Shutdown Policies with Power Capping for Large Scale Computing Systems

Anne Benoit, Laurent Lefèvre, Anne-Cécile Orgerie, Issam Raïs

Shutdown policies with power capping for large scale computing systems

Anne Benoit[1], Laurent Lefèvre[1], Anne-Cécile Orgerie[2], and Issam Raïs[1]

[1] Univ. Lyon, Inria, CNRS, ENS de Lyon, Univ. Claude-Bernard Lyon 1, LIP
[2] CNRS, IRISA, Rennes, France

Abstract Large scale distributed systems are expected to consume huge amounts of energy. To solve this issue, shutdown policies constitute an appealing approach able to dynamically adapt the resource set to the actual workload. However, multiple constraints have to be taken into account for such policies to be applied on real infrastructures, in particular the time and energy cost of shutting down and waking up nodes, and power capping to avoid disruption of the system. In this paper, we propose models translating these various constraints into different shutdown policies that can be combined. Our models are validated through simulations on real workload traces and power measurements on real testbeds.[3]

Keywords: Large scale distributed systems, energy models, shutdown policies, simulations.

1 Introduction

Reducing the energy consumption of large scale distributed systems (high performance computing centers, networks, datacenters) is a mandatory step to address, in order to build a sustainable digital society. Since more than a decade, several technological solutions have been made available by systems designers to help reducing power, like shutdown and slowdown approaches. The first and most explored solution consists in shutting down and waking up some resources depending on platform usage. In this paper, the question on how resource providers and managers can be helped to validate their constraints while reducing the energy consumption using only the shutdown and wake-up of large amount of resources is addressed.

Resource providers and managers can be human who are responsible of the administration of large supercomputers, but they can also be software components that deal with resources (schedulers, resource management frameworks, etc.). Nowadays, hardware components of a datacenter or supercomputer (servers, network switches, data storage, etc.) are not yet energy proportional. In fact,

[3] This work is integrated and supported by the ELCI project, a French FSN ("Fond pour la Société Numérique ") project that associates academic and industrial partners to design and provide software environment for high performance computing.

the static part (i.e., the part that does not vary with workload) of the energy consumed for example by computing units, represents a high part of the overall energy consumed by the node. Therefore, shutting unused nodes or routers, that are idle and not expected to be used in a predicted duration, could lead to non negligible energy savings. This paper focuses on shutting down and waking up any kind of resources like servers, network devices, memory banks, cores, etc. For clarity's sake, here, the proposed models and validations focus on servers (called *nodes*).

Off-the-shelf software eco-systems are nowadays integrating (mainly basic) shutdown policies. Data center resource managers propose techniques or hooks to configure such capabilities. For example, Slurm [16], an open-source cluster management system, introduces a *SuspendTime*[4] that represents the minimum idle time after which it allows the node to be switched off. Then, the resource manager is responsible for deciding when to switch on and off servers. It takes decisions either based on pre-determined policy [16], on workload predictions [8], on queuing models [5] or on control theory approach [15].

Overall, shutdown seems to be an interesting leverage to save energy (referred to as *OnOff leverage*). But this technique cannot be applied at large scale if no constraint is respected on the target system. This is especially true if the resource providers take into account several types of constraints, such as the cost of shutdown and wake-up (in time and energy), or power-capping constraints imposed to the whole system. In particular, shutting down too many nodes could cause the power consumption to be under the minimum power capping decided with the electricity provider. Likewise, if too many nodes are waked-up, and if providers take into account the energy consumed during shutdown and wake-up sequences (which is far from being free), limits fixed by the electricity provider can be greatly exceeded. If providers do not take into account such constraints, they can put into danger machines composing the studied computing facility.

In this paper, we propose several models of shutdown that can be used under actual and future supercomputer constraints, and that takes into account the impact of shutting down and waking up nodes (time, power and energy) and the Idle and Off states observed after such actions as they impact the power usage. Our formalization allows for a mono or combined usage of models in order to help resource managers and providers respect several constraints at the same time. Several shutdown models that can be handled by resource providers and that deal with infrastructure constraints are explored:

- The *basic models* allow comparisons with several related works where shutting down and waking up nodes can either be free and immediate, or not allowed.
- The *sequence-aware models* account for the cost of shutting down or waking up nodes, in terms of time or energy.
- The *power-capping models* aim at respecting power capping requirements.

