Towards Energy-Efficient Cloud Computing: Prediction, Consolidation, and Overcommitment

Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani[†] and Ammar Rayes[‡]
Oregon State University, Corvallis, OR 97331, dabbaghm,hamdaoub@onid.orst.edu
[†] Qatar University, mguizani@ieee.org
[‡] Cisco Systems, San Jose, CA 95134, [‡] rayes@cisco.com

Abstract—**Energy consumption has become a great deal for cloud service providers due to financial as well as environmental concerns. As a result, cloud service providers are seeking innovative ways that allow them to reduce the amounts of energy that their data centers consume. They are calling for the development of new energy-efficient techniques that are suitable for their data centers. The services offered by the cloud computing paradigm have unique specificities that distinguish them from traditional services, giving rise to new design challenges as well as opportunities when it comes to developing energy-aware resource allocation techniques for cloud computing data centers. In this article, we highlight key resource allocation challenges, and present some potential solution approaches for reducing cloud data center energy consumption. Special focus is given to power management techniques that exploit the virtualization technology to save energy. Several experiments, based on real traces from a Google cluster, are also presented to support some of the claims we make in this article.**

I. Why Worry About Energy?

Energy efficiency has become a major concern in large data centers. In the United States, data centers consumed about 1.5% of the total generated electricity in 2006, an amount that is equivalent to the annual energy consumption of 5.8 million households [1]. In US dollars, this translates into power costs of 4.5 billion per year. Data center owners, as a result, are eager now more than ever to save energy in any way they can in order to reduce their operating costs.

There are also increasing environmental concerns that too call for the reduction of the amounts of energy consumed by these large data centers, especially after reporting that the Information and Communication Technology (ICT) itself contributes about 2% to the global carbon emissions [2]. These energy costs and carbon footprints are expected to increase rapidly in the future as data centers are anticipated to grow significantly both in size and in numbers due to the increasing popularity of their offered services. All of these factors have alerted industry, academia, and government agencies to the importance of developing and coming up with effective solutions and techniques that can reduce energy consumption in data centers.

Cloud data centers are examples of such large data centers whose offered services are gaining higher and higher popularity, especially with the recently witnessed increasing reliance of mobile devices on cloud services [3, 4]. Our focus in this article is then on energy consumption efficiency in cloud data centers. We start the article by first introducing the cloud paradigm. We then explain the new challenges and opportunities that arise when trying to save energy in cloud centers. We then describe the most popular techniques and solutions that can be adopted by cloud data centers to save energy. Finally, we provide some conclusions and directions for future work.

II. The Cloud Paradigm

In the cloud paradigm, a cloud provider company owns a cloud center which consists of a large number of servers also called physical machines (PMs). These PMs are grouped into multiple management units called clusters, where each cluster manages and controls a large number of PMs, typically in the order of thousands. A cluster can be homogeneous in that all of its managed PMs are identical, or it could be heterogeneous in that it manages PMs with different resource capacities and capabilities.

Cloud providers offer these computing resources as a service for their clients and charge them based on their usage in a pay-as-you-go fashion. Cloud clients submit requests to the cloud provider, specifying the amount of resources that they need to perform certain tasks. Upon receiving a client request, the cloud provider scheduler creates a virtual machine (VM), allocates the requested resources to it, chooses one of the clusters to host the VM and assigns the VM to one of the PMs within that cluster. Client requests are thus also referred to as VM requests. After this allocation process takes place, the client can then use its allocated resources to perform its tasks. Throughout the VM lifetime, the cloud provider is expected, as well as committed, to guarantee and ensure a certain level of quality of service to the client. The allocated resources are released only once the client's task completes.

