Fostering energy-awareness in scientific cloud users

Simon Ostermann*, Gabor Kecskemeti† and Radu Prodan*
*Institute of Computer Science, University of Innsbruck, Austria
†Institute of Computer Science and Control of the Hungarian Academy of Sciences, Hungary
Email: simon@dps.uibk.ac.at and kecskemeti.gabor@sztaki.mta.hu

***Abstract*—Academic cloud infrastructures are constructed and maintained so they minimally constrain their users. Since they are free and do not limit usage patterns, academics developed such behavior that jeopardizes fair and flexible resource provisioning. For efficiency, related work either explicitly limits user access to resources, or introduces automatic rationing techniques. Surprisingly, the root cause (i.e., the user behavior) is disregarded by these approaches. This paper compares academic cloud user behavior to its commercial equivalent. We deduce, that academics should behave like commercial cloud users to relieve resource provisioning. To encourage this behavior, we propose an architectural extension to academic infrastructure clouds. We evaluate our extension via a simulation using real life academic resource request traces. We show a potential resource usage reduction while maintaining the unlimited nature of academic clouds.**

Keywords*—*Scientific computing, distributed computing

I. Introduction

Academic computing infrastructures are built and maintained to support scientific users in their research endeavors. Introducing limitations on hardware usage in any ways would defeat the reason for the existence of these infrastructures. However, the more limitless a system is the more responsibility it requires from scientific users. For example, they must learn to eliminate their impact on other user's workings. Therefore, maintainers of such systems traditionally make the compromise of introducing such limitations for the users that stop unintentional obstructions on the work of others [1]. Meanwhile, for future systems, computer science tries to reduce the limitations and their impact on scientific users.

Infrastructure as a service (IaaS) cloud computing systems [2] are amongst the recent developments in this field. Pricing is an essential aspect of commercial IaaS systems [3] that academic solutions did not copy. Thus academic providers who apply such academic solutions will appear as offering unlimited resources for free to academic users. This promise is tempting for the users as it lifts one of their last remaining limitations. Unfortunately, this setting leads to an unprecedented demand of resources that is often latent (e.g., users maintaining demand for resources similarly to pilot jobs in grids [4]).

Academic providers have to fulfill these demands with the limited physical resources they are operating on [5]. To meet the demands with the infrastructure's real capabilities they usually apply two solutions: (i) access rationing, (ii) under provisioning (N to 1 mapping of virtual to physical resources). Both approaches were utilized in academic infrastructures even before the cloud era, but they both have serious downsides for academic uses. First, access rationing directly intrudes the freedom of researchers' access to the infrastructure [6]. Second, providers with under provisioning policies promise resources that are heavily shared amongst users [7], therefore these shared resources could vary in performance over time.

Instead of the past solutions, we propose to direct users to self-rationing. We derived the rationing problem from the missing pricing in academic clouds and argue that it is possible to construct academic systems featuring similar behavior to commercial clouds but still promise unlimited resources and unprecedented software configurability. We achieve this behavior with an architecture exposing energy efficiency metrics to users. First, our architecture provides the foundations for various leader boards where academics can compete with each other on how energy efficiently they use computing resources. Second, to ignite rivalry on leader boards, we suggest providers to allow the specification of energy related constraints on resource requests. Finally, we introduce the concept of engaging options (electronic representations of underused capacities) that allow users to attract others for particular resources.

The proposed architecture is built on three fundamental assumptions: (i) availability of energy readings, (ii) application of energy aware virtual machine placement, and (iii) leader board publicity. First, we require energy readings because we propose to publicize the accountable user consumption either directly or on a transformed way through leader boards. Second, users should be able to influence their leader board position, thus particular resource requests should have deterministic energy behavior. This behavior should be guaranteed by an energy efficient virtual machine placement policy. Finally, the expected effects of leader boards and engaging options are really dependent on their publicity, thus it is expected that they are soon adopted by a significant percentage of the community.

To test the feasibility of our architecture and its positive effects on the academic cloud communities, we have analyzed the behavior of typical academic users. We simulated the possible behavior of the academic users found in the Grid Workload Archives. Based on our simulations, we have concluded that there is a high chance of increasing energy efficiency and reducing resource demand on the provider side while still performing all user tasks. Our findings show, that the effect of our architecture could decrease the energy footprint of the provider's computing infrastructure by a factor of 2.6 at most.