[4] http://slurm.schedmd.com/power_save.html

The models are used as follows: knowing that there is an idle interval of length T_{gap} on a given node, the model decides whether the node should be shut down, given the enforced constraints.

The paper is organized as follows. Section 2 presents the modeling of the various shutdown (OnOff) policies for basic models, sequence-aware models, and finally models dealing with power-capping. It also deals with the usage and combination of these models. The experimental setup is described in Section 3 and experimental results are analyzed in Section 4. Section 5 presents related work on shutdown techniques for large scale systems. Section 6 concludes and presents future work.

2 Modeling shutdown policies

This section presents our characterization of the impact of shutting down and waking-up a node in terms of time and power consumption. It also introduces models acting on the OnOff leverage.

2.1 Model inputs

To monitor nodes' wake-up and shutdown sequences, an external power monitoring allowing us to trace power consumption of nodes is used. It has a rate of one power value per second. The sequences have been monitored to detect when every event happened. For the wake-up sequence, unfortunately, no information could be extracted between BIOS (Basic Input Output System) bootstrap and GRUB (Grand Unified Bootloader) loading. The first monitorable event in this sequence is the Kernel launch; this is displayed on Figure 1, which shows how the power evolves with time during a monitored boot sequence on a node. The time where kernel starts has been recovered with the *dmesg* tool (which is a logging of what happened during the launch of the kernel). The INIT monitoring is made by modifying the runlevel scripts.

These monitored profiles are modeled by a sequence for each node. For node i, $Seq_i = \{(t_0; AvrgP_0), \ldots, (t_n; AvrgP_n)\}$ is the set of timestamps and average power consumption measurements of a wake-up (or Off→On sequence) or shutdown (or On→Off sequence), where t_0 and t_n represent the starting and ending time respectively of sequence Seq on node i. The length of the sequence is therefore $t_n - t_0$, and at time-step t_k $(1 \leq k \leq n)$, $AvrgP_k$ is the average power consumption of node i.

2.2 Model definitions

Basic models. Two basic models are used by most papers in the literature: either the nodes are never shut down (No-OnOff model), or there is no cost (time, energy, power) to wake-up or shutdown a node (LB-ZeroCost-OnOff model: *Lower-Bound Zero-Cost OnOff Model*), making it very simple to shutdown a node (but very far from reality). In this context, the node consumes

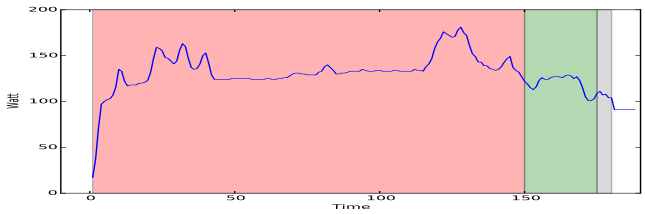

Figure 1: Monitored boot sequence of a node running Linux: BIOS-MBR-GRUB period in red; Kernel in green; Init in gray.

nothing when executing an On→Off or Off→On sequence. Thus, there is no cost nor time spent to switch state, and no power peak observed during the sequence. In this context, no influence could be derived from waking-up or shutting down nodes. This LB-ZeroCost-OnOff model therefore provides a theoretical upper bound on the gains that can be achieved by shutting down nodes.

Sequence-aware models. The sequence-aware models make sure that the sequence observed on a node or set of nodes during On→Off or Off→On sequences fits in time or are beneficial in energy. Therefore, information for every node composing the studied case needs to be recorded, in particular a record of the Off→On and On→Off sequences.

Time constrained. The first model, Seq-Aw-T (*Sequence-Aware-Time*), checks whether there is enough time to perform an On→Off sequence followed by an Off→On sequence on a node, given the available time when the node is idle. Let T_{gap} be the size of the "gap", i.e., the interval of idle time of the node. Then, Seq-Aw-T will allow the addition of sequences in this time slot if and only if $T_{\text{OnOff}} + T_{\text{OffOn}} \leq T_{\text{gap}}$, where T_{OnOff} (resp. T_{OffOn}) is the time spent by the node during an On→Off (resp. Off→On) sequence.