III. New Challenges, New Opportunities

Energy efficiency has been a hot topic even before the existence of the cloud paradigm where the focus was on saving energy in laptops and mobile devices in order to extend their battery lifetimes [5–7]. Many energy saving techniques that were initially designed for this purpose were also adopted by the cloud servers in order to save energy. Dynamic voltage and frequency scaling and power gating are examples of such techniques. What is different in cloud centers is that we now have a huge number of servers that need to be managed efficiently. What makes this further challenging is the fact that cloud centers need to support on-demand, dynamic resource provisioning, where clients can, at any time, submit VM requests with various amounts of resources. It is this dynamic provisioning nature of computing resources that makes the cloud computing concept a great one. Such a flexibility in resource provisioning gives rise, however, to several new challenges in resource management, task scheduling, and energy consumption, just to name a few. Furthermore, the fact that cloud providers are committed to provide and ensure a certain quality of service to their clients requires extreme prudence when applying energy saving techniques, as they may degrade the quality of the offered service, thereby possibly violating Service Level Agreements (SLAs) between the clients and the cloud provider.

The good news after mentioning these challenges is that the adoption of the virtualization technology by the cloud paradigm brings many new opportunities for saving energy that are not present in non-virtualized environments as we will see in this article. Furthermore, the fact that cloud clusters are distributed across different geographic locations creates other resource management capabilities that can result in further energy savings if exploited properly and effectively.

IV. Energy Conservation Techniques

We present in this section the most popular power management techniques for cloud centers by explaining the basic ideas behind these techniques, the challenges that these techniques face, and how these challenges can be addressed. We limit our focus on the techniques that manage entire cloud centers and that rely on the virtualization capabilities to do so rather than on those designed for saving energy in a single server, as the later techniques are very general and are not specific to cloud centers. Readers who are interested in power management techniques at the single server level may refer to [8] for more details.

Experiments conducted on real traces obtained from a Google cluster are also included in this section to further illustrate the discussed techniques. Some of these experiments are based on our prior work [9, 10] and others were conducted by us for the sake of supporting the explained techniques. As for the Google traces, they were publicly released in November 2011 and consists of traces collected from a cluster that contains more than 12 thousand PMs. The cluster is heterogeneous as the PMs have different resource capacities. The traces include all VM requests received by the cluster over a 29-day period. For each request, the traces include the amount of CPU and memory resources requested by the client, as well as a timestamp indicating the request's submission and release times. Since the size of the traces is huge, we limit our analysis to chunks of these traces. Further details on these traces can be found in [11].

The energy-efficient techniques for managing cloud centers that we discuss in this article are divided into the following categories: workload prediction, VM placement and workload consolidation, and resource overcommitment.

A. Workload Prediction

One main reason for why cloud center energy consumption is very high is because servers that are ON but idle do consume significant amounts of energy, even when they are doing nothing. In fact, according to a Google study [12], the power consumed by an idle server can be as high as 50% of the server's peak power. To save power, it is therefore important to switch servers to lower power states (such as sleep state) when they are not in use. However, a simple power management scheme that turns a PM to sleep once it becomes idle and switches a new PM ON whenever it is needed cannot be effective. This is due to the fact that switching a PM from a power state to another incurs high energy and delay overheads. As a result, the amount of energy consumed due to switching an idle PM back ON when needed can be much greater than the amount of energy saved by having the PM stay in an idle state (as opposed to keeping it ON) while not needed. This of course depends on the duration during which the PM is kept idle before it is switched back ON again. That is, if the PM will not be needed for a long time, then the energy to be saved by switching it off can be higher than that to be consumed to switch the PM back ON when needed.

In addition, clients will experience some undesired delay due to waiting for idle PMs to be turned ON before their requested resources can be allocated.

These above facts call for prediction techniques that can be used to estimate future cloud workloads so as to appropriately decide whether and when PMs need to be put to sleep and when they need to be awaken to accommodate new VM requests. However, predicting cloud workloads can be very challenging due to the diversity as well as the sporadic arrivals of client requests, each coming at a different time and requesting different amounts of various resources (CPU, memory, bandwidth etc.). The fact that there are infinite possibilities for the combinations of the requested amounts of resources associated with these requests requires classifying requests into multiple categories, based on their resource demands. For each category, a separate predictor is then needed to estimate the number of requests of that category, which allows to estimate the number of PMs that are to be needed. Using these predictions, efficient power management decisions can then be made, where an idle PM is switched to sleep only if it is predicted that it will not be needed for a period long enough to compensate the overhead to be incurring due to switching it back ON later when needed.

