Discrete-Time MDP Policy for Energy-Aware Data Center

Léa Bayati
Laboratoire d'Algorithmique Complexité et Logique, Université Paris-Est Créteil (UPEC),

Keywords: Data Center, Markov Decision Process, Energy Saving, Performance, Queues.

Abstract: This paper presents a stochastic model for Dynamic Power Management (DPM) that consists in dynamically switching-on/off servers in the data center to ensure a reasonable energy consumption with a good Quality of Services (QoS). In this work, arrival jobs are specified with histograms which are discrete identically independently distributions obtained from real traces, empirical data, or incoming traffic measurements. We model a data center by a queue, then we formulate the optimization problem by a discrete time Markov Decision Process (MDP) to find the optimal policy. We prove also some structural properties of the optimal policy. Our approach was applied and tested for several data center parameters with arrivals modeled by histograms obtained from real Google traffic traces.

1 INTRODUCTION

The recent expansion of *Clouds* and *Data Centers* is causing energetic problems and digital pollution issues. More than 1.3% of the global energy consumption is due to the electricity used by data centers. Additionally, one data center server can produce more than 10 kg of CO_2 per day, rates that are increasing, revealed by surveys conducted in (Koomey, 2011), saying a lot about the increasing evolution of data centers. Data centers are designed to support the expected peak traffic load, however the global load is about 60% of the peak load (Benson et al., 2010). In fact, an important number of servers are not under load and still consume about 65%-70% of the maximal energy consumption (Greenberg et al., 2009). Studies like (Lee and Zomaya, 2012) show that much of the energy consumed in the data center is mainly due to the electricity used to run the servers and to cool them (70% of total cost of the data center). Thus, the main factor of this energy consumption is related to the number of operational servers. Many efforts have focused on servers and their cooling. Works have been done to build better components and low-energy-consumption processors (Grunwald et al., 2000), more efficient energy network (Benson et al., 2010), more efficient cooling systems (Patel et al., 2003), and optimized kernels (Jin et al., 2012). That being said, another complementary saving energy approach is to consider a power management strategy to manage the switching-on/off of servers in a data center to ensure both a good performance of services offered by these data centers and reasonable energy consumption. Two requirements are in conflict: (i) Increasing the Quality of the Service (QoS). (ii) Saving energy. For the first requirement we need to turn-on a large number of servers which consume more energy and leads to less waiting time and decreases the rate of losing jobs but requires a high energy consumption. For the second requirement we need to turn-on a small number of servers which leads to less energy consumption, but causes more waiting time and increases the rate of losing jobs. Thus, the goal is to design better power management algorithms which take into account these two constraints to minimize waiting time, loss rate and energy consumption.

In a data center every job may generates a profit, and the average profit per job can be computed as a ratio of the total profit over the number of served jobs. For instance, 10^6 requests (page views) may bring \$1000 of revenue. Thus, it can be said that each job brings $\$10^{-3}$ on average. Work in (Dyachuk and Mazzucco, 2010) suggests that each successfully processed job generates a profit around $6.2 \times \$10^{-6}$. In this case, a lost job costs $6.2 \times \$10^{-6}$. Otherwise, according to research published by Dell and Principled Technologies, a single server consumes around something between 384 and 455 Watts. Other works evaluate that the power consumption of each server ranges between 238 and 376 Watts (Mazzucco et al., 2010). Rajesh et al. (Rajesh et al., 2008) estimate the cost

Bayati, L.
Discrete-Time MDP Policy for Energy-Aware Data Center.
DOI: 10.5220/0011846300003491
In *Proceedings of the 12th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2023)*, pages 89-97
ISBN: 978-989-758-651-4; ISSN: 2184-4968

of one kWh of energy to \$0.0897. These values may vary depending on where the data center is located and how electricity is generated. Using that baseline a one server costs around \$300 per year to run.

In order to save energy, the strategy presented by Mazzucco et al in (Mazzucco and Dyachuk, 2012) is a dynamic allocation policy that checks the system periodically, collects statistics, estimates arrival rate and average service time, and finally allocates a minimal number of servers which should meet the QoS requirements. Numerical experiments and simulations based on Wikipedia traces (November 2009) show a significant improvement of energy saving.

Works as in (Mitrani, 2013) present a theoretical queuing model to evaluate the trade-off between waiting time and energy consumption if only a subset of servers is active all the time and the remaining servers (reserve-servers) are enabled when the number of jobs in the system increases and exceeds some threshold *Up*. The reserve-servers will be turned-off again if the number of jobs decreases and becomes less than another threshold *Down*. Jobs arrive according to an identically independently distributed (i.i.d) Poisson process then served according to an exponentially distributed service time. Analytic equation and experimental simulation results show that energy consumption is significantly reduced (from 20% to 40%) while still having an acceptable waiting time.

In (Bayati et al., 2016) we present, with other coauthors, a tool to study the trade-off between energy consumption and performance evaluation. The tool is based on threshold policy. It is numerical rather than analytical or simulation. Some measurements of real traffic are used to model the job arrivals in a more accurate manner. The arrival process is assumed to be stationary for short periods of time and change between periods. This allowed us to model for instance hourly variations of the job arrivals. The tool uses real traffic traces to produce a discrete distribution that models arrival jobs. Threshold policy checks every slot the number of waiting jobs in the buffer, then turn-on more additional servers if this number exceeds threshold U, and turn off some server if it's less then threshold D. The tool analyzes all possible couples (U,D) then returns the best couple that minimize a cost function combining performance measures and energy consumption.