Energy constrained. The Seq-Aw-E model (*Sequence-Aware-Energy*) further refines Seq-Aw-T by checking whether changing the state of the node is beneficial in terms of energy. The minimum time T_s of the gap is now further constrained by the energy savings:

$$T_s = \max\left(T_{\text{OnOff}} + T_{\text{OffOn}}, \frac{E_{\text{OnOff}} + E_{\text{OffOn}} - P_{\text{off}}(T_{\text{OnOff}} + T_{\text{OffOn}})}{P_{\text{idle}} - P_{\text{off}}}\right),$$

where:

- P_{idle} is the power consumption when the node is in the Idle state (unused, but powered on);
- P_{off} is the power consumption when the node is switched off (typically not null and lower than P_{idle});
- E_{OnOff} is the energy consumed during the On→Off sequence;
- E_{OffOn} is the energy consumed during the Off→On sequence.

The first term states, as for SEQ-AW-T, that at least a time $T_{\text{OnOff}} + T_{\text{OffOn}}$ is needed to shutdown the node (and to wake it up) during the idle interval. The second term ensures that there will be gains in energy: the energy saved by running at P_{off} rather than P_{idle} is $T_s(P_{\text{idle}} - P_{\text{off}})$ during the interval, but the additional energy due to the On→Off and Off→On sequences is $E_{\text{OnOff}} + E_{\text{OffOn}} - P_{\text{off}}(T_{\text{OnOff}} + T_{\text{OffOn}})$. Therefore, if $T_{\text{gap}} > T_s$, then it is beneficial to shutdown (at the beginning of the gap) and to wake up (at the end of the gap) the node, in terms of energy consumption.

Power-capping-aware models. The POWER-CAP model (*Power-Capping-Aware*) aims at maintaining an average power budget and guaranteeing minimal or maximal electrical power consumption. Indeed, shutting down and waking up components could lead to hard power-capping disruptions. Such actions energetically stress the node, whether it is in an upper or lower way. Here, information about the power capping that should be respected is provided.

A minimum power capping (PC_Min) represents a constraint set by the electrical provider, and is defined by providing a lower bound on power. A maximum power capping (PC_Max) represents power limit fixed by the electrical provider, and it is defined by an upper limit on power. These minimum and maximum power capping values may be a function of the time, i.e., the requirements may change in time.

We introduce the function $PowerSum(X, t)$, which returns the sum of the power consumed by nodes in set X at time-step t. We denote by ALL the set of all nodes. Nodes in X can be shut down or waked up only if $PowerSum(ALL, t) \geq PC\_Min$ and $PowerSum(ALL) \leq PC\_Max$ at all time during the sequence.

2.3 Model usages

Several models have been derived, assuming that local knowledge about the node reservations is available, i.e., the current state of each node: *On* (*Working* or *Idle*) or *Off*. At current time-step T_c, a model aims at deciding whether this node can be shut down and then waked up, while respecting the constraints of the system. If the node is neither in a *Working* or *Off* state, it is in an *Idle* interval of length T_{gap}. Therefore, an entity giving advice on changing the state of a node (or set of nodes) is provided, making sure that the overall system responds to the described constraints.

3 Experimental setup

To instantiate our models in various configurations, we developed a simulator capable of replaying a real datacenter trace, with real node and job calibrations (time, power, energy). Grid'5000 [1], a large-scale and versatile testbed for experiment-driven research in all areas of computer science, was used as a testbed. Grid'5000 deploys clusters linked with dedicated high performance networks in several cities in France (Lille, Nancy, Sophia, Lyon, Nantes, Rennes,

Grenoble). On the Lyon site, the energy consumption of every node from all available clusters (Orion, Taurus, Sagittaire) is monitored through a dedicated wattmeter, exposing one power measurement per second with a 0.125 Watts accuracy. Therefore, detailed traces concerning the energy consumption of jobs at any time step are available. An average power consumption of each job was obtained. Thanks to these traces, realistic replays of jobs with their corresponding energy consumption is performed. Taurus nodes were monitored to calibrate in time, energy and power the Off→On and On→Off sequences, as explained in Section 2 (see Table 1).

Taurus	Features		Parameters	Values
Server model	Dell PowerEdge R720		E_{OffOn} (Joules)	23,683
CPU model	Intel Xeon E5-2630		T_{OffOn} (seconds)	182
Number of CPU	2		E_{OnOff} (Joules)	1,655
Cores per CPU	6		T_{OnOff} (seconds)	15
Memory (GB)	32		P_{idle} (Watts)	91
Storage (GB)	2 x 300 (HDD)		P_{off} (Watts)	8
T_s (seconds)	286.29			

Table 1: Calibration nodes' characteristics and energy parameters for On→Off and Off→On sequences (average on 50 experimental measurements).

