Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 337 (2018) 173–191

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

# Evaluation of Advanced Data Centre Power Management Strategies

Björn F. Postema[1,2] Boudewijn R. Haverkort[3]

*Design and Analysis of Communication Systems*
*University of Twente*
*Enschede, the Netherlands*

**Abstract**

In recent work, we proposed a new specification language for power management strategies as an extension to our AnyLogic-based simulation framework for the trade-off analysis of power and performance in data centres. In this paper, we study the quality of such advanced power management strategies based on both power and performance measurement data collected during system operation. These strategies take a wide variety of state variables into account. In order to ensure the quality of new strategies, they are studied for stability, efficiency, adaptability and robustness; these qualities will be formally defined. This paper presents an evaluation approach for these qualities for several power management strategies inspired by strategies presented in the literature (and extensions thereof). We show that the choice of power management strategy depends both on which qualities are given the highest priority and on the used state information. The new power management strategies show significant reductions in energy consumption in our case of up to 54% energy (compared to an "always on" strategy) for a typical data centre workload for a small 30-server cluster.

*Keywords:* power management, strategies, qualities, evaluation, stability, efficiency, adaptability, robustness, discrete-event simulation, agent-based simulation, data centre.

## 1 Introduction

One way to reduce overall energy consumption in data centres is *Power Management* (PM). PM allows to switch between power states of servers to reduce power consumption, while trying to keep the performance intact [5]. Power proportionality, i.e., power consumption is proportional to utilisation, has proven to be one of the three main areas of improvement on data centre energy-efficiency in the last years [4,14]. Since PM software and hardware has been improving, scaling back

[1] The work in this paper has been supported by the Dutch national NWO project Cooperative Networked Systems (CNS), as part of the program "Robust Design of Cyber-Physical Systems" (CPS), including the industrial partners Target Holding B.V. and Better.be.

[2] Email: b.f.postema@utwente.nl

[3] Email: b.r.h.m.haverkort@utwente.nl

https://doi.org/10.1016/j.entcs.2018.03.040

1571-0661/© 2018 The Author(s). Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

idle servers, as a consequence of no power proportionality, has become a reasonable practice nowadays.

The three PM categories, as stated in the widely used open standard *Advanced Configuration and Power Interface* (ACPI) [8], are: (i) Dynamic PM, e.g., sleep and hibernate; (ii) *Dynamic Voltage and Frequency Scaling* (DVFS), e.g., scaling frequencies and voltage of CPUs; and (iii) device PM, e.g., suspension of GPUs and hard disks. This paper discusses dynamic PM that focusses on the suspension of idle or underutilised servers [3].

Our new evaluation method is embedded in our recent work [13] that introduced a power management module and specification in a data centre simulation framework [12] using the multi-method simulation software AnyLogic [1]. The module extends power management functionality by proposing an interface to easily specify multiple strategies. Moreover, such a specification allows to use many state information variables with regard to traffic, system service, power, performance and thermodynamics, to formulate all kinds of more advanced power management strategies/policies that could lead to significant improvements in energy-efficiency, while other Service-Level Agreement (SLAs) demands are met.

This paper uses this specification method to study multiple power management strategies and proposes a set of metrics to evaluate the quality of advanced dynamic PM strategies simulated in our framework. We study PM strategies for efficiency and stability including minor variations (robustness) and the impact of adapted workloads (adaptability); these quality measures will be formally defined.

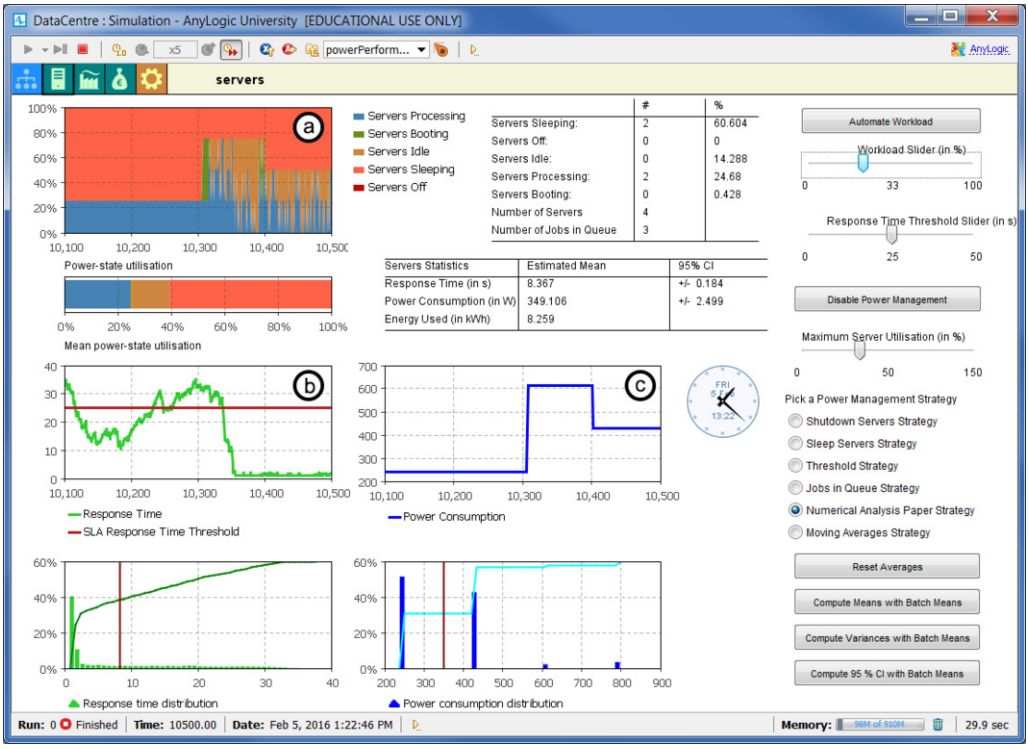
The paper is further organised as follows. An overview of the approach is provided in Section 2. Section 3 describe strategy qualities that are helpful to find the best and most suitable strategy for a given data centre configuration. Section 4 shows the evaluation in action with a typical job and data centre configuration and several interesting power management strategies. Conclusions and future work are provided in Section 5.

## 2 Overall Approach

Our approach to analysis of dynamic PM strategies is subdivided into three steps:

i. combine job and data centre *characteristics* to form an overall model;
ii. structurally describe a dynamic PM strategy using a *language*;
iii. *evaluate* dynamic PM strategy for relevant power and performance metrics.

Our first step is attained with the aid of an existing data centre simulation framework [12]. Section 2.1 describes this framework and its features, followed by Section 2.2, that elaborates a dynamic PM extension for this framework that allows for control of power states of servers strategically using various observable quantities. Secondly, an existing specification of dynamic PM strategies using various state information variables within this simulation framework, which is elaborated in Section 2.3. Section 2.4 describes an example strategy using the specification as illustration and serves as a running example in the section that follows. The final step of this




| | # | % |
|---|---|---|
| Servers Sleeping: | 2 | 60.604 |
| Servers Off: | 0 | 0 |
| Servers Idle: | 0 | 14.288 |
| Servers Processing: | 2 | 24.68 |
| Servers Booting: | 0 | 0.428 |
| Number of Servers | 4 | |
| Number of Jobs in Queue | 3 | |

| Servers Statistics | Estimated Mean | 95% CI |
|---|---|---|
| Response Time (in s) | 8.367 | +/- 0.184 |
| Power Consumption (in W) | 349.106 | +/- 2.499 |
| Energy Used (in kWh) | 8.259 | |

Fig. 1. The AnyLogic dashboard

approach is in the remainder of the paper.

