Exploiting Heterogeneity in Grid Computing for Energy-Efficient Resource Allocation

Saurabh Kumar Garg and Rajkumar Buyya
The Cloud Computing and Distributed Systems
Department of Computer Science and Software Engineering
The University of Melbourne, Australia
Email: {sgarg, raj}@csse.unimelb.edu.au

***Abstract*—The growing computing demand from industry and academia has lead to excessive power consumption which not only impacting long term sustainability of Grids like infrastructures in terms of energy cost but also from environmental perspective. The problem can be addressed by replacing with more energy efficient infrastructures, but the process of switching to new infrastructure is not only costly but also time consuming. Grid being consist of several HPC centers under different administrative domain, make problem more difficult. Thus, for reduction in energy consumption, we address the challenge by effectively distributing compute-intensive parallel applications on grid. We presented a metascheduling algorithm which exploits the heterogeneous nature of Grid to achieve reduction in energy consumption. Simulation results show that out algorithm HAMA can significantly improve the energy efficiency of global grids by a factor of typically 23% and as much as a factor of 50% in some cases while meeting user's QoS requirements**

I. Introduction

From last many years, global grid is serving as a mainstream High Performance Computing (HPC) platform to provide massive computational power to execute large-scale and compute-intensive scientific and technological applications. Enlarging the existing global grid infrastructure to meet the increasing demand from grid users can progressively speed up the advancement of science and technology. But the growing environmental and economic impact due to high energy consumption of HPC platforms has become a major bottleneck in expansion of grid like platforms.

In April 2007, Gartner estimates that the ICT industry is liable for 2% of the global CO_2 emissions annually, which is equal to the aviation industry [1][2]. In addition to that, the high power consumption has not only lead to rapid increase in utility bills but also affecting the reliability of servers due to high concentrated heat loads. The power efficiency of a HPC center depend on number of factors such as processor's power efficiency, cooling and air conditioning system, infrastructure design and lighting/physical system. A recent study [3] done by Lawrence Berkeley National Laboratory shows the cooling efficiency (the ratio of computer power : cooling power) of data centers varies drastically from a low of 0.6 to a high of 3.5. Thus, the sustainable and environmental-friendly solutions must be employed by current HPC community to increase the energy efficiency of HPC systems which can more effectively make use of electricity.

While a lot of research has been performed to increase efficiency of individual clusters at various levels such as processor level (CPU) [4][5], in virtualization based resource managers [6], and cluster resource managers [7][8], the research on improving the energy efficiency of global systems such as grid is still in its infancy. Most of the existing grid meta-schedulers, such as Maui/Moab scheduling suite [9], Condor-G [10], and GridWay [11], focus on improving system-centric performance metrics such as utilization, average load and application's turnaround time. Others such as Gridbus Broker [12] focus on deadline and budget constrained scheduling. Thus, this paper examines how a grid meta-scheduler can exploit the heterogeneity of the global grid infrastructure to achieve reduction in energy consumption of overall grid. In particular, we focus on designing a meta-scheduling policy that can be easily adopted by existing grid meta-schedulers without many changes in current grid infrastructure. This work will also have relevance to emerging cloud computing paradigm when scaling of application across multiple clouds is considered [13]. The key contributions of this paper are:

1) It defines a novel Heterogeneity Aware Meta-scheduling Algorithm (HAMA) that considers various factors contributing to high energy consumption of grids, including cooling system efficiency and CPU power efficiency.
2) It demonstrates through extensive simulations using real workload traces that the energy efficiency of global grids can be improved as much as 23% with HAMA.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 defines the grid meta-scheduling model. Section 4 describes HAMA. Section 5 explains the evaluation methodology and simulation setup for comparing HAMA with existing meta-scheduling policies. In Section 6, the performance results of HAMA are analyzed. Section 7 concludes the paper and presents future work.

II. Related Work

This section presents related work on energy-efficient/power-aware scheduling on grids. To the best of our knowledge, no previous work has proposed a meta-scheduler that explicitly addresses energy efficiency of grids from a global perspective.

Currently, for global grids, meta-schedulers in operation, such as GridWay [11] use heuristics such as First Come First

Serve (FCFS). Moab also has a FCFS batch scheduler with easy backfilling policy [9], [14]. Condor-G [10] uses either FCFS or matchmaking with priority sort [15] as scheduling policies. These schedulers mostly schedule jobs with several goals such as minimizing job completion time and achieving load balancing. The issue of energy consumption emission by the grids still need to be addressed.

There are several research efforts on power-aware resource allocation to optimize energy consumption at a single resource site, typically within a single cluster or data center. The power usage reduction within the resource site is achieved through two methods: by switching off parts of the cluster that are not utilized [16], [17], [18], [7]; or by Dynamic Voltage Scaling (DVS) to slow down the speed of CPU processing [19], [20], [21], [22], [8], [23], [24], [7]. Hence, these efforts help reduce the energy consumption of one resource site such as cluster or server farm, but not across multiple resource sites distributed geographically.