For our evaluation, two real workload traces of the Grid'5000 Lyon site were extracted. Traces only contains nodes that received jobs during the chosen period to avoid doping of the energy savings results with node that will potentially always be shutdown. The first one runs from October 24, 2016 at 7 pm to November 1, 2016 at 8 am, thus representing approximately one week of resource utilization on this site. During this period, the number of used nodes is 76. The second one runs from October 29, 2016 at 7 pm to December 1, 2016 at 3 am, thus representing approximately one month of resource utilization. The number of used nodes is 69. Nodes are considered homogeneous concerning the P_{idle}, P_{off} characteristics, Off→On and On→Off sequences. This study is focused on shutting down nodes, scheduled jobs are considered fixed. Moreover, we only add sequences if possible, thus not impacting nor overlaying scheduled jobs. For the week trace, there are $1,768$ jobs with an average power consumption of 167.5 Watts and an average job size of $13,776.96$ seconds (approximately four hours). For the month trace, there are $5,505$ jobs with an average power consumption of 166.5 Watts and an average job size of $14,203.42$ seconds (approximately four hours). Yet, these traces exhibit high workload variations as shown on Figures 2 and 3, where the upper curve corresponds to the No-OnOff model.

4 Experimental validation

This section presents the simulation results for the proposed setup as proposed in Section 3. All graphs represent a trace replay of the application of one or multiple

combined models of Section 2. Table 2 presents the energy consumption in Joules of all models in the figures included in this section. This section presents results of the simulator on the extracted traces with calibration of Taurus nodes while applying previously defined models. The first trace, which spreads over one week, will be used to extensively study behavior of models. The second trace, over one month, will validate observations and model tendencies at a larger scale.

For the experiments involving Power-Cap, Seq-Aw-T is always combined with this model to ensure that the node is in the *On* state when a scheduled job starts. Shutting down a node is allowed only when it can be waked up before the end of the idle time interval, to ensure that no job is delayed.

Model	Total energy consumed	# (On→Off & Off→On)	% Saved
Grid'5000 trace, 1 week			
No-OnOff	6,083,698,688	0	0,0
LB-ZeroCost-OnOff	3,983,408,384	1794	34.52
Seq-Aw-T	4,015,736,064	964	33.99
Seq-Aw-E	4,015,201,024	844	34.00
Power-Cap 2000 min	4,401,067,520	855	27.65
Power-Cap 4000 min	4,593,668,096	761	24.49
Power-Cap 6000 min	5,059,857,408	617	16.82
Grid'5000 trace, 1 month			
No-OnOff	22,866,315,264	0	0.0
LB-ZeroCost-OnOff	12,935,132,160	5,559	43.43
Seq-Aw-T	13,038,270,464	3,819	42.9804
Seq-Aw-E	13,037,558,784	3,605	42.9835
Power-Cap 4000 min	17,864,194,048	2,376	21.87

Table 2: Trace replay's energy consumption (in Joules), number of added double sequences (On→Off and Off→On), and percentage of energy saved compared to No-OnOff replay.

4.1 Sequence-aware models: Seq-Aw-T and Seq-Aw-E

Figures 2 and 3 show results of the different models, namely No-OnOff, Seq-Aw-T, Seq-Aw-E, and LB-ZeroCost-OnOff on, respectively, one week and one month traces. Between the two sequence-aware models, minor differences can be witnessed on the complete replay. For example on Figure 2, for October 31 at 4:40 am, Seq-Aw-T allows more Off→On sequences to be scheduled. Both of these models lead to major energy savings, respectively 34.00% and 39.9% of energy savings on the one week trace compared to No-OnOff, as shown in Table 2. In comparison with the No-OnOff trace replay, major power peaks are witnessed because of the application of these models. For example in the one week trace, for October 31 at 12:00pm, after a peak of work around $12,000W$, a very low peak is witnessed under $1,000W$. Such a behavior could be witnessed

in an amplified way on Figure 3 for example around November 12, November 3 and November 21. Such behaviors could lead to abrupt thermal changes and thus to hot and cool spots, so to possible deterioration of the nodes.

