12]. However, only few works considered the energy consumed within data center. Heller et *al.* [13] tried to find a trade off between energy efficiency and resilience. They considered a fat-tree architecture which is similar to [14]. According to their approaches, they disabled additional switches used for backup paths. These switches will be activated in case of high load or network failures. For instance, *OpenFlow* protocol is used as switch management. Nevertheless, only switches are disabled and not servers. The authors of [15] showed that these devices only consume between 5% and 10% of the overall energy consumption.

More recent studies have also studied dynamic cases [6], [7], [15] that activate and turn off servers according to the current load. In [6], the authors aim to find a trade-off between the power consumed and the quality of service by introducing a policies that dynamically adapt the number of running servers. The authors of [7] propose a simple and multi-server model, where inter-arrival times and service time are exponentially distributed. Guenter et *al.* [16] considered a similar problem,

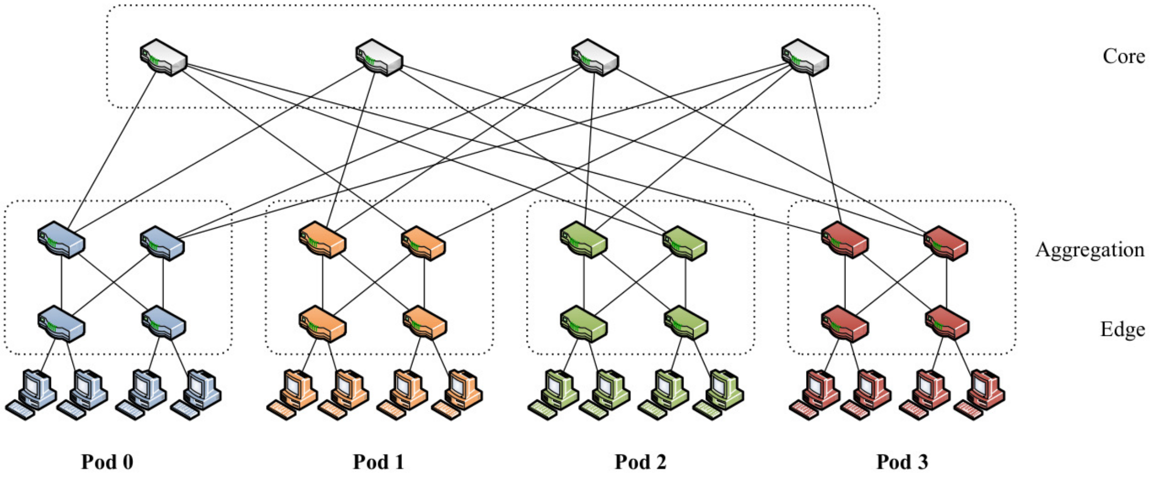

Figure 1. Fat-tree topology

but proposed to predict the future service request by using Markov chain to determine the number of active servers.

Schwartz et *al.* [9] split the number of servers into two groups, the first group is always activated and the second group is dynamically enabled/disabled. By activating the number of servers dynamically according to the current load and a fixed thresholds they can reduce the energy consumption. Based on the queue size, they are able to turn off the second group of servers. Dan et *al.* works [10] are similar to [9] and both consider a tree-third architecture. They split the number of servers into three groups. The first group is still activated and the remaining groups are dynamically switched ON according to the activation thresholds based on the queue size. It is worth noticing that the heuristics proposed in [10] outperforms the former works depicted in [9]. The authors of [10] argue that it is better to maintain a small group of servers if they can support the load peak as well as achieve an acceptable response time. By so doing, they are able to reduce the power consumption.

III. BACKGROUND ON GREENPOD

GreenPOD is based on mathematical models and aims to reduce the power consumption in data centers. It considers a k-array fat-tree multistage topology [11] [8] [17] where k represents the number of POD within the considered topology. In fact, each level-n network can have several level-$(n-1)$ fat-tree subnetworks called *POD* of the level-n network. For instance, Figure 1 depicts a fat-tree topology [17]. In fact, the total number of servers is $\frac{k^3}{4}$ and a quarter of the servers and switches are started as follows : $\frac{k}{4} * \frac{k^2}{4}$ servers and $\frac{k}{4} * k$ switches. Based on a given activation thresholds, the other groups of servers, which represent the number of servers in a given POD, are turning ON/OFF.

Initially, only core switches and one quarter of the PODs are activated. Afterwards, we are able to reduce energy consumption at servers and switches level. Furthermore, the activation threshold is in correlation with the queue size. Also, GreenPOD is able to check whether target servers have tasks in progress before to turn it off. By so doing, we are able to avoid requests lost. Otherwise, arrival requests can be suspended pending by putting them in queue and thus GreenPOD can switch OFF a fixed set of servers.