Classifying requests into multiple categories can be done via clustering techniques. Fig. 1 shows the categories obtained from applying clustering techniques on a chunk of observed traces from the Google data. These four categories capture the resource demands of all requests submitted to the Google cluster. Each point in Fig. 1 represents an observed request and the two dimensional coordinates correspond to the requested amounts of CPU and memory resources. Each request is mapped into one and only one category and different color/shape are used to differentiate the different categories. Category 1 represents VM requests with small amounts of CPU and small amounts of memory; Category 2 represents VM requests with medium amounts of CPU and small amounts of memory; Category 3 represents VM requests with large amounts of memory (and any amounts of requested CPU). Category 4 represents VM requests with large amounts of CPU (and any amounts of requested memory). The centers for these categories are marked by 'x'.

Once clustering is done, the number of requests to be received in each category is then predicted, and each predicted request is assumed to have demanded resource amounts equal to those corresponding to the category center it belongs to. Recall that these predictions can be a little off, leading to an under- or over-estimation of the number of requests. Under-estimating the number of future requests results in extra delays when allocating cloud resources to clients due to the need for waking up machines upon arrival of any unpredicted request(s). In order to reduce the occurrences of such cases, a safety margin can be added to the number of predicted requests to accommodate such variations. The cost of this safety margin is that some PMs will need to be kept idle even though they may or may not be needed. We propose to use a dynamic approach for selecting the appropriate

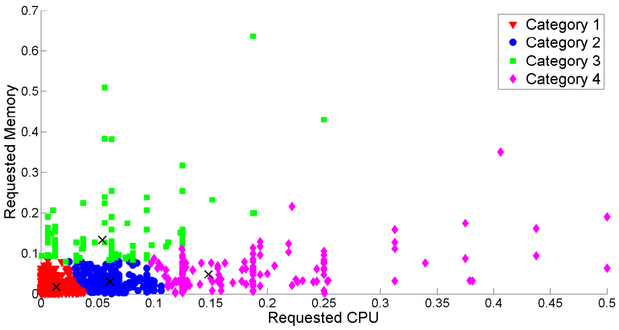

Fig. 1: The resulting four categories for Google traces.

safety margin value, where the value depends on the accuracy of predictors—it increases when the predictions deviate much from the actual number of requests and decreases otherwise. Fig. 2 shows both the actual and the predicted (with and without safety margin) requests of the third category that were received at the Google cluster. Observe that the predictions with safety margin form an envelope above the actual number of requests, and the more accurate the predictions are, the tighter the envelop is.

The question that arises now is how much energy can one save by applying such a prediction-based power management? To answer this question, we measure and plot in Fig. 3 the amount of energy saved when using the prediction-based technique when compared to the case when no power management is employed—no power management means that the cluster leaves all PMs ON as it does not know how many PMs will be needed in future. For the sake of comparison, we also plot in the same figure the amount of energy saved when optimal power management is employed, which corresponds to the case when predictors know the exact numbers of future VM requests, as well as the exact amounts of CPU and memory associated with each request (i.e., perfect prediction). This represents the best-case scenario and serves here as an upper bound. The figure shows that the prediction-based power management achieves great energy savings, and that the amount of saved energy is very close to the optimal one. The gap between the prediction-based power management and the optimal one is due to prediction errors and to the redundant PMs that are left ON as a safety margin.

It is worth mentioning that the energy savings of the prediction-based power management plotted in Fig. 3 vary depending on the workload demands. These savings are high under light workloads as many redundant PMs will be switched to sleep, thereby increasing the energy savings. And they decrease as the workload increases, since the higher the workload, the greater the number of PMs that are predicted to be kept ON, and hence, the lesser the energy savings the prediction-based approach makes over the no-power management approach; i.e., when compared to when all PMs are kept ON.