### 2.1 General Description of the Simulator

In [12], a simulation framework has been proposed that allows for the analysis of power and performance trade-offs in data centres to save energy via power management. High-level simulation models allow us to estimate data centre power consumption and performance. The framework is developed in AnyLogic, that allows implementation of and cooperation between a combination of discrete-event and agent-based models. The framework features an intuitive dashboard to actively control and obtain insight during each simulation run, as illustrated in Figure 1. As can be seen in this figure, transient behaviour can be analysed for (a) *power-state utilisation*, (b) *response times* and (c) *power consumption*, and steady-state behaviour is depicted with averages of these three in automatically updated tables. Additionally, AnyLogic has the option for a parameter variation experiment that allows for parallel computation of multiple simulation runs and is often better for rapid computation of steady state behaviour.

### 2.2 Dynamic Power Management

Dynamic PM switches between global/sleep power states to reduce energy consumed while performance is kept intact (e.g. put underutilised servers to sleep). In order to maintain good performance, strategic decisions need to be made based on various

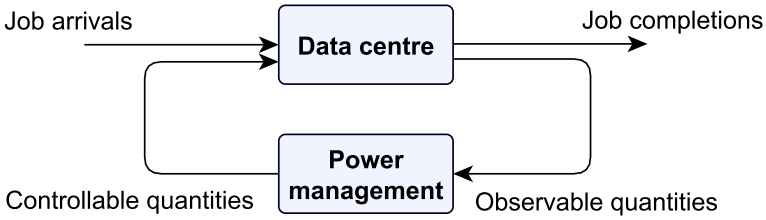



Fig. 2. System overview

state information variables. Figure 2 shows an overview of a data centre equipped with power management.

The information of the data centre offered to the power management module are the observable quantities of the system. With a growing number of sensors that collect data, much more information can be used in decision-making: (i) *power state utilisation* (PU) describes the fraction of time spent in a particular power state; (ii) *power consumption* (PC) describes the power consumed by servers and the data centre infrastructure (in Watt), which is related to the *expected power consumption* (E[P]) of all servers and the *expected energy consumption* (E[E]) by all servers over the duration of the simulation; (iii) *response time* (RT) is a measure of performance, that indicates the total time a job takes from a request of a user to enter a job to the system, to the completion of that job in the system, which is related to the *expected response time* (E[R]) of a job; (iv) *temperature* (TM) indicates the temperature of the servers (in degrees Celsius); (v) *traffic* indicate variables concerning the arrivals of jobs to the system; and (vi) *system service* indicates variables concerning the service of jobs in the system.

The observable quantities allow us with some computation to control quantities

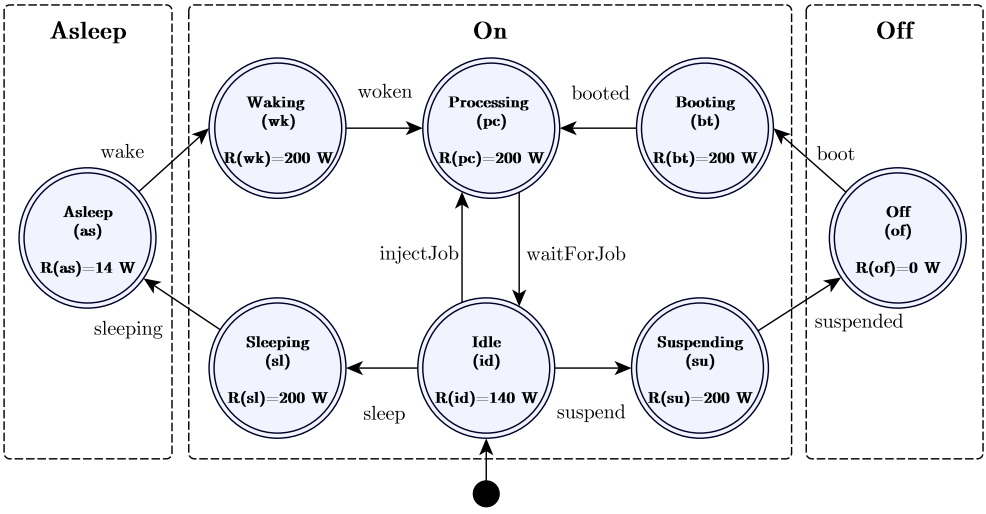



Fig. 3. Model for switching between three global power states `asleep`, `on` and `off` with a label and in each state with abbreviation $i$ an indication of the power consumed $R(i)$

of the data centre. The controllable quantities taken into account are as follows: (i) *power state switching* is a power management feature, that allows to switch between power states for particular servers; and (ii) *job scheduling* allows the distribution of workload among its service units, which has great effect on the quality of a power management strategy.

Figure 3 shows a *deterministic finite automaton* (DFA) that is used in the framework to switch between three global power states inside each server (as can be found in the ACPI open standard [8]). All state transitions in this model are invoked by either dynamic PM or by jobs entering or leaving the system. Combining the time spent in each state with the per-state power consumptions (rewards) allows for the computation of (average) power consumption and energy consumption. There are three important effects that occur when switching power states: (a) job processing suspends/continues, (b) transition from one power state to another takes time and consumes power, and (c) power consumption decreases/increases. The system is *on* in all states of the model except for the states `of` (*off*) and `as` (*asleep*). Note that the time spent in the states `sl` (*sleeping*), `id` (*idle*), `wk` (*waking*), `bt` (*booting*) and `su` (*suspending*) is considered overhead and, therefore, should be minimised.

So, formally the DFA (introduced in [9]) for switching between power states consists of the 5-tuple $M = (Q, \sum, \delta, q_0, F)$, where $Q = \{\texttt{as}, \texttt{wk}, \texttt{pc}, \texttt{bt}, \texttt{sl}, \texttt{id}, \texttt{su}, \texttt{of}\}$, $\sum = \{\texttt{waitForJob}, \texttt{injectJob}, \texttt{wake}, \texttt{woken}, \texttt{sleep}, \texttt{sleeping}, \texttt{boot}, \texttt{booted}, \texttt{suspend}, \texttt{suspended}\}$, $\delta : Q \times \sum \to Q$ (cf. all transitions in Figure 3), $q_0 = \{\texttt{Idle}\}$, $F = Q$. Additionally, the state rewards is defined as a function $R : Q \to \mathbb{R}$. For this DFA, the state rewards are depicted in the figure as well.

Figure 4 shows the dispatcher that is used in the framework to schedule all the jobs to one of the $M$ servers using *Shortest Queue Next* (SQN). Each server comprises a $G|G|1|\infty|\infty$ queue with a FIFO buffer. As a consequence of power management, the number $M$ of servers available for handling jobs varies over time.