A. System model

We assume that the requests arrival follow a *Poisson* distribution with mean rate λ and each server processes only one request with an exponential distributed service time with mean rate μ. The system can be modeled as a $M/M/\frac{k^3}{4}$ queue. When a new job that arrives in the system is sent to a server in idle state among the $\frac{k^3}{16}$ which constitute the quarter of the servers. These servers are still switched ON whatever the number of requests in the system. If the servers are busy, the request is queued until a server is available. Server's behavior that belong to this group is illustrated in Figure 2.

The first servers group behave as a default data center where all servers and networking devices are still activated. Therefore, they are in standby state which means "idle" or "busy" where the power consumption is high. Figure 3 illustrates the behaviour of the second group of servers which are activated/deactivated with respect to the current traffic load. Servers can be either at "OFF" state where the power consumption is zero, or "idle" or "busy".

Let P a random variable which represents the number of jobs in the system and P_j the probability to have j jobs in the system. In each POD, $\frac{k}{2}$ (respectively $(\frac{k}{2})^2$) edge and aggregation switches (respectively switches core) are switched on at the beginning. It is worth noticing that we have $\frac{k^3}{4}$ servers in our system and e_busy means power consumption where the server is busy. Since each server serves on job, if we have

Level	Number of activated servers	Number of jobs waiting in queue	Number of jobs in the system	Deactivation threshold
0	$\frac{k^3}{16}$	–		
1	$\frac{k^3}{16}+\frac{k^2}{4}$	$\theta_1 \in \left[S_0, S_0+\frac{k^2}{4}\right]$	$S_0+\theta_1$	S_0
2	$\frac{k^3}{16}+\frac{k^2}{2}$	$\theta_2 \in \left[S_1, S_1+\frac{k^2}{4}\right]$	$S_1+\theta_2$	S_1
3	$\frac{k^3}{16}+\frac{3k^2}{4}$	$\theta_3 \in \left[S_2, S_2+\frac{k^2}{4}\right]$	$S_2+\theta_3$	S_2
…	…	…	…	…
i	$\frac{k^3}{16}+\frac{ik^2}{4}$	$\theta_i \in \left[S_{i-1}, S_{i-1}+\frac{k^2}{4}\right]$	$S_{i-1}+\theta_i$	S_{i-1}

Table I. NUMBER OF ACTIVATED SERVERS AND DEACTIVATION THRESHOLD AS FUNCTION OF LEVEL-i

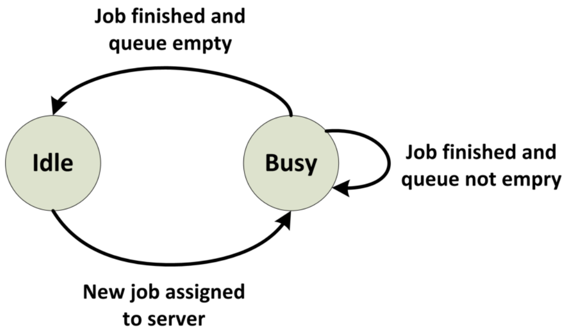

Figure 2. State models of the first group of servers

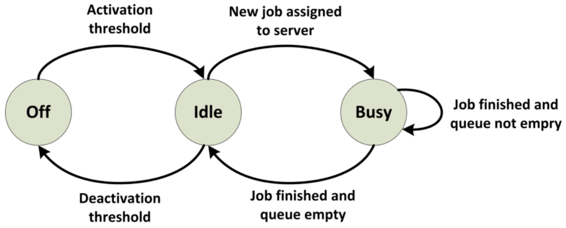

Figure 3. State models of the remaining group servers

j jobs in the system and $j < \frac{k^3}{4}$ then $\frac{k^3}{4} - j$ servers are in idle state.

According to a default data center as baseline, the overall power consumption called E_{max}, which is an upper bound, is computed as follows:

$$E_{max} = \left[\sum_{j=0}^{\frac{k^3}{4}} P_j * (j * e_{busy} + (\frac{k^3}{4} - j) * e_{idle}) + e_{busy} * \sum_{j=\frac{k^3}{4}+1}^{\infty} P_j\right] + \frac{k^2}{2}(e_{busy}(S_e) + e_{busy}(S_a)) + \frac{k^2}{4}(e_{busy}(S_c)) \quad (1)$$

In Eq. 1, e_{busy} (*resp.* e_{idle}) represents the power consumed by an activated server (*resp.* a server in idle state); $e_{busy}(S_e)$, $e_{busy}(S_a)$, and $e_{busy}(S_c)$ represent respectively the power consumption when edge, aggregation and core switches are busy.