B. VM Placement and Workload Consolidation

Cloud centers are typically made up of multiple clusters distributed in different geographic locations. When a cloud provider receives a VM request, its scheduler has to first decide which cluster should host the submitted request. The geo-diversity of the clusters' locations can be exploited to reduce the center's electricity bills by assigning submitted requests to clusters with the cheapest power prices. Since electricity prices exhibit temporal variations, developing good power price prediction techniques is needed and can be very beneficial to make efficient cluster selections. Clusters' sources of energy can also be considered during this selection process where a request can be assigned to the cluster with the highest reliance on green sources of power in order to reduce carbon emissions.

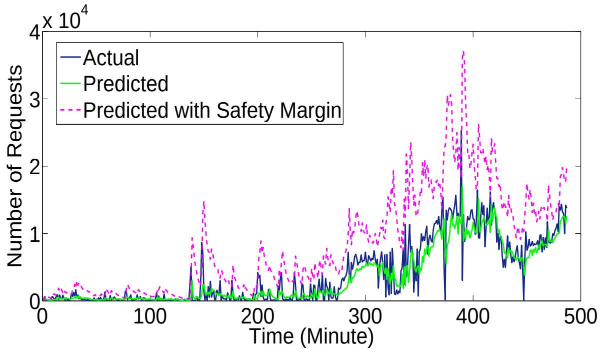

Fig. 2: Actual versus predicted number of requests for category 3.

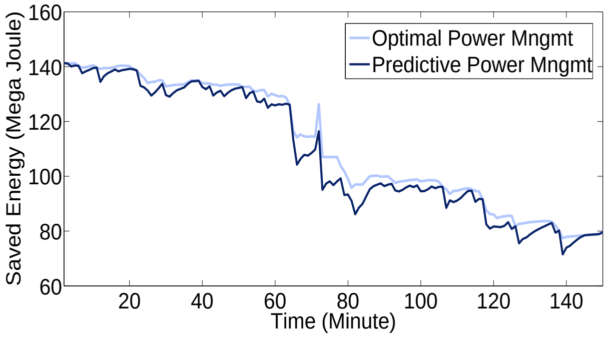

Fig. 3: Energy savings.

Once a cluster is selected, the next question that arises is which PM within the cluster should be chosen to host the submitted request? One of the great advantages that virtualization technology has is that it allows to place multiple requests, possibly coming from different clients, on the same PM. This enables to consolidate workloads on fewer servers, resulting in saving energy by turning to sleep as many servers as possible. The problem of deciding which PMs the submitted VM requests should be assigned to such that the number of ON PMs is minimized is referred to as the VM consolidation problem.

The VM consolidation problem is treated as the classical online Bin Packing (BP) optimization problem, which views VMs as objects and PMs as bins, and where the objective is to pack these objects in as few bins as possible. The objects (VMs) have different sizes (resource demands) and the bins (PMs) have different capacities (resource capacities). The problem here is an ’online’ problem because VM requests arrive in real time, and must be assigned to PMs as they arrive. The online BP problem is known to be NP-hard [13], and thus approximation heuristics, such as First Fit, Next Fit, and Best Fit, have been proposed instead to make VM-PM placement decisions. These heuristics tend to turn a new PM ON to host a submitted VM request only when the request cannot be fitted in any already ON PMs. However, they differ in how they select the PM (to host the submitted request) among the multiple ON PMs that fits the submitted request. The Best Fit heuristic, for example, chooses the ON PM with the least free space (least slack) that can fit the submitted request. The intuition here is that placing requests on the PM with the least slack results in leaving other ON PMs with large slack for supporting future requests with larger sizes.

Another technique, also pertaining to virtualization, that turns out to be very useful for VM consolidation is *VM Migration*, where already assigned VMs can be migrated (moved) from one PM to another. VM migration enables new VM-PM mappings, which can be used to concentrate the sparse workload (caused by the release of some VMs) on a smaller subset of PMs, thereby allowing the rest to be turned to sleep to save energy. One key problem with this technique is that VM migration incurs energy overhead [14], and thus should be performed only when the performance gains due to migration outweigh the overhead to be incurred when performing such a migration.

