Queueing Models and
Service Management
Vol. 3, No. 1, page 111-135, 2020

QMSM
© PU 2018

Asymptotic Performance of an Energy-Aware G/G/C Queue with General Setup Times

Vincent J. Maccio[1] and Douglas G. Down[2,*]

[1]Department of Mathematical and Computational Sciences
University of Toronto, Mississauga, Ontario L5L 1C6, Canada
[2]Department of Computing and Software
McMaster University, Hamilton, Ontario L8S 4L7, Canada

(Received June 2019; accepted February 2020)

Abstract: An intuitive solution to address the immense energy demands of datacentres is to turn servers off to incur less costs. However, when to turn a specific server off and when to then turn that server back on are far from trivial questions. As such, many different authors have modeled this problem as an $M/M/C$ queue where each server can be turned on with an exponentially distributed setup time or turned off instantaneously. Due to the complexity of the model analysis, authors often examine a specific policy. Moreover, different authors examine different policies under different cost functions. This in turn causes difficulties when making statements or drawing conclusions regarding competing policies. We analyse this well established model under the asymptotic regime where the number of servers approaches infinity, i.e. $C \to \infty$, while the load remains fixed, i.e. $0 < \lambda/(C\mu) < 1$. Furthermore, we relax the assumptions regarding the underlying exponential distributions. That is, we consider a $G/G/C$ queue that has generally distributed setup times. To address the issue of comparing competing policies, it is shown that under the aforementioned asymptotic regime and generality, not only are many of the policies in the literature equivalent, but they are also optimal under any cost function which is non-decreasing in the expected energy cost and response time.

Keywords: Asymptotic analysis, energy-aware systems, many server limits, multiserver queues.

1. Introduction

Over the past several years, energy concerns in datacentres have driven an interest in queueing systems where individual servers can be turned on to improve performance and turned off to save on costs. This interest has led to different authors studying the same, or similar, queueing models. However, due to the complexity of the problem, i.e. the choice of cost function, policy implemented, model details, etc., different conclusions can be drawn from similar underlying problems. One consequence of the variety in the problems studied and the corresponding variety of insights is that it is difficult to confidently draw conclusions which are overarching across the problem domain. To address this issue this work presents a result under an asymptotic regime which states that when the system

*Corresponding author
Email : downd@mcmaster.ca

parameters are appropriately scaled up, a general class of policies is optimal under a reasonable family of cost functions.

To the best of our knowledge, Chen *et al.* [2] and Sledger *et al.* [24] were the first to apply queueing models in the context of energy-aware datacentres. This work introduced a popular queueing model which extends the traditional $M/M/C$ queue where each of the C servers can be switched on after a setup delay (to improve performance) and instantly switched off (to increase efficiency). The question remained however, when should servers be turned on and when should they be turned off. This question was and currently remains a topic of interest. Gandhi *et al.* [3-6] produced a body of work examining this model under the *staggered setup* policy, where the number of jobs in the system is equal to the number of servers on and in setup when possible, and servers turn off when idle. Furthermore, they also studied the *delayed off* policy, which extends the staggered setup policy by allowing an idle server to wait an exponentially distributed period of time before it turns off. Mitrani [18] studied this model where a reserved set of servers were brought into setup when the number of jobs in the system exceeds a threshold and then turns those servers off once the number of jobs drops below another threshold. This policy was further studied in Hu and Phung-Duc [8]. Xu and Tian [26] examined the model where e servers will turn off when d servers idle.

From here more sophisticated policies began to emerge. Specifically, the use of a threshold decision variable regarding the number of jobs in the system has often been employed. In [16] and [13], we derived several structural properties pertaining to the optimal policy to allow for more confidence in existing policies, as well as to create guidelines when determining a new policy to study. Kuehn and Mashly [12] looked at the system which turns on servers if some threshold k is met and turns them off when idle, under the constraint of a finite buffer. In [16] and [14], we studied two policies both of which allow for a static provisioning of servers (servers which are always on) and the remaining servers turn off the moment they idle. The first of the policies examined was the *bulk setup* policy, which begins the setup process of all available servers simultaneously, cancelling the remaining setups once one server has turned on. The second, *staggered threshold*, is in the same vein as [12]. Informally, a dedicated number of servers always remain on, and of the remaining servers, the ith server will begin its setup process once ik jobs have accumulated in the queue, where k is a specified threshold. Phung-Duc [22] has also considered this model where he analysed the policies studied by Gandhi et al., and Phung-Duc and Kawanishi [20] examined the $s-$*staggered setup* policy which turns on s servers for each job waiting in the queue. More recently Phung-Duc and Kawanishi [21] analysed this system with abandonments and derived the waiting time distribution. Moreover, Ren *et al.* [23] also studied the staggered threshold policy under a slight modification, in the context of virtual machines. This model also has applications to or is studied in problems which arise in other fields such as manufacturing, logistics, and vacation models [1, 17, 25].

As mentioned previously and as one may infer from the literature review, an existing problem is that analysing a specific policy, or family of policies, is far from trivial. As such, the examination and analysis of a single policy could span an entire work. Moreover, when evaluating one of these policies it may do well for a specific cost function but poorly under another. Therefore, saying one policy is strictly better than another can be a difficult claim to justify. With the goal in mind for one to be able to make broad claims across a large set of policies and cost functions, this work makes the following contributions:

1. It is shown that as the number of servers and arrival rate are appropriately scaled to infinity such that the load on the system remains fixed at some value less than one, then there exists a set of policies such that any policy belonging to this set will simultaneously minimize both the expected response time and expected energy costs, and therefore will be optimal under all cost functions which are non-decreasing in those metrics. This family has an unbounded number of members.
2. This set of policies is formally described and observed to include many of the policies currently examined in the literature and discussed previously in this section.
3. A formal proof is provided that the previous two points hold in great generality, i.e. when the interarrival, service, and setup times of the model are generally distributed.
4. Numerical experiments are conducted to determine how quickly the asymptotic behaviours of these systems are reached, and it is shown that particular choices of policy parameters may be used to induce faster convergence to the minimum values.

