Energy consumption of scheduling policies for HTC jobs in the Cloud

Osama Nasser Alrajeh
School of Computing Science
Newcastle University, UK
o.alrajeh1@ncl.ac.uk

Nigel Thomas
School of Computing Science
Newcastle University, UK
nigel.thomas@ncl.ac.uk

ABSTRACT
Organisations often exploit idle time on existing local computing infrastructure through High Throughput Computing (HTC) to perform computation. More recently the same approach has been employed to make use of cloud resources in large-scale computation. To date, the impact of HTC scheduling policies within a cloud environment have received limited attention in the literature. The key focus of this paper is to extend an existing cloud simulation system to incorporate energy measurement, and evaluate the energy and performance impact of existing policies for scheduling HTC jobs to cloud resources. We demonstrate through trace-driven simulation the trade-off between energy consumption and system performance for a number of HTC scheduling policies.

Categories and Subject Descriptors
H.3.4 [**Information Systems**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*; I.6.8 [**Simulation and Modelling**]: Types of Simulation—*discrete event*

Keywords
Cloud Computing, Green Computing, Efficient Energy, Datacentre, HTCondor

1. INTRODUCTION
High Throughput Computing (HTC) systems are a popular choices in organisations, frequently employed to leverage existing, idle, infrastructure to perform computation in an *'cycle stealing'* fashion. Latterly, these same approaches are increasingly used to meet peak requirements for computation by leveraging the scale offered by Cloud Computing [1] which could not otherwise be satisfied using local infrastructure alone.

Meanwhile, global energy demand has increased from 10 thousand terawatt hours in 1990 to 20 thousand terawatt hours today. By 2040, global demand is expected to approach 40 thousand terawatt hours [12]. Power saving initiatives are motivated by a number of factors, including governmental pressure and the need to save money. For example, the British government aims to lower CO2 emissions by 80% in the next 35 years [16].

Thus, it is very important that data centres improve the efficiency with which they use their power and a great deal of innovation has taken place to this end. For instance, cloud computing takes advantage of allocation and migration policies in order to promote the use of physical resources, and can change the amount of computing power allocated to users in line with their needs.

Existing research has considered the use of the Cloud in this context to meet exceptional capacity which cannot be met on local infrastructures [17, 13, 5], and also for running full workloads [6]. However, this is often without consideration for the energy consumption incurred by the cloud provider.

In this paper, we build on previous work to evaluate the energy and performance impact of scheduling policies for HTC workloads to cloud resources. We employ an existing trace-driven simulation provided [14], extending it to model energy consumption such that we may evaluate the energy impact of policies governing the scheduling of HTC jobs to cloud instances. Then, we compare the energy consumption of the same workload on servers whose performance and energy consumption characteristics differ.

We evaluate policies using trace data obtained from a University HTCondor high throughput cluster. HTCondor [10] is an open source high throughput computing software which provides workload management system technology for grid computing jobs. HTCondor has a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. HTCondor chooses where and when to start running the jobs that are submitted by users dependent upon its policy.

The rest of the paper is structured as follows. Section 2 introduces related work. In Section 3 we present the policies we evaluate through this work. Section 4 discusses our approach to adapting an existing simulation environment for calculating energy consumption. We present and evaluate the results of our preliminary experimentation in Section 5, before concluding and discussing future work in Section 6.

SIMUTOOLS 2015, August 24-26, Athens, Greece

DOI 10.4108/eai.24-8-2015.2261103

2. RELATED WORK

Deelman *et al.* [6] perform an evaluation of the cost of using the Cloud for the execution of a single scientific application using Amazon's Elastic Compute Cloud (EC2) and Amazon's Simple Storage Service (S3). Our work differs as we seek to service the workloads of multiple users comprising heterogeneous applications.

De Alfonso *et al.* [4] deployed a virtual cluster in cloud environment and compared the cost of physical cluster and virtual cluster. The High Performance Computing (HPC) was deployed as virtual cluster on Amazon EC2. They construed the energy consumption in their comparison by applying different energy policies into the physical cluster. They concluded that in some cases virtual cluster could be chapter than physical cluster.