All previous works and others (Gebrehiwot et al., 2016) consider models under specific management mechanisms, that leads to suboptimal strategies. In (Maccio and Down, 2015), MDP based models were considered in order to device optimal strategies.

More recent works can be found in (Khan, 2022; Rteil et al., 2022; Peng et al., 2022; An and Ma, 2022).

Markov Decision Process (MDP) is a formalism allowing to model a decision-making system. Basically it is a Markov chain augmented by a set of decision-making actions. Applying an action moves the system from a state to another state and gives rise to some cost. Thus, the probability that the system moves into its new state is influenced by the chosen action. In the context of Dynamic Power Management, the cost of the consumed energy depends on the number of operational servers and the QoS cost depends on the number of waiting and/or rejected jobs. Additionally, from every state of the system we have to consider all the following actions: {turn off all servers, turn on only 1 server, turn on only 2 servers, ... , turn on all servers}. The optimal strategy consists in finding the best sequence of actions to minimize the total cost accumulated during a finite period of time, or to minimize the expected cost for an infinite period of time.

MDP models can be considered in continuous or discrete time context. In the case of discrete time MDP the value iteration algorithm that implements the Bellman's backup equations (Bellman, 1957) computes the optimal policy. However, continuous time MDP are considered when the arrival jobs are modeled by a continuous time distribution obtained by fitting the empirical data or by asserting some assumptions that leads usually to a Poisson distribution (Maccio and Down, 2015).

One of the main uses of Markov decision process methods is to establish the existence of optimal policies with simple structure. The importance of founding structured policy among optimal strategies lies in its advantage in making decisions, in its implementation facility, and in its efficient computation. When the optimal policy has a simple form, specialized algorithms can be device to search only among policies that have the same form as the optimal policy. This avoids the need for using iteration algorithm. Works as in (Topkis, 1978; Lu and Serfozo, 1984; Serfozo, 1979; Hipp and Holzbaur, 1988; Plum, 1991) investigate the structure of optimal policy in the context of Markovian control process and give conditions to check properties like hysteresis, monotony, or isotony of the policy.

In our work, real incoming traffic traces are sampled then used directly to build an empirical discrete distribution called histogram that may model the job arrivals in a more accurate way. In fact, we can accommodate less regular processes than the Poisson process considered in several works like in (Mitrani, 2013; Schwartz et al., 2012). Thus, the Markovian assumptions (Poisson arrivals, exponential services)

and the infinite buffer capacity are not required for our approach analysis and its optimization procedure. We already used histograms in (Bayati, 2016; Bayati et al., 2016), and similar works as in (Tancrez et al., 2009) used them for network traffic.

The target of our approach is request based services data center. A data center is modeled by a discrete time queue with a finite buffer capacity where we set the length of the time slot to the length of the sampling time used to sample the traffic traces to obtain the histogram. We formulated the problem of energy-QoS optimization by a discrete time MDP without passing by uniformization process. Then we study the energy consumption and its evolution over a finite (hour, day, month) or infinite period of time. As value iteration algorithm is an efficient dynamic programming implementation for solving an MDP model (for both finite and infinite horizon), we used it to compute the optimal control policy of our MDP. We assume identical servers, identical job duration, job duration largely higher than time to switch on a server, and servers switch-on immediately which is particularly a strong assumption.

The rest of this paper is organized as follows. Section 2 models the system by simple queue. Then, Section 3 formulates the optimization problem as discrete time MDP. After that, in Section 4 we prove some structural properties related to the optimal policy. Finally, Section 5 analyzes systems with arrivals modeled by discrete distribution obtained from real Google traffic traces (Wilkes, 2011).

2 QUEUE MODEL

Here we deal with discrete time model. Let *DC* be a data center composed of *max* identical servers working under the FIFO[1] discipline. *DC* receives jobs requesting the offered service. The maximal number of jobs that can be served by one server in one slot is assumed to be constant and denoted by d. Thus, the queuing model is a batch arrival queue with constant services and finite capacity buffer b (buffer size). In practice, data centres handle jobs of varying service tiers, each with its own revenue model. In this work we assume that the data centre handles only one tier of jobs. We model arrival jobs by finite structure called *histogram* which is based on discrete distribution.

Definition 1 (histogram)**.** *Let A be a discrete random variables taking values in $\mathbb{N}$. The couple $\mathcal{H}_A = (S_A, P_A)$ denotes the histogram of A where $P_A : \mathbb{N} \to [0,1]$ is the probability mass function of A and $S_A = \{i \in \mathbb{N} : P_A(i) > 0\}$ is the support of P_A.*

[1]First In First Out.

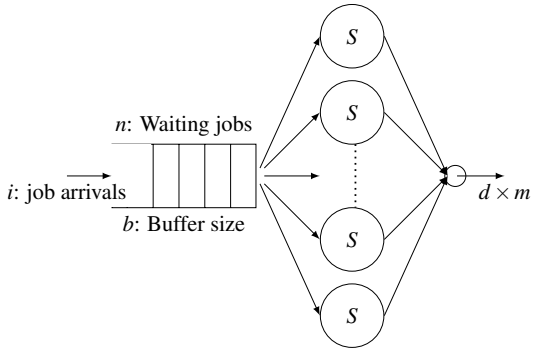

Figure 1: Illustration of the queuing model.

