Research Paper on Green Cloud Computing

Mr. Manoj Kumar[1] Mr. Deepak Gupta[2] Dr. Sanjay Jain[3]
[1]PG Student [2]Assistant Professor [3]Associate Professor
[1,2,3]Department of Computer Engineering
[1,2]Siddhant College of Engineering, India [3]ITM University Gwalier, India

Abstract— Cloud computing is a highly scalable and cost-effective infrastructure for running HPC, enterprise and Web applications. However, the growing demand of Cloud infrastructure has drastically increased the energy consumption of data centers, which has become a critical issue. High energy consumption not only translates to high operational cost, which reduces the profit margin of Cloud providers, but also leads to high carbon emissions which is not environmentally friendly. Hence, energy-efficient solutions are required to minimize the impact of Cloud computing on the environment. In order to design such solutions, deep analysis of Cloud is required with respect to their power efficiency. Thus, in this chapter, we discuss various elements of Clouds which contribute to the total energy consumption and how it is addressed in the literature. We also discuss the implication of these solutions for future research directions to enable green Cloud computing. The chapter also explains the role of Cloud users in achieving this goal.

Key words: TCP/IP-Based Mobile Cloud, Nonconvex Optimization, Internet Data Centers, Dynamic Voltage Frequency Scaling, Resource Provisioning, Energy-Efficiency

I. INTRODUCTION

The commercialization of these developments is defined currently as Cloud computing [2], where computing is delivered as utility on a pay-as-you-go basis. Traditionally, business organizations used to invest huge amount of capital and time in acquisition and maintenance of computational resources. The emergence of Cloud computing is rapidly changing this ownership-based approach to subscription-oriented approach by providing access to scalable infrastructure and services on-demand. Users can store, access, and share any amount of information in Cloud. That is, small or medium enterprises/organizations do not have to worry about purchasing, configuring, administering, and maintaining their own computing infrastructure. They can focus on sharpening their core competencies by exploiting a number of Cloud computing benefits such as on-demand computing resources, faster and cheaper software development capabilities at low cost. Moreover, Cloud computing also offers enormous amount of compute power to organizations which require processing of tremendous amount of data generated almost every day. For instance, financial companies have to maintain every day the dynamic information about their hundreds of clients, and genomics research has to manage huge volumes of gene sequencing data.

Configuration costs in Internet-based virtualized DCs which utilize end-to-end TCP/IP mobile energy constrained connections under hard limits on the per-job total processing time. Our scheduler performs dynamic load balancing and uses online job decomposition for adaptive resource management. It leads to the optimum VM processing speed and bandwidth rates, as well as the proper workload quota for each VM on a per-job basis. Consider that our energy model is a no convex model; hence, we develop a mathematical approach to turn no convexity into convexity. A remarkable feature of our scheduler is its adaptive nature and scalability. The reminder of this paper is organized as follows. After presenting the model of the considered TCP/IP-based mobile Cloud platform in Section II, the approach leading to the proposed joint scheduler are developed in Sections III and IV. The numerical results attained through extensive tests are presented in Section V.

II. CLOUD & ENVIRONMENTAL SUSTAINABILITY

Cloud computing is an evolving paradigm which is enabling outsourcing of all IT needs such as storage, computation and software such as office and ERP, through large Internet. The shift toward such service-oriented computing is driven primarily by ease of management and administration process involving software upgrades and bug fixes. It also allows fast application development and testing for small IT companies that cannot afford large investments on infrastructure. Most important advantage offered by Clouds is in terms of economics of scale; that is, when thousands of users share same facility, cost per user and the server utilization. To enable such facilities, Cloud computing encompasses many technologies and concepts such as virtualization, utility computing, pay as you go, no capital investment, elasticity, scalability, provisioning on demand, and IT outsourcing.

Due to such varying properties of Cloud computing, there are many informal definitions, none of which fully describes it. The literary meaning of “Cloud computing” can be “computing achieved using collection of networked resources, which are offered on subscription”. In terms of qualities of real “Clouds,” which do not have any definite shape or position, Cloud computing is also called “Cloud” since a Cloud server can have any configuration and can be located anywhere in the world. Internet is a fundamental medium through which these Cloud services are made accessible and delivered to end user.