This work in an extension of [15], where we were able to show the same formal results for the exponential case only. A similar asymptotic approach has been employed by Mukherjee *et al.* [19]. They consider a model that consists of C queues in parallel, where a routing decision must be made upon a job's arrival to the system and all underlying distributions are exponential. They identify a combined routing and server control policy that is optimal in the fixed-load, many-server regime. We consider a system with a central queue, so that the server control decisions can be coordinated. The combination of the work presented here and the work in [19] provides a complete picture of the control problem (at least asymptotically) for both central and parallel queue architectures.

2. Model

The model under study is a $G/G/C$ queue where each server can be switched on and off, and where turn-offs are instantaneous, but turn-ons take a generally distributed setup time. This is described formally as follows. Jobs arrive to a central queue, where interarrival times are independent and identically distributed with finite mean $1/\lambda$ and finite, nonzero variance. These jobs are processed on a first come first served basis and have processing times (job sizes) which are independent and identically distributed with finite mean $1/\mu$ and finite, nonzero variance. Furthermore, there are C homogeneous servers present, each of which can be in one of four energy states: *off, setup, idle*, or *busy*. For ease of exposition this work often refers to a server being busy, idle, off, or in setup as shorthand for a server being in the corresponding energy state. Furthermore, it is often convenient to refer to a server as simply being *on*. For the remainder of this work when we say a server is on, we are saying it is in the energy state *idle* or *busy*. Regarding definitions and transitions, a server is *idle* if and only if it is on and not processing a job. Moreover, a server can only begin serving a job if it is currently *idle*, in which case the server becomes *busy*. At any time, a server can be switched *off*. Regarding the process of turning a server on, an *off* server can transition to *setup*. Once in *setup*, the server will remain there for a time generally distributed with finite mean $1/\gamma$ and finite, nonzero variance. When the setup is complete the server will become *idle*, at this moment it may instantly transition to *busy* if there is a job waiting to be served. The setup times are assumed to be independent and identically distributed. A system which meets the criteria of the above model is said to be a *general energy-aware system*.

This work refers to a general energy-aware system as a four-tuple $(C, \lambda, \mu, \gamma)$, where C is the number of servers, and λ, μ, and γ are the arrival, processing, and setup rates, respectively. The system load ρ is defined as $\rho = \lambda/(C\mu)$. Moreover, the well known $G/G/C$ queue is referred to by a three-tuple (C, λ, μ), with the traditional interpretation of those parameters. As such, one may view a general energy-aware system as an extension of a $G/G/C$ queue. To fully understand how a specific general energy-aware system behaves, one must also know when, or how it is determined when, each server is turned on and off. Such a description of the server behaviour is referred to as a policy. In this work a specific policy is denoted by π. Some examples of a policy π are: the number of servers that are on or in setup equals the number of jobs in the system, turn on all servers once there are k jobs in the system and turn them off when they idle, keep all servers on all the time, etc. It is natural to want to compare such policies against each other. That is, to determine whether one policy is better than another.

In order to compare different policies, one must have metrics to evaluate. This work examines the trade-off between efficacy and efficiency. The expected response time, denoted by $\mathbb{E}[R]$ is employed to evaluate efficacy, while the expected energy cost to process a single job, or simply the expected energy cost, denoted by $\mathbb{E}[E]$, is employed to evaluate efficiency. The expected response time is the expected amount of time a job spends in the system, from arrival to departure. The expected energy cost takes a little more care to define. Each of the energy states (*off, idle, busy*, and *setup*) have a corresponding power consumption. Let these rates be denoted by $\mathcal{P}_{\text{Off}}$, $\mathcal{P}_{\text{Idle}}$, $\mathcal{P}_{\text{Busy}}$, and $\mathcal{P}_{\text{Setup}}$, respectively.

Furthermore, let the random variables C_{Off}, C_{Idle}, C_{Busy}, and C_{Setup} denote the number of servers which are off, idle, busy, or in setup, respectively. Then, letting $\mathbb{E}[\mathcal{P}]$ be the expected power used by the system:

$$\mathbb{E}[\mathcal{P}]=\mathcal{P}_{\text{Off}}\mathbb{E}[C_{\text{Off}}]+\mathcal{P}_{\text{Idle}}\mathbb{E}[C_{\text{Idle}}]+\mathcal{P}_{\text{Busy}}\mathbb{E}[C_{\text{Busy}}]+\mathcal{P}_{\text{Setup}}\mathbb{E}[C_{\text{Setup}}] \tag{1}$$

and

$$\mathbb{E}[E]=\frac{\mathbb{E}[\mathcal{P}]}{\lambda}. \tag{2}$$

Without loss of generality, it is assumed that $\mathcal{P}_{\text{Busy}}=1$, and the remaining rates are appropriately normalized. Furthermore, it is also assumed that $\mathcal{P}_{\text{Idle}}<\mathcal{P}_{\text{Setup}}$, $\mathcal{P}_{\text{Idle}}<\mathcal{P}_{\text{Busy}}$, and $\mathcal{P}_{\text{Off}}=0,$ although the latter could be relaxed to account for lower powered states where the server cannot process jobs, e.g. sleep states.