Thus, the number of jobs arriving to the data center during a slot is modeled by a histogram $\mathcal{H}_A$ where $P_A(i)$ gives the probability to have i arrival jobs per a slot. Note that in this paper, arrival jobs are assumed independent, and their distribution P_A is obtained from real traces, empirical data, or incoming traffic measurements.

Example 1. *Assume that, per slot, we have a probability of* 0.59 *to receive one arrival job, and* 0.41 *to receive no arrival jobs. In this case, arrivals are modeled by histogram $\mathcal{H}_A = (S_A, P_A)$ where $S_A = \{0,1\}$, $P_A(0) = 0.41$, and $P_A(1) = 0.59$.*

The number of waiting jobs in the buffer is denoted by n. The number of operational servers is denoted by m. The number of rejected (lost) jobs is denoted by l. We assume that initially the number of operational servers, the number of waiting jobs, and the number of rejected jobs are 0. The maximal number of servers that can be operational is *max*. The number of waiting jobs n can be computed by induction where the exact sequence of events during a slot have to be described. First, the jobs are added to the buffer then they are executed by the servers. The admission is performed per job according to the Tail Drop policy: a job is accepted if there is a place in the buffer, otherwise it is rejected. The following equations give the number of waiting jobs in the buffer and the lost jobs. For a number of i arrival jobs, we have:

$$\begin{cases} n & \leftarrow & \min\{b, \max\{0, n+i-d\times m\}\} \\ l & \leftarrow & \max\{0, n+i-d\times m-b\} \end{cases} \quad (1)$$

It is assumed that the input arrivals are i.i.d. and under these assumptions, the model of the queue is a time-homogeneous Discrete Time Markov Chains.

The problem we have to consider is to find a trade-off between the performance (i.e. waiting and loss jobs) and the energy consumption (i.e. number of operational servers). However, as the number of servers

changes with time, the system becomes more complex to analyze. The number of servers may vary according to the traffic and performance indexes. More precisely, n, l and m are considered and then some decisions are taken according to a particular cost function.

The energy consumption takes into account the number of operational servers. Each server consumes some units of energy per slot when a server is operational and it costs an average of $c_M \in \mathbb{R}^+$ monetary unit. A server may consume a very low amount of energy when it is turned-off. However, during the latency period a server may consume an additional amount of energy that costs an average of $c_{On} \in \mathbb{R}^+$ monetary unit which is the energetic cost needed to switch-on a server. Additionally, we consider that a server switches-on immediately. The total consumed energy is the sum of all units of energy consumed among a specific period. QoS takes into account the number of waiting and lost jobs. Each waiting job costs $c_N \in \mathbb{R}^+$ monetary unit per slot, and a rejected job costs $c_L \in \mathbb{R}^+$ monetary unit. Notice that in practice energy costs may vary with time of day, depending on the prevailing demand from other users, however in this work we assume that there is no variation energy costs over a day.

3 MARKOV DECISION PROCESS

In order to find the optimal strategy and then analyze the performance and the energy consumption of the data center under this optimal strategy, we will use the concept of *Markov Decision Process* to formulate our optimization problem. Notice that (n,l) and m are mutually dependent. If we decrease m, we save more energy, but both n and l increase, so we will have an undesirable diminution of QoS and vice versa.

Let $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{C})$ be an MDP where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the set of actions, $\mathcal{P}$ is the transition probability, and $\mathcal{C}$ the immediate cost of each action. Let $\mathcal{H}_A = (S_A, P_A)$ be the histogram used to model the arrival jobs. The state of the system is defined by the couple $(m,(n,l))$ where m is the number of operational servers, n is the number of waiting jobs, and l is the number of lost job. Indeed the state space $\mathcal{S}$ is defined as:

$$\mathcal{S} = \{(m,(n,l)) \mid m \in [0..max],\ n \in [0..b],\ l \in [0..\max(S_A)]\} \quad (2)$$

At the beginning of each slot, and based on the current state of the system, an action $\alpha_j \in \mathcal{A}$ will be made to determine how many servers will be operational during the current slot. In fact the action space $\mathcal{A}$ is defined as $\mathcal{A} = \{\alpha_j \mid 0 \le j \le max\}$, where action α_j consists of keeping exactly j operational servers during the current slot. We have a probability of $\mathcal{P}_{ss'}^{\alpha_j}$ to move from state $s = (m,(n,l))$ to $s' = (j,(n',l'))$ under action α_j. This probability is defined as:

$$\mathcal{P}_{ss'}^{\alpha_j} = \sum_{i \in S_A} P_A(i) \quad (3)$$

For each $i \in S_A$ satisfying: $\begin{cases} n' &= \min\{b, \max\{0, n+i-d\times j\}\} \\ l' &= \max\{0, n+i-d\times j-b\} \end{cases}$

Consequently moving from state $s = (m,(n,l))$ to $s' = (j,(n',l'))$ under action α_j induces immediately a cost $\mathcal{C}_s^{\alpha_j}$ defined as:

$$\mathcal{C}_s^{\alpha_j} = j \times c_M + \max\{0, j-m\} \times c_{On} + n \times c_N + l \times c_L \quad (4)$$

The immediate cost $\mathcal{C}_s^{\alpha_j}$ includes four parts:

1. The first part is $j \times c_M$, where c_M is the cost of energy consumption of one working server per slot and j is the number of working servers during the current slot. This part presents the total cost of energy consumed by the operational servers during the current slot.
2. The second part is $\max\{0, j-m\} \times c_{On}$, where c_{On} is the energetic cost of switching-on one server from stopping mode to working mode and $\max\{0, j-m\}$ is the number of servers switched-on at the beginning of the slot. This part presents the total cost of energy used to switch-on servers at the beginning of the current slot.
3. The third part is $n \times c_N$, where c_N is the cost of keeping one job in the buffer during the current slot and n is the number of waiting jobs. This part presents the total cost of maintaining waiting jobs in the buffer during the current slot.
4. The last part is $l \times c_L$, where c_L is the cost of loosing one job during the current slot and l is the number of lost jobs. This part presents the total cost of loosing jobs during the current slot.