Beloglazov *et al.* [2] evaluate various energy-efficient resource allocation policies and scheduling algorithms in the cloud computing systems. They consider the evaluation of power consumption and Quality of Service (QoS) impacts, though they do not consider HTC workloads in their evaluation.

Duy *et al.* [7] assess the performance of a neural networks predictor for enhancing server power utilisation in the Cloud. They utilise predictors of future load demand predicated on historical demand. The outcomes demonstrated that 46.7% of energy consumption could be reduced.

Kliazovich *et al.* [9] explore the energy-aware cloud computing datacentres through simulation, modeling the energy consumption of all datacentre components including servers, switches, and links. The simulation results showed that 66% of server energy consumption was by idle servers, and networking accounted for 30% of total energy consumption.

Liu *et al.* [11] studied the power efficiency of intensively managing data grids on distributed systems. They created a power-efficient management simulator called Distributed Energy-Efficient Scheduler (DEES) that provides energy savings by incorporating scheduling jobs with data assignment approaches. Information duplication and job transferals are lowered, reducing the power drain.

CycleServer and CycleCloud [1] are software tools provided by Cycle Computing, and are used for managing and accessing cloud resources. CycleServer manages and monitors the jobs in HTCondor, while CycleCloud provides access to the resources though web services such as Amazon EC2. These tools have been used at large-scale to provide a huge number of instances from a public cloud provider, which can reduce the time of running and the cost.

SPECpower_ssj2008 [15] is an industry-standard benchmark for server performance and measured power expenditure. This metric signifies the combination of the performance measured at individual stages or load levels (in ssj_ops) divided by the total of the average power (in ssj_ops/watts). The benchmark itself instills progressive stages of load on a given machine friom active-idle (0%) to peak (100%) load at 10% progressive load levels.

3. POLICY

McGough *et al.* [14] have previously proposed policies governing the scheduling of HTC jobs to Cloud instances, aiming to reduce cost and overheads. In this paper we seek to quantify the energy impact of these policies on the Cloud provider.

P1: limiting the maximum number of Cloud instances: This policy seeks to limit the potential cost on the cloud by imposing a maximum number of instances which can be active at a given time. If this limit has been reached, any arriving jobs will be queued until a resource becomes available. This policy seeks to reduce cloud cost and cloud idle time, at the expense of increased overheads for jobs submitted during busy periods.

P2: merging of different users' jobs: For reasons of security, different users of the cloud commonly wish to maintain separation from other users' workloads. In Policy P2 we relax this assumption and allow HTC jobs from multiple users to execute on the same running instance. This should reduce overheads, cost and energy consumption.

P3: instance keep-alive: This policy enables an idle instance at the end of the billing stage to stay 'hired' for the next stage with probability *f(p)*, allowing it to serve the latest next in line jobs more efficiently. It is problematic to predict the arrival of subsequent jobs; consequently there are three alternative policies that can determine if an instance needs to stay within the alive and define *f(p)* for each:

Fixed: instances will be kept alive with probability *f(p)=p*.

Idle: instances will be kept alive with a probability proportional to the number of currently idle instances: $f(p) = (\frac{1-i}{t})p$, where i is the number of idle cloud instances at decision time and t is the total number of cloud instance at this time.

Load: instances will be kept alive with a probability proportional to the current load on the system:

$$f(p) = \frac{\int_{t-T}^{t} u_i d_i}{\int_{t-T}^{t} a_i d_i} p \qquad (1)$$

where t is the decision time, T is total simulation, u_i the quantity of active Cloud instances at time i, and a_i is the number of cloud instances active at time i.

Such a policy could have additional ramifications on overheads, and as such a commencing job would have an increased chance of arriving to an idle instance. The number of instance hours consumed is expected to increase due to some instances running in the absence of jobs.

P4: delaying the start of instances: seeks to minimise idle time arising as a consequence of short running jobs using only partial hours on cloud instances. Under this policy, jobs which arrive to the system which cannot be allocated to an idle instance are queued. If the job does not receive an instance within t minutes of submission, a new instance is then created. The policy should incur overheads but reduce cloud cost and energy consumption.