With the cost metrics defined, one can then create a cost function dependent on these metrics and begin to compare policies. There is no bound on the number of cost functions which can be defined, but common cost functions do arise in the literature, e.g. $\mathbb{E}[R]\mathbb{E}[E]$ and $\mathbb{E}[R]+\beta\mathbb{E}[E]$. Due to the diversity of the set of possible cost functions, conclusions which span across many cost functions are often difficult to make. Moreover, niche behaviours are often easy to invoke by tweaking parameters within a cost function, such as β in $\mathbb{E}[R]+\beta\mathbb{E}[E]$. For example, one could imagine a system with a small number of servers where the policy which keeps all servers on would be optimal when β is small, since this minimizes the expected response time. On the other hand, when β is large, the optimal policy for the same system may be one where servers are kept off for long periods of time while they wait for a large number of jobs to accumulate before expending the energy to turn on. As such, this work strives to draw conclusions applicable to large sets of cost functions, and in fact does so for all *well-formed cost functions*.

Definition 1. ***Well-Formed Cost Function:*** *A cost function $\mathcal{C}(\cdot)$ is a well-formed cost function if it is non-decreasing in, dependent on, and only dependent on, the expected response time, $\mathbb{E}[R]$, and the expected energy costs, $\mathbb{E}[E]$.*

As stated previously, the point of these cost functions is to allow the comparison of policies applied to the same general energy-aware system. Similar to how this work strives to make conclusions across a large set of cost functions, it also strives to make conclusions across large sets of policies. To this point, two important sets of policies are defined below.

Definition 2. ***Class A Policy:*** *A policy is said to be a Class A policy if the following conditions are met:*

1. Server setups are invoked following a threshold scheme.

2. A server will never turn off if it is busy, or if at the moment it completes processing *a job another job is waiting in the queue.*

To elaborate on the definition of a Class A policy, a threshold scheme pertaining to server turn-ons implies that each server i, $0 \le i \le C$, has a corresponding threshold value $k_{i,j}$, $0 \le j < i$, such that while there are j servers currently on, if the number of jobs in the system is greater than or equal to $k_{i,j}$ and server i is currently off, then server i begins its setup process, and if the number of jobs in the system is less than $k_{i,j}$ and server i is in setup, then it is switched off. It is worth noting that due to the homogeneity of the servers, from the subscript i and j one can infer how many servers are also in setup. As an example, if the system is turning on its third server while one server is already on, then the second server must also be in setup. We would argue that the properties of Class A policies are intuitively appealing, and moreover, from Theorems 1 and 2 in [16] it is known that optimal policies are contained within the set of Class A policies.

Before the second class of policies is given, another definition must first be introduced. Because this work examines the system as $C \to \infty$ it may be the case that while a policy gives a criterion to turn on some server s, the probability of this criterion being met approaches 0. Therefore to reason about these cases (and others), the following framework is introduced. Let $X_{\mathcal{E}}(s,t)$ be an indicator variable such that

$$X_{\mathcal{E}}(s,t) = \begin{cases} 1, & \text{if server } s \text{ is in energy state } \mathcal{E} \text{ at time t;} \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{E} \in \{\text{off, setup, idle, busy}\}$. Then it is said that s is an always $\mathcal{E}$ server if and only if as $t \to \infty$, $P(X_{\mathcal{E}}(s,t) = 1) \to 1$. As an example, if a server s has a criterion which turns it on and it is known that the server will always eventually turn off but the probability that the turn on criterion is met approaches 0 as $t \to \infty$, s would be called an always off server since as $t \to \infty$, $P(X_{\text{off}}(s,t) = 1) \to 1$. With these notions in mind the second class of policies is defined as follows.

Definition 3. ***Class B Policy:*** *A policy is said to be a Class B policy if the following conditions are met:*

1. *It is a Class A policy.*
2. *There exists an $\alpha < 1$ such that the number of always idle servers is less than $(1-\rho)C^{\alpha}$.*
3. *Each arrival can initiate at most a finite number of setups, each departure can cancel at most a finite number of setups, and each setup completion can cancel at most a finite number of setups.*

The second condition for Class B policies states that the number of servers which are always idle cannot be on the same order as the total number of servers. The third condition protects against known suboptimal behaviour, see Theorem 3 of [13]. That is, it is never the case that when a server switches off it immediately begins its setup process. It is worth noting that most policies studied in the literature are Class B policies, i.e. the policies of focus in [3-6, 12, 14, 16, 20-23, 26] are all Class B policies. There are some exceptions however. For example, the *bulk setup* policy first described in [13] is not a Class B policy since the number of setups which a single arrival can initiate is not finite (when the number of servers is infinite) and thus violates condition 3 of the definition. To elaborate further on condition 3 of the definition, it is there to guard against behaviours that work well under certain distributions but are problematic under others. Again, take the bulk setup policy given in [13] as an example. Here, when a job queues the system will switch all off servers into setup and then cancel all remaining setup processes once any server completes its setup process. This is a reasonable thing to do (in fact an optimal thing to do) when setup times are exponentially distributed, however, if setup times are degenerate (constant) such a policy could be far from optimal. The sets of Class A and Class B policies are denoted by Π_A and Π_B respectively. Furthermore, for a specific policy π, $\mathbb{E}[R^{\pi}]$ and $\mathbb{E}[E^{\pi}]$ denote the expected response time and expected energy costs under policy π, respectively.

3. Main Results

This work examines the behaviour of general energy-aware systems under a fixed-load, many-server asymptotic regime. This asymptotic regime is nontrivial, due to the fact that the servers can be turned on and off. In this regime, for a general energy-aware system $S=(C,\lambda,\mu,\gamma)$, the metrics $\mathbb{E}[R]$ and $\mathbb{E}[E]$ are evaluated as $C\to\infty$ while $\rho=\lambda/(C\mu)$ is held constant. Formally, we consider a sequence of energy-aware systems, indexed by n, where each system has a fixed ρ, and as $n\to\infty$, $C\to\infty$. All proofs of the results are given in Section 4.