Notice that the number of state of the MDP is in $O(max \times b \times \max(S_A))$, and the number of transition of the MDP is in $O(max^2 \times b \times |S_A| \times \max(S_A))$. This can be proved as following. Every state of the MDP includes three element:

1. the number of operational servers which is between 0 and max,
2. the number of waiting jobs in the buffer which is bounded by b, and
3. the number of rejected jobs which can be at most equals to the maximum number of arrival jobs given by $\max(S_A)$.

So, $|\mathcal{S}|$ is bounded by $(max+1)\times(b+1)\times(\max(S_A)+1)$. Additionally from each state of the MDP we have at most $(max+1)$ action, and each action leads to a number of transition equals to $|S_A|$ (one transition for each bin in the support of the arrival distribution). In fact, as the number of state is in $O(max\times b\times\max(S_A))$, we deduce that the number of transition is bounded by $max\times b\times\max(S_A)\times(max+1)\times|S_A|$.Table 1 resumes parameters used in our model and our MDP formulation.

Table 1: Model and MDP Parameters.

Parameters	Description
h	duration of analysis
max	total number of servers
d	processing capacity of a server
b	buffer size
m	number of operational servers
n	number of waiting jobs
l	number of rejected jobs
$\mathcal{H}_A$	histogram of job arrivals
c_{On}	energetic cost of switching-on 1 server
c_M	energetic cost of 1 working server during 1 slot
c_N	cost of 1 waiting job in buffer during 1 slot
c_L	cost of 1 lost job during 1 slot
$\mathcal{S}$	set of all possible states
$\mathcal{A}$	set of all possible actions
$s=(m,(n,l))$	system state
$s_0=(0,(0,0))$	starting state
α_j	action to keep exactly j operational servers
$\mathcal{P}_{ss'}^{\alpha_j}$	probability transition from s to s' under action α_j
$\mathcal{C}_s^{\alpha_j}$	immediate cost from s under action α_j

Example 2. *To illustrate our formalization let's show an MDP for a very simple data center of one server with a buffer size equals one. Which means that* $max=1$ *and* $b=1$. *Job arrivals are modeled by histogram of Example 1. So, MDP=* $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{C})$ *where:*

$$\begin{cases} \mathcal{S}=\{(0,(0,0)),(0,(1,0)),(0,(1,1)),(1,(0,0)) \\ ,(1,(1,0)),(1,(1,1))\} \\ \mathcal{A}=\{\alpha_0,\alpha_1\} \end{cases}$$

and $\mathcal{P}$ *can be deduced from the graph of Figure 2.*

4 OPTIMAL STRATEGY STRUCTURE

As we formulate our optimization problem as an MDP, an action consists in turning-on each unit of time a specific number of servers and turning-off the rest of the servers. The optimal strategy is the best sequence of actions to be done in order to minimize the overall cost during a finite period of time called *horizon* and noted h. More generally, the value function

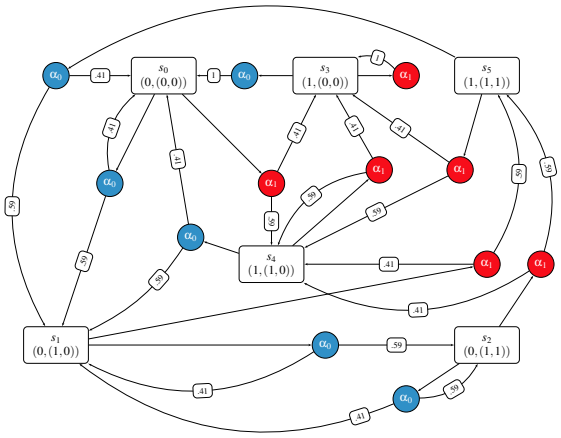

Figure 2: MDP example. For instance, state $s_3=(1,(0,0))$ means that the system is running by one server, and no waiting jobs are in the buffer nor lost jobs. Action α_0 switches-off the server and α_1 switches-on the server.

$V:\mathcal{S}\times[0..h]\rightarrow\mathbb{R}^+$ has as objective minimizing the expected sum of costs over time:

$$V(s,t)=\min_{\pi}\mathbb{E}\left[\sum_{k=0}^{t}\mathcal{C}_{s_k}^{\pi(s_k,k)}\right] \quad (5)$$

The value function can be seen as a Bellman equation (Bellman, 1957; Puterman, 1994; Bertsekas, 1995):

$$V(s,t)=\min_{\alpha_j}\left\{\mathcal{C}_s^{\alpha_j}+\sum_{s'\in\mathcal{S}}\mathcal{P}_{ss'}^{\alpha_j}V(s',t-1)\right\} \quad (6)$$

Where α_j is the action taken by the system, and $\mathcal{P}_{ss'}^{\alpha_j}$ is the transition probability from state s to state s'. In this case the optimal policy for each state s is:

$$\pi^*(s,t)=\underset{\alpha_j\in\mathcal{A}}{\operatorname{argmin}}\left\{\mathcal{C}_s^{\alpha_j}+\sum_{s'\in\mathcal{S}}\mathcal{P}_{ss'}^{\alpha_j}V(s',t-1)\right\} \quad (7)$$

As the value iteration algorithm is an efficient dynamic programming, implementation for solving Bellman equation, we used it to compute the optimal control policy of our MDP.