Orgerie et al. [16] propose a prediction algorithm to reduce the power consumption in a large-scale computational grid such as Grid'5000 by aggregating the workload and switching off unused CPUs. They focus on reducing CPU power consumption to minimize the total energy consumption. As the power efficiency of grid sites can vary across the grid, reducing CPU power consumption itself may not necessary lead to a global reduction in the energy consumption by the entire grid. We focus on conserving energy of grids from a global perspective.

Meisner et al. [19] show that in the case of high and unpredictable workload, it is difficult to exploit the power on/off facility even though it is ideal to simply switch off idle systems. Thus, DVS-enabled CPUs will be much better in saving energy in this case. Therefore, in this work we use DVS to reduce the energy consumption of CPUs since our main focus is on large-scale computational grid resource sites which generally have unpredictable workload.

III. Grid Meta-scheduling Model

A. System Model

A grid meta-scheduler acts as an interface to grid resource sites and schedules jobs on the behalf of users as shown in Figure 1. It interprets and analyzes the service requirements of a submitted job and decides whether to accept or reject the job based on the availability of CPUs. Its objective is to schedule jobs so that the energy consumption of grid can be reduced while the Quality of Service (QoS) requirements of the jobs are met. As grid resource sites are located in different geographical regions, they have different power efficiency of CPUs and cooling systems. Each resource site is responsible for updating this information at the meta-scheduler for energy efficient scheduling. The two participating parties, grid users and grid resource sites, are discussed below along with their objectives and constraints:

1) **Grid Users:**
   Grid users submit parallel jobs with QoS requirements to the grid meta-scheduler. Each job must be executed on an individual grid resource site and does not have preemptive priority. The reason for this requirement is that the synchronization among various tasks of parallel jobs can be affected by communication delays when jobs are executed across multiple resource sites. The user's objective is to have his job completed by a deadline. Deadlines are hard, i.e., the user will benefit only if the job completes before its deadline [25]. To facilitate the comparison between the algorithms described in this work, the estimated execution time of a job provided by the user is considered to be accurate [26]. Several models, such as those proposed by Sanjay and Vadhiyar [27], can be applied to estimate the runtime of parallel jobs. In this work, a job execution time is inversely proportional to the CPU operating frequency.

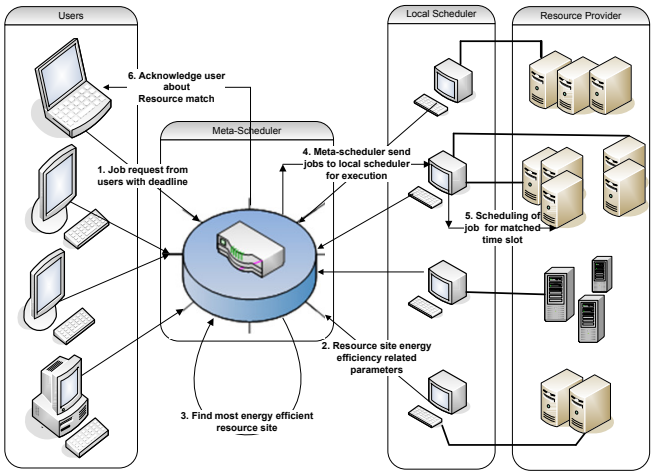


Fig. 1. Meta-scheduling protocol

2) **Grid Resource Sites:**
   Grid resource sites consist of clusters at different locations, such as the sites of the Distributed European Infrastructure for Supercomputing Applications (DEISA) [28] with resource sites located in various European countries and LHC Grid across the world [29]. Each resource site has a local scheduler that manages the execution of incoming jobs. Each local scheduler periodically supplies information about available time slots (t_s, t_e, N) to the meta-scheduler, where t_s and t_e are the start time and end time of the slot respectively and N is the number of CPUs available for the slot. To facilitate energy efficient computing, each local scheduler also supplies information about cooling system efficiency, CPU power-frequency relationship, and CPU operating frequencies of the grid resource site. All CPUs within a single resource site are homogeneous, but CPUs can be heterogeneous across resource sites.

B. Grid Resource Site Energy Model

The major contributors for total energy usage in grid resource site are computing devices (CPUs) and cooling system

which constitute about 80% of total energy consumption. Other systems such as lighting are not considered due to their negligible contribution to the total energy cost.