Theorem 1. *All policies in Π_A are asymptotically optimal with regards to expected response time. In other words, given a general energy-aware system, for any $\pi_a\in\Pi_A$, as $\lambda, C\to\infty$ while μ is held constant and $\lambda/\mu C$ is fixed to be ρ, where $0<\rho<1$, $\mathbb{E}[R^{\pi_a}]\to 1/\mu$.*

While perhaps surprising at first, such a result becomes intuitive as one considers the details of the system behaviour. Informally, there is a significant proportion of jobs which are served immediately on arrival and therefore a significant proportion of jobs have a response time equal to their service time. And while it is true that some jobs will have to wait to be served, whether it be for a server to complete a job or finish a setup, the number of these jobs turns out to be negligible under the asymptotic regime. It is worth noting that

Theorem 1 would not necessarily hold for policies which turned servers off while there are waiting jobs that they could process. On the other hand, belonging to Π_A is not necessary for minimizing the expected response time. With optimal policies now known for $\mathbb{E}[R]$, our focus shifts to the second cost metric, $\mathbb{E}[E]$.

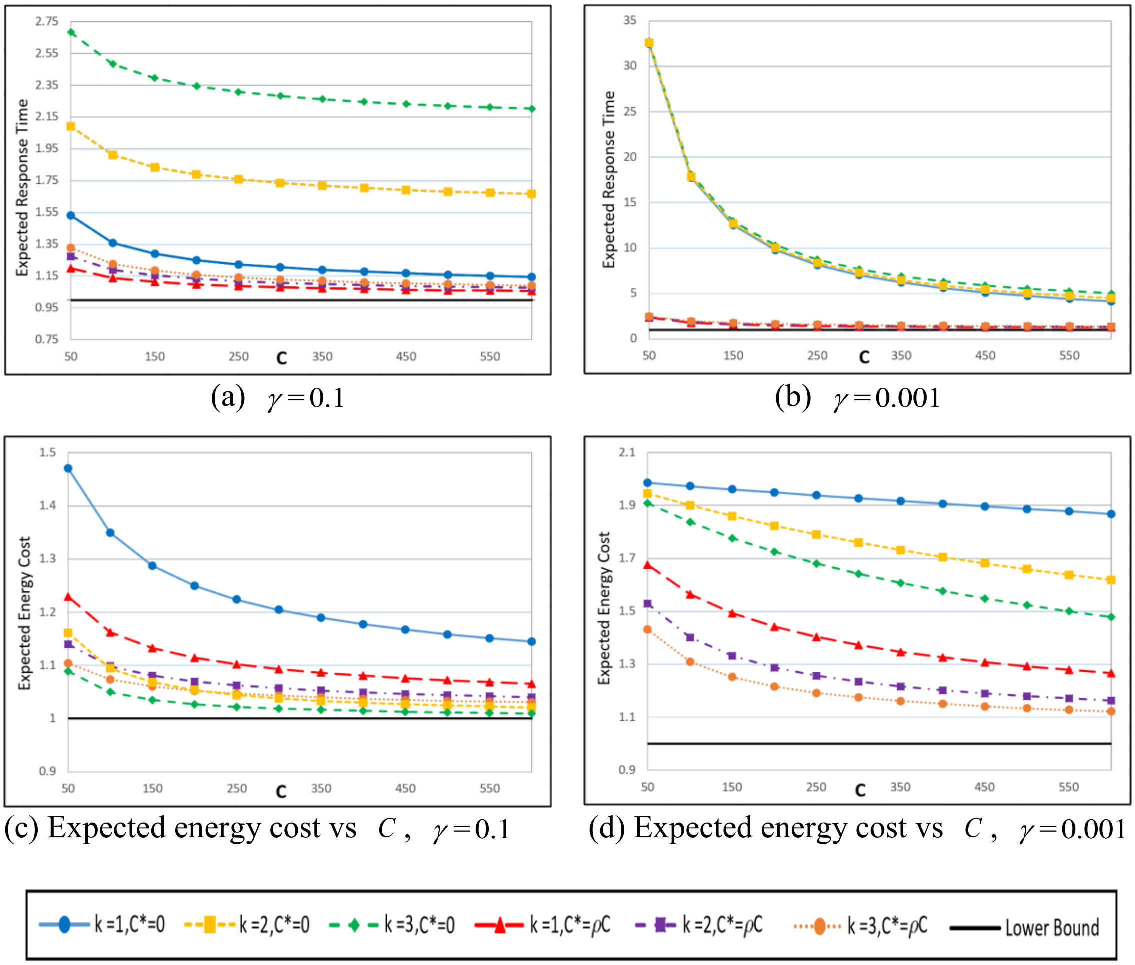
Theorem 2. *All policies in* Π_B *are asymptotically optimal with regards to expected energy cost. In other words, given a general energy-aware system, for any* $\pi_b \in \Pi_B$*, as* $\lambda, C \to \infty$ *while* μ *is held constant and* $\lambda / \mu C$ *is fixed to be* ρ*, where* $0 < \rho < 1$*,* $\mathbb{E}[E^{\pi_b}] \to \mathcal{P}_{\text{Busy}} / \mu$.

Corollary 1. *All Class B policies are asymptotically optimal under any well-formed cost function.*

The optimality result for the expected energy cost is arguably more surprising than the result for the expected response time. One may have the intuition that some of these policies would regularly have an unbounded number of servers in a specific energy state, such as setup, which could in turn incur strictly greater cost than some other policy (see (1) and (2)). Reasoning directly about the number of servers in a specific energy state is problematic here, as even for optimal policies one cannot bound the number of servers in energy states that lead to excess energy usage. Instead, we argue about relative proportions of servers in each energy state. In particular, under the asymptotic regime, servers which find themselves in setup or idle are negligible when compared to those which spend all of their time busy. In fact, as will be seen in Sections 3.1 and 4, reasoning about the energy cost as an even share of the server pool's power demands, as was seen in (1), results in complexities which are washed away when the energy costs are viewed from more convenient vantage points. Once these observations are made, these seemingly complex systems become simple to reason about.