4. SIMULATION ENVIRONMENT

In this paper we use a Java based trace-driven simulation to evaluate the performance of job management policies of HTC jobs to Cloud instances. We extend the existing simulation environment to evaluate the energy consumption of these policies. Section 4.1 discusses changes to the cloud resource model and Section 4.2 discuss metrics considered in the remainder of the report, while Section 4.3 describes the scenario we simulate in the remainder of this paper.

4.1 Resource Model

We extend the resource model of the existing simulation framework to include energy consumption and performance figures, as described below.

4.1.1 Energy consumption:

We select readily available metrics from the SPECpower benchmark to obtain energy consumption values for servers in idle, booting and working modes. These may be then multiplied by the amount of time spent in each mode in order to calculate total energy consumption of the system under a given policy. Here we assumed the load level for booting and working jobs is 100%.

4.1.2 Performance scaling:

Based on performance measures provided by the SPECpower benchmark for each server, we are able to scale the execution time of our workload to more closely model the performance observable if the workload were run on that hardware. In doing so we assume the SPECpower benchmark to be representative of the original HTC workload.

We scale the duration of jobs in our workload as follows:

$$D'^{s}_{j}(s,j) = D_j \times \frac{P_s}{P_b} \quad (2)$$

where P_b is the baseline server performance, P_s is the performance of one of the other selected servers, and D_j is the job duration which is the start time job minus end time job.

4.2 Metrics

Here we outline a number of key metrics reported by the simulation, which are subsequently used in Section 5 to evaluate the policies outlined in Section 3.

4.2.1 Overhead

Our main performance metric is average overhead for jobs within the system, which is calculated as the difference between the execution time of the job D'^{s}_{j} and the amount of time the job took to run in the simulation. Overheads may be incurred through waiting for an idle resource.

4.2.2 Cloud hours

We report the number of *'instance hours'* arising from each policy under investigation. This paper focuses primarily on provider-side so we do not provide financial cost to users; however, this may be trivially calculated by multiplying the number of instance hours by the hourly price.

4.2.3 Infrastructure Energy Consumption

We further extend the simulation to make use of the industry-standard Power Usage Effectiveness (PUE) metric [13] to calculate the total facility power. The PUE metric provides a ratio of energy consumption attributed to computing equipment compared to the infrastructure required to support it, including cooling and air conditioning.

$$PUE = \frac{TotalFacilityPower}{TotalITEquipmentPower} \quad (3)$$

We go further to incorporate the energy impact of networking infrastructure into our model of total facility energy consumption. While other efforts [9] have performed detailed analysis of the networking available in data centre environments, we chose to leverage an observation by Brown [3] that approximately 5% of the average data centre's total power use is absorbed by the network. We included this supposition into our simulation to acquire an estimation of the total facility power.

4.3 Simulation Scenario

In the remainder of the paper we evaluate scheduling policies from [14] using historical logs for 409,479 completed jobs from the HTCondor cluster located at Newcastle University, to calculate the energy consumption by applying varying job management policies of HTC jobs to Cloud instances.

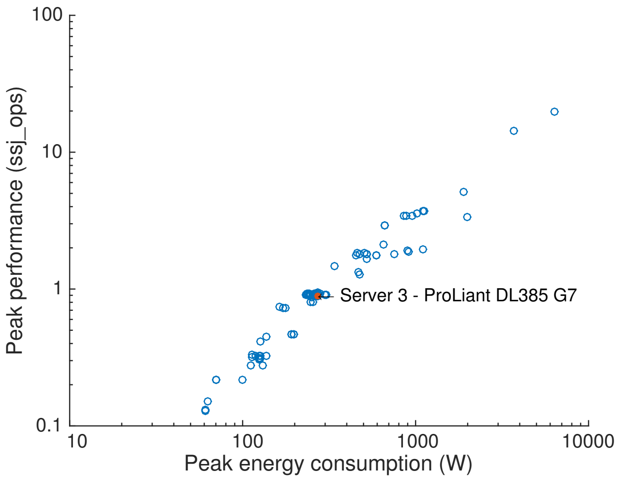

Figure 1: P1

Table 1 shows the server types we consider in the rest of this paper. We select Server 3 (*'ProLiant DL385 G7'*) to represent the baseline server against which task duration is scaled as in Equation 2. Server 3 was selected as the original HTCondor dataset originates from 2010, and the chosen server exhibits close to average peak power and performance characteristics from all published SPECpower results in 2010, as shown in Figure 1. The remaining servers were selected to offer a variety of performance and power characteristics, with this diversity of machines helping us indicate which policy is the best fit for the server based on its performance and energy consumption.