The power consumption P of a CPU at a grid resource site is composed of dynamic and static power [21][7]. The static power includes the base power consumption of the CPU and the power consumption of all other components. Thus, the CPU power P is approximated by the following function (similar to previous work [21][7]): $P = \beta + \alpha f^3$, where β is the static power consumed by the CPU, α is the proportionality constant, and f is the frequency at which the CPU is operating. We consider that CPUs support DVS facility and thus their frequency can be varied from a minimum of f^{min} to a maximum of f^{max} discretely. Let N_i be number of CPUs at a resource site i. Thus, if the CPU j running at frequency f_j for t_j time, then the total energy consumption due to computation is given by:

$$E_{c,i} = \sum_{N_i}^{j} (\beta_i + \alpha_i f_j^3) t_j. \quad (1)$$

The energy cost of an cooling system depends on the Coefficient Of Performance (COP) factor of the cooling system [30]. COP is indication of efficiency of cooling system which is defined as the ratio of the amount of energy consumed by CPUs to the energy consumed by the cooling system. The COP is however not constant and varies with cooling air temperature. We assume that COP will remain constant during scheduling cycle and resource sites will update meta-scheduler whenever COP changes. Thus, the total energy consumed by cooling system is given by:

$$E_{h,i} = \frac{E_{c,i}}{COP_i} \quad (2)$$

Thus, the resultant total energy consumption by a grid resource site is given by:

$$E_i = (1 + \frac{1}{COP_i}) E_{c,i} \quad (3)$$

IV. Heterogeneity Aware Meta-scheduling Algorithm (HAMA)

This section gives the details of our Heterogeneity Aware Meta-scheduling Algorithm (HAMA) which enables the grid meta-scheduler to select the most energy efficient grid resource site. The grid meta-scheduler runs HAMA periodically to assign jobs to grid resource sites. HAMA achieves this by first selecting the most energy efficient grid resource site and then by using DVS for further reduction in the energy consumption. Algorithm 1, described next, shows the pseudo-code for HAMA. At each scheduling interval, the meta-scheduler collects information from both grid resource sites and users (Algorithm 1: Line 2–3). Considering that a grid consists of n resource sites (supercomputer centers), all parameters associated with each resource site i are given in Table I. A user submits his QoS requirements for a job j in the form of a tuple $(d_j, n_j, e_j, f_{m,j})$, where d_j is the deadline to complete job j, n_j is the number of CPUs required for job execution, and e_j is the job execution time when operating at the CPU frequency $f_{m,j}$. In addition, let f_{ij} be the initial frequency at which CPUs of a grid resource site i operate while executing job j. HAMA, then, sorts the incoming jobs based on Earliest Deadline First (EDF) (Algorithm 1: Line 4). The grid resource sites are sorted in order of their power efficiency (Algorithm 1: Line 5) which is calculated by Cooling system efficiency $\times$ CPU Efficiency, i.e., $(1+\frac{1}{COP_i}) \times (\frac{\beta_i}{f_i^{max}} + \alpha_i (f_i^{max})^2)$. Then, meta-scheduler assigns jobs to resource sites according to this ordering (Algorithm 1: Line 7–11).

Algorithm 1: HAMA

```
1  while current_time < next_schedule_time do
2      RecvResourcePublish(P)
       //P contains information about grid resource sites
3      RecvJobQoS(Q)
       //Q contains information about grid users
4  Sort jobs in ascending order of deadline
5  Sort resource sites in ascending order of
   (1 + 1/COP_i) × (β_i/f_i^max + α_i(f_i^max)^2)
6  foreach job j ∈ RecvJobQoS do
7      foreach resource site i ∈ RecvResourcePublish do
           //find time slot for scheduling job j at resource site i
8          if FindTimeSlot(i,j) then
9              Schedule job j on resource site i using DVS;
               Update available time slots at resource site i
10             break
11
```

TABLE I
Parameters of a Grid Resource Site i

Parameter	Notation
Average Cooling system efficiency	COP_i
CPU power	$P_i = \beta_i + \alpha_i f^3$
CPU frequency range	$[f_i^{min}, f_i^{max}]$
Time slots (start time, end time, number of CPUs)	(t_s, t_e, n)

The energy consumption is further reduced by scheduling jobs using DVS at the CPU level which can save energy by scaling down the CPU frequency. Thus, when the grid meta-scheduler assigns a job to a grid resource site, it also decides the time slot in which jobs should be executed at the minimum frequency level to decrease energy consumption by CPU (Algorithm 1: Line 8). If the job deadline is violated, the meta-scheduler scales up the CPU frequency to the next level and then again tries to find the free slot to execute the job. The execution time an application is considered to increase linearly with the decrease in CPU frequency. Thus, in next CPU frequency level, since CPU will be executing application at higher frequency level, the time slot required will be shorter.

As at a resource site CPUs may or may not have the DVS facility, thus the scheduling at the local scheduler level can be of two types: CPUs run at the maximum frequency (i.e. without DVS); or CPUs run at various frequency using DVS

(i.e. with DVS). If the meta-scheduler fails to schedule the job on the resource site because no free slot is available, then the job is forwarded to the next energy efficient resource site for scheduling.