Rather than resorting to new VM-PM mappings to address workload sparsity, another potential solution would be to consider VMs' release times when making PM placement decisions. The idea here is to place VMs with similar release times together on the same PM, allowing PMs hosting VMs with short lifetimes to be turned to sleep quickly. Of course here it is assumed that the completion/release times of VM requests are known when VMs are submitted. This could be specified directly by the client or could be predicted based on the type of task the VM will be performing and/or based on the previous behavior of the client.

Fig. 4 shows the number of PMs needed to be ON to support all VM requests that were submitted to the Google cluster when different heuristics are used to make VM-PM placement decisions. The Random heuristic places each submitted request in any ON PM that can fit the request, whereas the Best Fit heuristic places the submitted request on the ON PM with the least slack. The Release-Time Aware Heuristic, on the other hand, accounts for the release time of VMs when deciding where to place VMs. The results in Fig. 4 clearly show that the PM selection strategy has a significant impact on the number of PMs in the cluster that need to be kept active/ON. The Random heuristic uses the largest number of PMs, as it encounters many cases where the submitted requests were too large to fit any already ON PMs, and thus forcing a new PM to be switched ON. Whereas by selecting the PM with the least slack, the BF heuristic is able to pack the workload more tightly, thus reducing the number of PMs that are needed to be active. Knowing the time at which the requests are to be released gives the Release-Time Aware heuristic an advantage by allowing it to place short-lived VMs on the same PMs so that they can be turned to sleep early to save energy. The corresponding energy costs associated with running the Google cluster throughout the 30-hour period are also reported in Fig. 5 where the costs are normalized with respect to the Random heuristic costs. The BF and the Release-Time Aware heuristics save respectively around 20% and 30% of the total costs when compared to the Random heuristic.

It is worth mentioning that the Release-Time Aware heuristic is an enhanced version of the Best Fit heuristic in which the time dimension is considered and the release time of VMs is taken into account in order to make more efficient placement decisions. When the release times of VMs are not known ahead of time, the Release-Time Aware heuristic drops the time dimension and thus behaves similarly to the Best Fit heuristic; i.e., it makes placement decisions similar to those made by the Best Fit heuristic.

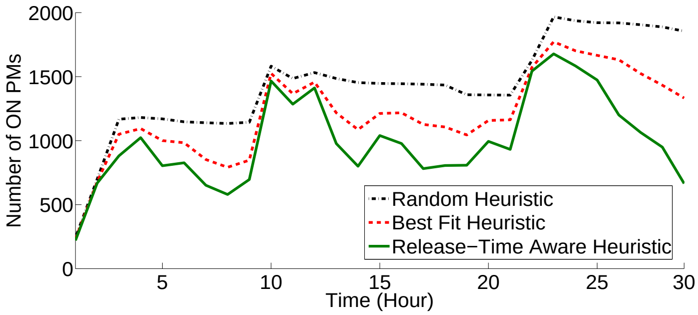

Fig. 4: Number of ON PMs over time when different heuristics are used to place the requests submitted to the Google cluster.

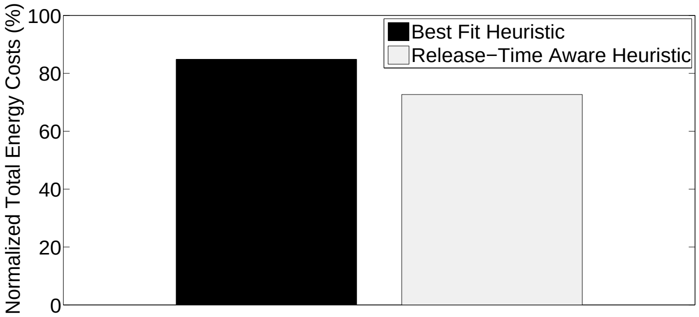

Fig. 5: Total energy overhead incurred from running the Google cluster under different placement heuristics (normalized w.r.t. the Random heuristic).