LB-ZeroCost-OnOff, the model with immediate On→Off with zero cost is also presented for the comparison. There is no significant difference in energy consumption observed when the cost of On→Off and Off→On sections are accurately described. However, the number of On→Off that are effectively triggered is significantly lower, implying that LB-ZeroCost-OnOff allows the addition of sequences when T_{gap} is smaller that $T_{\text{OnOff}} + T_{\text{OffOn}}$.

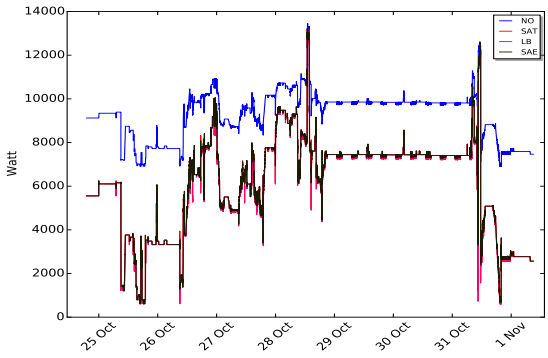

Figure 2: Replay for a week of Grid'5000 trace with No-OnOff (NO) and with Seq-Aw-T (SAT), Seq-Aw-E (SAE), LB-ZeroCost-OnOff (LB) models.

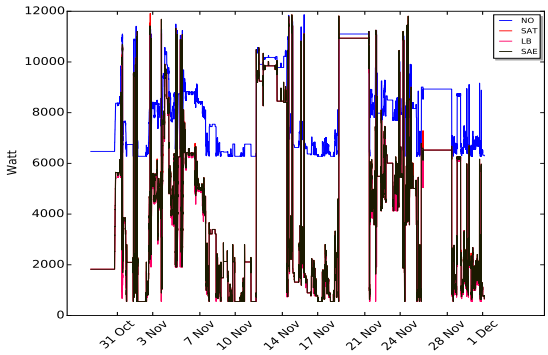

Figure 3: Replay for a month of Grid'5000 trace with No-OnOff (NO) and with Seq-Aw-T (SAT), Seq-Aw-E (SAE), LB-ZeroCost-OnOff (LB) models.

4.2 Power-capping model

Next, a focus on the POWER-CAP model is made. A maximum and a minimum power cap is set throughout the simulation. Modulation of the minimum power cap to see how it acts with the trace replay is then simulated. Recall that since only the OnOff leverage is evaluated, scheduled jobs are fixed. Therefore, maximum power cap does not vary because it highly depends on jobs and also because the difference between P_{idle} and P_{off} is more important than the difference between the peaks witnessed during the Off→On or On→Off sequences and P_{idle}.

Figure 4 shows results of NO-ONOFF, SEQ-AW-T and POWER-CAP with various PC_Min scenarios (2000, 4000 and 6000) during the one week trace. Even with the highest minimum power cap, here $6,000W$, important energy savings (around 16.82 % compared to NO-ONOFF) are made. The stratified power usage for every respected power cap was expected. In fact, a lower power cap permits more sequences to be scheduled and thus, more energy savings. The lowest cap constraint ($2,000W$) shows that a minimum power capping could be always respected and would still have a close to minimum consumption. Finally, Figure 5 shows results for larger scale replay (one month) on NO-ONOFF, SEQ-AW-T and POWER-CAP with $4,000$ as PC_Min, and leads to the same conclusions.

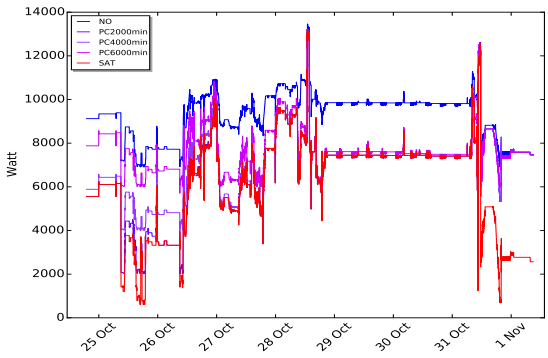

Figure 4: Replay for a week of Grid'5000 trace with NO-ONOFF (NO) and with POWER-CAP (with $PC\_Min = 2000, 4000, 6000$),SEQ-AW-T (SAT) models.