III. FORMULATION OF THE OPTIMIZATION PROBLEM

The proposed scheduler aims at minimizing the per-job total Energy consumption in by selecting the best resource allocation based on the current load level state of the TCP/IP mobile connection. In detail, we build a solution which is able to obtain the optimum values for the processing speeds of the VMs, and the transmission rates of the VLAN and the transmission rate of the mobile TCP/IP connection.

IV. ADAPTIVE SOLUTION OF THE AFFORDED PROBLEM

Algorithm 1 is an instance of the bisection method to solve (13.2) by using higher and lower range for each VM: fi/Li and obtain a rough approximation to a solution which is S.

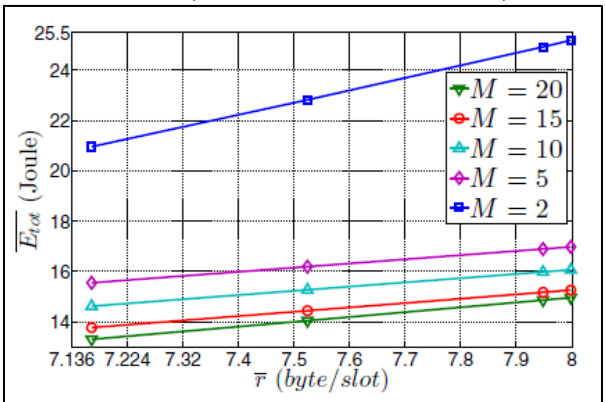
Algorithm 1 AdmCtrlFunc()

1: **Input** t-th Job
2: **Output** $L^{*}_{tot}(t)$, Admission Control parameters
3: Compute $\bar{r}(t)$ by using (15) as Jacobson formula
4: Compute $L^{*}_{tot}(t)$ by using (14)
5: Compute $\lambda(t)$ using $\min\{Job(t), N_I - s(t) + L^{*}_{tot}(t)\}$
6: Compute $r^{*}(t)$ using modified (16)
7: Compute $E^{*}_{W}(t)$ using second part (17)
8: Compute $MU(t+1)$= $[MU(t) + (k/t)(E^{*}_{W}(.) - E_{ave})]$
9: Compute $s(t+1)$ and $q(t+1)$ using (1) and (2), respectively
return $L^{*}_{tot}(t)$, Admission Control parameters

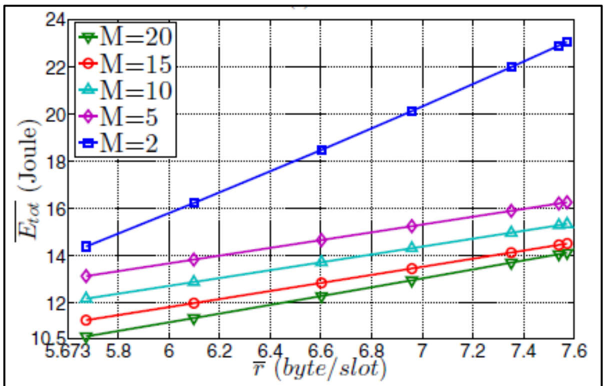
Algorithm 1: AdmCtrlFunc()