C. Resource Overcommitment

We have discussed so far the case where the cluster scheduler allocates, for each created VM, the exact amount of resources that is requested by the client, and locks these allocated resources for the VM during its entire lifetime (i.e., reserved resources are released only when the VM completes). A key question that arises now, which is the main motivation behind using the technique to be discussed in this section as a way of saving energy, is: what is the amount/percentage of these reserved resources that is actually being utilized? In order to answer this question, using real Google data, we measure and show in Fig. 6 the percentage of the utilized (CPU and memory) resources allocated by a Google cluster to its VM requests during one day. Observe that only about 35% of the requested CPU resources and 55% of the requested memory resources are actually utilized.

Our measurement study indicates that cloud resources tend to be overly reserved, leading to substantial CPU and memory resource wastage. In other words, many PMs are turned ON, but utilized only partially, which in turn translates into substantial energy consumption. Two reasons, among others, are behind such a resource over-reservation tendency:

1) Clients usually do not know the exact amount of resources their applications would need. Thus, they tend to reserve

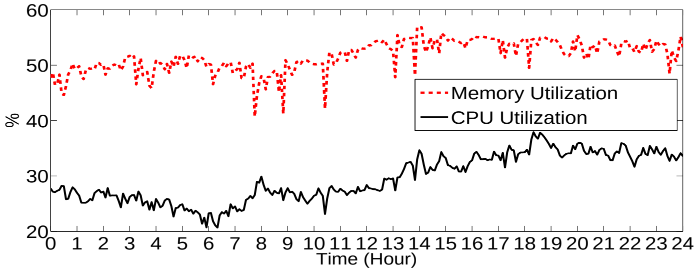

Fig. 6: One-day snapshot of aggregate utilization of the reserved CPU and memory resources of the Google cluster.

more resources than needed in order to guarantee safe execution of their applications.

2) The utilization of VM resources, by nature of some of the applications hosted on these VMs, may vary over time, and may rarely be equal to its peak. For example, a VM hosting a web server would possibly be utilizing its requested computing resources fully only during short periods of the day, while during the rest of the day, the reserved resources are way under-utilized.

Resource overcommitment [15] is a technique that has been adopted as a way for addressing the above-mentioned resource under-utilization issues. It essentially consists of allocating VM resources to PMs in excess of their actual capacities, expecting that these actual capacities will not be exceeded since VMs are not likely to utilize their reserved resources fully. Overcommitment has great potential for increasing overall PM resource utilization, resulting thus in making great energy savings as VMs are now hosted on a smaller number of PMs, which allows more PMs to be turned into lower power states.

One major problem that comes with overcommitment is PM overload, where an overload occurs when the aggregate amount of resources requested by the scheduled VMs exceeds the hosting PM capacity. When an overload occurs, some or all of the VMs running on the overloaded PM will witness performance degradation, and some VMs may even crash, possibly leading to the violation of SLAs between the cloud and its clients. The good news is that the virtualization technology allows to avoid, or to at least handle, these overloads. It does so by migrating VMs to under-utilized or idle PMs whenever a PM experiences or is about to experience an overload.

In essence, there are three key questions that need to be answered when it comes to developing resource overcommitment techniques that can be used to save energy in cloud centers:

i) What is the overcommitment level that the cloud should support? What is an acceptable resource overcommitment ratio, and how can such a ratio be determined?
ii) When should VM migrations be triggered to reduce/avoid the performance degradation consequences that may result from PM overloading?
iii) Which VMs should be migrated when VM migration decisions are made, and which PMs should these migrated VMs migrate to?

One potential approach that can be used to address the first question is prediction. That is, one can predict future resource utilizations of scheduled VMs, and use these predictions to determine the overcommitment level that the cloud should support. As for addressing the second question, one can also develop suitable prediction techniques that can be used to track and monitor PM loads to predict any possible overload incidents, and use these predictions to trigger VM migrations before overloads can actually occur. It can be triggered when for e.g. the aggregate predicted demands for the VMs hosted on a specific PM exceeds the PM's capacity.