II. Related work

Cloud computing is interesting for the scientific community from the beginning of the transformation of Amazon Web Services towards the Amazon Elastic Compute Cloud. Early evaluations investigated how scientists can benefit from this new technological infrastructure compared to Grids [8], especially with respect to storage [9] and computation [10] costs.

As analyzed in [11], maximizing revenue from infrastructure operations is an important objective of Cloud providers. However, this analysis focused on SLA relationships and did not consider possible energy savings that may further help in reducing the costs. Despite [12] has already shown that with a slight impact on overall performance providers can save power and increase their revenues. The work in [13] demonstrates that user awareness of power consumption also results in savings. Academic clouds are not adapting such power optimizations promptly, because the persons responsible for the resource usage are not responsible for the incurred electricity costs.

Instead academic clouds mostly apply rigid allocation techniques, such as accounting based limits that restricts many cloud and grid computing use cases and [6] reduce scientific productiveness. Contrary to credit or accounting based limitations, our approach motivates the users to optimize their resource usage through reduced power consumption achieved with our leader boards that increased user energy awareness.

[14] evaluated and quantified the impact of user feedback on power savings as 5% - 15%. To allow feedback on the power consumption, there is a need for a model that maps the physical power consumption to the virtual machines. Such models are provided in [15], [16] and show sufficient accuracy to be employed in our proposed architecture. Measuring the individual power consumption of VMs executed on a physical machine results in a higher power consumption per VM if the machine is under utilized and the static idle power is shared [17], [18]. We apply this knowledge in our approach so we can present academics valuable information on how they can improve their energy efficiency measures.

There is also an interest in saving power in Cloud infrastructures by turning off unused virtual machines to optimize resource use [19], [20]. This case is relevant for commercial providers, as users are not willing to pay for the idle times of their virtual machines. Optimized VM placement can save up to 55% of energy using the approach in [21]. Combining the savings possible through awareness of the energy consumption and optimized VM placement can significantly reduce the power consumption and further increase the productivity. Unfortunately, these techniques are only achieving these significant savings if rational users are utilizing the infrastructure, which is not the case with current academic users. Thus to fully utilize the effects of these techniques this paper aims at transforming academic users to behave more rational (i.e. like the price constrained commercial cloud users).

III. The architecture

A. Behavioral differences amongst cloud users

In response to the unprecedented latent demand from users, state of the art research focuses on changing the operation of academic cloud providers so they no longer appear to provide unlimited resources to their users. Unfortunately, this approach does not provide the cure for the root cause: the misbehavior of the users. This misbehavior is caused by the provider's promise about unlimited resources, and can be characterized as follows:

- Unnaturally and unnecessarily long infrastructure leases: E.g., academics often run their VMs even if they don't use their resources just to avoid the often long times they have to wait for the resource and its preparation for their particular need.
- Academic users also tend to prefer resources with the highest performance frequently disregarding other properties of the acquired resources (e.g., availability, energy efficiency, effect on other users).

In contrast, users of commercial clouds behave more rationally, since resource prices imposed by providers ensure the dissolution of uneconomical virtual infrastructures. These are the relevant characteristics of commercial users:

- They *delay the instantiation* of their virtual machines. For example, until these machines are an absolute necessity for the further progression of user tasks.
- They *ensure continuous use* of acquired VMs (completely utilizing resources while they lease the VM).
- They *terminate VMs early* on (considering the billing periods – e.g., on Amazon there is no use to terminate a VM before one hour). Thus they immediately terminate a VM that has no further tasks or it is not expected to have a task for it in the foreseeable future.
- They make a *compromise between price and performance* and allow increased task makespans. For example, a smaller priced instance could still be capable to perform the necessary tasks within the billing period of the provider, thus if the tasks are not time critical, they could take a little longer. Or they would postpone the execution of a new task until an existing VM could run them (assuming that serializing the tasks is more cost effective than having a new VM for the new task).

Commercial providers (like Amazon) establish such user characteristics through the following three pricing models: (i) *Standard* where the resource usage is paid with a constant hourly price, (ii) *Reserved* where a annual upfront payment reduces the hourly prices and (iii) *Spot* where the compromise of a possible VM abortion and higher instantiation time potentially lowers the prices. In Table I, we show how the aforementioned user characteristics are achieved by these pricing models. We also reveal rational user actions one can observe under a particular pricing model. Fortunately, all these user actions are possible in current academic clouds (i.e., these actions can be accomplished through the usually available IaaS interfaces). Therefore, if academics would have the incentive for these actions then they would bear similar characteristics as commercial users. And commercial like user characteristics would allow academic clouds to maintain sufficient balance between their users and resources and result in power aware users.