V. Performance Evaluation

We use workload traces Feitelson's Parallel Workload Archive (PWA) [31] to model the global grid workload. Since this paper focuses on studying the application requirements of grid users, the PWA meets our objective by providing job traces that reflect the characteristics of real parallel applications. The experiments utilize the jobs in the first week of the LLNL Thunder trace (January 2007 to June 2007). The LLNL Thunder trace from the Lawrence Livermore National Laboratory (LLNL) in USA is chosen due to its highest resource utilization of 87.6% among available traces to ideally model a heavy workload scenario. From this trace, we obtain the submit time, requested number of CPUs, and actual runtime of jobs. However, the trace does not contain the service requirement of jobs (i.e. deadline). Hence, we use a methodology proposed by Irwin et al. [32] to synthetically assign deadlines through two classes namely Low Urgency (LU) and High Urgency (HU).

A job i in the LU class has a high ratio of $deadline_i/runtime_i$ so that its deadline is definitely longer than its required runtime. Conversely, a job i in the HU class has a deadline of low ratio. Values are normally distributed within each of the high and low deadline parameters. The ratio of the deadline parameter's high-value mean and low-value mean is thus known as the high:low ratio. In our experiments, the deadline high:low ratio is 3, while the low-value deadline mean and variance is 4 and 2 respectively. In other words, LU jobs have a high-value deadline mean of 12, which is 3 times longer than HU jobs with a low-value deadline mean of 4. The arrival sequence of jobs from the HU and LU classes is randomly distributed.

Provider Configuration: The grid modelled in our simulation contains 8 resource sites spread across five countries derived from European Data Grid (EGEE) testbed [29]. The configurations assigned to the resources in the testbed for the simulation are listed in Table II. The configuration of each resource site is decided so that the modelled testbed would reflect the heterogeneity of platforms and capabilities that is normally the characteristic of such installations. Power parameters (i.e. CPU power factors and frequency level) of the CPUs at different sites are derived from Wang and Lu's work [7]. Current commercial CPUs only support discrete frequency levels, such as the Intel Pentium M 1.6 GHz CPU, which supports 6 voltage levels. We consider discrete CPU frequencies with 5 levels in the range $[f_i^{min}, f_i^{max}]$. For the lowest frequency f_i^{min}, we use the same value used by Wang and Lu [7], i.e. f_i^{min} is 37.5% of f_i^{max}. Each local scheduler at a grid site use Conservative Backfilling with advance reservation support as used by Mu'alem and Feitelson [33]. The grid meta-scheduler schedules the job periodically at a scheduling interval of 50 seconds, which is to ensure that the meta-scheduler can receive at least one job in every scheduling interval. The cooling system efficiency (COP) value of resource sites is randomly generated using a uniform distribution between [0.5, 3.6] as indicated in the study conducted by Greenberg et al. [3].

Grid Meta-scheduling Algorithms: We examine the performance of HAMA in terms of job selection and resource allocation of the grid meta-scheduler. We compare our job selection algorithm with EDF-FQ which prioritize jobs based on deadline and submit jobs to resource site in earliest start time (FQ) manner with the least waiting time. We also compare HAMA with another version of HAMA i.e. HAMA-withoutDVS to analyze the affect of DVS facility on energy consumption.

Performance Metrics: We consider two metrics: average energy consumption and workload (i.e. amount of workload executed). Average power consumption shows the amount of energy saved by using HAMA in comparison to other grid meta-scheduling algorithms, whereas workload shows HAMA affect on the workload executed successfully by grid.

Experimental Scenarios: We run the experiments in two scenarios 1) urgency class and 2) arrival rate of jobs. For the urgency class, we use various percentages (0%, 20%, 40%, 60%, 80%, and 100%) of HU jobs. For instance, if the percentage of HU jobs is 20%, then the percentage of LU jobs is the remaining 80%. For the arrival rate, we use various factors (10, 100, and 1000) of submit time from the trace. For example, a factor of 10 means a job with a submit time of 10s from the trace now has a simulated submit time of 1s. Hence, a higher factor represents higher workload by shortening the submit time of jobs.

Equation 3, we know that the performance of HAMA is highly dependent on the CPU efficiency and Cooling System efficiency of grid resource sites. We compare performance of our algorithm in worst case scenario (HL) i.e., when resource site with the highest CPU power efficiency has the lowest COP, and best case scenario (HH) i.e., when resource site with the highest CPU power efficiency has the highest COP (HH).

VI. Performance Results

A. Affect on Energy consumption

This section compares energy consumption of HAMA with other meta-scheduling algorithms for grid resource sites with HH and HL configurations. The figure 2 shows how energy consumption varies with deadline urgency and arrival rate of jobs. HAMA has clearly outperformed its competitor EDF-FQ by saving about 17%-23% energy in worst case and about 52%in best case.

