Performance Evaluation 90 (2015) 36–52

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

On optimal policies for energy-aware servers

V.J. Maccio, D.G. Down *

McMaster University, Hamilton, Ontario, Canada

ARTICLE INFO

Article history:
Received 7 June 2014
Received in revised form 7 April 2015
Accepted 21 April 2015
Available online 4 May 2015

Keywords:
Green computing
Energy-aware server
Queueing analysis
Routing

ABSTRACT

As energy costs and energy used by server farms increase, so does the desire to implement energy-aware policies. Although under some metrics, optimal policies for single as well as multiple server systems are known, a number of metrics remain without sufficient knowledge of corresponding optimal policies. We describe and analyse a model to determine an optimal policy for on/off single server systems under a broad range of cost functions that are based on expected response time, expected energy costs, and expected wear and tear costs. We leverage this model in the problem of routing jobs to one of two servers to show a range of non-trivial optimal routing probabilities and server configurations when energy concerns are a factor.

1. Introduction

The relative as well as absolute energy consumed by servers have been steadily increasing in North America and has become a problem of considerable interest [1]. As systems grow and expand, energy concerns become a major factor for server farm managers from both environmental and economic viewpoints. However, the task of creating feasible optimal or near optimal policies is a daunting problem due to the sheer complexity these systems exhibit. Even for single server systems, when energy is a factor, optimal policies remain unknown for a number of cost functions considered in the literature. We focus on developing a model in the context of and using tools and results from queueing theory, that allows one to determine an optimal policy for a single server system under a broad range of metrics. In particular, we consider cost functions constructed from the expected response time of a job in the system ($\mathbb{E}[R]$), the expected energy consumed by the system ($\mathbb{E}[E]$), and the steady state rate that the server cycles between two states, i.e. turning off and on ($\mathbb{E}[C]$), where the expected cycle rate can be thought as the expected wear and tear on the server. This paper extends the work [2].

By now, the field of green computing has a rich literature. We will focus our discussion on work which is concerned with moving servers to different energy states to increase (or decrease) performance. For example, the work in [3–5] looked at determining the optimal configuration of a server farm when the job sizes are known at arrival, and the decision to turn servers off or keep them on is made at discrete time intervals. This is then formulated as an optimization problem and solved. The article [3] was the first to appear and accounts for wear and tear cost on the servers by allowing for a term similar to $\mathbb{E}[C]$ in the cost function. The article [6] added the variation that jobs can be routed to different geographical locations where energy costs may differ. The work in [4] took a different viewpoint where customers pay a cost, based on a function of the response time of that job. The work of [7] looked at a similar problem where jobs are routed to separate on/off queues, and the problem was solved using Markov Decision Processes (MDPs). In this field the issue of speed scaling also arises, where one can use more energy to improve the performance (response time) of the system. This is examined in [5,8]. While

---

* Corresponding author.
E-mail address: downd@mcmaster.ca (D.G. Down).

http://dx.doi.org/10.1016/j.peva.2015.04.002

these frameworks use stochastic models and results, our model puts much heavier emphasis on analysing these systems in a queueing theory context.

When analysing these systems as continuous time Markov chains (CTMCs), they can often be viewed as or reduced to some form of a vacation model, where a vacation is interpreted as the server being off. Many of these models are considered in [9–11]. It will be seen that our work has elements that are of a similar spirit, i.e. the decomposition of system metrics. The work of [12] looked at specialized vacation models capturing multi-server behaviours, although this is done for specific policies which in general do not capture the optimal policy. To the best of our knowledge, the vacation model which most closely relates to the model analysed in this work is that described in [13]. In general the vacation model presented there is not optimal under the previously mentioned metrics. However, due to a closer coupling to the arrival process, our model allows one to describe the optimal policy.