B. IaaS extensions to support behavioral change

To encourage such user behavior, we propose to motivate the academics through presenting them the energy impact of their operations. Our architectural extensions to academic cloud environments not only collect and present the energy consumption data to academic users but also provide them information on how to increase their efficiency. Figure 1 presents these extensions to an existing IaaS software stack (shown in the bottom right corner) and shows their relations to academic users. As the extensions are aimed at academics,

TABLE I. THE EFFECTS OF THE VARIOUS PRICING MODELS IN COMMERCIAL CLOUDS

User characteristics	Pricing models offered by commercial providers		
	Standard	Reserved	Spot
Delay instantiation	Need triggered	Limit tasks to reserved VMs	Price & need triggered
Ensure continuous use	Use the VM regularly	Prolonged use is desired	Burst VM use until abortion
Early VM termination	Terminate if not needed	If unjustified, sell reservation and switch model	Terminate if not needed
Performance compromise	Instantiate a smaller VM	Bound to a resource type	Abrupt VM abortion

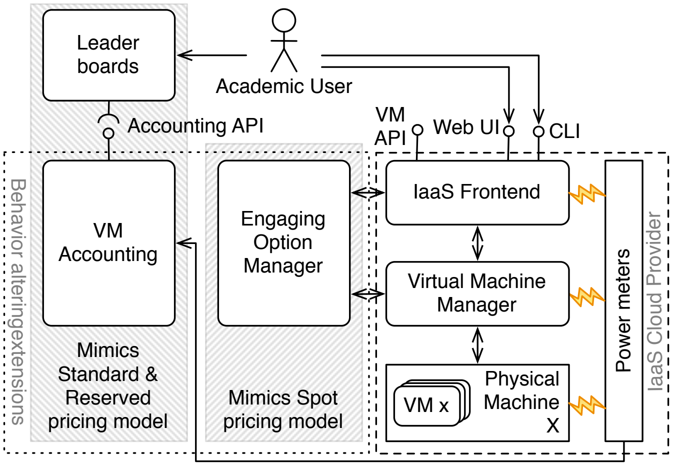

Fig. 1. Overall view of the proposed architecture

other actors like system administrators are not shown. In fact, these extensions do not even change the ways the users would interact with the existing cloud environment, they just add an optional functionality that – if used widely – could lead users towards self-rationing. So the users still utilize their usual command line interfaces or web portals, but by having access to our extensions they are expected to change their behavior towards these interfaces. These extensions are built on two cornerstones: (i) *leader boards* (for disseminating and comparing how energy efficient are the users) and (ii) *engaging options* (to ensure that VM requests are arriving in groups more suitable for those physical machines that will host the VMs). The following paragraphs discuss how these cornerstones are related to user behavior and show their basic properties.

First, similarly to the account statements of commercial providers, with our extensions, academic cloud providers collect the energy consumption of their users. Our architecture ensures that they publish this data through an *Accounting API*. Through this API, trusted third parties are allowed to query the energy consumption accounted to particular virtual machines (e.g., a user can query the consumption of its own VMs). Leader boards are special users of this API who collect data regarding every registered user. They aggregate energy consumption data for users. Depending on the intended user base, each leader board can use their own method for aggregation. At the end, to each user they assign a score that is comparable with other scores in the same leader board. Every board presents a ranking list for its users (the more energy efficient a user is the higher his/her ranking is). This list is our major motivational instrument as users often compare themselves to their peers and try to improve their ranking.

Next to the list, leader boards present users with techniques that could increase their score. Recommended techniques reveal how to accomplish similar behavior to the users of commercial clouds. Table I also acts as a summary for the recommended techniques. So, for example, academics are expected to terminate their unused virtual machines. Unfortunately, there are some user behavior (e.g., price and need triggered VM instantiation or the use of dedicated VMs for repetitive and prolonged tasks) that the accounting API and the leader boards cannot impose on academics.

Motivation can be further increased with the formation of user groups within a leader board [22]. In general, groups allow their members to compete. If groups also receive an overall score based on their member's scores, then group members are encouraged to pursue higher scores together by enabling the competition of groups based on overall group scores. For higher overall scores, enthusiastic users will try to convince more resistant users to revise their resource usage patterns. As group interaction is essential to strengthen the overall motivation in the system, our leader boards automatically form groups based on user affiliation and interest (e.g., groups are created for departments or computer scientists in general). These groups build up the initial momentum towards widespread user behavioral changes, but users are not bound to them. New groups can be formed by enthusiasts also.