The effect of job urgency on energy consumption can be clearly seen from figure 2(a) and 2(b). As the percentage of HU jobs with more urgent (shorter) deadline increases, the energy consumption (Figure 2(a) and 2(b)) also increases due to more urgent jobs running on resource sites with lower power efficiency and at the highest CPU frequency to avoid deadline violations. On the other hand, the effect of job arrival rate on

TABLE II
CHARACTERISTICS OF GRID RESOURCE SITES

Location of Grid Site	CPU Power Factors			No. of CPUs	MIPS Rating
		β	α	f_i^{max}	
RAL, UK	65	7.5	1.8	2050	1140
Imperial College (UK)	75	5	1.8	2600	1200
NorduGrid (Norway)	60	60	2.4	650	1330
NIKHEF (Netherlands)	75	5.2	2.4	540	1176
LYON (France)	90	4.5	3.0	600	1166
Milano (Italy)	105	6.5	3.0	350	1320
Torina (Italy)	90	4.0	3.2	200	1000
Padova (Italy)	105	4.4	3.2	250	1330

energy consumption (Figure 2(c) and 2(d) is minimal with a slight increase when more jobs arrive.

For grid resource sites without DVS, HAMA-without can reduce up to 15-21% of the energy consumption (Figure 2(a)) in the HL configuration and 28-50% of energy consumption (Figure 2(b)) in the HH configuration compared to EDF-FQ which also doesn't consider the DVS facility while scheduling across the entire grid. This highlights the importance of the power efficiency factor in achieving energy-efficient meta-scheduling. In particular, HAMA can reduce energy consumption (Figure 2(a) and 2(b) even more when there are more LU jobs with less urgent (longer) deadline and arrival rate is low.

When we compare HAMA and HAMA-withoutDVS, we observe that by using DVS energy saving has increased by about 11% when % of job with urgent deadline and job arrival rate is high. This is because for the scenario when DVS facility is available jobs can run at lower CPU frequency to save energy.

B. Affect on Workload Executed

Figure 3 shows the total amount of workload successfully executed according to user's QoS. The workload of a job refer to multiplication of its execution time and the number of CPU required. The affect of job urgency and arrival rate on workload executed can be clearly seen from Figure 3(a) and 3(d). All meta-scheduling algorithm shows consistent decrease in workload execution particularly in scenario of job urgency. The reason is rejection of more jobs due to deadline miss when all jobs are of high urgency. The amount of workload executed by EDF-FQ is less than HAMA because of the reason that while scheduling using EDF-FQ, the local scheduler execute the jobs using conservative backfilling without any consideration of job deadline. While in case of HAMA, meta-scheduler send job to a resource site only if a time slot is available to execute job before deadline.

VII. CONCLUSION

With the increasing demand of global grids, the energy consumption of grid infrastructure has escalated to the degree that grids are becoming a threat to the society rather than an asset. The carbon footprint of grids may continue to increase unless the problem is addressed at every level, i.e., from local (within a single grid site) to global (across multiple grid sites). Moreover, the immediate and significant reduction in CO_2 emissions is required for the future sustainability of global grids.

In this paper, we have addressed the energy efficiency of grids at the meta-scheduling level. We proposed Heterogenity Aware Meta-scheduling Algorithm (HAMA) to address the problem by scheduling more workload with urgent deadline on resource sites which are more power-efficient. Thus, HAMA considers crucial information of global grid resource sites, such as cooling system efficiency (COP) and CPU power efficiency. HAMA address the problem in two steps: 1) allocating jobs to more energy-efficient resource sites and 2) scheduling using DVS policy at the local resource site to further reduce energy consumption.

Results show that our HAMA can reduce up to 23% energy consumption in worst case and upto 50% in best case as compare to other algorithms (EDF-FQ). Moreover, even if DVS facility is not available, HAMA-withoutEDF can still result in considerable amount of power savings of upto 21%. Particularly, our HAMA algorithm can work very well when the deadline of jobs is less urgent and arrival rate of jobs is not high. Thus, HAMA can also compliment the efficiency of existing power-aware scheduling policies for clusters.

In future, we will investigate how HAMA can address the energy consumption problem in virtualized environments such as clouds, which is the emerging platform for hosting business applications. We will also integrate HAMA with existing grid meta-schedulers and conduct experiments on real grid and cloud resources. We will also extend our current meta-scheduling model to resources such as the storage disks and the switching devices.