The third question can be addressed by simply migrating as few VMs as possible, reducing then energy and delay overheads that can be incurred by migration. To avoid new PM overloads, one can for e.g. select the PM with the largest free slack to be the destination of the migrated VMs. These are just few simple ideas, but of course a more thorough investigation needs to be conducted in order to come up with techniques that can address these challenges effectively.

In summary, resource overcommitment has great potential for reducing cloud center energy consumption, but still requires the investigation and development of sophisticated resource management techniques that enable it to do so. Not much research has been done in this regard, and we are currently working on developing techniques that address these challenges.

It is also worth mentioning that overcommitment would not be possible without the capabilities brought by the virtualization technology, which enables real-time, dynamic/flexible allocation of resources to hosted VMs and eases the migration of VMs across PMs in order to avoid PM overloads.

V. Conclusion

We discussed in this article the key challenges and opportunities for saving energy in cloud data centers. In summary, great energy savings can be achieved by turning more servers into lower power states and by increasing the utilization of the already active ones. Three different, but complementary, approaches to achieve these savings were discussed in the article, which are: workload prediction, VM placement and workload consolidation, and resource overcommitment. The key challenges that these techniques face were also highlighted, and some potential ways that exploit virtualization to address these challenges were also described with the aim of making cloud data centers more energy efficient.

VI. Acknowledgment

This work was made possible by NPRP grant # NPRP 5-319-2-121 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] R. Brown et al., "Report to congress on server and data center energy efficiency: Public law 109-431," 2008.

[2] C. Pettey, "Gartner estimates ICT industry accounts for 2 percent of global co2 emissions," 2007.

[3] T. Taleb, "Toward carrier cloud: Potential, challenges, and solutions," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 80–91, 2014.

[4] T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks.," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.

[5] X. Ge, X. Huang, Y. Wang, M. Chen, Q. Li, T. Han, and C. Wang, "Energy-efficiency optimization for MIMO-OFDM mobile multimedia communication systems with QoS constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2127–2138, 2014.

[6] B. Hamdaoui, T. Alshammari, and M. Guizani, “Exploiting 4G mobile user cooperation for energy conservation: challenges and opportunities,” *IEEE Wireless Communications*, October 2013.

[7] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, M. Karimzadeh, and T. Magedanz, “EASE: EPC as a service to ease mobile core network,” *IEEE Network*, 2014.

[8] H. Hajj, W. El-Hajj, M. Dabbagh, and T.R. Arabi, “An algorithm-centric energy-aware design methodology,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2013.

[9] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, “Energy-efficient cloud resource management,” in *Proceedings of IEEE INFOCOM Workshop on Moblie Cloud Computing*, 2014.

[10] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, “Release-time aware VM placement,” in *Proceedings of IEEE GLOBECOM Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA)*, 2014.

[11] C. Reiss, J. Wilkes, and J. Hellerstein, “Google cluster-usage traces: format+ schema,” *Google Inc., White Paper*, 2011.

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

[13] E. Man Jr, M. Garey, and D. Johnson, “Approximation algorithms for bin packing: A survey,” *Jouranl of Approximation Algorithms for NP-Hard Problems*, pp. 46–93, 1996.

[14] H. Liu, C. Xu, H. Jin, J. Gong, and X. Liao, “Performance and energy modeling for live migration of virtual machines,” in *Proceedings of the Int’l Symp. on High performance Distributed Computing*, 2011.

[15] R. Ghosh and V. Naik, “Biting off safely more than you can chew: Predictive analytics for resource over-commit in IaaS cloud,” in *Proceedings of IEEE International Conference on Cloud Computing (CLOUD)*, 2012.

Mehiar Dabbagh received his B.S. degree from the University of Aleppo, Syria, in 2010 and the M.S. degree from the American University of Beirut (AUB), Lebanon, in 2012, all in Electrical Engineering and Computer Science. During his Master’s studies, he worked as a research assistant in Intel-KACST Middle East Energy Efficiency Research Center (MER) at the American University of Beirut (AUB), where he developed techniques for software energy profiling and software energy-awareness. Currently, he is a Ph.D. student in Electrical Engineering and Computer Science at Oregon State University (OSU), where his research focus is on how to make cloud centers more energy efficient. His research interests also include: Cloud Computing, Energy-Aware Computing, Networking, Security and Data Mining.