No	Server Name	Cores	Chips	Peak Power (W)	Idle Power (W)	ssj_ops	Scaling ratio	Test Date
1	PRIMERGY RX2560 M1	36	2	267	40.1	3,256,040	3.66	Mar-15
2	Express5800/A1080a-E	64	8	1749	993	3,647,000	4.10	Dec-10
3	ProLiant DL385 G7	48	4	271	101	888,819	1.00	Mar-10
4	PRIMERGY TX150 S7	4	1	112	24.3	276,514	0.31	Jan-10
5	Proliant DL580 G5	16	4	387	271	359,523	0.40	Dec-07

Table 1: Selected server from SPECpower_ssj 2008 published results [22]

5. RESULTS

In this section, we present the results of our preliminary experimentation, and explore the performance and energy impacts of the policies outlined in Section 3.

5.1 Limiting the number of Cloud instances

Figure 2 shows the impact of varying the maximum number of cloud instances. We see significant reductions in average overheads through increasing the instance count. The impact of the different performance levels of the servers is evident in these policies, with overheads much lower for servers such as 1 and 2 which exhibit better performance, and greater overheads observed for slower servers.

When considering the number of instance hours (hence cost incurred), we see for each of the computers that by 150 for higher performance servers, and by 300 instances for slower servers, the number of instance hours consumed plateaus and does not increase much further. Similar trends are evident when considering the energy consumption, with the exception of servers 4 and 5, which see an initially high energy consumption for lower number of instances, due to their substantially lower performance than the benchmark server.

5.2 Merging of different users' jobs

In Figure 3 we demonstrate the gains possible by merging workloads from multiple users, for policy P1 varying the maximum number of cloud instances. We see the merging of workloads leads to reduced average overheads, with jobs more likely to enter the system to find an available idle instance.

We acknowledge significant variability for this policy due to random effects in the simulation. As the maximum number of cloud instances increasing this tends towards zero. We see Policy P2 is capable of modest reduction of the number of cloud hours required. Similarly, we observe general trends that Policy P2 is capable of reducing energy consumption. Both the difference in number of instance hours and difference in energy efficiency will tend to zero as the number of cloud instances increases.

5.3 Instance keep-alive

In Figure 4 we explore the impact of Policy P3, which governs whether instances are to be kept-alive and remain active during a subsequent billing period to serve jobs arriving jobs. Here we show results for our baseline server (*ProLiant DL385 G7*) and a maximum instance count of 500. These results are representative of those for the other servers we consider; we omit results for each server due to space limitations.

In [14], the authors acknowledge this policy reduced average overheads significantly only under the assumption that cloud instances take 10 minutes to provision, with benefits minimal for smaller provisioning durations. Our results reinforce this finding, and we further demonstrate the policy leads to increased energy consumption. The authors also observe this policy is dependent on the characteristics of the offered workload. They present experimental results using synthetic workloads and demonstrate the potential for savings. We see this as an important area of future work, and discuss this further in Section 6.

Figure 5 demonstrates the relative impact of the three variants of Policy P3. The 'Fixed' policy clearly offers the most compelling overhead reductions but at the expense of the number of instance hours (and hence cost) consumed, and also has the most detrimental impact on the energy consumption of the system. The 'Idle' policy is the most promising policy, in that is is capable of reducing average overheads while incurring the least negative impact on the other metrics we consider. The 'Load' policy is shown to perform better in terms of overhead reduction, but at the expense of instance hours used and energy consumption.