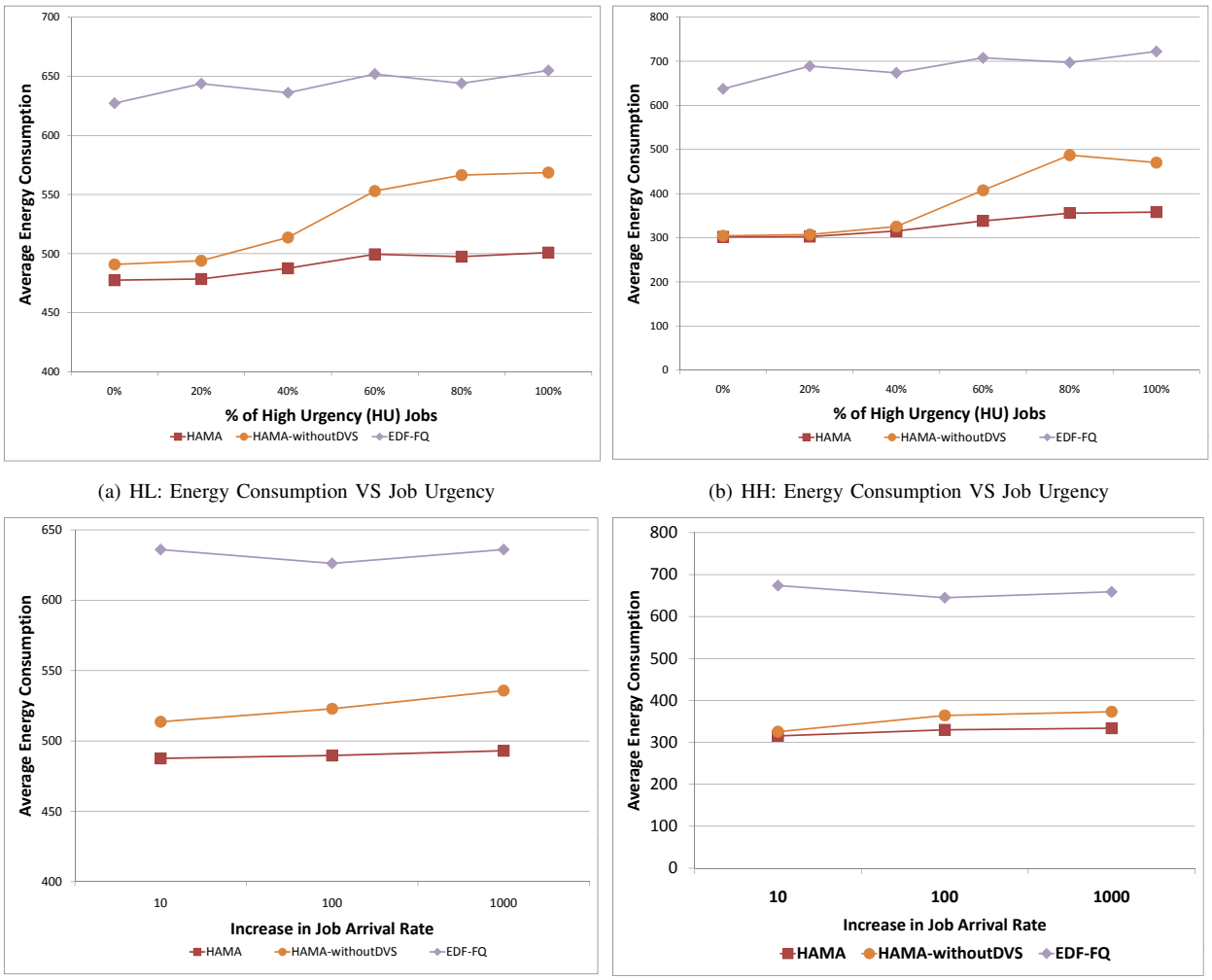
ACKNOWLEDGEMENTS

We would like to thank Chee Shin Yeo for his constructive comments on this paper. This work is partially supported by research grants from the Australian Research Council (ARC) and Australian Department of Innovation, Industry, Science and Research (DIISR).

REFERENCES

[1] Gartner, "Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions," http://www.gartner.com/it/page.jsp?id=503867.

[2] J. G. Koomey, "Estimating total power consumption by servers in US and world," http:/ /enterprise.amd.com/Downloads/svrpwrusecompletefinal.pdf.



(a) HL: Energy Consumption VS Job Urgency

(b) HH: Energy Consumption VS Job Urgency

(c) HL: Energy Consumption VS Job Arrival Rate

(d) HH: Energy Consumption VS Job Arrival Rate

Fig. 2. Comparison of HAMA with other meta-scheduling algorithms

[3] S. Greenberg, E. Mills, B. Tschudi, P. Rumsey, and B. Myatt, "Best practices for data centers: Results from benchmarking 22 data centers," in *Proc. of the 2006 ACEEE Summer Study on Energy Efficiency in Buildings*, Pacific Grove, USA, 2006, http://eetd.lbl.gov/emills/PUBS/PDF/ACEEE-datacenters.pdf.