The most significant implication of Theorems 1 and 2 is that under the asymptotic regime the apparent trade-off between $\mathbb{E}[R]$ and $\mathbb{E}[E]$ is in fact not a trade-off at all. That is, not only are both cost metrics minimized across a large set of policies, but over all well-formed cost functions. This is a powerful result, since if a system is *close* to this asymptotic regime, then a manager can confidently employ a Class B policy knowing that it will be reasonably close to optimal. Of course this begs the question, what does it mean for a system to be *close* to the system of study? This work addresses this question by numerically inspecting energy-aware systems with a finite number of servers C to see how quickly the cost metrics approach their bounds described in Theorems 1 and 2.

3.1. Numerical experiments



(a) $\gamma = 0.1$

(b) $\gamma = 0.001$

(c) Expected energy cost vs C, $\gamma = 0.1$

(d) Expected energy cost vs C, $\gamma = 0.001$

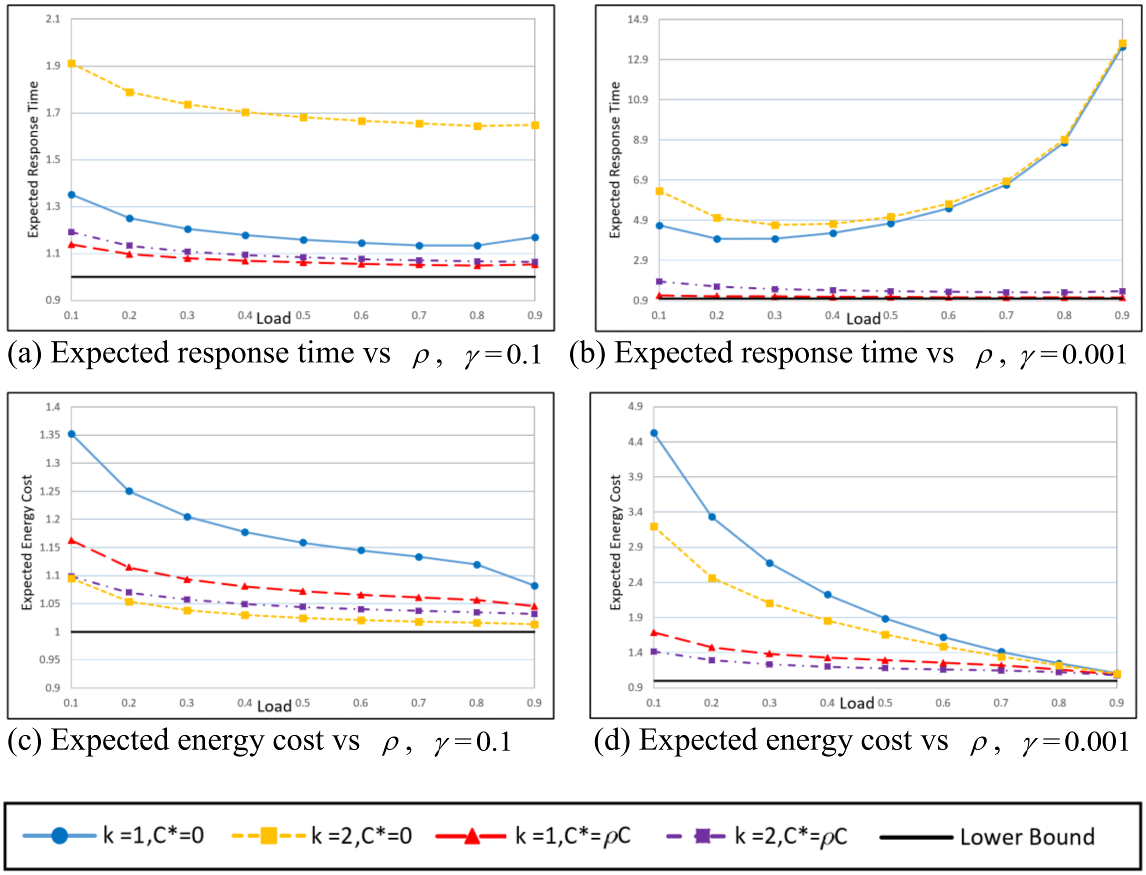
Figure 1. Expected response time and expected energy cost vs C for $\lambda = C/2$, $\mu = 1$.

All numerical experiments presented here are done for an energy-aware system employing a *staggered threshold* policy with a specific instantiation of its decision variables k, and C^*. A brief description of this policy is as follows. Regardless of the system state, C^* of the servers always remain on, the remaining $(C - C^*)$ servers will turn off the moment they idle, and the number of servers in setup is determined by the threshold value dependent on k, which equals $\{\lfloor \{j - C^*\}^+ / k \rfloor - i\}^+$, where $C^* + i$ is the number of servers currently on and j is the number of jobs in the system. Informally, the greater the value of k, the more jobs are required to accumulate before a server will begin its setup. A more in depth examination and analysis of this policy can be found in [16]. It is worth noting that by appropriately choosing the decision variables, other studied policies can be instantiated. As an example, letting $k = 1$ and $C^* = 0$ results in the staggered setup policy mentioned in Section 1.

Letting the system state be (i, j), where i is the number of servers on (idle or busy), and j is the number of jobs in the system, and assuming interarrival, processing, and setup times are exponentially distributed allows for the underlying CTMC to be expressed as a quasi birth-death process. This can be analysed using any of a number of well understood methods. We chose to analyse the CTMCs using the RRR technique described in [3]. Therefore, all numerical results are exact and were evaluated using standard Matlab libraries. The source code for the numerical analysis can be found at [27]. In general, if one were to not assume underlying exponential distributions one would have to rely on simulations for similar experiments. The purpose of these numerical experiments is firstly to ensure that exact analysis agrees with our results pertaining to the system under the asymptotic regime, and secondly to examine how quickly the system approaches the corresponding optimal behaviour as the parameters are appropriately scaled up.