As shown previously, the size of the MDP is important, and the computation of the optimal policy can be hard even impossible for a big data center. In fact, it is essential to analyze the structural properties of the optimal policy to make the computation efficient. In the following we will be interested in some properties around the optimal policy, and we show in particular that the property of the double-threshold structure does not hold for our heterogeneous data center model.

Definition 2 ((Hipp and Holzbaur, 1988)). *A policy* π *is called to be* **hysteretic** *if* $\forall t\in[0..h]$, $\exists m\in$

$[0..max]$, $\exists \alpha_j \in \mathcal{A}$ *such as* $\pi((m,(n,l)),t) = \alpha_j \implies \pi((j,(n,l)),t) = \alpha_j$.

Theorem 1. *The optimal policy (7) is hysteretic.*

Proof. For each slot t, let's define function $f_t(m,j) : [0..max] \times [0..max] \to \mathbb{R}^+$ as the cost for switching the number of operational servers from m to j. And function $w_t(m,(n,l)) : \mathcal{S} \to \mathbb{R}^+$ presents the expected and the possibly future cost starting with n waiting jobs in the buffer, losing l jobs, serving with m operational servers during one slot, and then following an optimal policy. So, the our optimal policy (7) can be formulated as:

$$\pi^*(s,t) = \underset{\alpha_j \in \mathcal{A}}{\operatorname{argmin}} \{ f_t(m,j) + w_t(j,(n,l)) \} \tag{8}$$

Where:

$$\begin{cases} w_t(j,(n,l)) & = j \times c_M + n \times c_N + l \times c_L + \\ & \quad \sum_{s'} \mathcal{P}_{ss'}^{\alpha_j} V(s',t) \\ f_t(m,j) & = \max\{0, j-m\} \times c_{On} \end{cases} \tag{9}$$

According to Theorem 1 of (Hipp and Holzbaur, 1988), if the function f_t satisfies the following condition:

$$\begin{cases} \forall m \in [0..max] : f_t(m,m) = 0 \\ \forall m,p,q \in [0..max] : f_t(m,q) \leq f_t(m,p) + f_t(p,q) \end{cases} \tag{10}$$

then the optimal policy π^* is a hysteretic policy. Thus, to prove that our optimal policy is hysteretic we need just to prove that f_t satisfies conditions 10. We have $f_t(m,m) = \max\{0, m-m\} \times c_{On} = 0$ which implies the first condition of 10, and the following resumes all possible cases for the second condition of 10:

1. if $m \geq q$ we have $f_t(m,q) = 0$ and as f_t is positive then $\forall p \in [0..max] : f_t(m,p) + f_t(p,q) \geq 0 = f_t(m,q) \implies f_t(m,q) \leq f_t(m,p) + f_t(p,q)$.
2. if $m \leq p \leq q$ we have $f_t(m,p) + f_t(p,q) = (p-m) \times c_{On} + (q-p) \times c_{On} = (q-m) \times c_{On} = f_t(m,q) \implies f_t(m,q) \leq f_t(m,p) + f_t(p,q)$.
3. if $m \leq q \leq p$ we have $f_t(m,p) + f_t(p,q) = (p-m) \times c_{On} + 0 = (p-m) \times c_{On} \geq (q-m) \times c_{On} = f_t(m,q) \implies f_t(m,q) \leq f_t(m,p) + f_t(p,q)$.
4. if $p \leq m \leq q$ we have $f_t(m,p) + f_t(p,q) = 0 + (q-p) \times c_{On} = (q-p) \times c_{On} \geq (q-m) \times c_{On} = f_t(m,q) \implies f_t(m,q) \leq f_t(m,p) + f_t(p,q)$.

In conclusion, conditions 10 hold and our optimal policy is hysteretic. □

Definition 3 ((Lu and Serfozo, 1984))**.** *A policy* π *is called to be* **monotone** *if* π *is hysteretic and,* $\forall t \in [0..h]$, $\forall m \in [0..max]$, $\exists\, D_m, U_m \in [-1..b+1]$, *and* $D_m \leq U_m$ *such that for every* $s = (m,(n,l))$ *we have:* $\pi((m,(n,l)),t) =$

$$\begin{cases} \pi((\max\{0, m-1\},(n,l)),t) \text{ if } n < D_m \\ \alpha_m \text{ if } D_m \leq n \leq U_m \\ \pi((\min\{max, m+1\},(n,l)),t) \text{ if } U_m < n \end{cases} \tag{11}$$

Theorem 2. *The optimal policy (7) is not monotone.*

Proof. In this proof we give a counterexample that shows that monotony of the optimal policy (7) does not hold in general. Let's model the arrival by the histogram of Example 1. We set $b = 5$, $max = 5$, $c_M = 9$, $c_N = 8$, $c_{On} = 7$, $d = 1$, and $h = 7$. In order to simplify the counter example[2], we set $c_L = 0$ so we don't need to consider rejected job in the MDP model. Under this parameters, solving the optimality equation 6 leads to the following optimal policy for $t = 5$:

