

An Adaptive Policy to Minimize Energy and SLA Violations of Parallel Jobs on the Cloud

Javier Celaya
Universidad de Zaragoza, Spain

Rizos Sakellariou
The University of Manchester, UK

Abstract—Energy consumption for Cloud providers and data centers is a major problem. Dynamic Power Management is a common solution to this problem, switching off and on idle servers as needed. However, failing to predict the impact of switching costs may adversely affect energy and/or SLA violations. This paper contributes a policy that adaptively decides when to switch servers on and off under a workload of parallel jobs. Its objective is to minimize both the energy consumption and the number of SLA violations. Experimental results using CloudSim show that our proactive policy strikes a good balance between consumed energy and the number of SLA violations and compares favorably with other policies from the literature.

Keywords—energy; SLAs; parallel jobs;

I. INTRODUCTION

This short paper contributes a policy that adaptively decides when to switch servers on and off under a workload of parallel jobs. These jobs consist of several, tightly coupled or interdependent tasks that must run in parallel and have SLA requirements in the form of duration, release time and deadline by which they should end. Our policy minimizes energy consumption and the number of SLA violations *proactively* by periodically predicting the suitable extra capacity needed at any time. Experimental results using CloudSim show that the proactive policy strikes a good balance between consumed energy and the number of SLA violations, and compares favorably with the policies in [1].

II. PROBLEM DESCRIPTION

We consider the scheduling of parallel jobs with SLA requirements among the servers of a data center in an energy-efficient way. Each job A_i has the following SLA requirements: (i) A release time r_i , at which this job is submitted, ready to be scheduled and executed. (ii) A deadline D_i . The job must finish before the deadline otherwise there is an SLA violation. (iii) A duration d_i in seconds. (iv) A number of tasks n_i that must run in parallel, each one with exclusive access to its processor core. An important property of an SLA-bound job is its *tightness* [2] $T_i = d_i/(D_i - r_i)$. The nearer this value gets to one, the less flexibility the scheduler has to finish the job before its deadline.

As workload, we consider a constant flow of parallel jobs that is fed into the scheduler. The scheduler only discovers about job A_i at r_i . We generate this workload using the

generator and model in [2]. We have modified the release time model to produce four test workloads with the following characteristics: (i) *slow*: slowly varying job rate. (ii) *burst*: alternating short periods of low and high job rates. (iii) *spike*: long periods of low job rate with seldom short peaks of very high job rate. (iv) *plateau*: alternating long periods of low and high job rates, with quick transitions between them.

We assume a typical data center, with a set of identical, multicore servers. Servers can be on or off, with setup and shutdown times that the scheduler must consider. We use a power model based on [3]. The power consumed at time t by server j is $P_j(t) = (P_{max} - P_{min})U_j(t) + P_{min}$, where $U_j(t)$ is its utilization, as the fraction of cores in use over the total. P_{min} and P_{max} are the power consumed by a server when its utilization is 0% and 100%, respectively.

III. ENERGY-AWARE SCHEDULING OF PARALLEL JOBS

We consider a centralized architecture with online scheduling, as depicted in Figure 1. Periodically, a scheduler assigns the tasks of the submitted jobs to the processor core queues. The figure shows the four actions performed during such a reschedule operation. The last step is to decide which additional servers to switch on and off. We use three well-known policies as reference, as described in [1]: (i) *Always-on policy*, that keeps on as many servers as needed to cope with the highest job rate. (ii) *Reactive policy*, that switches servers on and off as soon as they are needed or idle, respectively. Due to switching costs, extra capacity is allocated by using a *number of spare servers* (which are kept on, in addition to the ones that are actually working) and a *delay before shutdown* (do not switch servers off immediately, in case

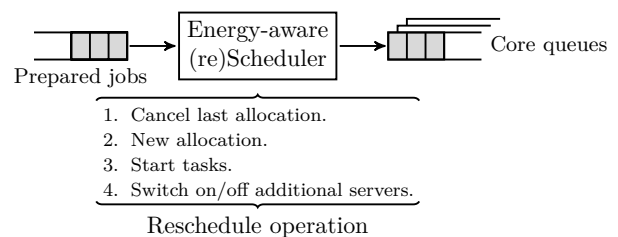


Figure 1. Scheduler architecture and reschedule operation.