It should be noted that the power consumption of deactivated servers is zero and it is represented by e_{off}. Therefore, the lower bound of the power consumption according to number of job j is expressed as follows:

$$E_{min} = \left[\sum_{j=0}^{\frac{k^3}{4}} P_j * (j * e_{busy} + (\frac{k^3}{4} - j) * e_{off}) + e_{busy} * \sum_{j=\frac{k^3}{4}+1}^{\infty} P_j\right] + N_{POD}(e_{busy}(S_e) + e_{busy}(S_a)) + \frac{k^2}{4}(e_{busy}(S_c) \quad (2)$$

where N_{POD} represents the number of active PODs.

GreenPOD considers "i" activation/deactivation thresholds. The system is split into k server groups (PODs). Initially, the $(\frac{k}{4})$ group of servers are activated and the number of activated servers S_0 at level-0 is equal to $\frac{k}{4} \times (\frac{k}{2})^2$, where $\frac{k}{4}$ represents the number of activated POD and $(\frac{k}{2})^2$ the number of used servers within each POD. Consequently, the total number of activated servers is $\frac{k^3}{16}$.

The i^{-th} server group can be activated if the number of requests waiting in the system exceeds the threshold θ_i with $\theta_i \in [S_{i-1}, S_{i-1} + (\frac{k}{2})^2]$ and S_{i-1} is the number of activated servers at level-$i-1$. In other words, as soon as there are $S_{i-1} + \theta_i$ requests in the system, group i is activated. This group of servers still remain activate until the total number of waiting requests in the queue is equal to S_{i-1}.

Table I shows the number of activated servers with respect to a fixed deactivation threshold. The number of switched ON servers is illustrated as function of level-i.

B. Analytical model

Figure 4 describes our queuing theory-based model which enables to evaluate the power consumption in data centers. The considered model is as a set of systems S_i^j, where j is the

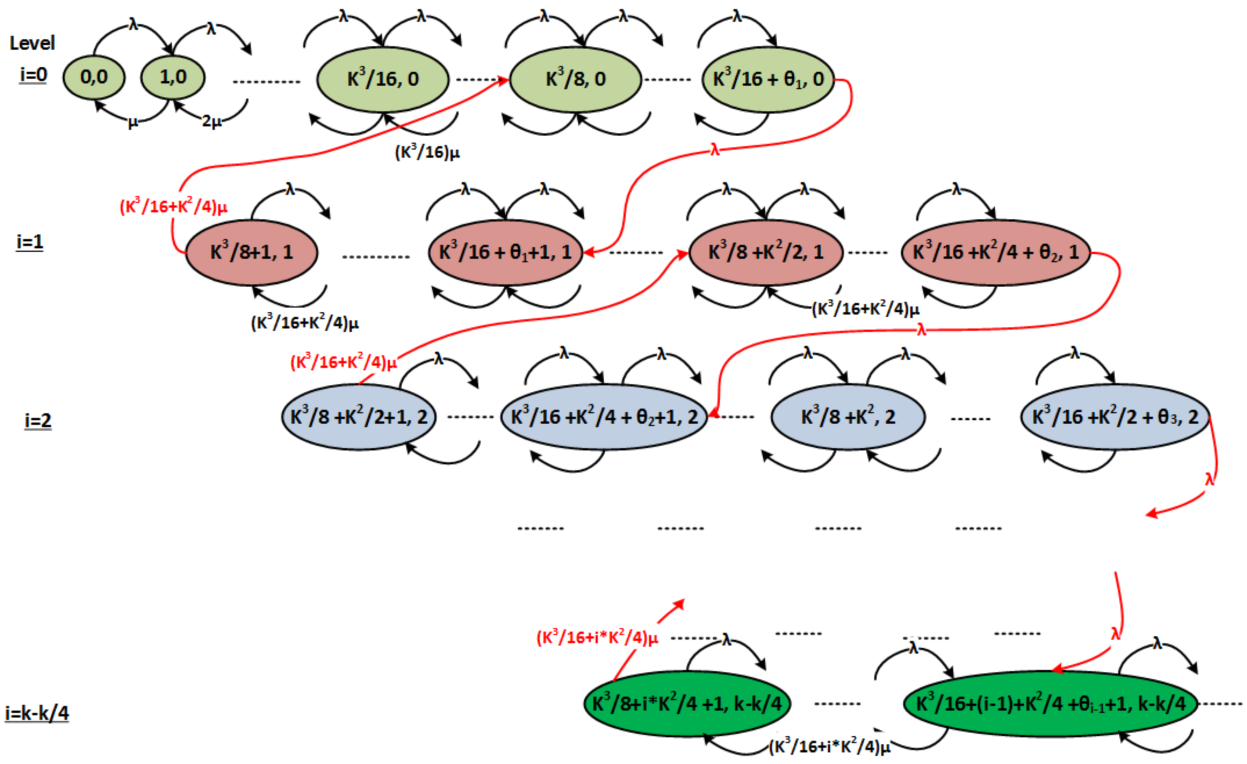

Figure 4. System model: M/M/$\frac{k^3}{4}$

number of jobs in the system and i the group servers at level-i.