5.4 Delaying the start of instances

Figure 6 demonstrates the impact of Policy P4 on average overhead, number of cloud hours consumed and energy consumption. The policy seeks to reduce the negative impact of short-running jobs on the system, which would otherwise occupy only partial hours on a resource. By delaying the creation of a cloud instance, it is hoped that subsequent jobs will arrive and be able to make use of idle time on the hired cloud resource.

We demonstrate results here with a maximum cloud instance limit of 500, and vary the maximum job delay between zero and thirty minutes. We demonstrate that as expected, this policy results in a detrimental on average overheads, but leads to reductions in the number of instance hours consumed and hence energy consumption. There is clearly a trade-off between overheads incurred for the HTC workload and cost/energy consumption which must be reconciled.

Focusing on the impact on energy consumption, we see for each of the computers considered, imposing a maximum job delay of 30 minutes results in reductions of over 50%, but in exchange for an intolerable increase in task overheads. Imposing a delay of up to 5 minutes would reduce energy consumption by 20% in exchange for slightly less than a doubling in average overhead, which would appear more acceptable.

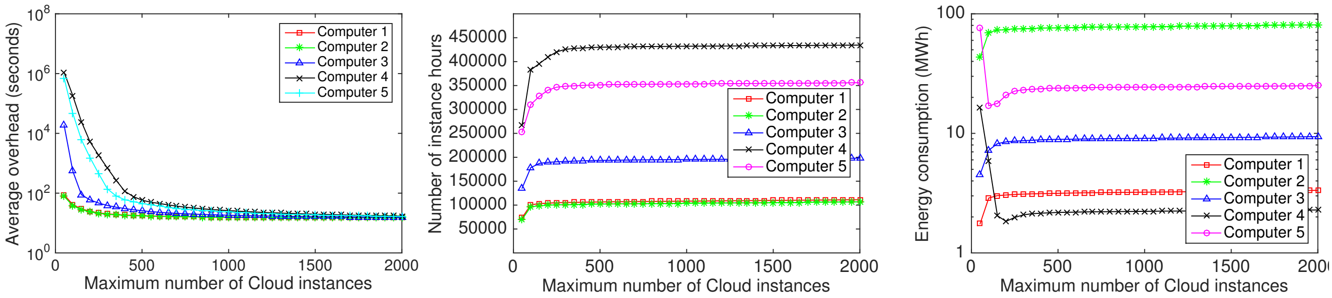

Figure 2: Impact of Policy P1 on average overhead, number of instance hours, and energy consumption.

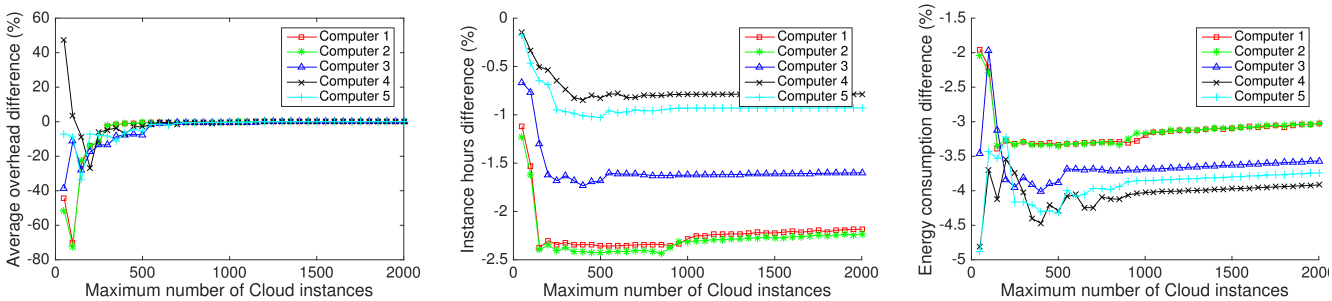

Figure 3: Impact of Policy P2 on average overhead, number of instance hours, and energy consumption.