To guide users towards these behavioral patterns we introduce the concept of engaging options. These options are such electronic documents that represent inefficiencies in the system. If a user is assigned to a physical machine that's power efficiency could be increased then he receives some engaging options. The received options then can be exchanged with other users. Amongst the techniques listed on the leader boards, users will be noted that if they would share/wait for such options, then they could increase their scores. Thus users will act similarly as those in commercial clouds: they will wait for a suitable engaging option to appear before instantiating their VMs (just like commercial users would behave for spot pricing). Thus, our architecture extends current IaaS systems with the management of the entire life-cycle of engaging options (from their issuing to their dissolution). Figure 1 refers to this extension as the *Engaging Option Manager*.

To make sure of the highest scores possible on the leader boards, users with immediate computing needs would acquire engaging options (representing highly energy efficient resources) either from their fellow academics or from marketplaces. When requesting a virtual machine from the provider, users would also present the acquired engaging options. If multiple engaging options are used for a single virtual machine request, then users must ensure that the presented options have identical identifier for the hosting machine (otherwise their VM request cannot be fulfilled). At the provider, this will result in a VM request composed from all the resources represented by the specified engaging options. In its next step, the provider will create the new user's virtual machine on the host referred by the engaging options presented in the VM request. Because of the placement of this new virtual machine, the energy consumption accounted to the other VMs will be reduced in their remaining lifetime (thus engaging options positively impact the scores of those who shared them).

IV. Evaluation

In this section, we reveal our analyze of how scientific infrastructures are impacted while academic user behavior is transformed to a more commercial like behavior. We briefly turn our attention on the infrastructure's general properties, then we discuss the simulated user behavior on the system.

In all of our evaluations, we used the infrastructure cloud simulator called DISSECT-CF [23]. This simulator can deal with the internals of the infrastructure clouds on levels required to deliver accurate energy consumption readings under all evaluated user behaviors. In all of our simulations, we have set up a cloud infrastructure consisting of such amount and kind of physical machines that were capable to host even the maximum number of VMs that could occur in parallel under a particular workload. We ensured that the simulation only requested VMs of a single type with resource requirements equalling to the 1/8th of all available resources (CPU cores, RAM, disk, network) of a single physical machine. We applied the simulator's simplest first fit VM scheduler that placed new VMs on the first physical machine with available resources. If there are no physical machines with free capacities then a new physical machine is switched on and the new VM is placed on that physical machine. If a physical machine does not host a single VM then it is switched off. The applied physical machine scheduler also ensured that amongst the switched on physical machines only one is not fully utilized. When this is not the case, the scheduler automatically migrated the VMs from less utilized to more utilized physical machines to minimize the count of necessary physical machines switched on. With this simulated infrastructure configuration, we ensure that the momentary resource utilization will not experience large variance because of the internals of the simulated IaaS system. The elimination of this variance is important because otherwise it would be impossible to tell whether resource utilization reductions are caused by internal IaaS operations or by the user behavioral changes we intended to evaluate.

To simulate user behavior, we have turned our attention towards workload traces offered by the scientific community. We have looked for traces that fulfill the following three criteria: (i) represent extensive durations because the usage patterns for particular users become more clear in the long term, (ii) focus on a large user base so we will have a chance to see the behavior of a multitude of users and (iii) offer a description which clearly associates activities and users. After employing these criteria, we have selected the grid workload archive (GWA – [24]) as a good representative for the behavior of our expected users (unfortunately our last criteria eliminates the possibility of using some of the well known traces like PlanetLab). Although, this archive is not cloud oriented, we assumed that the frequency of the user tasks and the user's overall behavior regarding the use of the infrastructure is independent from the underlying technologies. In the following subsections we first present how we processed the scientific workload in GWA for our analysis. Then we present an in depth overview on the expected impacts of our architecture on the resource utilization of academic clouds.

A. *Trace processing methods*

Before our analysis, we have processed the traces so we can simulate both commercial and academic user characteristics (in terms on how and when they would instantiate virtual machines for their tasks). Throughout the simulation, we have focused our attention to three of the main user characteristics listed in Table I: (i) delay instantiation, (ii) early VM termination and (iii) ensure continuous use. As the exact resource utilization pattern of user tasks were not available, our simulations have had a single VM type usable, thus the analyze of the performance compromise characteristic was out of scope.