Observed state $(m,(n,l))$		**Action**
$(m=0,(n=0,l=0))$	$\to$	α_1
$(m=0,(n=1,l=0))$	$\to$	α_2
$(m=0,(n=2,l=0))$	$\to$	α_2
$(m=0,(n=3,l=0))$	$\to$	α_3
$(m=0,(n=4,l=0))$	$\to$	α_4
$(m=0,(n=5,l=0))$	$\to$	α_5

Observed state $(m,(n,l))$		**Action**
$(m=1,(n=0,l=0))$	$\to$	α_1
$(m=1,(n=1,l=0))$	$\to$	α_1
$(m=1,(n=2,l=0))$	$\to$	α_2
$(m=1,(n=3,l=0))$	$\to$	α_2
$(m=1,(n=4,l=0))$	$\to$	α_3
$(m=1,(n=5,l=0))$	$\to$	α_4

Observed state $(m,(n,l))$		**Action**
$(m=2,(n=0,l=0))$	$\to$	α_1
$(m=2,(n=1,l=0))$	$\to$	α_1
$(m=2,(n=2,l=0))$	$\to$	α_1
$(m=2,(n=3,l=0))$	$\to$	α_2
$(m=2,(n=4,l=0))$	$\to$	α_2
$(m=2,(n=5,l=0))$	$\to$	α_3

Observed state $(m,(n,l))$		**Action**
$(m=3,(n=0,l=0))$	$\to$	α_1
$(m=3,(n=1,l=0))$	$\to$	α_1
$(m=3,(n=2,l=0))$	$\to$	α_1
$(m=3,(n=3,l=0))$	$\to$	α_1
$(m=3,(n=4,l=0))$	$\to$	α_2
$(m=3,(n=5,l=0))$	$\to$	α_3

[2]The counter example holds even for some positive value of c_L.

Observed state $(m,(n,l))$		Action
$(m=4,(n=0,l=0))$	$\rightarrow$	α_1
$(m=4,(n=1,l=0))$	$\rightarrow$	α_1
$(m=4,(n=2,l=0))$	$\rightarrow$	α_1
$(m=4,(n=3,l=0))$	$\rightarrow$	α_1
$(m=4,(n=4,l=0))$	$\rightarrow$	α_1
$(m=4,(n=5,l=0))$	$\rightarrow$	α_2

Observed state $(m,(n,l))$		Action
$(m=5,(n=0,l=0))$	$\rightarrow$	α_1
$(m=5,(n=1,l=0))$	$\rightarrow$	α_1
$(m=5,(n=2,l=0))$	$\rightarrow$	α_1
$(m=5,(n=3,l=0))$	$\rightarrow$	α_1
$(m=5,(n=4,l=0))$	$\rightarrow$	α_1
$(m=5,(n=5,l=0))$	$\rightarrow$	α_1

It is clear that lower and upper thresholds of the optimal policy for $m=2$ should be $D_2=3$ and $U_2=4$. If the policy is monotone, the following must hold:

if $n<D_2 \implies \pi^*((m=2,(n,l)),t)=\pi^*((m=1,(n,l)),t)$. Let's check that for $n=2$. We have $n<D_2$, if the optimal policy is monotone we have to conclude that $\pi^*((m=2,(n=2,l)),t)=\pi^*((m=1,(n=2,l)),t)$. Unfortunately this last equality does not hold, because from the above tables we have $\pi^*((m=2,(n=2,l)),t)=\alpha_1$ however: $\pi^*((m=1,(n=2,l)),t)=\alpha_2$. □

Corollary 1. *The optimal policy (7) is not isotone.*

Proof. From Definition of isotony of (Serfozo, 1979), monotony is a necessary condition for isotony. However form Theorems 1 and 2, we deduce that the optimal policy (7) is not isotone. □

5 EXPERIMENTAL RESULTS

To model arrivals, this work uses real traffic traces based on the open *clusterdata-2011-2* trace (Wilkes, 2011). We focus on the part that contains the job events corresponding to the requests destined to a specific Google data center for the whole month of May 2011. The job events are organized as a table of eight attributes; where column *timestamps* refers to the arrival times of jobs expressed in μ-sec. This traffic trace is sampled with a sampling period equal to the slot duration. Thus, we consider frames of one minute to sample the trace and construct one empirical distribution. The obtained distributions are formed respectively of a number of bins between 20 and 100. And the average of arrival jobs is around 46 jobs per slot. As PRISM can be used for the specification and analysis of a Markov decision process (MDP) model, in this section we use this probabilistic model-checker software tool to perform our experimentations. In the following of this section we use PRISM to model and then analyze data centers with various parameters shown in Table 2. Notice that the buffer sizes are small because to avoid a hug PRISM model.