6. CONCLUSIONS

This paper has evaluated the energy consumption of a number of policies governing the scheduling of a HTC jobs to cloud instances. These policies exhibit varying impacts on energy consumption and average overheads, and in all cases have demonstrated the criticality of the trade-off between energy consumption and performance/cost. We evaluate policies under a number of assumptions made about the hardware used within a cloud datacentre. Through this work have extended an existing simulation tool to allow practitioners to reason over resource allocation decisions in a cloud setting. The development of this tool will ultimately support the development and evaluation new algorithms for HTC workload scheduling in cloud environments.

In our ongoing work we are extending our simulation framework to support the modelling of virtualised resources. This will allow us to evaluate policies governing the consolidation of HTC workloads onto virtualised resources in a cloud environment. Furthermore, in our current work we consider a data centre of heterogeneous resources; in the future we hope to relax this assumption to evaluate classes of policy which leverage differences in servers to make a selection on which to use.

The authors of [14] have recently released HTC-Sim [8], a high level trace-driven simulation to evaluate the energy consumption and performance of different HTC operating policies. HTC-Sim currently has a focus on local dedicated and non-dedicated resources, but we see the goals of this initiative to be complementary with ours, and we seek to incorporate and extend our modelling of cloud resources within this environment.

7. REFERENCES

[1] Cycle Computing (Homepage). http://cyclecomputing.com/, 2014.

[2] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5):755–768, 2012.

[3] R. Brown. Report to congress on server and data center energy efficiency: Public law 109-431. *Lawrence Berkeley National Laboratory*, 2008.

[4] C. De Alfonso, M. Caballer, F. Alvarruiz, and G. Moltó. An economic and energy-aware analysis of the viability of outsourcing cluster computing to a cloud. *Future Generation Computer Systems*, 29(3):704–712, 2013.

[5] M. D. de Assuncao, A. di Costanzo, and R. Buyya. Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. HPDC '09, pages 141–150, New York, NY, USA, 2009. ACM.

[6] E. Deelman, G. Singh, M. Livny, 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*, page 50. IEEE Press, 2008.

[7] T. V. T. Duy, Y. Sato, and Y. Inoguchi. Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In *IPDPSW*, pages 1–8. IEEE, 2010.

[8] M. Forshaw, N. Thomas, and S. McGough. Trace-driven simulation for energy consumption in high throughput computing systems. In *18th IEEE/ACM International Symposium on Distributed*

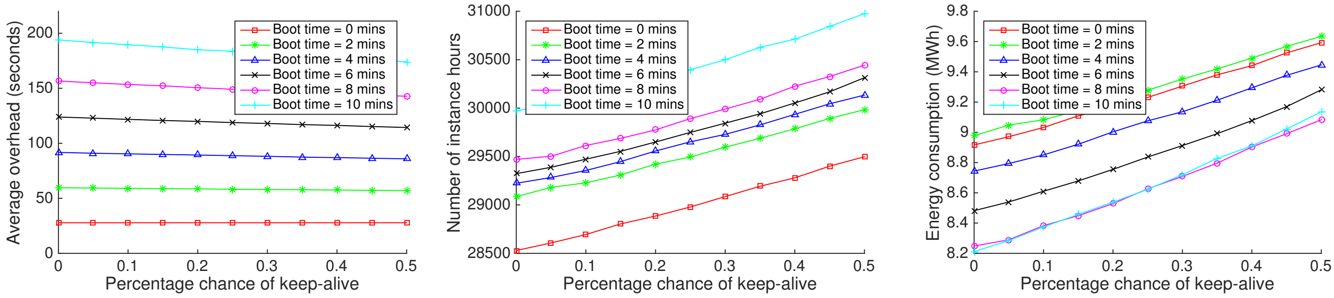

Figure 4: Impact of Policy P3 on average overhead, number of instance hours, and energy consumption. Varying chance of keep-alive and boot time for Server 3.