$$i = \begin{cases} 0 \text{ if only the base servers } S_0 \text{ are activated} \\ 1, 2, ...3 \times \frac{k^3}{4} \text{ if remaining servers are activated} \end{cases}$$

The system activates a group of servers i if there are at least θ_i jobs in the queue. Put simply, if we have $S_{i-1} + \theta_i$ jobs in the system. Furthermore, the server group at level-i is deactivated if the number of job in the queue is equal to S_{i-1}. Let P a random variable and S_i^j the probability to have j job in the system at level-i. We begin by formulating the first equations according to S_0^j.

$$j\mu P_j^0 = \lambda P_{j-1}^0$$

$$S_0\mu P_j^0 = \lambda P_{j-1}^0$$

$$S_0\mu P_j^0 = S_1\mu P_{2S_0}^1 + \lambda P_{j-1}^0$$

for i← 1 to $\frac{3k}{4}$

$$S_i\mu P_j^i = \lambda P_{j-1}^i + \lambda P_{S_{i-1}+\theta_i}^{i-1}$$

$$S_i\mu P_j^i = \lambda P_{j-1}^i$$

$$S_i\mu P_j^i + S_{i+1}\mu P_{2S_i}^1 = \lambda P_{j-1}^i$$

$$\frac{k^3}{4}\mu P_j^{\frac{k^3}{4}} = \lambda P_{j-1}^{\frac{k^3}{4}}$$

C. Calculating probabilities

In order to compute performance metrics such as response time and power consumption, firstly, we evaluate the distribution probability as follows:

$$P_j^0 = \left(\frac{\lambda}{\mu}\right)^j \times \frac{1}{j!}P_0^0$$

$$P_j^0 = \frac{\left(\frac{\lambda}{\mu}\right)^j}{S_0!S_0^{j-S_0}}P_0^0$$

$$P_j^0 = \left(\frac{\lambda}{S_0\mu}\right)^{j-2S_0} \times \frac{\left(\frac{\lambda}{\mu}\right)^{2S_0}}{S_0!S_0^{S_0}}P_0^0$$

$$-\sum_{m=0}^{j-2S_0-1} \frac{S_1}{S_0}\left(\frac{\lambda}{S_0\mu}\right)^m P_{2S_{0+1}}^1$$

$$\Rightarrow P_j^0 = \left(\frac{\lambda}{S_0\mu}\right)^{j-2S_0} * \frac{\left(\frac{\lambda}{\mu}\right)^{2S_0}}{S_0!S_0^{S_0}}P_0^0 - \frac{S_1}{S_0} \times \frac{\left(1-\frac{\lambda}{S_0\mu}\right)^{j-2S_0}}{1-\frac{\lambda}{S_0\mu}}$$

$$With \quad \lambda P_{S_0+\theta_1}^0 = S_1\mu P_{2S_0+1}^1,$$

$$P^1_{2S_0+1} = \frac{\left(\frac{\lambda}{S_0\mu}\right)^{\theta_1 - S_0} * \frac{\left(\frac{\lambda}{\mu}\right)^{2S_0}}{S_0!S_0^{S_0}}}{\frac{S_1}{S_0} \times \frac{\left(1-\frac{\lambda}{S_0\mu}\right)^{\theta_1 - S_0}}{1-\frac{\lambda}{S_0\mu}} + \frac{S_1\mu}{\lambda}} P^0_0$$

for i$\leftarrow 1\ to\ \frac{3k}{4}$

$$P^i_{2S_{i-1}+1} = \frac{\left(\frac{\lambda}{S_{i-1}\mu}\right)^{\theta_i - S_{i-1}}}{\frac{S_i\mu}{\lambda} + \frac{S_i}{S_{i-1}} \times \sum\limits_{m=0}^{\theta_i - S_{i-1} - 1} \left(\frac{\lambda}{S_{i-1}\mu}\right)^m} P^{i-1}_{2S_{i-1}}$$

$$P^i_j = \sum_{m=0}^{j-2S_{i-1}} \left(\frac{\lambda}{S_i\mu}\right)^m P^i_{2S_{i-1}+1}$$