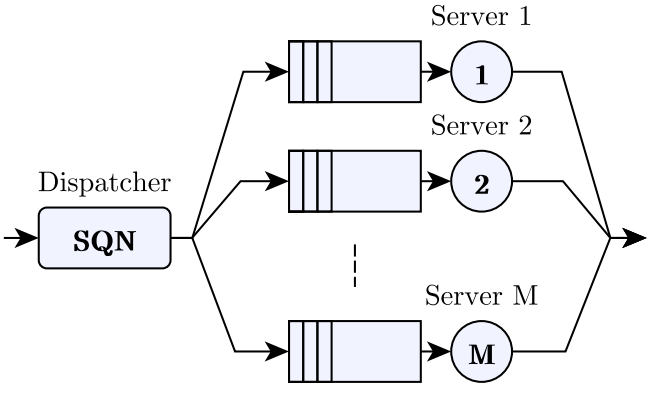



Fig. 4. A dispatcher schedules jobs to the queues of the servers 1 to $M$ using *Shortest Queue Next* (SQN)

### 2.3 Strategy Specification

Dynamic power management strategies describe a high-level plan to achieve certain goals while operating, e.g., reasonable performance, stable power consumption or evenly distributed temperature over all servers. This plan describes when servers are to switch power states based on the observed quantities. In order to structurally describe a dynamic PM strategy a specification is used.

A full description of the specification language is as given in [13]. The specification allows us to define a PM strategy $\Theta$ using a 3-tuple that contains global power states G (e.g., asleep, on and off), global-level satisfiers $\Phi_{\mathcal{S}}$ and server-level constraints $\Phi_{\mathcal{C}}(s)$, where $s$ is a server. These satisfiers and constraints are part of a two-step approach to decide if service units need to switch global power states. In the first step, satisfiers determine if certain goals/thresholds are met globally, e.g., response times determined with a moving average exceeds one second. In the second step, the constraints check all eligible servers for server-level constraints, e.g., server temperatures exceeds 30 °C. This procedure repeats every $r$ seconds, or can be triggered by events relevant for the observed variables.

Strategic power management decisions can be made using instantaneous (*ins*) observed values. As a consequence of stochastic workloads, however, such instantaneous decisions may lead to overly "aggressive" power state switching. To prevent such undesired behaviour, it is often better to use derivatives of the observed values, e.g., steady-state averages with batch means method (*savg*), or (exponentially) moving window averages (*eavg*/*mavg*).

Useful definitions for this strategy specification are included in Appendix A.

### 2.4 Example Strategy

For illustrative purposes, we give here an example power management strategy, denoted $\Theta^{que}$, that ensures a good power and performance with the aid of a threshold $q$ by waking and sleeping servers based on queue size (QS) observations. Use of this queue size threshold aims to reduce the (expected) waiting time, and thus the overall (expected) response time. For comparison purposes, the queue sizes threshold is varied between 100 and 1500, with stepsize 100. The satisfier formulas $\Phi_{\mathcal{S}}^{que}$ for this queue size threshold strategy with usable global power states $\mathrm{G}^{que} = (as, on)$ is as follows:

$$\Phi_{\mathcal{S}}^{que} = \begin{pmatrix} \phi_{\mathcal{S}}^{as} := (\mathrm{QS} \leq q) \\ \phi_{\mathcal{S}}^{on} := (\mathrm{QS} > q) \end{pmatrix}, \tag{1}$$

where $q \in \{100 \cdot i \mid 1 \leq i \leq 15, i \in \mathbb{N}\}$.

This strategy has server constraints $\Phi_{\mathcal{C}}^{que}(s)$ that only allows servers to sleep when the queue of that server has no jobs and servers are only woken when these are actually in power state (PS) asleep, as follows:

$$\Phi_{\mathcal{C}}^{que}(s) = \begin{pmatrix} \phi_{\mathcal{C}}^{as}(s) := (\mathrm{QS}(s) = 0) \\ \phi_{\mathcal{C}}^{on}(s) := (\mathrm{PS}(s) = \texttt{as}) \end{pmatrix}. \tag{2}$$

A recurrence time $r$ of 5.0 s is set for this power management strategy. Smaller values for the recurrence time increase the overhead; higher values of the recurrence time would make the strategy less responsive.

# 3 Strategy Qualities

We assess power management strategies using a combination of four qualities, namely: (i) *efficiency*, (ii) *stability*, (iii) *robustness*, and (iv) *adaptability*. Since every data centre has different clients and environment, the importance of their four qualities might differ. In such a case, decision analysis techniques could assist with finding the best strategy. The meaning of each quality and its quantitative interpretation are elaborated below. To ease the explanation, the strategy from Section 2.4 serves as a running example. The four qualities are discussed in the subsequent subsections 3.1-3.4, followed by a discussion in Section 3.5.

## *3.1 Efficiency*

*Efficiency* in power management strategies addresses how well performance and energy goals are met. The goal of performance management is to obtain the lowest response time possible, while the goal of power management is to have the lowest power consumption possible. Since both values are relevant, efficiency is often expressed as the *performance per Watt* (PPW). However, many approaches like [2] struggle with expressing a combination of power consumption and performance in a meaningful way. Since in some cases a power and performance trade-off exists, as can be seen in earlier work [6,11,15], both power and performance are indicated.

To illustrate the power-performance trade-off, the effect of varying the queue

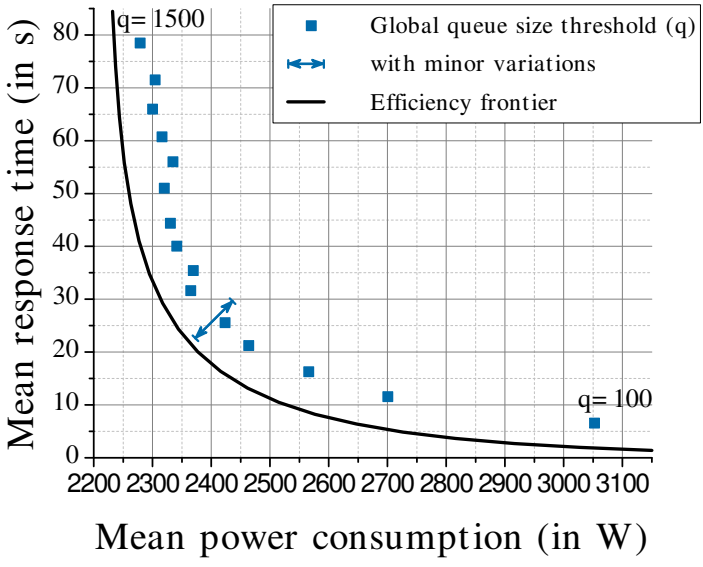



Fig. 5. Varying threshold of the queue ($q$) that illustrates (i) an efficiency frontier, and (ii) the effect of minor variations in workload and data centre characteristics

size threshold (as seen in Section 2.4) is shown in Figure 5. Each dot represents a single simulation run with a different queue size threshold. The mean power consumption (on the $x$-axis) ranges between 2 279 W and 3 053 W and the response time (on the $y$-axis) ranges between 6.54 s and 78.44 s. The scatter plot shows that by growing or shrinking the queue size threshold performance can be traded for power consumption.

Note that in the same figure an efficiency frontier is drawn that illustrates in which direction the optimal values of the power-performance are found. An efficient dynamic PM strategy is considered to have both low mean power consumption and low mean response times. Other details of this figure will be elaborated upon with the other qualities.