5 Related work

Pioneering work on studying the energy-related impacts of shutdown techniques started in 2001 [2,11]. These early efforts did not consider any transition cost for switching between *Idle* and *Off* states, but they nonetheless showed the potential impact of such techniques. Yet, aggressive shutdown policies are not always the best solution to save energy [9].

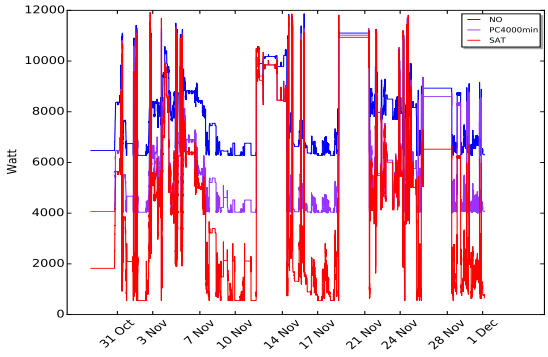

Figure 5: Replay for a month of Grid'5000 trace with No-OnOff (NO) and with Power-Cap (with $PC\_Min = 4000$), Seq-Aw-T (SAT) models.

Demaine et al. examine the power minimization problem where the objective is to minimize the total transition costs plus the total time spent in the active state [3]. They develop a $(1+2\alpha)$-approximation algorithm, with α the transition cost. However, the parameters considered for this transition cost highly vary across the literature. Gandhi et al. take into account the energy cost of waking up servers (no shutting down cost as it is estimated to be negligible in comparison with the waking-up cost) [5]. This energy cost is assumed to be equal to the transition time multiplied by the power consumption while in the *Idle* state. Lin et al. take into account the energy used for the transition, the delay in migrating connections or data, the increased wear-and-tear on the servers, and the risk associated with server toggling [7]. We go one step beyond and carefully assess the cost of shutting down or waking up a node, in terms of time, power and energy.

Moreover, supporting shutdown and wake-up of large amount of resources can be risky as it impacts the whole infrastructure of supercomputers, such as the cooling system [18]. Shutdown techniques can also be used for limiting the dark silicon effect, i.e., the under-utilization of the device integration capacity due to power and temperature effects [4]. This issue has lead to the introduction of user-specified, dynamic, hardware-enforced processor power bounds, as for the Intel's Sandy Bridge family of processors for instance [13]. At a data center level, it translates into power budgeting, where the total power budget is partitioned among the cooling and computing units, and where the cooling power has to be sufficient to extract the heat of the computing power. Given the computing power budget, Zhan et al. propose an optimal computing budgeting technique based on knapsack-solving algorithms to determine the power caps for the individual servers [17].

Shutdown policies are often combined with consolidation algorithms that gather the load on a restricted number of servers to favor the shutdown of the others. Employing either reactive or proactive scheduling options [6,10], consoli-

dation algorithms increase the energy gains brought by shutdown techniques at a cost of a trade-off with performance [14]. The rich diversity in power management techniques and levers can lead to substantial issues if they are not coordinated at the data center level [12]. In this paper, shutdown policies (i.e., when to shutdown) are studied, without combining them to scheduling algorithms and consolidation approaches in order to evaluate the impacts of such policies without interfering with the workload of real platforms and with the users' expected performance.

6 Conclusion

In this paper, the OnOff leverage is explored as a technique to save energy on large scale computing systems. While it is often assumed that nodes can change state at no cost, realistic scenarios are explored where several constraints (time and energy of changing the state of a node, global power capping on the platform) may prevent us from shutting down a node at a given time. This paper presents a formal definition of such models, targeting various scenarios on selected architectures.

A possible application of these models, either alone or combined, is provided through a set of simulations on real workload traces, showing the gain in energy that can be achieved, given the constraints on the platform, and providing clear guidelines about when a node can change state. Overall, the gain of the non-realistic model, where nodes are instantaneously shut down during an idle period, is very small over the sequence-aware model, that shutdowns a node only if there is time to wake it up again before the next computation (Seq-Aw-T), and accounts for the power consumption during the Off→On and On→Off sequences (Seq-Aw-E). On top of previous models, power-capping constraints (Power-Cap) could be enforced, thus reducing the number of added On-Off sequences, and hence leading to losses in energy, but fully respecting the imposed constraints.