Figure 3.1 (a) and (b) shows the behaviour of $\mathbb{E}[R]$ as the system size increases. A preliminary observation is that for the curves where $C^*=0$, the corresponding value of $\mathbb{E}[R]$ can be far from optimal even for large values of C. As an example, the curve where $k=3$ and $C^*=0$ in Figure 3.1-(a) is more than double that of the optimal value even for the largest values of C analysed. And perhaps even worse than that, the curves where $C^*=0$ have extremely slow convergence rates. On the other hand, one may also note that when $C^*=\rho C,$ $\mathbb{E}[R]$ becomes reasonably close to its optimal value relatively quickly. This effect is accentuated further in Figure 3.1-(b), where the setup times are large. Here, all curves which share the same choice of C^* are visually grouped together, and moreover, when $C^*=0$ the expected response time can be far from optimal even for larger values of $C.$ On the other hand, the curves which have a number of servers which are forced to be on, i.e. $C^*=\rho C,$ become much closer to the minimum value. In other words, the convergence rate is sensitive to the choice of C^* while relatively insensitive to the threshold value k, especially when setup times are large. Shifting focus to the expected energy costs, i.e. Figure 3.1 (c) and (d), all observations made regarding $\mathbb{E}[R]$ seem to carry over. That is, when C^* is forced to take on the value it practically approaches under the asymptotic regime, $\mathbb{E}[E]$ becomes closer to its optimal value. The notable exception would be that $\mathbb{E}[E]$ seems to be more sensitive to the choice of k, and the lower the value of k, the faster it approaches asymptotic behavior. Due to $\mathbb{E}[R]$'s relative insensitivity to k, this would suggest that for this particular policy one should err on the side of k being larger (assuming that C^* is intelligently selected).

The appealing choice of forcing $\lambda/\mu=\rho C$ servers to always remain on is interesting, since as will be seen in Section 4, the number of servers which are always busy approaches $\lambda/\mu=\rho C$ under the asymptotic regime. In other words, when the system parameters are finite, setting $C^*=\rho C$ forces the system to behave in a manner in which it is known to behave under the asymptotic regime. As such, it is intuitive that when the system is constrained to invoke certain asymptotic behaviour, i.e. $C^* = \rho C$, the corresponding values of $\mathbb{E}[R]$ and $\mathbb{E}[E]$ are closer to values which would be seen under the asymptotic regime.



(a) Expected response time vs ρ, $\gamma = 0.1$ (b) Expected response time vs ρ, $\gamma = 0.001$

(c) Expected energy cost vs ρ, $\gamma = 0.1$ (d) Expected energy cost vs ρ, $\gamma = 0.001$

Figure 2. Expected response time and Expected energy cost vs ρ for $C = 500$, $\mu = 1$.

Another aspect of these systems which warrants attention is how sensitive the convergence rate is to the load. This is seen in Figure 3.1. Examining the load's effect on the expected response time, one can observe that when the setup times are relatively short, the convergence rate is relatively insensitive to choice of load. Furthermore, the most sensitive parts of the curves are when the load is light or heavy, especially in Figure 3.1 (b) where the setup times are longer. This makes some intuitive sense, since when the loads are light or heavy the system is more likely to exhibit behaviour that is not described by the asymptotic regime. When the load is light, the system has a significant chance to be empty, and in turn has a significant chance to have the minimum number of servers on. When jobs arrive it begins to overcompensate with more setups than are needed and the servers begin to thrash. On the other hand, when the system load is high there is a significant chance that there will be more than C jobs in the system. So even if all servers were on, jobs would still have to wait. Having a non-negligible number of servers regularly thrashing or having more jobs in the system than servers are two characteristics which are not captured by a system under the asymptotic regime. Therefore it is intuitive that a system under light or heavy loads would be slower to exhibit asymptotic behaviour than that of a system with a medium load.

With this sensitivity in mind, one can still clearly note that the previous observation regarding having ρC servers always on induces the asymptotic behaviour to occur sooner. The only curve not to agree with this notion is the case where $k=2$ and $C^*=0$ in Figure 3.1 (c). In this case the system is approaching the minimum value of $\mathbb{E}[E]$ slightly quicker than the curves where $C^*=\rho C.$ This is a product of the servers thrashing, causing most jobs to see the system when many other jobs are present and therefore little energy is wasted. This is only achieved with a significantly larger $\mathbb{E}[R]$ and therefore this configuration would not be suggested.

4. Proofs

Here the proofs of Theorems 1 and 2 are presented in detail. Before this can be done some preliminary systems used throughout the proofs must first be defined. Consider the following three sequences of systems with $0<\rho<1$:

1. Let S_1 be a sequence of n general energy-aware systems, where the ith general energy-aware system is given by $S_{1,i}=(C_{1,i},\lambda_{1,i},\mu_{1,i},\gamma_{1,i})$, which employs some policy $\pi_i\in\Pi_A$ where $(\forall i\ s.t.\ 0<i\le n: \mu_{1,i}=1, C_{1,i}=i, \text{and } \lambda_{1,i}/C_{1,i}=\rho)$, and as $n\to\infty$, $\lambda_{1,n}\to\infty$.
2. Let S_2 be a sequence of n $G/G/C$ queues, where the ith $G/G/C$ queue is denoted by $S_{2,i}=(C_{2,i},\lambda_{2,i},\mu_{2,i})$ where $(\forall i\ s.t.\ 0<i\le n: \mu_{2,i}=1, C_{2,i}=i, \text{and } \lambda_{2,i}/C_{2,i}=\rho)$, and as $n\to\infty$, $\lambda_{2,n}\to\infty$.
3. Let S_3 be a sequence of n $G/G/C$ queues, where the ith $G/G/C$ queue is denoted by $S_{3,i}=(C_{3,i},\lambda_{3,i},\mu_{3,i})$ where $(\forall i\ s.t.\ 0<i\le n: \mu_{3,i}=1, \text{and } C_{3,i}=\lambda_{3,i}+(\lambda_{3,i})^{0.5+\epsilon})$, where $0<\epsilon<0.5$, and as $n\to\infty$, $\lambda_{3,n}\to\infty$.