An indication of efficiency is the *overhead ratio* (OR), i.e., the mean power-state utilisation of the idle (PU(`id`)), booting (PU(`bt`)), waking (PU(`wk`)), suspending (PU(`su`)) and sleeping (PU(`sl`)) 'overhead' power states divided by the total power-state utilisation ($\sum_i \mathrm{PU}(i)$). In an efficient power management strategy, the time spent in those 'overhead' power states is minimised, because power states should only be switched when really necessary. We can express the OR, as follows:

$$\mathrm{OR} = \frac{\mathrm{PU}(\texttt{id}) + \mathrm{PU}(\texttt{bt}) + \mathrm{PU}(\texttt{su}) + \mathrm{PU}(\texttt{wk}) + \mathrm{PU}(\texttt{sl})}{\sum_i \mathrm{PU}(i)}. \tag{3}$$

### *3.2 Stability*

A *stable* power management strategy ensures acceptable power consumption and performance that does not fluctuate too much as a consequence of incorrect switching between power states of servers. As a consequence of stability, data centres eventually benefit, since less peaks are observed in power consumption, which leads to lower power consumption capacity demands. Customers of data centres are often ensured to have service of good quality via their SLAs, a certain maximum response time threshold below which, for instance, 95% of all response times observed should lie. This demand implicitly requires stable performance.

Recall that the running example (as seen in Section 2.4) varies queue size thresholds to obtain a power-performance trade-off. To actually reach the best efficiency for each simulation run, a stable power management strategy is required. A stable power management strategy makes it possible to move each simulation run (a dot) closer to the efficiency frontier (as illustrated in Figure 5). The main reason that dots are moving closer to the efficiency frontier is that only necessary power state switching occurs. As a consequence, the time spent in the 'overhead' power states is reduced, which thus improves efficiency.

A convenient method used to meet these SLA demands involves a response time threshold. Counting the number of violations and dividing this number over the total number of samples, estimates the *percentage of jobs that violate the SLA* (SLAv). This method is used in practice by one of our partners. Better power management strategies have a low number of SLA violations.

Another valuable measure is *power state switching frequency* (PSSF), i.e., the

number of power state switches per unit time. The PSSF is often the main cause of strong oscillations, because switching between power states leads to changes in the power consumption and performance. One 'power state switch' is recorded as soon as a service unit reaches the power state *on*, *as* or *of* (as seen in Figure 3). The PSSF is then determined by the *number of power state switches* (#PSS) as a fraction of the *time* ($t$) elapsed in the entire simulation. We can express the PSSF, as follows:

$$\text{PSSF} = \frac{\#\text{PSS}}{t}. \tag{4}$$

### *3.3 Robustness*

A power management strategy is considered to be *robust* if it is capable of having acceptable stable and efficient performance under minor variations of its data centre configuration. Robustness is relevant for a power management strategy to be applicable under realistic circumstances. Workload often fluctuates during the day and service times of resources vary. These variations include changes in inter-arrival rates $\lambda$, service times $1/\mu$ and power state switching time-outs $\alpha$. So, first a set of relevant minor variants should be formulated. For this, we take the original values, and allow addition or subtraction of 10% of $\lambda$, $1/\mu$ and $\alpha$ to their original values. With less than 10% the variants would be too minor to be significant, while with more than 10% would make it harder to compare. Next, stability and efficiency are then compared to the original configuration. In the comparison, the difference in PSSF ($\Delta\text{PSSF} = |\ \text{PSSF}_{\text{original}} - \text{PSSF}_{\text{variant}}\ |$) and OR ($\Delta\text{OR} = |\ \text{OR}_{\text{original}} - \text{OR}_{\text{variant}}\ |$) values gives a useful indication of robustness. Note that, observation of differences between solely the mean values is considered not to be a correct indication for robustness of the power management strategy, e.g., changing the number of servers will impact the performance per Watt. Therefore, PSSF and OR are more configuration-independent metrics.

In terms of the running example and power-performance trade-off graph (cf. Figure 5), a robust power management strategy with minor variations shifts the data points closer to or away from the efficiency frontier. Note that for variations with regard to the number of servers, a new efficiency frontier is established. Therefore, more generic metrics are required than mean power consumption and mean response

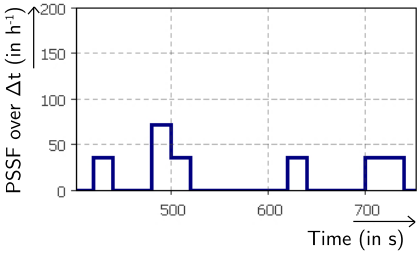



Fig. 6. A PSSF over $\Delta t$ of a simulation run without workload fluctuations other than the usual stochastic fluctuations in the workload (as specified in Section 2.2)

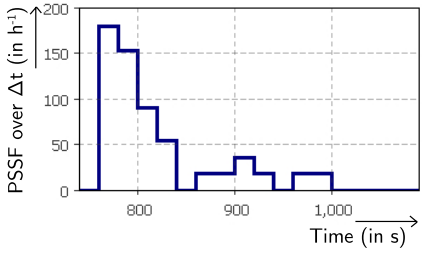



Fig. 7. A PSSF over $\Delta t$ of a simulation run with workload fluctuation caused by a 30% decrease of the arrival rate

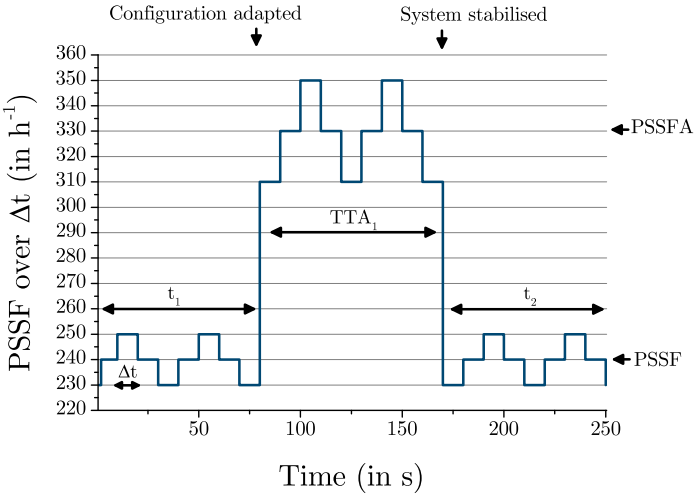



Fig. 8. An example adaptation period with PSSF over $\Delta t$

times. This is where PSSF and OR give a good indication by comparison with the original model parameters.

### *3.4 Adaptability*

When changing from one configuration to another there is a period of adaptation. A power management strategy is considered to be *adaptable* if it adapts itself stably and fast to a change in circumstances. So, an adaptable strategy adds configuration settings to its parameters to be able to determine the right number of servers and the right moment to adapt. The quality adaptability is relevant to consider, because an adaptable strategy is more generally applicable.

Figure 6 and Figure 7 show the effect of workload fluctuations on the PSSF over small blocks of time ($\Delta t = 20\,\text{s}$) during a simulation run [4]. Both plots display the PSSF over $\Delta t$ (in $h^{-1}$) on the $y$-axis and simulation time (in $s$) on the $x$-axis. We observe from comparing the two plots, that PSSF over $\Delta t$ is temporally much higher with workload fluctuation. The reason for this behaviour is that the employed power management strategy tries to only switch power states when necessary.

Figure 8 illustrates a system that enters ($t = 80$ s) and stabilises after an adaptation period ($t = 170$ s), and shows the PSSF over small blocks of time ($\Delta t = 20\,\text{s}$). The adaptation period is entered by a change in one of the configuration settings (as indicated in the plot). As can be seen in the plot, during the adaptation period the PSSF over $\Delta t$ is higher and has stronger oscillations.