$$P^i_j = \left(\frac{\lambda}{S_i\mu}\right)^{j-(S_{i-1}+\theta_i+1)} P^i_{S_{i-1}+\theta_i+1} +$$

$$\sum_{m=0}^{j-1-(S_{i-1}+\theta_i+1)} \left(\frac{\lambda}{S_i\mu}\right)^m \times$$

$$\frac{1}{S_i\mu} \left(S_i\mu P^i_{S_{i-1}+\theta_i+1} - \lambda P^{i-1}_{S_{i-1}+\theta_i} - \lambda P^i_{S_{i-1}+\theta_i}\right)$$

$$P^{\frac{3k}{4}}_j = \left(\frac{\lambda}{\frac{k^3}{4}\mu}\right)^{j-2S_{\frac{3k}{4}}+1} P^{\frac{3k}{4}}_{2S_{\frac{3k}{4}}}$$

$$With \sum_{j=0}^{S_0+\theta_i} P^0_j + \sum_{i=1}^{\frac{3k}{4}} \sum_{j=2S_{i-1}+1}^{S_{i-1}+\theta_i} P^i_j + \sum_{j=2S_{\frac{3k}{4}}-1}^{\infty} P^{\frac{3k}{4}}_j = 1$$

D. Performance metrics

Based on computed probabilities in Section III-C, we are able to evaluate the power consumption and the response time metrics. Therefore, the power consumption of the overall data center is computed as follows:

$$E = \left(\left[\sum_{j=0}^{S_0+\theta_i} P^0_j \left(je_{busy} + (S_0 - j)\, e_{idle} + \beta\right) + \right.\right.$$

$$\sum_{j=S_0}^{2S_0} P^0_j \left(S_0 e_{busy} + \beta\right)$$

$$+ \sum_{i=1}^{\frac{3k}{4}-1} \sum_{j=S_{i-1}+\theta_i+1}^{2S_i} P^i_j \left(\left(S_i e_{busy} + \left(S_{\frac{3k}{4}} - S_i\right) e_{off}\right)\right.$$

$$\left.\left. + P^i_{j>\frac{k^3}{4}} S_{\frac{3k}{4}} e_{busy}\right] + \alpha\right)$$

where $\alpha = N_{POD} \times \left(\frac{k}{2} e_{busy}(S_a) + \frac{k}{2} e_{busy}(S_e)\right) + \frac{k^2}{4} e_{busy}(S_c)$

and $\beta = \left(S_{\frac{3k}{4}} - S_0\right) e_{off}$

The number of jobs waiting in queue is given by:

$$\Omega = \sum_{i=0}^{\frac{k}{4}-1} \left(\sum_{j=S_i}^{S_i+\theta_{i+1}} (j - S_i)\, P^i_j\right) + \sum_{j=S_{\frac{3k}{4}}}^{\infty} \left(j - S_{\frac{3k}{4}}\right) P^{\frac{3k}{4}}_j$$

According to Little's formula, the average waiting time for the system is: $T_a = \frac{\Omega}{\lambda}$

IV. PERFORMANCE EVALUATION

A. Experimental settings

We considered a system with 432 servers ($k = 12$ for a fat-tree topology). Each server has a service time equal to 0.2 second. The power consumption according to a busy server called e_{busy} is equal to 240 $watt$. The power consumption of a server in idle state called e_{idle} is equal to 150 $watt$. A server in OFF state consumes zero power ($e_{off} = 0\ watt$).

We compare GreenPOD with Dan et *al.* work [10]. For both proposals, a fixed group of servers is activated at the beginning of the simulation. According to the queue size, we can enable or disable new servers group. In [10], servers are subdivided into three groups of n, $m1$ and $m2$ servers. As soon as the activation thresholds θ_1 and θ_2 are reached, the next group is activated. We considered the same activation and deactivation thresholds. According to GreenPOD, a new group of servers is activated as soon as the queue size exceeds a given number of servers after activation. We deactivate a group of servers as soon as the queue size is less than the enabled number of servers. Nevertheless, GreenPOD checks whether target servers have tasks in progress before to turn it OFF.

B. Numerical results

Figure 5 illustrates the power consumption as a function of the simulation time for a relatively uninvolved system (utilization rate = 0.45). The average power consumption for GreenPOD (*resp.* Dan et *al.*) is equal to 24167.61 $watt$ (respectively 29567.61 $watt$). The obtained results exhibit clearly that GreenPOD outperforms Dan et *al.* in situation of low utilization rate. In contrast, for a default data center without any energy-aware mechanism and where all servers are started at the same time, the average power consumption is estimated at 72767.61 $watt$. According to non-energy-aware data center, GreenPOD is able to save up to 66.79%.