V. TEST RESULTS & PERFORMANCE COMPARISONS

This section presents the tested energy performance of the proposed scheduler for a set of synthetic and real workloads and compare it with the most recent DVFS-based techniques in [13], the Lyapunov-based method in [9] and NetDC method in [6]. Simulation Setup. The simulations have been carried out by exploiting the numerical software of the MATLAB platform. They emulate four Dell PowerEdge servers, equipped with 3.06 GHz Intel Xeon CPU and 4GB of RAM. All the emulated servers are connected through commodity Fast Ethernet NICs. In all carried out tests, we configure the VMs with 512MB of memory and utilize the TCP New Reno suite for implementing the needed VM-to-VM transport connections. The simulated parameters are listed in Table I. Simulation Results. In the first test scenario, we have run the proposed scheduler and evaluate Etot under a synthetic workload with Job = 8 Mbyte (input workload before input buffer in Fig. 1) for 2000 slots. Fig. 2 reports the average total consumed energy for various values of M. Specifically, Fig. 2 points out that: i) while M increases, the Etot decreases; and, ii) while r increases, according to (17) and AdmCtrlFunc() function EW and Etot increase. Furthermore, in Fig. 3, we evaluate the admission control (Algorithm 1) under different Jacobson's parameter JP and various values of Eave. Fig. 3 shows that, in the steady state, Ltot is the same as r, λ. S decreases. Furthermore,



(a)



(b)

TABLE I: Simulation parameters for the proposed scheduler.

Parameters	
$\Delta = 1\ (s)$	$f_i = \{0, 5, 50, 70, 90, f_i^{max}\}$
$J_P = \{0.25, 0.75\}$	$\lambda_{max} = 10^3$
R_{max}=15(Mb/s)	$T_t = 5\ (s)$
$N_I = N_O = 240\ (Mbyte)$	$\overline{L}_{tot} = \overline{Job} = 8\ (Mbyte)$
$k_e = \{0.05\}\ (J/(MHz)^2)$	$R_t = 100\ (Mb/s)$
$f_i^{max} = 105(Mbit/s)$	$E_c^{max}(i) = 60(Joule)$
$MSS = 120$	$r_{max} = 16\ (Mbyte/slot)$
$E_{AVE} = \{0.125, 0.25, 0.5, 1\}(Joule)$	
$r_{min} = 0.01\ (Mbyte/slot)$	$n = \#itr^{max} = 10^6$

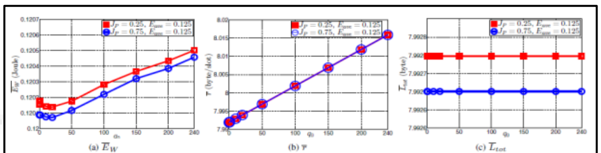


Fig. 3: Admission control: q_0 denotes output buffer length in initial slot, with r_{min}=3.4323, $max(CRTT)$= 8.448 and Δ_{JP}^{max}= 10.

Fig. 3 points out that changing JP has no effect on r and Ltot, so that, the system is capable to work, even under huge input workload. In the last simulations, in order to evaluate the energy reduction due to scaling up/down of the computing and reconfiguration rates, we have also implemented three well-known recent schedulers [6], [9], [13]. The resulting comparisons are presented in Fig. 4. According to Fig. 4, the average energy savings of the proposed scheduler (i.e., the green colored continue plot with −∇−) over the Lyapunov-based one in [9] (i.e., the two upper most plots marked by − ◦ − and labeled as "[9]"), NetDC method [6] (i.e., black dashed plots marked by −_− and labeled by "[6]")and gradient-based iterative method [13] (i.e., dashed red and cyan continue plot marked by −_− and labeled by "[13]") are about 60%, 10%, and 33%, respectively. This confirms that the proposed method is capable to adapt to the incoming five-varying behaviors of the input workload, whilst increasing the VM number faster than NetDC method, which focuses onthe optimum frequency in each time-slot. Lastly, consider that[6], [9] are able to control the processed speed by using DVFS but do not perform TCP/IP optimization, which is performed by our scheduler. Besides, [9] manages time in a average manner and it is unable to manage the online/instantaneous job fluctuations, which is handled by our approach. The final point is that the complexity of the proposed joint scheduler is linear in the number M of the utilized VMs and is lower than the schedulers in [6], [9], because our scheduler takes less iterations for the convergence and, due to using bisection method, compared to iterative gradient based method [13] which uses series of iterative instructions.