To observe how adaptable the strategy is, first the *total time of adaptation* (TTA) from change of configuration to a stable situation and the PSSF during this period (denoted as PSSFA) is determined. The PSSFA is the number of power state switching during the adaptation periods (#PSSA) as a fraction of the time spent adapting ($\sum_{i=1}^{n} \text{TTA}_i = \text{TTA}_1 + \text{TTA}_2 + \ldots + \text{TTA}_n$). The PSSF is the

[4] Using job and data centre characteristics from Section 4.1 and ADVANCED PM strategy from Section 4.2.3.

| Quality | Computable values | Abbr. |
|---|---|---|
| **Efficiency** | Mean values | E |
| | Overhead ratio | OR |
| **Stability** | SLA violations percentage | SLAv |
| | Power state switching frequency | PSSF |
| | Standard deviation | $\sigma$ |
| **Robustness** | OR difference | $\Delta$OR |
| | PSSF difference | $\Delta$PSSF |
| **Adaptability** | Total time of adaptation | TTA |
| | PSSF of adaptation | PSSFA |

Table 1
Observed values for each power management strategy quality (lower is better)

total number of power state switches during the entire simulation excluding the switches made during adaptation as a fraction of the time spent in a stabilised system ($\sum_1^n t_i = t_1 + t_2 + \ldots + t_n$). We now express the PSSFA and a more detailed PSSF, as follows:

$$\text{PSSFA} = \frac{\#\text{PSSA}}{\sum_{i=1}^{n} \text{TTA}_i}, \text{PSSF} = \frac{\#\text{PSS} - \#\text{PSSA}}{\sum_{i=1}^{n} t_i}. \tag{5}$$

To compute the TTA and PSSFA, a start and an end of the adaptation period has to be determined. The start of this period is easy to detect, since parameters change. The end of this period is determined by observing whether the situation is again stable and efficient as before. Therefore, the strategy has to stabilise to at least the PSSF that it had before the adaptation period started. This requires the strategy to minimally being capable of stabilising.

### *3.5 Discussion*

An overview of all computable values for each of the qualities is provided in Table 1 based on the previous Section 3.2-3.4. Also worth noting is that lower values are considered better for all these values.

In the literature [7], the notions of OR and PSSF are related to elasticity in resource management of cloud systems. The report [7] states that elastic adaptation cannot be described with the traditional performance metrics (response times and utilisation). As a consequence, [7] presents a new set of metrics that surpasses current approaches on this subject. Our OR is related to a combination of their *accuracy* and *timeshare* metric. Our PSSF is related to their *jitter* metric. Thus, both approaches give expression to similar observations.

An aspect left out of the scope of the quality evaluation is the notion of complexity, which is still open for research. The satisfiers and constraints used for the PM (cf. Section 2) require additional sensors, extra computation and/or adjusted infrastructure. This introduces additional overhead by storing additional data, that requires extra space, and additional processing for sensing, computing or storing data, that require extra time and energy. Especially computing satisfiers with hysteresis adds space complexity, since this requires to keep track of information in the past. Moreover, the sampling frequency of observable quantities also requires

additional storage and are more computational intensive.

# 4 Evaluation Example

To illustrate the full evaluation of the qualities of a power management strategy, a data centre configuration and its job characteristics are described in Section 4.1 as the first step of the three-step approach from Section 2. Subsequently, Section 4.2 specifies five power management strategies. In Section 4.3, the last step of the approach evaluates the quality of these five strategies for the given data centre and job characteristics.

## *4.1 Job and Data Centre Characteristics*

We consider a data centre in which jobs arrive according to a Poisson process such that the inter-arrival times distribution is exponential with rate $\lambda$ (job/s). The service time ($1/\mu$ s) of each job is exponential, i.e., the server finishes jobs in a varying amount of time, because of varying job sizes. By default, $\lambda$ is set to 20.0 job/s and $\mu$ is set to 1.0 job/s. Otherwise, the overruling values are stated explicitly.

Figure 9 indicates daily server utilisation of a typical business data centre based on data from [5]. Since the processing speed of these servers remains the same, this server processing utilisation could be rewritten to a time-dependent arrival process with a job arrival rate $\lambda$ for all servers of $\frac{(\mu \cdot n)}{100} \cdot \mathrm{PU}(\mathtt{pc}, ins)$, where $\mathrm{PU}(\mathtt{pc}, ins)$ is the currently observed processing utilisation (in %) and $(\mu \cdot n)$ is the maximum allowable number of jobs arriving per second. This assumes that overhead caused by scheduling jobs is negligible.

The advantage of this approach is that data centres are simulated by only analysing its day-night patterns. This can be done by deriving the inter-arrival

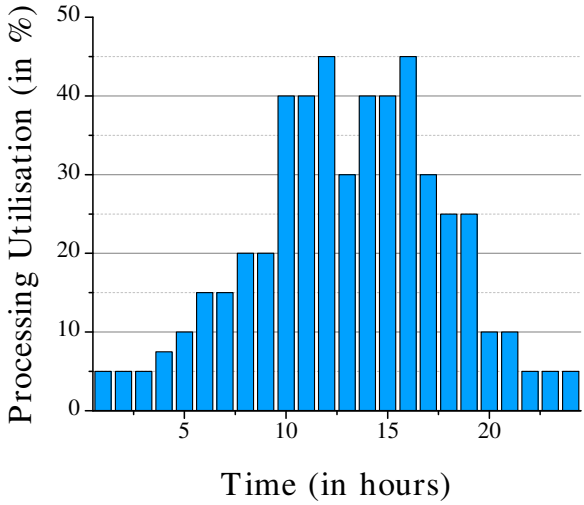



Fig. 9. Daily processing power state utilisation of a typical business data centre [5]

times of jobs for each hour together with its job characteristics such as demands, priorities and job sizes. Validation and sensitivity analysis of such parameters is considered to be future work.

Job scheduling is an essential part of the data centre configuration, since it influences the performance a lot. Jobs arrive at the data centre and are distributed via a dispatcher. The dispatcher uses the default scheduling of jobs via *shortest queue next*. However, jobs can only be scheduled to servers in the idle and processing power states.

The model for power state switching supports the three global power states, as in Figure 3. Each state in the model has a specific power consumption. By default, the time required for a server to shut down $\alpha_{\texttt{su}}$ is set to 100.0 s, boot $\alpha_{\texttt{bt}}$ is set to 100.0 s, wake $\alpha_{\texttt{wk}}$ is set to 10.0 s and sleep $\alpha_{\texttt{sl}}$ is set to 10.0 s and each time is deterministic. The awareness of power states in the models allows us to compute power consumption ($P$) by rewarding each power state with a power consumption. Note that processing is the only power state in which jobs are served.

The mean power consumption ($E[P]$) and mean response times ($E[R]$) are computed using the batch means method. This method requires the model time ($t_{\text{sim}}$) to be very long, which is usually around 86 400 virtual seconds (exactly 1 day/night cycle), and the system should be finish its start-up phase after some warm-up ($wup$) period. The maximum number of servers available is 30. The Server-Level Agreements (SLAs) demand a response time threshold ($R_{\text{SLA}}$) set to 20s. Violations of this threshold could result in a penalty or are considered to be unacceptable. The window size for the (exponentially) moving averages are set to one second of sample data. Analysis of different window sizes is part of future work.