Our first trace processing step analyzed the tasks and identified those that have the highest chance to be delayed and grouped together in a virtual machine (so that the tasks can be grouped like a commercial user would group them). We assumed that no user would delay tasks for indefinite time. Thus, after some time, even the most scoring conscious users would make a compromise on resources for their long delayed tasks. The more commercial like users are more willing to delay tasks hoping for more energy efficient (thus better scoring) resource openings. To identify those tasks that together could lead to the minimum amount of VM instantiations, we have processed the traces with a data mining algorithm called K-Means. Based on submission time proximity, the algorithm clusters such tasks together that are the least likely to cause impatience (we assumed that a user becomes impatient when he would have to wait for more than two hours for a more energy efficient resource). In the five selected traces, we have found over 28000 clusters. From these clusters, we have selected those that could statistically represent an entire trace and we only kept those for further analysis (the properties of these clusters are revealed in parentheses). These clusters contained over 1000 tasks (with a median of 1719) and they also contained references to users between 1-98 (with a median of 25 users). The rest of the paper focuses our analysis on these clusters only.

As the next processing step, for each cluster we have run a user simulator which assigned the user's tasks in the cluster to virtual machines (ensuring that users can only assign tasks to their own VMs). Based on the commercial user characteristics, we have identified three assignment approaches: (i) act as a non-transformed academic user, (ii) employ early VM termination if possible, (iii) delay tasks until they would form a continuous block and create a VM for them. In the next paragraphs, we detail these three options.

With the first approach, we simulated the academic users before their transformation: we assumed users would create a VM as soon as there is a task to be executed and the user does not have a VM that could execute it immediately. This could lead to several VMs in parallel depending on the number of tasks that must be executed in a particular instance of time. When a task finishes in a particular VM, the user could choose to terminate the VM. This is only done when the remaining tasks of the cluster have a parallelism less than the amount of virtual machines the user currently has. For example, when the trace has three simultaneous tasks for a while, the user would have three VMs, but immediately when there is no more

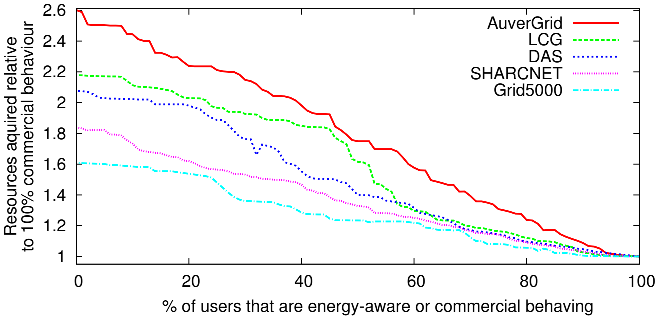

Fig. 2. Transforming randomly chosen users to commercial like behavior

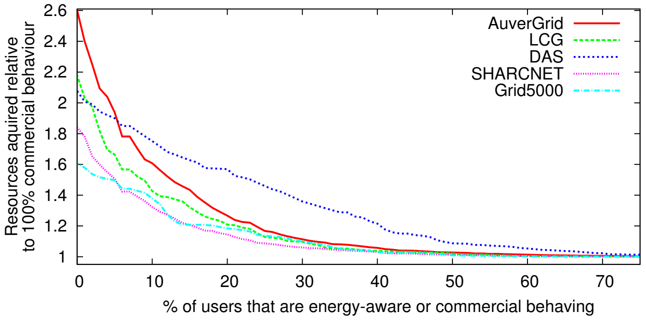

Fig. 3. Prioritizing transformation of the heaviest users

chance to have three simultaneous tasks within the cluster, the user would terminate one of his/her VMs.

The second approach partially simulates commercial like behavior: the users terminate their VMs as soon as feasible. The simulation of this approach only differs marginally from the previous solution. We set up an initial set of virtual machines utilizing that approach. Then we split these VMs by repeating the following procedure until there are no new VMs formed: (i) we identify the longest gap between the tasks of the original VM, (ii) if the length of this gap is over a threshold then we remove the original VM from the set of necessary VMs and add two new ones (one that includes all tasks before the gap, one for the rest). We define this threshold as the shortest time period for which two new VMs would render bigger overall score for the user than the original one.