[4] V. Salapura *et al.*, "Power and performance optimization at the system level," in *Proc. of the 2nd conference on Computing frontiers*, Ischia, Italy, 2005.

[5] A. Elyada, R. Ginosar, and U. Weiser, "Low-complexity policies for energy-performance tradeoff in chip-multi-processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 9, pp. 1243–1248, 2008.

[6] A. Verma, P. Ahuja, and A. Neogi, "pmapper: Power and migration cost aware application placement in virtualized systems," in *Proc. of the 9th ACM/IFIP/USENIX International Conference on Middleware*, Leuven, Belgium, 2008.

[7] L. Wang and Y. Lu, "Efficient Power Management of Heterogeneous Soft Real-Time Clusters," in *Proc. of the 2008 Real-Time Systems Symposium*, Barcelona, Spain, 2008.

[8] K. Kim, R. Buyya, and J. Kim, "Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters," in *Proc. of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, Rio de Janeiro, Brazil, 2007.

[9] B. Bode *et al.*, "The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters," in *Proc. of the 4th Annual Linux Showcase and Conference*, Atlanta, USA, 2000.

[10] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids," *Cluster Computing*, vol. 5, no. 3, pp. 237–246, 2002.

[11] E. Huedo, R. Montero, and I. Llorente, "A framework for adaptive execution in grids," *Software Practice and Experience*, vol. 34, no. 7, pp. 631–651, 2004.

[12] S. Venugopal, K. Nadiminti, H. Gibbins, and R. Buyya, "Designing a resource broker for heterogeneous grids," *SoftwarePractice & Experience*, vol. 38, no. 8, pp. 793–825, 2008.

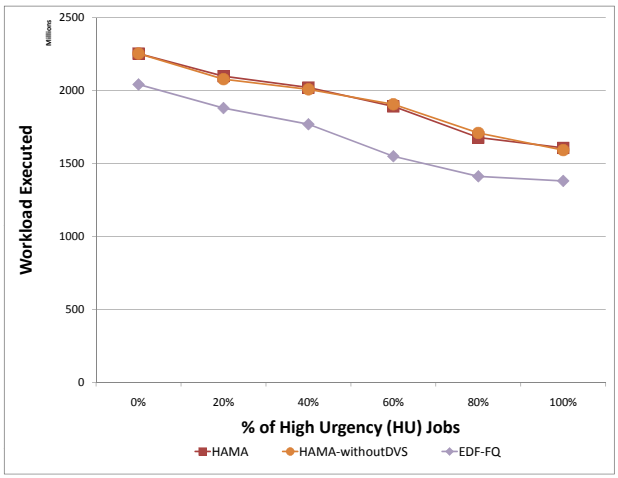
[13] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[14] Y. Etsion and D. Tsafrir, "A Short Survey of Commercial Cluster Batch Schedulers," Technical Report 2005-13, Hebrew University, May 2005, Tech. Rep.

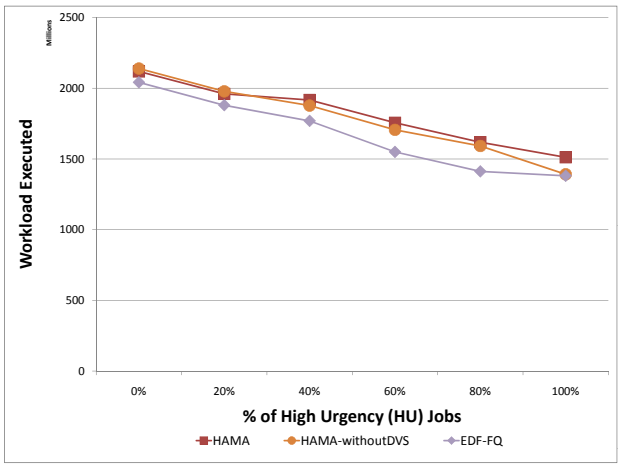
[15] R. Raman, M. Livny, and M. Solomon, "Resource Management through Multilateral Matchmaking," in *Proc. of the 9th IEEE Symposium on High Performance Distributed Computing*, Pittsburgh, USA, 2000.

[16] A. Orgerie, L. Lefèvre, and J. Gelas, "Save Watts in Your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems," in *Proc. of the 2008 14th IEEE International Conference on Parallel and Distributed Systems*, Melbourne, Australia, 2008.