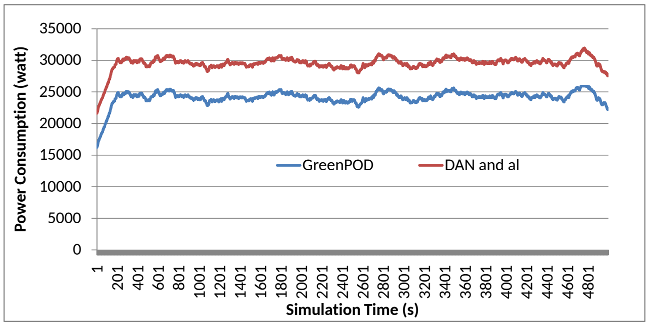

Figure 5. Power consumption of a low-demand system

Figure 6 shows the power consumption as a function of simulation time when we consider a moderately loaded system (utilization rate = 0.65). The GreenPOD (*resp.* Dan et *al.*) average power consumption is estimated at 29242.66 $watt$ (*resp.* 33189.05 $watt$). According to a default data center, the power consumption is equal to $76389.05\ watt$. In comparison with a default data center, we save up to 61.72% of energy whereas Dan et *al.* gain is 56.55%.

Figure 7 depicts the power consumption and the waiting time as a function of simulation time. One can notice that each time that we activate a group of servers, the waiting time decreases slightly. In contrast, during a couple of deactivations, we notice a sudden increase of the waiting time. This is due to the fact that current jobs in progress are put at the top of the queue.

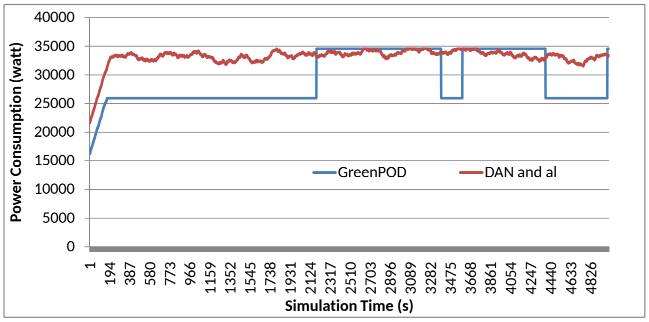

Figure 6. Power consumption of a moderately loaded system

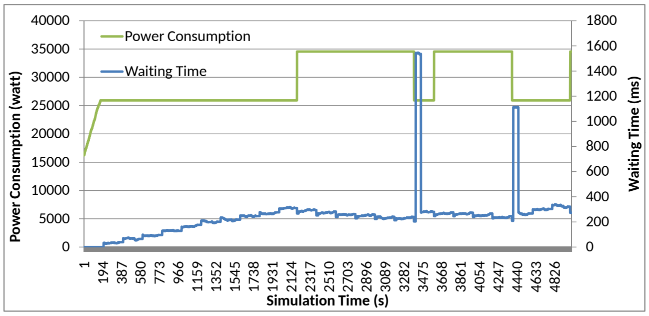

Figure 7. GreenPOD: power consumption vs waiting time according to a moderately loaded system

Figure 8 shows the power consumption as a function of the simulation time for both proposals. For a highly demanded system (utilization rate = 0.95), the GreenPOD (*resp.* Dan et *al.*) average power consumption is $42652.39\ watt$ (*resp.* $45447.26\ watt$). In contrast, by considering a default data center, the computed average power consumption is $81637.16\ watt$. According to a default data center, the amount of energy saved by GreenPOD (*resp.* Dan et al.) is 47.75% (*resp.* 44.33%).

Figure 9 shows the power consumption and waiting time of GreenPOD for a highly demanded system. The same trend is observed with respect to a moderately loaded system. We also note the variation of the waiting time when a group of servers is switched OFF.

Figure 10 illustrates the obtained response time with respect to both approaches. The average GreenPOD (*resp.* Dan et *al.*) waiting time is equal to $290.38ms$ (*resp.* $159.36ms$). The obtained response time with respect to Dan et *al.* outperforms GreenPOD. It is worth noticing that in Dan et *al.* approach when the queue size increases they activate speedily new group of servers. As results, they increased of the power consumption.