Then, our last approach provides our most sophisticated approximation of commercial like behavior. Here we assume users postpone their tasks so they form a continuous block (one task promptly starts after the previous has finished). To ensure that we can postpone tasks indefinitely within the clusters, we assume no dependencies between user tasks (this information is not included in the available traces). Thus we first delay the tasks as much as possible, then we apply the second approach to define the shortest lifetime VMs. This combination of task delays and early VM termination will result in VMs that are in continuous use for their entire lifetime.

B. *Analysis*

To prove our architecture's effectiveness, we have set up a simulation environment where every selected cluster can be evaluated. A single experiment was done as follows: first we allowed the specification of user behavior (e.g., non-transformed academic, commercial like), then based on this behavior we determined the needed VMs and their properties (request and termination time), next we run the virtual machine placement algorithm for these VMs in our simulated cloud infrastructure, finally we have collected the energy readings and the total CPU hours of each VM in the simulation.

In our first experiments, we assumed uniform likeliness that our architecture transforms a selected user. Thus we have specified a random order of users in which they are transformed to power-aware. At first, we have set all users to behave like commercial ones. We run an initial experiment and collected the total CPU hours spent in the system. This formed our baseline. During our first experiments, we have collected the total CPU hour figures from the experimental runs and divided by the previously acquired baseline. Figure 2 shows the possible extra resource demands (compared to our baseline) for the five analyzed traces depending on the percentage of the users who behave as non-transformed academics. Although we have evaluated the trends for all selected clusters, we presented them as an average per analyzed trace. Based on the results, we see that if only non-transformed users are using the system, then the energy and resource consumption is at its maximum. AuverGrid is impacted the most by user behavior. About 2.6 times more resources are used if all users apply their past bad practices. The figure also shows, that significantly (23%–33%) smaller infrastructure can serve the users even if just half of them convert to a more commercial user like behavior.

When we looked at the non averaged trends, it clearly showed that some users have heavy impact on the resource utilization in the system. We evaluated a scenario where heavier users are the first to adopt power-aware behavior. We expect that targeted leader board advertisements and the hall of shame will increase the likeliness of such a scenario. With this new adoption order, we have re-evaluated our first experiments and presented the new results in Figure 3. The figure not only highlights the impact of heavy users, but also reveals that converting just half of the users result in an even more dramatic decrease in the required infrastructure size as before. The reductions are now between 37% and 61%. As per the figure, only a few users are responsible for the vast majority of the extra energy consumption. Thus providers should put extra effort advertising the leader boards to them. Also, setting up a hall of shame – an inverse ranking leader board – might be beneficial to prominently highlight those users who misbehave.

Our evaluation has shown the possible transformations of academic user towards more commercial like behavior and their impact on the overall energy and resource usage. In real life systems, we expect the introduction of the leader boards will immediately turn some users towards the behavior of commercial users. Thus even though our experiments start from 100% non-transformed users, it is only a hypothetical case. The saving curve that can be expected in realistic scenarios is expected to be close to the one shown in Figure 2.

V. Conclusion & Future Work

Regrettably, even if an academic cloud provider applies similar resource management approaches as commercial providers, their resource needs and energy footprint hardly decreases because of the bad practices the academic users developed in the unconstrained environments of the past. Resources are occupied longer than needed as the limiting factor from commercial cloud providers is missing: cost. We have proposed an architecture that helps to raise the resource and power usage awareness of academic cloud users by two technologies: (i) leader boards which provide direct feedback on how closely academics behave compared to commercial users and (ii) engaging options that direct users towards more economical resource usage.

Our evaluation shows potentially high savings both in terms of utilized resources and energy if users are willing to change their usage patterns. Our architecture tries to increase their willingness to change by allowing academics to compete with fellow scientists. This competition encourages academics to self-ration their resource usage for better scores on the leader boards. As a result of self rationing, academic providers will have a chance to utilize state of the art resource management approaches to reduce energy expenses just like commercial providers. Such resource provisioning algorithms can have high influence on power draw of infrastructure and in combination with energy aware users this can lead to considerable power savings. Also, self-rationing practices often increase the throughput of the infrastructure, thus academics can use the infrastructure more productively than before the application of our architectural extensions.

We plan to continue our work first by evaluating different scoring schemes in detail and to analyze their impact on the overall resource and energy utilization. We will investigate high level leader board organizational issues like proposing new schemes to reduce possible imbalances caused by leader boards between individual providers in a cloud federation. Next, we will evaluate the possible positive effects of engaging option brokers and marketplaces on user behavior and we will investigate new behavioral patterns made possible by advanced engaging option handling.