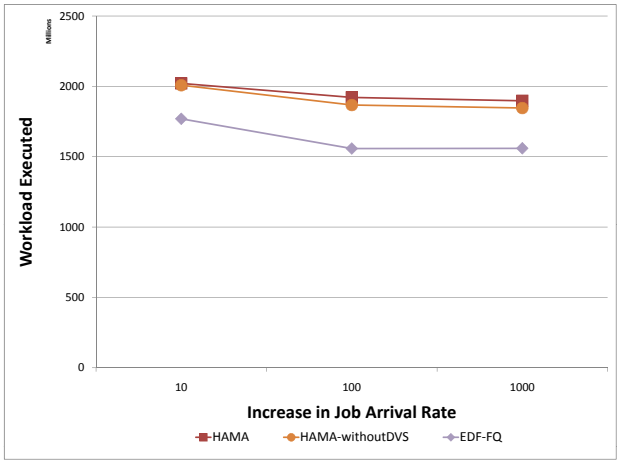
[17] D. Bradley, R. Harper, and S. Hunter, "Workload-based power management for parallel computer systems," *IBM Journal of Research and Development*, vol. 47, no. 5, pp. 703–718, 2003.



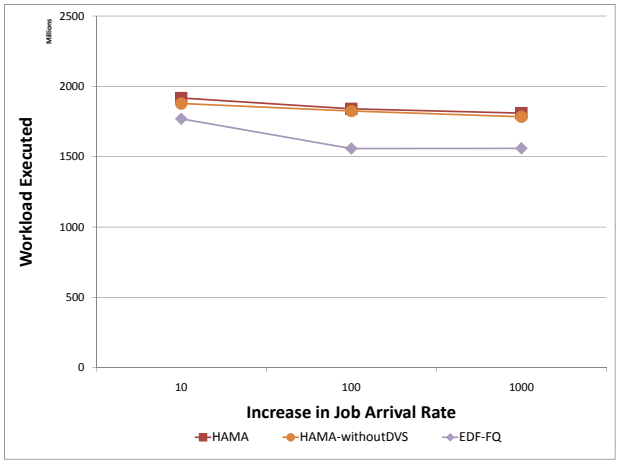
(a) HL: Workload Execution VS Job Urgency



(b) HH: Workload Execution VS Job Urgency



(c) HL: Workload Execution VS Job Arrival Rate



(d) HH: Workload Execution VS Job Arrival Rate

Fig. 3. Comparison of HAMA with other meta-scheduling algorithms

[18] B. Lawson and E. Smirni, "Power-aware resource allocation in high-end systems via online simulation," in *Proc. of the 19th annual international conference on Supercomputing*, Cambridge, USA, 2005, pp. 229–238.

[19] D. Meisner, B. Gold, and T. Wenisch, "PowerNap: eliminating server idle power," in *Proceeding of the 14th international conference on Architectural support for programming languages and operating systems*, Washington, USA, 2009.

[20] G. Tesauro *et al.*, "Managing power consumption and performance of computing systems using reinforcement learning," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2007.

[21] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 303–314, 2005.

[22] A. Verma, P. Ahuja, and A. Neogi, "Power-aware dynamic placement of HPC applications," in *Proc. of the 22nd annual international conference on Supercomputing*, Athens, Greece, 2008, pp. 175–184.

[23] N. Kappiah, V. Freeh, and D. Lowenthal, "Just in time dynamic voltage scaling: Exploiting inter-node slack to save energy in MPI programs," in *Proc. of the 2005 ACM/IEEE conference on Supercomputing*, Seattle, USA, 2005.

[24] C. Hsu and W. Feng, "A power-aware run-time system for high-performance computing," in *Proc. of the 2005 ACM/IEEE conference on Supercomputing*, Seattle, USA, 2005.

[25] R. Porter, "Mechanism design for online real-time scheduling," in *Proc. of the 5th ACM conference on Electronic commerce*, New York, USA, 2004, pp. 61–70.

[26] D. G. Feitelson, L. Rudolph, U. Schwiegelshohn, K. C. Sevcik, and P. Wong, "Theory and practice in parallel job scheduling," in *Job Scheduling Strategies for Parallel Processing*, London, UK, 1997, pp. 1–34.

[27] H. A. Sanjay and S. Vadhiyar, "Performance modeling of parallel applications for grid scheduling," *J. Parallel Distrib. Comput.*, vol. 68, no. 8, pp. 1135–1145, 2008.

[28] "Distributed European Infrastructure for Supercomputing Applications (DEISA)," http://www.deisa.eu.

[29] Enabling Grids for E-sciencE, "EGEE project," http://www.eu-egee.org/, 2005.

[30] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": temperature-aware workload placement in data centers," in *Proceedings of the 2005 Annual Conference on USENIX Annual Technical Conference, Anaheim, CA*, 2005.

[31] D. Feitelson, "Parallel workloads archive," http://www.cs.huji.ac.il/labs/parallel/workload.

[32] D. Irwin, L. Grit, and J. Chase, "Balancing risk and reward in a market-based task service," in *Proc. of the 13th IEEE International Symposium on High Performance Distributed Computing*, Honolulu, USA, 2004.

[33] A. W. Mu'alem and D. G. Feitelson, "Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling," vol. 12, no. 6, pp. 529–543, Jun. 2001.