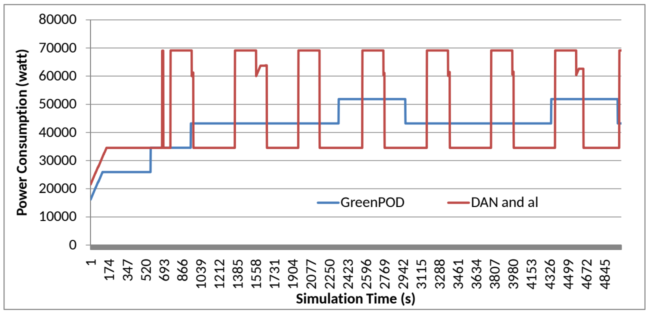

Figure 8. Power consumption according to a highly demanding system

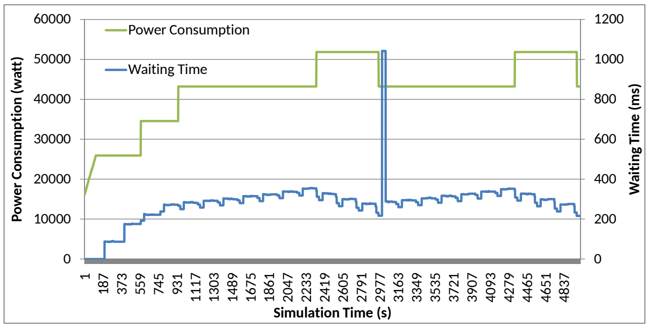

Figure 9. GreenPOD: power consumption vs waiting time according to a highly demanded system

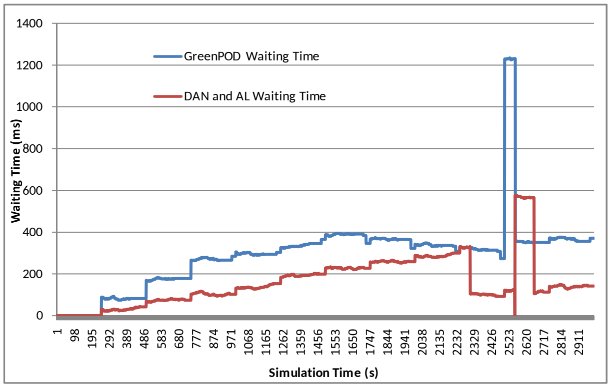

Figure 10. Waiting time evaluation

Figure 11 shows the total power consumption (Switches and servers) for to both proposals. Indeed, when the number of POD k increase the gap between GreenPOD and Dan et *al.* is more important. The same trend is noted in Figure 12 where we illustrated GreenPOD total power consumption with respect to a default data center.

Based on Figures 11 and 12, for $k = 14$, GreenPOD saves up to 48% of power consumption compared to a default data center, whereas Dan et *al.* saves 44%. For instance, for $k = 36$,

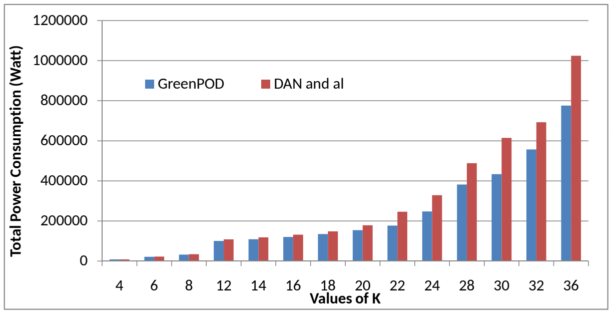

Figure 11. Total power consumption (switches + servers)

GreenPOD (*resp.* Dan et *al.*) reduces up to 65% (*resp.* 54%).

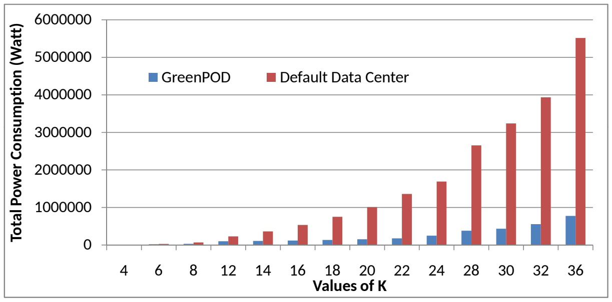

Figure 12. Total power consumption (switches + servers)

Figure 13 depicts the power consumption depending on the system load and the same time gives the simulation time for each case. In this figure, we have only represented the power consumption of the servers. GreenPOD still outperforms Dan et *al.* with respect to power consumption.