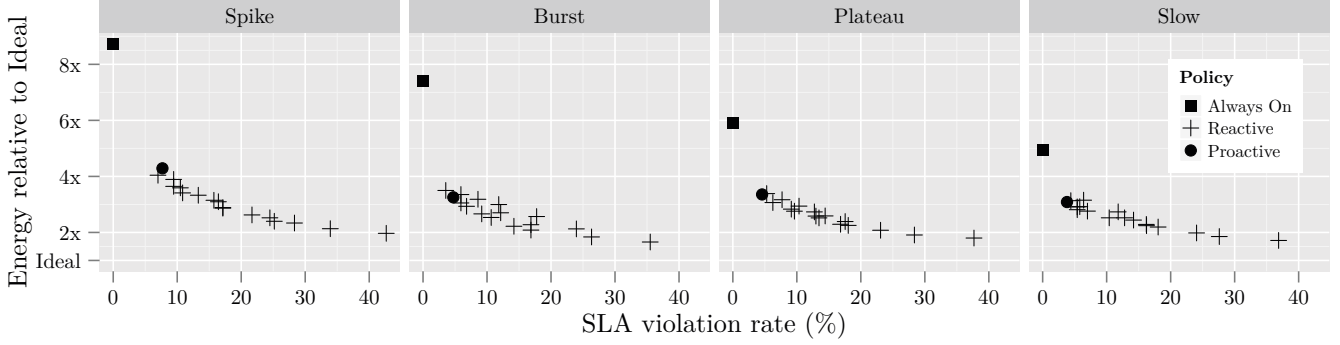


Figure 2. Energy relative to ideal against SLA violation rate, by workload type.

new jobs arrive shortly after). (iii) *Ideal policy*, which does not miss deadlines and uses just as much energy as needed.

The *proactive* (adaptive) policy proposed in this paper is based on the reactive one, but automatically adapts to sudden changes in the workload by periodically *predicting the next suitable extra capacity*. It uses two Linear Regression Models (LRM), for the number of spare servers and the delay before shutdown. They use eight features, measured in periods of 120 seconds: (i) Mean and standard deviation of the tightness in the last period. (ii) Mean and standard deviation of the task duration in the last period. (iii) Job rates in the last two periods. (iv) Task rates in the last two periods.

In the last step of the reschedule operation, the proactive policy does the following: 1. Train the LRMs with the last observations. 2. Take new measures of the features from historical information. 3. Use the LRMs to predict the suitable extra capacity for the current situation, and apply the result. 4. Add the new measures to the set of last observations. The LRMs are continuously trained with the observations of the last reschedule operation, but for testing purposes, we trained the LRMs in advance.

IV. RESULTS

To evaluate the performance of the reference policies and our contribution, we have used CloudSim and its extension for power and energy models [4]. We have simulated the always-on, ideal, reactive and proactive policies with all four test workloads, for 16 hours on 16 dual-core servers. Besides, we have simulated the reactive policy with 16 different combinations of its parameters, as defined in Section III. The result of each simulation is the percentage of jobs that violated their SLA, and the amount of energy consumed during all the simulation.

Figure 2 plots the energy consumption ratio, relative to the ideal policy, against the percentage of SLA violations, of the always-on (filled square), the proactive (filled circle) and the reactive policies (cross, one for each different combination of number of spare servers and delay before shutdown). This plot highlights that our proposed proactive policy is able to achieve a good balance between consumed energy and SLA

commitment. All this, without the need for manual tuning of the spare servers or the delay before shutdown parameters. Instead, to obtain a similar result with the reactive policy, one would need to test different combinations of these two parameters by trial and error.

V. CONCLUSION

This paper presented a policy for switching on and off servers in a data center. By predicting the extra capacity needed to cope with sudden changes in the workload, it achieves a balance between estimated energy consumption and SLA violations. Through simulation, we have compared the performance of our proposal against other two well-known policies, the reactive and always-on policies. Whereas the reactive policy must be manually tuned by trial and error to obtain the desired results, our policy adapts to the characteristics of the current workload.

Acknowledgments: The first author is partially supported by project TIN2013-40809-R of the Spanish government. Most of this work was carried out while he was visiting the University of Manchester. Partial support from the FP7 SUCRE project is also acknowledged.

REFERENCES

- [1] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. Kozuch, "AutoScale: Dynamic, Robust Capacity Management for Multi-Tier Data Centers," *ACM Transactions on Computer Systems*, vol. 30, no. 4, pp. 1–26, Nov. 2012.
- [2] V. Yarmolenko and R. Sakellariou, "An evaluation of heuristics for SLA based parallel job scheduling," in *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. IEEE, 2006, p. 8.
- [3] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2008.
- [4] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.