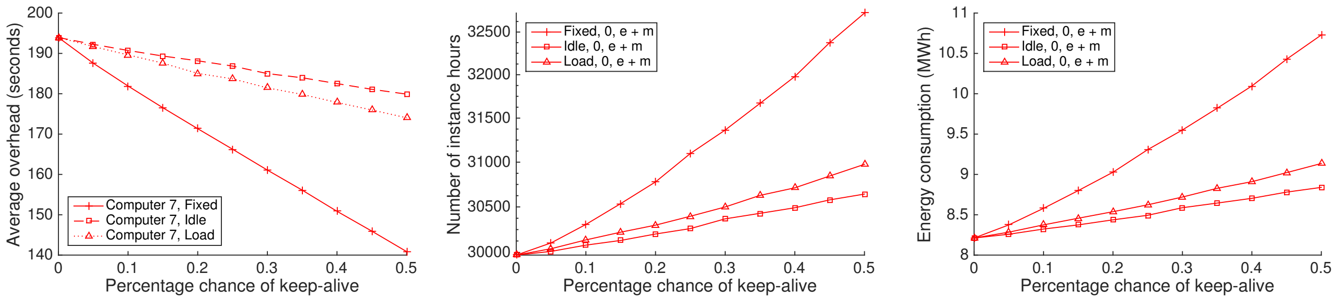

Figure 5: Impact of Policy P3 on average overhead, number of instance hours, and energy consumption. Varying chance of keep-alive for a boot time of ten minutes.

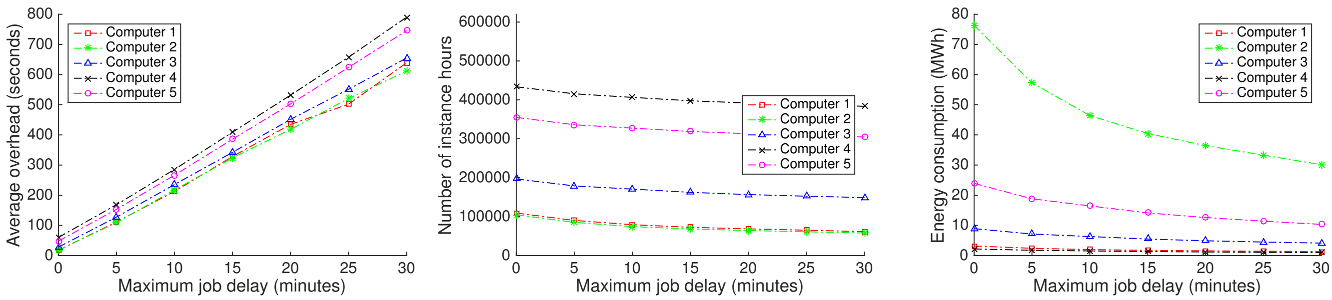

Figure 6: Impact of Policy P4 on average overhead, number of instance hours, and energy consumption.

Simulation and Real Time Applications (DS-RT), 2014.

[9] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan. Greencloud: A packet-level simulator of energy-aware cloud computing data centers. In *GLOBECOM*, pages 1–5, 2010.

[10] M. Litzkow, M. Livney, and M. W. Mutka. Condor-a hunter of idle workstations. ICDCS '88, pages 104–111, 1998.

[11] C. Liu, X. Qin, S. Kulkarni, C. Wang, S. Li, A. Manzanares, and S. Baskiyar. Distributed energy-efficient scheduling for data-intensive applications with deadline constraints on data grids. In *IPCCC*, pages 26–33. IEEE, 2008.

[12] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos. Cloud computing: Survey on energy efficiency. *ACM Computing Surveys (CSUR)*, 47(2):33, 2014.

[13] M. Mattess, C. Vecchiola, and R. Buyya. Managing peak loads by leasing cloud infrastructure services from a spot market. HPCC '10, pages 180–188, 2010.

[14] A. S. McGough, M. Forshaw, C. Gerrard, S. Wheater, B. Allen, and P. Robinson. Comparison of a cost-effective virtual cloud cluster with an existing campus cluster. *Future Generation Computer Systems*, 41(0):65 – 78, 2014.

[15] SPEC. SPECpower_ssj2008. http://www.spec.org/power$_{s}sj$2008/.

[16] UK Research Council End Use Energy Demand (EUED) Centres. http://www.eued.ac.uk/, 2014.

[17] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove. Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In *Cloud Computing*, 2010.