Bechir Hamdaoui (S’02-M’05-SM’12) is presently an Associate Professor in the School of EECS at Oregon State University. He received the Diploma of Graduate Engineer (1997) from the National School of Engineers at Tunis, Tunisia. He also received M.S. degrees in both ECE (2002) and CS (2004), and the Ph.D. degree in Computer Engineering (2005) all from the University of Wisconsin-Madison. His research focus is on distributed resource management and optimization, parallel computing, cooperative & cognitive networking, cloud computing, and Internet of Things. He has won the NSF CAREER Award (2009), and is presently an AE for IEEE Transactions on Wireless Communications (2013-present), and Wireless Communications and Computing Journal (2009-present). He also served as an AE for IEEE Transactions on Vehicular Technology (2009-2014) and for Journal of Computer Systems, Networks, and Communications (2007-2009). He served as the chair for the 2011 ACM MobiCom’s SRC program, and as the program chair/co-chair of several IEEE symposia and workshops (including ICC 2014, IWCMC 2009-2014, CTS 2012, PERCOM 2009). He also served on technical program committees of many IEEE/ACM conferences, including INFOCOM, ICC, GLOBECOM, and PIMRC. He is a Senior Member of IEEE, IEEE Computer Society, IEEE Communications Society, and IEEE Vehicular Technology Society.

Mohsen Guizani (S’85-M’89-SM’99-F’09) is currently a Professor and Associate Vice President of Graduate Studies at Qatar University, Qatar. Previously, he served as the Chair of the Computer Science Department at Western Michigan University from 2002 to 2006 and Chair of the Computer Science Department at the University of West Florida from 1999 to 2002. He also served in academic positions at the University of Missouri-Kansas City, University of Colorado-Boulder, Syracuse University and Kuwait University. He received his B.S. (with distinction) and M.S. degrees in Electrical Engineering; M.S. and Ph.D. degrees in Computer Engineering in 1984, 1986, 1987, and 1990, respectively, all from Syracuse University, Syracuse, New York. His research interests include Wireless Communications and Mobile Computing, Computer Networks, Mobile Cloud Computing and Smart Grid. He currently serves on the editorial boards of many International technical Journals and the Founder and EIC of "Wireless Communications and Mobile Computing" Journal published by John Wiley (http://www.interscience.wiley.com/ jpages/1530-8669/). He is also the Founder and General Chair of the International Conference of Wireless Communications, Networking and Mobile Computing (IWCMC). He is the author of nine books and more than 300 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE Journals and Magazines. He also served as member, Chair, and General Chair of a number of conferences. He was selected as the Best Teaching Assistant for two consecutive years at Syracuse University, 1988 and 1989. He was the Chair of the IEEE Communications Society Wireless Technical Committee and Chair of the TAOS Technical Committees. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005. Dr. Guizani is Fellow of IEEE, member of IEEE Communication Society, IEEE Computer Society, ASEE, and Senior Member of ACM.

Ammar Rayes Ph.D., is a Distinguished Engineer at Cisco Systems and the Founding President of The International Society of Service Innovation Professionals, www.issip.org. He is currently chairing Cisco Services Research Program. His research areas include: Smart Services, Internet of Everything (IoE), Machine-to-Machine, Smart Analytics and IP strategy. He has authored / co-authored over a hundred papers and patents on advances in communications-related technologies, including a book on Network Modeling and Simulation and another one on ATM switching and network design. He is an Editor-in-Chief for "Advances of Internet of Things" Journal and served as an Associate Editor of ACM "Transactions on Internet Technology" and on the Journal of Wireless Communications and Mobile Computing. He received his BS and MS Degrees in EE from the University of Illinois at Urbana and his Doctor of Science degree in EE from Washington University in St. Louis, Missouri, where he received the Outstanding Graduate Student Award in Telecommunications.