### *4.2 Five Strategies*

#### *4.2.1 Base Case Strategy:* AlwaysOn

The AlwaysOn ($\Theta^{all}$) strategy is defined to set a base case for discussion of the impact of using a power management strategy. The satisfier and constraint formulas for $\Theta^{all}$, with $\mathrm{G}^{all} = (on)$, are as follows:

$$\Phi_{\mathcal{S}}^{all} = (\phi_{\mathcal{S}}^{on} := \ (true)), \Phi_{\mathcal{C}}^{all}(s) = (\phi_{\mathcal{C}}^{on}(s) := \ (true)). \tag{6}$$

So, for this base case strategy all servers are considered to be on at all times. This is expected to lead to high performance, but also to high energy consumption for the job and data centre characteristics in Section 4.1.

#### *4.2.2 Literature Inspired Strategies:* Optimal *and* Demotion

An example of a typical satisfier formula that belong to our Optimal ($\Theta^{opt}$) and Demotion ($\Theta^{dem}$) strategies (based on [10,13]) with $\mathrm{G}^{opt} = \mathrm{G}^{dem} = (as, on, of)$,

are defined as follows:

$$\Phi^{opt}_{\mathcal{S}} = \Phi^{dem}_{\mathcal{S}} = \begin{pmatrix} \phi^{as}_{\mathcal{S}} := (\mathrm{RT}(mavg) \leq R_{\mathrm{SLA}}) \\ \phi^{on}_{\mathcal{S}} := (\mathrm{RT}(mavg) > R_{\mathrm{SLA}}) \\ \phi^{of}_{\mathcal{S}} := (true) \end{pmatrix}, \quad \Phi^{opt}_{\mathcal{C}}(s) = \begin{pmatrix} \phi^{as}_{\mathcal{C}}(s) := (\mathrm{PS}(s) = \texttt{id}) \\ \phi^{on}_{\mathcal{C}}(s) := (\mathrm{PS}(s) = \texttt{as}) \\ \phi^{of}_{\mathcal{C}}(s) := (false) \end{pmatrix}, \tag{7}$$

$$\Phi^{dem}_{\mathcal{C}}(s) = \begin{pmatrix} \phi^{as}_{\mathcal{C}}(s) := (\mathrm{TO}(s, \texttt{id}) \leq t_{\texttt{idle}}) \\ \phi^{on}_{\mathcal{C}}(s) := (\mathrm{PS}(s) = \texttt{as} \vee \mathrm{PS}(s) = \texttt{of}) \\ \phi^{of}_{\mathcal{C}}(s) := (\mathrm{TO}(s, \texttt{as}) \leq t_{\texttt{asleep}}) \end{pmatrix}. \tag{8}$$

The Optimal strategy only uses moving average response time threshold $R_{\mathrm{SLA}}$ to determine if servers should switch between power states *on* and *as*. The Demotion strategy adds additional constraints to prevent overly active power state switching with the aid of a time-out ($t_{\texttt{idle}}$ and $t_{\texttt{asleep}}$) before switching power states, and servers can now also be shut down.

These examples are expected to improve the energy-efficiency, in comparison to the base case strategy, by a lower energy consumption as a consequence of sleeping and suspended servers. However, performance is expected to be negatively impacted by power state switching of the servers.

#### *4.2.3 Fine Tuned Strategies:* Strong *and* Advanced

Essentially, these Optimal and Demotion strategies maximise the number of sleeping and off machines while performance is kept intact with the aid of response time moving averages and a minimum amount of time spent in particular power states. The fine tuned strategies Strong and Advanced reduce unnecessary power state switching and SLA violations even further by (i) reduction of fluctuations in the response time observations using exponentially moving averages, (ii) limiting the number of servers to be in particular power states, and (iii) the use of a safety margin caused by fluctuations below the actual SLA response time threshold.

The Strong extension has been developed after running instances of the Optimal and Demotion variant. These runs showed that these strategies did not perform well with a small number of servers. Therefore, the Strong strategy never shuts down the last 20% of the total number of servers. In order to reduce the PSSF of the strategy, a server can only be turned back on if it has been sleeping for (at least) 100 seconds. Furthermore, fluctuations in the response times are reduced even more using exponentially moving averages. A safety margin below the actual SLA response time threshold is used for any remaining fluctuations. The satisfier

and constraint formulas for $\Theta^{str}$, with $\mathrm{G}^{str} = (as, on)$, are as follows:

$$\Phi^{str}_{\mathcal{S}} = \begin{pmatrix} \phi^{as}_{\mathcal{S}} := ((\mathrm{RT}(eavg) \leq \frac{3}{4} \cdot R_{\mathrm{SLA}}) \\ \qquad \wedge (\mathrm{PU}(\mathtt{id}, ins) \leq 0.2)) \\ \phi^{on}_{\mathcal{S}} := (\mathrm{RT}(eavg) > \frac{3}{4} \cdot R_{\mathrm{SLA}}) \end{pmatrix}, \quad \Phi^{str}_{\mathcal{C}}(s) = \begin{pmatrix} \phi^{as}_{\mathcal{C}}(s) := (\mathrm{PS}(s) = \mathtt{id}) \\ \phi^{on}_{\mathcal{C}}(s) := (\mathrm{TO}(s, \mathtt{as}) \leq 100.0) \end{pmatrix}. \tag{9}$$

In the Advanced extension servers can be suspended such that even less energy is consumed, as can be seen in the Demotion strategy. If the processing power state utilisation of the data centre (Figure 9) is really typical, then 50% of the resources is expected to be idle. So, while taking a safety bound of 20%, around 30% of all servers can still be turned off. The satisfier and constraint formulas for $\Theta^{adv}$, with $\mathrm{G}^{adv} = (as, on, of)$, are as then follows:

$$\Phi^{adv}_{\mathcal{S}} = \begin{pmatrix} \phi^{as}_{\mathcal{S}} := (\mathrm{RT}(eavg) \leq \frac{3}{4} \cdot R_{\mathrm{SLA}} \\ \qquad \wedge \mathrm{PU}(\mathtt{id}, \mathrm{ins}) \leq 0.2) \\ \phi^{on}_{\mathcal{S}} := (\mathrm{RT}(eavg) > \frac{3}{4} \cdot R_{\mathrm{SLA}}) \\ \phi^{of}_{\mathcal{S}} := (\mathrm{PU}((\mathtt{id}, ins) \leq 0.2) \\ \qquad \wedge (\mathrm{PU}(\mathtt{of}, ins) \leq 0.3) \\ \qquad \wedge (\mathrm{PU}(\mathtt{bt}, ins) \leq 0.05)) \end{pmatrix}, \quad \Phi^{adv}_{\mathcal{C}}(s) = \begin{pmatrix} \phi^{as}_{\mathcal{C}}(s) := (\mathrm{PS}(s) = \mathtt{id}), \\ \phi^{on}_{\mathcal{C}}(s) := ((\mathrm{PS}(s) = \mathtt{as} \vee \mathrm{PS}(s) = \mathtt{of}) \\ \qquad \wedge (\neg(\mathrm{PU}(\mathtt{as}, ins) \geq 0.0) \\ \qquad \wedge (\mathrm{PS}(s) = \mathtt{as})) \\ \phi^{of}_{\mathcal{C}}(s) := (\mathrm{TO}(s, \mathtt{as}) \leq 100.0) \end{pmatrix}. \tag{10}$$

### *4.3 Quality Evaluation*

A global overview of an assessment of the above power management strategies is provided in Table 2. Each row in the table represents a power management strategy and each column a quality. In each cell, several measurements for that specific strategy and quality combination are provided. Beside the measurements itself, each measurement is ranked by comparing it to the other strategies, where the highest ranks (lowest numbers are best) indicate the most optimal values. We combine measurements by giving equal weight to each relevant computed value and average its rank for each of the qualities. In practice, these weights could be adjusted if some values are considered to be unacceptable.