In order to compare and reason about these systems, let it also hold that: $(\forall i\ s.t.\ 0<i\le n: \lambda_{1,i}=\lambda_{2,i}=\lambda_{3,i}=\lambda_i)$. Note that imposing such a constraint implies that $(\forall i\ s.t.\ 0<i\le n: C_{1,i}=C_{2,i}=C_i)$. In other words, the arrival rates of the systems across all three sequences are equal, and the total number of servers in the systems of sequences 1 and 2 are also equal.

Let $B_{1,i}$, $B_{2,i}$, and $B_{3,i}$ denote the number of always busy servers in the ith system of sequence S_1, S_2, and S_3 respectively. This work now provides a Lemma which is key to understanding the behaviours of these energy-aware systems under the asymptotic regime.

Lemma 1. *Given a sequence of general energy-aware systems where each system employs a policy $\pi\in\Pi_A$, C_n, λ_n, and μ_n denote the number of servers, arrival rate, and* *processing rate of the* n *th system respectively, and as* $n \to \infty$, $\lambda_n, C_n \to \infty$ *while* $\lambda_n / (\mu_n C_n)$ *is fixed to some* ρ *where* $0 < \rho < 1$, *and for all* n*,* $\mu_n = \mu$, *it holds that for the number of always busy servers in the* n *th system (denoted by* B_n*),*

$$\lim_{n \to \infty} \frac{B_n}{\lambda_n} = \frac{1}{\mu}.$$

Proof. Without loss of generality one can set $\mu = 1$. Therefore, to show Lemma 1, it is equivalent to prove

$$\lim_{n \to \infty} \frac{B_{1,n}}{\lambda_n} = 1.$$

This is done via a sample path argument regarding the sequences S_1 and S_2. Consider the systems $S_{1,n}$ and $S_{2,n}$ as $n \to \infty$. At any point in time the number of servers currently available in $S_{1,n}$ is less than or equal to the number of servers available in $S_{2,n}$. This follows from the fact that $S_{1,n}$ may have some of its C_n servers off or in setup, while $S_{2,n}$ has C_n servers on at all times. Therefore, taking the same arrival stream and job sizes for both systems, the number of jobs in $S_{2,n}$ is less than or equal to the number of jobs in $S_{1,n}$. Therefore, if s is busy in $S_{2,n}$, then s has enough workload to also be busy in $S_{1,n}$, but may not be busy due to it being switched off or in setup. Therefore, if s is an always busy server in $S_{2,n}$, then s has enough workload to be always busy in $S_{1,n}$. However, as $S_{1,n}$ is employing a policy from Π_A, specifically, from the second condition in the definition of Class A policies it is known a server will never turn off if there is work to do and from the first condition a server will eventually turn on as a consequence of the threshold scheme. Then it follows that almost surely the servers which can be always busy, will be always busy. That is to say, if s is an always busy server in $S_{2,n}$, then s is an always busy server in $S_{1,n}$. Therefore,

$$\lim_{n \to \infty} B_{1,n} \geq \lim_{n \to \infty} B_{2,n}. \tag{3}$$

Furthermore, it is known that

$$\lim_{n \to \infty} \frac{B_{2,n}}{\lambda_n} = 1.$$

This is shown via the following argument. Let $N_{2,n}(t)$ denote the number of jobs in $S_{2,n}$ at time t and let a corresponding diffusion be denoted by $\hat{N}_{2,n}(t)$, where

$$\hat{N}_{2,n}(t) = \frac{N_{2,n}(t) - \lambda_n}{\sqrt{n}}. \tag{4}$$

Note, as $n \to \infty$, for all t, $\mathbb{E}[N_{2,n}(t)] \to \lambda_n / \mu_{2,n} = \lambda_n$, see Section 5 of [10]. Then, (4) is of the form of the diffusion given in Section 1 of [11] where (using their notation) $Y_n(t) = N_{2,n}(t)$

and $h = \rho$. Then, from Theorem 4.3 of [11], it is known that $\hat{N}_{2,n}(t)$ weakly converges to a Gaussian process. Note, all conditions which need hold in order to apply Theorem 4.3 are shown to be satisfied in Section 5.2 of [11]. One detail which should be addressed is that in [11] the author is dealing with a $G/G/\infty$ queue where the arrival rate is being scaled up to infinity with n, while we are dealing with the slightly different system where the arrival rate and number of servers are being scaled together with n. This is of little consequence however, since due to the order of which the deviation from the mean is on, i.e. $\hat{N}_{2,n}(t)$ being on the order of $\sqrt{n}$, both systems (our scaling and the $G/G/\infty$ queue) have equivalent behaviour as $n \to \infty$. This claim is made more formally in Section 5 of [10]. As an aside, if service and interarrival times were exponentially distributed then (4) is equivalent to the diffusion given in [9]. Returning to (4), after some elementary algebra,

$$N_{2,n}(t) = \sqrt{n}\hat{N}_{2,n}(t) + \lambda_n \quad \Rightarrow \quad \frac{N_{2,n}(t)}{\lambda_n} = \frac{\hat{N}_{2,n}(t)}{\sqrt{\rho\lambda_n}} + 1 .$$

Since $n \to \infty$, $\hat{N}_{2,n}(t)$ weakly converges to a Gaussian process, it follows $\lim_{n\to\infty} \hat{N}_{2,n}(t)/\sqrt{\rho\lambda_n} = 0$. This immediately implies

$$\lim_{n\to\infty} \frac{N_{2,n}(t)}{\lambda_n} = 1 .$$

Moreover, one can say that as $n \to \infty$ if at all time points t there are almost surely at least x jobs in the system, then as $n \to \infty$ at all time points t there are almost surely at least x always busy servers. Therefore,

$$\lim_{n\to\infty} \frac{N_{2,n}(t)}{\lambda_n} = 1 \quad \Rightarrow \quad \lim_{n\to\infty} \frac{B_{2,n}}{\lambda_n} \geq 1 .$$