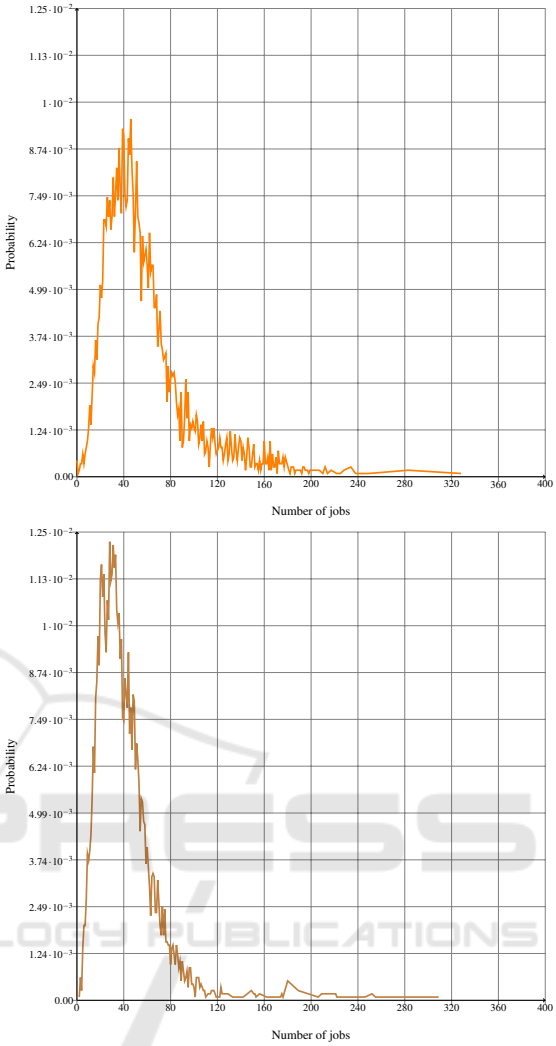

Figure 3: Example of arrival jobs distribution P_A for two different days.

Table 2: Settings of different data centers parameters.

Setting	DC	# servers	Buffer	# bins
First	Small	$max=21$	$b=31$	$\lvert S_A\rvert=23$
Second	Medium	$max=51$	$b=59$	$\lvert S_A\rvert=47$
Third	Big	$max=103$	$b=111$	$\lvert S_A\rvert=98$

An analysis period of a day ($h=1440$ slots) is considered. Initially, in order to have an experimentation in which the importance of energy and QoS is the same, we set the unitary costs to the same value:

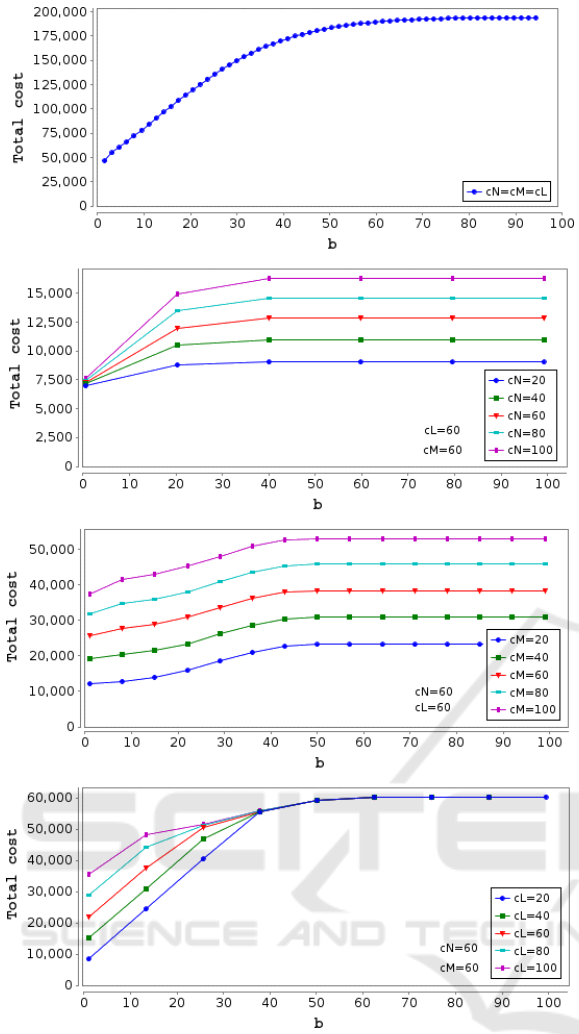

Figure 4: Value of total cost when varying buffer size b for several values of c_M, c_N and c_L.

$c_M = c_N = c_L$. Figure 4 resumes the obtained results for this configuration. However, we have done other experimentations to analyze the impact of varying the unitary costs c_M, c_N, and c_L on the total cost. Figure 4 resume the obtained results. The first sub-figure of Figure 4 shows the result of experiments in which we are analyzing the total cost over one day when varying the buffer size b. The second sub-figure presents analysis for several values of c_N where keeping c_M and c_L constants. The third sub-figure shows an analysis for several values of c_M where keeping c_N and c_L constants. The last sub-figure shows an analysis for several values of c_L where keeping c_N and c_M constants.

Observation 1. *Total cost increase when b is less than some value. However, when b is bigger than this value the total cost seams to be convergent.*

We can explain this behavior as following: for a small size of the buffer, the number of waiting jobs is low. This leads to a small number of served jobs. In fact the system switches-on a less number of servers. So the energetic and also the waiting jobs are low. When the buffer size is bigger, the number of waiting jobs is more important. Which leads to a bigger number of served jobs. In fact the system switches-on more servers. So the energetic and also the waiting jobs increase. As the arrival jobs are bounded, when the buffer size is bigger than some value, the number of waiting jobs and the number of served job become stationary which leads to a stationary number of running servers and necessarily to a stationary total cost.

Observation 2. *For big values of b, unitary cost for loosing jobs c_L does not affect the behavior of the system, especially the total cost.*

We can explain that by the fact that large value of b leads to a low rejection rate. In fact the total cost will not increase so much even if we increase c_L. However, for small value of b the rejection rate is more important and increasing c_L leads to a higher overall cost.