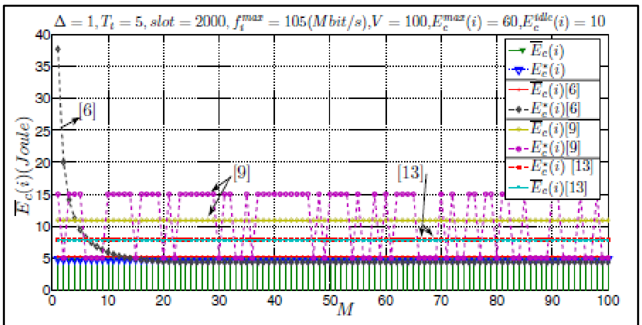
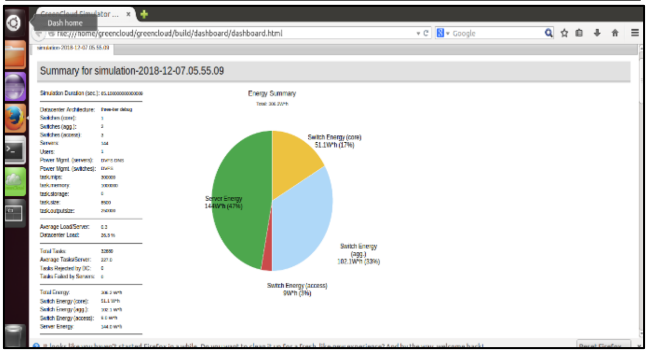


Fig. 4: $\overline{E}_c(i)$ $(Joule)$ for the proposed DVFS-equipped method-vs.- NetDC method in [6]-vs.- Lyapunov method in [9] and vs.- gradient-based iterative method in [13].



VI. Conclusion and Hint for the Future Researches

The goal of this paper is to present an online energy aware resource provisioning scheduler for TCP/IP-based mobile cloud applications. In the process, we identified sources of energy consumptions in data centers and presented high-level solutions. The carried out tests highlight that the average computing savings provided by the proposed joint scheduler over the NetDC, Lyapunov-based, and gradient-based ones may be of 60%, 10%, 33% respectively. The proposed adaptive scheduler may manage not-only the online workload but also the reconfiguration costs, besides the energy needed for transferring the processed data over the mobile TCP/IP connection. Optimizing live migration of VMs without resorting to exhaustive NP-hard numerical approaches could be an interesting topic for future work.

References

[1] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," ACM Computing Surveys (CSUR), vol. 44, no. 3, p. 15, 2012.

[2] Mishra, R. Jain, and A. Durresi, "Cloud computing: networking and communication challenges," Communications Magazine, IEEE, vol. 50, no. 9, pp. 24–25, 2012.

[3] J. Baliga, R. W. Ayre, K. Hinton, and R. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," Proceedings of the IEEE, vol. 99, no. 1, pp. 149–167, 2011.

[4] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Cloud computing networking: challenges and opportunities for innovations," Communications Magazine, IEEE, vol. 51, no. 7, pp. 54–62, 2013.

[5] M. Hirzel, R. Soul´e, S. Schneider, B. Gedik, and R. Grimm, "A catalog of stream processing optimizations," ACM Computing Surveys (CSUR), vol. 46, no. 4, p. 46, 2014.

[6] N. Cordeschi, M. Shojafar, and E. Baccarelli, "Energy-saving selfconfiguring networked data centers," Computer Networks, vol. 57, no. 17, pp. 3479–3491, 2013.

[7] Z. Qian, Y. He, C. Su, Z. Wu, H. Zhu, T. Zhang, L. Zhou, Y. Yu, and Z. Zhang, "Timestream: Reliable stream computation in the cloud," in EuroSys'13. ACM, 2013, pp. 1–14.

[8] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, "Discretized streams: Fault-tolerant streaming computation at scale," in SOSP'13. ACM, 2013, pp. 423–438.

[9] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, "Dynamic resource allocation and power management in virtualized data centers," in NOMS. IEEE, 2010, pp. 479–486.

[10] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated arq over wireless links," Wireless Communications, IEEE Transactions on, vol. 3, no. 5, pp. 1746–1755, 2004.

[11] S. Faruque, "Traffic engineering for multi rate wireless data," in