A relevant observation from Table 2 is the 54% energy reduction for the Advanced strategy (53.02 kWh for 1 day) compared to using the AlwaysOn strategy (115.10 kWh for 1 day). However, such an energy reduction caused by power management has effect on other quantities of the data centre. Optimal and Demotion both show already a large improvement in energy efficiency with reasonable performance, stability, robustness and adaptability. With additional fine tuning, Strong

| Strategy | Efficiency | Stability | Robustness | Adaptability |
|---|---|---|---|---|
| AlwaysOn ($\Theta^{all}$) | **(11) = $\sum_{\text{ranks}}$** | **(5)** | **(6)** | **(2)** |
| (cf. Section 4.2.1) | (5) E[E] = 115.10kWh<br>E[P] = 4 796W<br>(1) E[R] = 1.269 s<br>(5) OR = 66.86% | (2) $\sigma_{(P)}$ = 163.910 W<br>(1) $\sigma_{(R)}$ = 0.328 s<br>(1) SLAv = 0 %<br>(1) PSSF = 0 $h^{-1}$ | (5) avg.ΔOR = 1.38%<br>(1) avg.ΔPSSF = 0 $h^{-1}$ | (1) avg.PSSFA = 0 $h^{-1}$<br>(1) avg.TTA = 0 s |
| Optimal ($\Theta^{opt}$) | **(12)** | **(20)** | **(6)** | **(10)** |
| (cf. Section 4.2.2)<br>(inspired by Horvath et al. [10]) | (4) E[E] = 59.09kWh<br>E[P] = 2 462W<br>(5) E[R] = 8.761 s<br>(3) OR = 4.69% | (5) $\sigma_{(P)}$ = 494.238 W<br>(5) $\sigma_{(R)}$ = 5.590 s<br>(5) SLAv = 2.57 %<br>(5) PSSF = 241.2 $h^{-1}$ | (2) avg.ΔOR = 0.21%<br>(4) avg.ΔPSSF = 6.12 $h^{-1}$ | (5) avg.PSSFA = 331.2 $h^{-1}$<br>(5) avg.TTA = 150.00 s |
| Demotion ($\Theta^{dem}$) | **(10)** | **(16)** | **(8)** | **(8)** |
| (cf. Section 4.2.2)<br>(inspired by Horvath et al. [10]) | (2) E[E] = 55.06kWh<br>E[P] = 2 294W<br>(4) E[R] = 8.665 s<br>(4) OR = 4.78% | (4) $\sigma_{(P)}$ = 487.528 W<br>(4) $\sigma_{(R)}$ = 5.463 s<br>(4) SLAv = 1.879 %<br>(4) PSSF = 237.6 $h^{-1}$ | (3) avg.ΔOR = 0.24%<br>(5) avg.ΔPSSF = 7.56 $h^{-1}$ | (4) avg.PSSFA = 293.8 $h^{-1}$<br>(4) avg.TTA = 145.83 s |
| Strong ($\Theta^{str}$) [13] | **(6)** | **(8)** | **(3)** | **(6)** |
| (cf. Section 4.2.3) | (3) E[E] = 56.14kWh<br>E[P] = 2 339W<br>(2) E[R] = 4.257 s<br>(1) OR = 3.59% | (1) $\sigma_{(P)}$ = 147.939 W<br>(3) $\sigma_{(R)}$ = 3.393 s<br>(2) SLAv = 0.0021 %<br>(2) PSSF = 9 $h^{-1}$ | (1) avg.ΔOR = 0.16%<br>(2) avg.ΔPSSF = 2.52 $h^{-1}$ | (3) avg.PSSFA = 51.48 $h^{-1}$<br>(3) avg.TTA = 110.42 s |
| Advanced ($\Theta^{adv}$) [13] | **(5)** | **(11)** | **(7)** | **(4)** |
| (cf. Section 4.2.3) | (1) E[E] = 53.02kWh<br>E[P] = 2 209W<br>(2) E[R] = 4.101 s<br>(2) OR = 3.90% | (3) $\sigma_{(P)}$ = 170.192 W<br>(2) $\sigma_{(R)}$ = 3.360 s<br>(3) SLAv = 0.0130 %<br>(3) PSSF = 14.4 $h^{-1}$ | (4) avg.ΔOR = 0.31%<br>(3) avg.ΔPSSF = 3.24 $h^{-1}$ | (2) avg.PSSFA = 33.84 $h^{-1}$<br>(2) avg.TTA = 106.25 s |

Table 2: Power management strategies qualities assessment (cf. Table 1)

adds even more stability and robustness, and a boost in performance. ADVANCED shows even less energy consumption with still fine efficiency, stability, slightly improved adaptability and slightly less robustness compared to STRONG.

All these observations together show that the right strategy depends on the demands of this data centre on each of the qualities. Some data centres find certain percentage of SLA violations unacceptable. Other data centres might have very steady workload characteristics, which makes robustness and adaptability less relevant. Therefore, these two examples might choose different power management strategies based on their quality demands.

## 5 Conclusions

For the purpose of analysing energy efficiency and performance in data centres, this paper introduces novel metrics for evaluation of power management strategies in four qualities: (i) efficiency, (ii) stability, (iii) robustness and (iv) adaptability.

First, the job and data centre characteristics have been described. Subsequently, five power management strategies have been specified to meet data centre quality demands for some global and server-level conditions with the aid of so-called satisfiers and constraints formulas. In the final step of the approach, these power management strategies have been evaluated with our novel metrics.

The various qualities are assessed for a data centre configuration with 30 servers and typical small business data centre workload. An energy reduction of 54% is obtained by power management strategies (compared to ALWAYSON strategy) inspired by the literature (OPTIMAL and DEMOTION) and fine tuning thereof (STRONG and ADVANCED). For these fine tuned strategies, energy efficiency is increased *and* performance, stability, adaptability and robustness are maintained as well.

## References

[1] AnyLogic, *AnyLogic: Multimethod Simulation Software* (2000).
URL http://www.anylogic.com/

[2] Azimzadeh, A. and N. Tabrizi, *A Dynamic Power Management Schema for Multi-Tier Data Centers* (2016).
URL http://arxiv.org/abs/1604.04320

[3] Benini, L. and G. d. Micheli, "Dynamic Power Management: Design Techniques and CAD Tools," Kluwer Academic Publishers, Norwell, MA, USA, 1998.
URL http://dl.acm.org/citation.cfm?id=551011

[4] Chao, J., *Data Centers Continue to Proliferate While Their Energy Use Plateaus* (2016).
URL http://newscenter.lbl.gov/2016/06/27/data-centers-continue-proliferate-energy-use-plateaus/

[5] Emerson Network Power, *Energy Logic: Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems*, White Paper of Emerson Electric Co (2009).
URL https://www.uk.insight.com/content/dam/insight/EMEA/uk/shop/emerson/energy-logic.pdf

[6] Haverkort, B. R. and B. F. Postema, *Towards Simple Models for Energy-Performance Trade-Offs in Data Centres*, in: *Proc. of Int. Work. on Demand Modeling and Quantitative Analysis of Future Generation Energy Networks and Energy Efficient Systems*, 2014, pp. 113–122.
URL https://opus4.kobv.de/opus4-bamberg/frontdoor/index/index/docId/6486