Acknowledgment

This work has been performed in the projects TRP 237-N23 funded by the Austrian Science Fund (FWF) and eRamp, co-funded by the grant 843738 from the Austrian Research Promotion Agency (FFG) and the ENIAC Joint Undertaking.

References

[1] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling," in *Proceedings of the 5th European conference on Computer systems*, ser. EuroSys '10. New York, NY, USA: ACM, 2010, pp. 265–278.

[2] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service (iaas)," *International Journal of engineering and information Technology*, vol. 2, no. 1, pp. 60–63, 2010.

[3] D. Durkee, "Why cloud computing will never be free," *Queue*, vol. 8, no. 4, pp. 20:20–20:29, Apr. 2010.

[4] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein, "The pilot way to grid resources using glideinWMS," in *WRI World Congress on Computer Science and Information Engineering*, vol. 2, 2009, pp. 428–432.

[5] G. Kecskemeti, S. Ostermann, and R. Prodan, "An architecture to stimulate behavioral development of academic cloud users," *Sustainable Computing: Informatics and Systems*, p. 15, 2014, *to appear*. [Online]. Available: http://dx.doi.org/10.1016/j.suscom.2014.04.002

[6] P.-O. Östberg, D. Espling, and E. Elmroth, "Decentralized scalable fairshare scheduling," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 130–143, 2013.

[7] S. Zhuravlev, S. Blagodurov, and A. Fedorova, "Addressing shared resource contention in multicore processors via scheduling," *SIGPLAN Not.*, vol. 45, no. 3, pp. 129–142, Mar. 2010.

[8] I. T. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, 2009, pp. 1–10.

[9] E. Deelman, G. Singh, M. Livny, G. B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, ser. SC '08. Piscataway, NJ, USA: IEEE Press, 2008, pp. 50:1–50:12.

[10] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, "Cost-benefit analysis of cloud computing versus desktop grids," in *IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, 2009, pp. 1–12.

[11] G. Feng, S. Garg, R. Buyya, and W. Li, "Revenue maximization using adaptive resource provisioning in cloud computing environments," in *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*, ser. GRID '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 192–200.

[12] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing cloud providers revenues via energy aware allocation policies," in *IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010, pp. 131–138.

[13] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?" *Energy Efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[14] S. Darby, "The effectiveness of feedback on energy consumption. a review for defra of the literature on metering, billing and direct displays." Working paper, Oxford Environmental Change Institute, p. 24 pp, April 2006.

[15] W. Chengjian, L. Xiang, Y. Yang, F. Ni, and Y. Mu, "System power model and virtual machine power metering for cloud computing pricing," in *Third International Conference on Intelligent System Design and Engineering Applications (ISDEA)*, 2013, pp. 1379–1382.

[16] P. Xiao, Z. Hu, D. Liu, G. Yan, and X. Qu, "Virtual machine power measuring technique with bounded error in cloud environments," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 818 – 828, 2013.

[17] Y. Jin, Y. Wen, and Q. Chen, "Energy efficiency and server virtualization in data centers: An empirical investigation," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2012, pp. 133–138.

[18] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing*, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 39–50.

[19] D. Huang, D. Yang, H. Zhang, and L. Wu, "Energy-aware virtual machine placement in data centers," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012, pp. 3243–3249.

[20] S. Takahashi, A. Takefusa, M. Shigeno, H. Nakada, T. Kudoh, and A. Yoshise, "Virtual machine packing algorithms for lower power consumption," in *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2012, pp. 161–168.

[21] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-efficient virtual machine provision algorithms for cloud systems," in *Fourth IEEE International Conference on Utility and Cloud Computing (UCC)*, 2011, pp. 81–88.

[22] M. A. Hogg and D. Abrams, *Group motivation: Social psychological perspectives.* Hertfordshire, HP2 7EZ, England: Harvester Wheatsheaf, 1993, ch. Towards a single-process uncertainty-reduction model of social motivation in groups., pp. 173–190.

[23] G. Kecskemeti, "DISSECT-CF: DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations," https://github.com/kecskemeti/dissect-cf, August 2014.

[24] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. J. Epema, "The grid workloads archive," *Future Generation Computer Systems*, vol. 24, no. 7, pp. 672–686, 2008.