While all previously mentioned works are related to this paper, the research which is most similar is [14–18]. They model these systems as CTMCs in both the single server and multi-server settings and analyse them under the specific metric which they refer to as the Energy Response Product (ERP), which unsurprisingly is defined to be the product of the expected response time, and expected energy consumed. With several new techniques (such as the RRR technique described in [14]) and observations, they are often able to arrive at closed form expressions even for multi-server systems, albeit under specific policies. Furthermore, they are able to arrive at the optimal policy for single server systems, however this is due to some convenient properties of the ERP cost function.

Our contributions offer a deeper understanding of the optimal policy for single server energy-aware systems, and are as follows.

- We perform our analysis under a large family of cost functions, based on the expected response time, expected number of jobs in the system, expected energy used, the expected turn-off rate of the server, and also the expectations of the product of these metrics. Furthermore, our analysis allows one to determine the optimal policy under any of these previously mentioned cost functions.
- We give an explicit solution to the underlying CTMC for our model. To the best of our knowledge, this CTMC has not been previously solved.
- We extend our analysis to considerable generality with respect to the underlying distributions. That is, we offer closed form solutions for all of our cost function metrics, under completely general server setup times, and job processing time distributions. We also offer several insightful observations pertaining to these metrics and how different system configurations relate to them.
- We offer several applications of our model. This includes applying our results to a multi-server system with random routing to show that in general when energy concerns are a factor, classical load balancing may be far from optimal.

The organization of this paper is as follows. In Section 2 a formal model of the system is presented. We continue by giving a detailed analysis of this model in Section 3. We firstly impose the assumption that all underlying distributions are exponential and therefore the model can be analysed as a CTMC. We progressively relax these assumptions and analyse the model under almost complete generality, offering a variety of insights and results. Section 4 gives several applications of our model while Section 5 shows how the model can be applied to a two server random routing setting.

2. Model

We wish to capture the behaviour of a single server system, where the server can be dynamically set to a low or high state. Furthermore, we wish to add the restriction that jobs may only be processed when the server is in its higher state. Such a system is modelled as being in one of four energy states: *LOW, SETUP, BUSY,* or *IDLE*. Each of these energy states has a corresponding rate of energy consumption E_{Low}, E_{Setup}, E_{Busy}, and E_{Idle}, respectively. For simplicity of analysis and understanding, if $E_{Low} = 0$, we rename *LOW* to *OFF*. We will see that optimal policies typically depend on the ratios of the energy costs rather than the values themselves. Therefore, we take these ratios with respect to E_{Busy}, and denote them as r_{Low}, r_{Setup}, and r_{Idle}, where $r_{Idle} < 1$. For the remainder of this paper we will often refer to moving to a higher or lower state as turning the server on or off, respectively. For further ease of reading, we often abuse our nomenclature for the energy states. For example we may refer to the server or system being *IDLE, OFF,* etc., where we implicitly mean that the server or system is in the energy state *IDLE, OFF,* etc.

Jobs arrive to a FIFO queue according to a Poisson process of rate λ. If the system is *LOW/OFF* when a job arrives, it checks how many jobs are currently waiting in the queue. If the number in the queue plus the arriving job is equal to a given threshold k, the system moves into *SETUP*. This corresponds to the server turning on. The time it takes to make this transition is exponentially distributed with rate γ. Once the server has completed making its transition, it leaves *SETUP* and becomes *BUSY*. Once *BUSY*, the server begins to process the accumulated jobs. The job processing times are exponentially distributed with rate μ. When a job is completed and no jobs remain in the queue, the system becomes *IDLE*. Once *IDLE*, the system begins to accumulate idling time, since the last time it was turned on. If no job arrives to the system once the server has accumulated a given amount of idling time, the system becomes *LOW/OFF*. If a job arrives while the system is *IDLE*, it becomes *BUSY*. The amount of idling time which will be accumulated before the system becomes *LOW/OFF*, is exponentially distributed with rate α. It is important to note that the time spent in *IDLE* before transitioning to *OFF* is not the time which it takes for a server to turn off, but rather the amount of time it is willing to wait before deciding to turn off. For the purpose