[7] Herbst, N., *Ready for Rain? A View from SPEC Research on the Future of Cloud Metrics*, Technical report, SPEC Research (2016).
URL https://arxiv.org/abs/1604.03470

[8] Hewlett-Packard, Intel, Microsoft, Phoenix Technologies and Toshiba, *Advanced Configuration and Power Interface Specification*, Technical report (2011).
URL http://acpi.info/DOWNLOADS/ACPIspec50.pdf

[9] Hopcroft, J. E., R. Motwani and J. D. Ullman, "Automata Theory, Languages, and Computation," Pearson Education, 2006, 24 int. edition.
URL http://www.academia.edu/download/31352670/19s_Automata_Theory.pdf

[10] Horvath, T. and K. Skadron, *Multi-Mode Energy Management for Multi-Tier Server Clusters*, in: *Proc. of the 17th Int. Conf. on Parallel Architectures and Compilation Techniques* (2008), pp. 270–279.
URL https://doi.org/10.1145/1454115.1454153

[11] Postema, B. F. and B. R. Haverkort, *Stochastic Petri Net Models for the Analysis of Trade-Offs in Data Centres with Power Management*, in: S. Klingert, M. Chinnici and M. Rey Porto, editors, *Proc. of 3rd Int. Work. on Energy-Efficient Data Centres (E2DC)*, LNCS **8945** (2014), pp. 52–67, https://link.springer.com/chapter/10.1007/978-3-319-15786-3_4.

[12] Postema, B. F. and B. R. Haverkort, *An AnyLogic Simulation Model for Power and Performance Analysis of Data Centres*, in: M. Beltrán, W. Knottenbelt and J. Bradley, editors, *Computer Performance Engineering*, LNCS **9272** (2015), pp. 258–272.
URL https://link.springer.com/chapter/10.1007/978-3-319-23267-6_17

[13] Postema, B. F. and B. R. Haverkort, *Specification of data centre power management strategies*, in: *Proc. of the 8th Int. Conf. on Future Energy Systems*, e-Energy '17 (2017), pp. 284–289.
URL http://doi.acm.org/10.1145/3077839.3084025

[14] Shehabi, A., S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo and W. Lintner, *United States Data Center Energy Usage Report*, Technical report (2016).
URL https://eta.lbl.gov/sites/all/files/publications/lbnl-1005775_v2.pdf

[15] van den Berg, F., B. F. Postema and B. R. Haverkort, *Evaluating Load Balancing Policies for Performance and Energy-Efficiency*, in: *Proc. of 14th Int. Work. on Quantitative Aspects of Programming Languages and Systems*, ETPCS **227**, Eindhoven, the Netherlands, 2016, pp. 98–117.
URL https://arxiv.org/abs/1610.08172

# A Strategy Specification Definitions

This section contains the necessary notation and definitions from our previous work [13] required to understand the formulas used to describe strategies in this paper. A power management strategy is formally defined as a 3-tuple:

$$\Theta = (\mathrm{G}, \Phi_{\mathcal{S}}, \Phi_{\mathcal{C}}(s)), \tag{A.1}$$

where the vector $\mathrm{G} = (g_1, \ldots, g_n)$ contains all possible global power states, the vector $\Phi_{\mathcal{S}} = \left(\phi_{\mathcal{S}}^{g_1}, \ldots, \phi_{\mathcal{S}}^{g_n}\right)$ contains satisfiers for each global power state to switch to, and the vector $\Phi_{\mathcal{C}}(s) = \left(\phi_{\mathcal{C}}^{g_1}(s), \ldots, \phi_{\mathcal{C}}^{g_n}(s)\right)$ contains constraints for each global power state to switch to for any power management enabled server $s$.

## *A.1 Satisfiers and Constraints*

A satisfier $\mathcal{S}$ can be one of the quantities observed at global-level (cf. Section 2.2), as follows:

$$\begin{aligned}\mathcal{S} := \mathrm{QS} \mid \mathrm{PU}(\delta,\gamma) \mid \mathrm{TO}(\delta) \mid \mathrm{RT}(\gamma) \mid \mathrm{PC}(\gamma) \mid \\ \mathrm{TM}(\gamma) \mid \mathrm{AR}(\gamma) \mid \mathrm{WK}(\gamma),\end{aligned} \tag{A.2}$$

where state $\delta \in \{\texttt{as}, \texttt{wk}, \texttt{pc}, \texttt{bt}, \texttt{sl}, \texttt{id}, \texttt{su}, \texttt{of}\}$ (cf. Figure 3) and computation method $\gamma \in \{\mathrm{ins}, \mathrm{mavg}, \mathrm{eavg}, \mathrm{savg}, \mathrm{per}\}$. In general, variable $\gamma$ indicates how the

satisfiers are computed, which are: instantaneous (ins), moving averages (mavg), exponentially moving averages (eavg), steady-state averages computed with the batch means method (bavg), and percentiles (per).

A constraint $\mathcal{C}(s)$ can be one of the quantities observed at server-level (cf. Section 2.2), as follows:

$$\mathcal{C}(s) := \mathrm{QS}(s) \mid \mathrm{PS}(s) \mid \mathrm{TO}(s,\delta) \mid \mathrm{TM}(s), \tag{A.3}$$

where $s$ is a server and $\delta \in \{\texttt{as}, \texttt{wk}, \texttt{pc}, \texttt{bt}, \texttt{sl}, \texttt{id}, \texttt{su}, \texttt{of}\}$.

Formula $\phi_{\mathcal{C}}^{g_i}(s)$ and $\phi_{\mathcal{S}}^{g_i}$ shows the expressiveness including negate ($\neg$) operator, conjunction ($\wedge$) operator, disjunction ($\vee$) operator and parentheses for respectively these *constraints* from $\mathcal{C}(s)$ for some server $s$ and observing quantities with the *satisfiers* in $\mathcal{S}$, as follows:

$$\begin{aligned}\phi_{\mathcal{C}}^{g_i}(s) := \mathcal{C}(s) \sqsubseteq \rho \mid \neg\phi_{\mathcal{C}}^{g_i}(s) \mid \phi_{\mathcal{C}}^{g_i}(s) \wedge \phi_{\mathcal{C}}^{g_i}(s) \mid \\ \phi_{\mathcal{C}}^{g_i}(s) \vee \phi_{\mathcal{C}}^{g_i}(s) \mid (\phi_{\mathcal{C}}^{g_i}(s)) \mid \phi_{\mathcal{S}}^{g_i} \mid \mathit{true} \mid \mathit{false},\end{aligned} \tag{A.4}$$

where $\sqsubseteq \in \{\leq, <, =\}$, $g_i$ is a global power state and the domain of $\rho$ depends on its constraint from $\mathcal{C}(s)$.

$$\begin{aligned}\phi_{\mathcal{S}}^{g_i} := \mathcal{S} \sqsubseteq \rho \mid \neg\phi_{\mathcal{S}}^{g_i} \mid \phi_{\mathcal{S}}^{g_i} \wedge \phi_{\mathcal{S}}^{g_i} \mid \\ \phi_{\mathcal{S}}^{g_i} \vee \phi_{\mathcal{S}}^{g_i} \mid (\phi_{\mathcal{S}}^{g_i}) \mid \mathit{true} \mid \mathit{false},\end{aligned} \tag{A.5}$$

where $\sqsubseteq \in \{\leq, <, =\}$ is a comparison operator, $g_i$ is a global power state and the domain of $\rho$ depends on its satisfier from $\mathcal{S}$.