We plan to further add several models by considering for example a cooling-aware model that accounts for the system temperature in order to avoid abrupt thermal changes (and thus hot and cool spots), provoked by changing the state of a large number of nodes. We also plan to deeply analyze combinations of shutdown models in order to jointly take into account more realistic constraints imposed to supercomputers.

References

1. Bolze, R., et al.: Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. International Journal of High Performance Computing Applications 20(4), 481–494 (2006), `https://hal.inria.fr/hal-00684943`
2. Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., Doyle, R.P.: Managing Energy and Server Resources in Hosting Centers. In: ACM Symposium on Operating Systems Principles (SOSP). pp. 103–116 (2001)

3. Demaine, E.D., Ghodsi, M., Hajiaghayi, M.T., Sayedi-Roshkhar, A.S., Zadimoghaddam, M.: Scheduling to Minimize Gaps and Power Consumption. In: ACM Symp. on Parallel Algorithms and Architectures (SPAA). pp. 46–54 (2007)
4. Esmaeilzadeh, H., Blem, E., St. Amant, R., Sankaralingam, K., Burger, D.: Power Limitations and Dark Silicon Challenge the Future of Multicore. ACM Transactions on Computer Systems (TOCS) 30(3), 11:1–11:27 (2012)
5. Gandhi, A., Gupta, V., Harchol-Balter, M., Kozuch, M.A.: Optimality analysis of energy-performance trade-off for server farm management. Performance Evaluation 67(11), 1155–1171 (2010)
6. Gmach, D., Rolia, J., Cherkasova, L., Kemper, A.: Resource Pool Management: Reactive Versus Proactive or Let's Be Friends. Computer Networks 53(17), 2905–2922 (2009)
7. Lin, M., Wierman, A., Andrew, L.L.H., Thereska, E.: Dynamic Right-sizing for Power-proportional Data Centers. IEEE/ACM Transactions on Networking (TON) 21(5), 1378–1391 (Oct 2013)
8. Orgerie, A.C., Lefèvre, L.: ERIDIS: Energy-Efficient Reservation Infrastructure for Large-scale Distributed Systems. Parallel Processing Letters 21(02), 133–154 (2011)
9. Orgerie, A.C., Lefèvre, L., Gelas, J.P.: Save Watts in Your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems. In: IEEE Int. Conf. on Parallel and Distributed Systems (ICPADS). pp. 171–178 (2008)
10. Pernici, B., et al.: Setting Energy Efficiency Goals in Data Centers: The GAMES Approach. In: Huusko, J., de Meer, H., Klingert, S., Somov, A. (eds.) Energy Efficient Data Centers, Lecture Notes in Computer Science, vol. 7396, pp. 1–12. Springer (2012)
11. Pinheiro, E., Bianchini, R., Carrera, E.V., Heath, T.: Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems. In: Workshop on Compilers and Operating Systems for Low Power. pp. 182–195 (2001)
12. Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., Zhu, X.: No "power" struggles: Coordinated multi-level power management for the data center. In: ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). pp. 48–59 (2008)
13. Rountree, B., Ahn, D.H., de Supinski, B.R., Lowenthal, D.K., Schulz, M.: Beyond DVFS: A First Look at Performance under a Hardware-Enforced Power Bound. In: IEEE International Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW). pp. 947–953 (May 2012)
14. Srikantaiah, S., Kansal, A., Zhao, F.: Energy aware consolidation for cloud computing. In: USENIX Conference on Power Aware Computing and Systems (HotPower). pp. 1–5 (2008)
15. Urgaonkar, R., Kozat, U.C., Igarashi, K., Neely, M.J.: Dynamic resource allocation and power management in virtualized data centers. In: IEEE Network Operations and Management Symposium (NOMS). pp. 479–486 (April 2010)
16. Yoo, A.B., Jette, M.A., Grondona, M.: International Workshop Job Scheduling Strategies for Parallel Processing (JSSPP), chap. SLURM: Simple Linux Utility for Resource Management, pp. 44–60. Springer (2003)
17. Zhan, X., Reda, S.: Techniques for energy-efficient power budgeting in data centers. In: ACM/EDAC/IEEE Design Automation Conference (DAC). pp. 1–7 (May 2013)
18. Zhang, W., Wen, Y., Wong, Y.W., Toh, K.C., Chen, C.H.: Towards Joint Optimization Over ICT and Cooling Systems in Data Centre: A Survey. IEEE Communications Surveys Tutorials 18(3), 1596–1616 (2016)