6 CONCLUSION

In this work we present an approach based on discrete-time Markov Decision Process to save energy in a data center. The system is modeled by a a queue where job arrivals are modeled by histograms obtained from an empirical trace. The optimal control policy is computed by value iteration algorithm. This optimal policy is used to define the Dynamic Power Management to insure the trade-off between performance and energy consumption. We prove that the optimal policy is hysteretic but neither isotone nor monotone, consequently, the optimal policy can not be designed as a double-threshold structure.

REFERENCES

Alagoz, O. and Ayvaci, M. U. Uniformization in markov decision processes. *Wiley Encyclopedia of Operations Research and Management Science*.

An, H. and Ma, X. (2022). Dynamic coupling real-time energy consumption modeling for data centers. *Energy Reports*, 8:1184–1192.

Bayati, M. (2016). Managing energy consumption and quality of service in data centers. In *Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems*, pages 293–301.

Bayati, M., Dahmoune, M., Fourneau, J., Pekergin, N., and Vekris, D. (2016). A tool based on traffic traces and stochastic monotonicity to analyze data centers and their energy consumption. *EAI Endorsed Trans. Energy Web*, 3(10):e3.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition.

Benini, L., Bogliolo, A., Paleologo, G. A., and De Micheli, G. (1999). Policy optimization for dynamic power management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):813–833.

Benson, T., Akella, A., and Maltz, D. A. (2010). Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 267–280. ACM.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA.

Dyachuk, D. and Mazzucco, M. (2010). On allocation policies for power and performance. In *2010 11th IEEE/ACM International Conference on Grid Computing*, pages 313–320. IEEE.

Gebrehiwot, M. E., Aalto, S., and Lassila, P. (2016). Optimal energy-aware control policies for fifo servers. *Performance Evaluation*, 103:41–59.

Greenberg, A. G., Hamilton, J. R., Maltz, D. A., and Patel, P. (2009). The cost of a cloud: research problems in data center networks. *Computer Communication Review*, 39(1):68–73.

Grunwald, D., Morrey, III, C. B., Levis, P., Neufeld, M., and Farkas, K. I. (2000). Policies for dynamic clock scheduling. In *Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation - Volume 4*, OSDI'00, pages 6–6, Berkeley, CA, USA. USENIX Association.

Hipp, S. K. and Holzbaur, U. D. (1988). Decision processes with monotone hysteretic policies. *Operations Research*, 36(4):585–588.

Jin, Y., Wen, Y., and Chen, Q. (2012). Energy efficiency and server virtualization in data centers: An empirical investigation. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 133–138. IEEE.

Khan, W. (2022). Advanced data analytics modelling for evidence-based data center energy management.

Koomey, J. (2011). Growth in data center electricity use 2005 to 2010. *A report by Analytical Press, completed at the request of The New York Times*, page 9.

Lee, Y. C. and Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 60(2):268–280.

Lu, F. and Serfozo, R. F. (1984). M/m/1 queueing decision processes with monotone hysteretic optimal policies. *Operations Research*, 32(5):1116–1132.

Maccio, V. J. and Down, D. G. (2015). On optimal control for energy-aware queueing systems. In *Teletraffic Congress (ITC 27), 2015 27th International*, pages 98–106. IEEE.

Mazzucco, M. and Dyachuk, D. (2012). Optimizing cloud providers revenues via energy efficient server allocation. *Sustainable Computing: Informatics and Systems*, 2(1):1–12.

Mazzucco, M., Dyachuk, D., and Dikaiakos, M. (2010). Profit-aware server allocation for green internet services. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, pages 277–284. IEEE.

Mitrani, I. (2013). Managing performance and power consumption in a server farm. *Annals OR*, 202(1):121–134.

Patel, C. D., Bash, C. E., Sharma, R., and Beitelmal, M. (2003). Smart cooling of data centers. In *Proceedings of IPACK*.

Peng, X., Bhattacharya, T., Mao, J., Cao, T., Jiang, C., and Qin, X. (2022). Energy-efficient management of data centers using a renewable-aware scheduler. In *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 1–8. IEEE.

Plum, H.-J. (1991). Optimal monotone hysteretic markov policies in anm/m/1 queueing model with switching costs and finite time horizon. *Mathematical Methods of Operations Research*, 35(5):377–399.

Puterman, M. L. (1994). *Markov Decision Processes. J.* Wiley and Sons.

Rajesh, C., Dan, S., Steve, S., and Joe, T. (2008). Profiling energy usage for efficient consumption. *The Architecture Journal*, 18:24.

Rteil, N., Burdett, K., Clement, S., Wynne, A., and Kenny, R. (2022). Balancing power and performance: A multi-generational analysis of enterprise server bios profiles. In *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, pages 81–85. IEEE.

Schwartz, C., Pries, R., and Tran-Gia, P. (2012). A queuing analysis of an energy-saving mechanism in data centers. In *Information Networking (ICOIN), 2012 International Conference on*, pages 70–75.

Serfozo, R. F. (1979). Technical note—an equivalence between continuous and discrete time markov decision processes. *Operations Research*, 27(3):616–620.

Tancrez, J.-S., Semal, P., and Chevalier, P. (2009). Histogram based bounds and approximations for production lines. *European Journal of Operational Research*, 197(3):1133–1141.

Topkis, D. M. (1978). Minimizing a submodular function on a lattice. *Operations research*, 26(2):305–321.

Wilkes, J. (2011). More Google cluster data. Google research blog. Posted at http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.

Yang, Z., Chen, M.-H., Niu, Z., and Huang, D. (2011). An optimal hysteretic control policy for energy saving in cloud computing. In *Global Telecommunications Conference*, pages 1–5. IEEE.