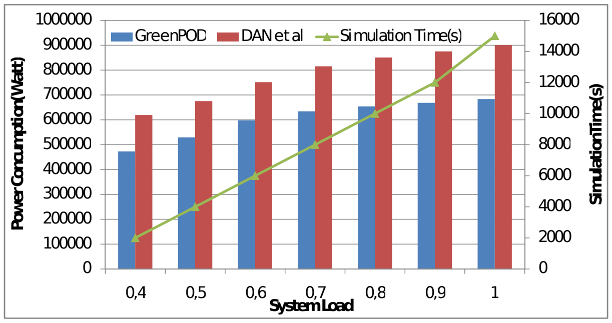

Figure 13. Power consumption and simulation Time according to the load

Figure 14 shows the power consumption as a function of system load. For each case, it gives the corresponding simulation time. The power consumption concerns both switches and servers. According to overall data center consumption, the same trend than Figure 13 is observed.

V. CONCLUSION

In this paper, by taking into account a Fat-tree topology, we proposed a new analytic model with different activation thresholds in order to reduce power consumption in data center. Our activation and deactivation threshold is based on current network load. We compared GreenPOD with Dan et *al.* energy-aware approach presented in [10]. GreenPOD saves more energy compared to Dan et *al.* approach while achieving an acceptable response time. In our future work, we plan to apply our solution by using a data center with heterogeneous servers. Also, we will analyze our model in a more complex environment by considering a batch of Poisson arrivals, different client classes and different service times.

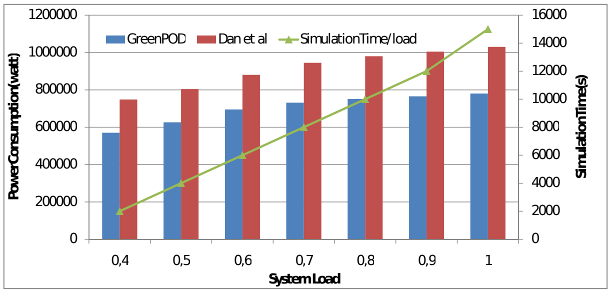

Figure 14. Total power consumption(switches + servers) and simulation Time according to the load

REFERENCES

[1] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.

[2] F. Baldé, B. Gueye, and H. Elbiaze., “Green cloud architecture: Low power routers for an energy-aware data transport,” in *IEEE Globecom CCSNA*, Dec. 2014.

[3] ——, “Leveraging network virtualization for energy-efficient cloud: Future directions,” in *39th IEEE LCN*, Sept. 2014.

[4] T. Benson, A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the wild,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’10. New York, NY, USA: ACM, 2010, pp. 267–280.

[5] M. Mazzucco, D. Dyachuk, and R. Deters, “Maximizing cloud providers’ revenues via energy aware allocation policies,” in *2010 IEEE 3rd International Conference on Cloud Computing*, July 2010, pp. 131–138.

[6] D. Dyachuk and M. Mazzucco, “On allocation policies for power and performance,” in *2010 11th IEEE/ACM International Conference on Grid Computing*, Oct 2010, pp. 313–320.

[7] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, “Optimality analysis of energy-performance trade-off for server farm management,” *Perform. Eval.*, vol. 67, no. 11, pp. 1155–1171, Nov. 2010.

[8] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[9] C. Schwartz, R. Pries, and P. Tran-Gia, “A queuing analysis of an energy-saving mechanism in data centers,” in *International Conference on Information Networking (ICOIN)*, Bali, Indonesia, 2 2012.

[10] D. Liao, K. Li, G. Sun, V. Anand, Y. Gong, and Z. Tan, “Energy and performance management in large data centers: A queuing theory perspective,” in *2015 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2015, pp. 287–291.

[11] C. G. Requena, F. G. Villamón, M. E. Gómez, P. López, and J. Duato, “Deterministic versus adaptive routing in fat-trees,” in *21th International Parallel and Distributed Processing Symposium (IPDPS 2007), Proceedings, 26-30 March 2007, Long Beach, California, USA*, 2007, pp. 1–8.

[12] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, “Bcube: A high performance, server-centric network architecture for modular data centers,” *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, Aug. 2009. [Online]. Available: http://doi.acm.org/10.1145/1594977.1592577

[13] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “Elastictree: Saving energy in data center networks,” in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 17–17.

[14] R. Pries, M. Jarschel, D. Schlosser, M. Klopf, and P. Tran-Gia, “Power consumption analysis of data center architectures,” in *GreenNets*, Colmar, France, 10 2011.

[15] M. Ghamkhari and H. Mohsenian-Rad, “Energy and performance management of green data centers: A profit maximization approach,” *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 1017–1025, June 2013.

[16] B. Guenter, N. Jain, and C. Williams, “Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning.” in *INFOCOM*. IEEE, 2011, pp. 1332–1340.

[17] R. HARRIS, “Fat trees and skinny switches,” Aug. 2008. [Online]. Available: https://storagemojo.com/2008/08/24/fat-trees-and-skinny-switches/