After realizing this property of S_2, one can begin examining the implication on the behaviour of S_1. Specifically from (3) it is known,

$$\lim_{n\to\infty} \frac{B_{2,n}}{\lambda_n} \geq 1 \quad \Rightarrow \quad \lim_{n\to\infty} \frac{B_{1,n}}{\lambda_n} \geq 1 .$$

Hence, all that remains to show Lemma 1 is to prove

$$\lim_{n\to\infty} \frac{B_{1,n}}{\lambda_n} \leq 1 .$$

This follows immediately after the observation that

$$\lim_{n\to\infty} \frac{B_{1,n}}{\lambda_n} > 1$$

can hold only if the arrival rate of the system is greater than λ_n, which is a direct contradiction to the system definition. Therefore,

$$\lim_{n\to\infty}\frac{B_{1,n}}{\lambda_n}=1. \qquad \blacksquare$$

4.1. Proof of Theorem 1

For simplicity of navigation, Theorem 1 is restated.

Theorem 1. *All policies in* Π_A *are asymptotically optimal with regards to expected response time. In other words, given a general energy-aware system, for any* $\pi_a \in \Pi_A$*, as* $\lambda, C \to \infty$ *while* μ *is held constant, and* $\lambda/\mu C$ *is fixed to be* ρ *where* $0<\rho<1$*,* $\mathbb{E}[R^{\pi_a}]\to 1/\mu$.

A high-level description of the proof is as follows. It is determined that if a job J is served by an always busy server in the system $S_{3,n}$ as $n\to\infty$, then the expected response time of job J approaches its expected service time. With this in mind, the system $S_{1,n}$ is compared to $S_{3,n}$ as $n\to\infty$. It is shown that the expected response time of these systems is dominated by jobs that are served by always busy servers. Moreover, the limits of the expected response time of these two systems are equal. Therefore, the expected response time of $S_{1,n}$ approaches the expected service time as $n\to\infty$.

Proof. As with the proof of Lemma 1, without loss of generality one can set $\mu=1$. Therefore, to prove the theorem it is enough to show that $\lim_{n\to\infty}\mathbb{E}[R_{1,n}]=1$, where $\mathbb{E}[R_{i,j}]$ denotes the expected response time corresponding to the system at the jth index of the ith sequence of systems. The equality of this limit is shown via a sample path argument regarding S_1 and S_3. Before this argument is made some properties of S_3 must be shown.

Consider the system $S_{2,n}$, but imagine we are only concerned with the first $\lambda_n+\lambda_n^{0.5+\epsilon}$ servers (out of the potential $C_{2,n}$), where $0<\epsilon<0.5$. One can think of this virtual server set as equivalent to the servers in $S_{3,n}$, i.e. $C_{3,n}=\lambda_n+\lambda_n^{0.5+\epsilon}$. As before, let $\hat{N}_{2,n}(t)$ equal the diffusion given in (4). Because it is known as $n\to\infty$ that $\hat{N}_{2,n}(t)$ weakly converges to a Gaussian process, as $n\to\infty$, $\hat{N}_{2,n}(t)$ is almost surely finite. Therefore,

$$\lim_{n\to\infty}\frac{\sqrt{\lambda_n}\hat{N}_{2,n}(t)}{\lambda_n^{0.5+\epsilon}}=\lim_{n\to\infty}\frac{\hat{N}_{2,n}(t)}{\lambda_n^{\epsilon}}=0.$$

where $\sqrt{\lambda_n}\hat{N}_{2,n}(t)$ can be thought of as the number of jobs deviating from the expectation, i.e. the number of jobs deviating from λ_n. Therefore, as $n\to\infty$, for all t, $P(N_{2,n}(t)>\lambda_n+\lambda_n^{0.5+\epsilon})=0$, where $N_{2,n}(t)$ is the number of jobs in $S_{2,n}$ at time t. This implies that some subset of the $\lambda_n+\lambda_n^{0.5+\epsilon}$ servers are always idle. Hence, $S_{3,n}$ can mimic the behaviour of $S_{2,n}$ almost exactly. This implies $P(N_{3,n}(t)>C_{3,n})$, where $N_{3,n}(t)$ is the number of jobs in $S_{3,n}$ at time t. In other words, in $S_{3,n}$ jobs almost never queue. Therefore,

$$\lim_{n\to\infty}\mathbb{E}[R_{3,n}]=\lim_{n\to\infty}\frac{1}{\mu_{3,n}}=1\,. \tag{5}$$

One can observe this as a version of the square root staffing rule, see Theorem 15.2 of [7]. Moreover, identical to the reasoning presented in Lemma 1, one can conclude

$$\lim_{n\to\infty}\frac{B_{3,n}}{\lambda_n}=1\,. \tag{6}$$

From here the server pool of $S_{3,n}$ is split into two distinct conceptual sets. The first set consists of the always busy servers and the second set consists of all remaining servers, i.e. the servers which spend some of their time idle. Let these two sets of servers be denoted by $A_{3,n}$ and $I_{3,n}$, respectively. Then the expected response time can be expressed as a sum of terms,

$$\mathbb{E}[R_{3,n}]=P_{A_{3,n}}\mathbb{E}[R_{3,n}^{A}]+P_{I_{3,n}}\mathbb{E}[R_{3,n}^{I}]\,, \tag{7}$$

where $P_{A_{3,n}}$ and $P_{I_{3,n}}$ denote the probability of a job being served from set $A_{3,n}$ or $I_{3,n}$, respectively, and $E[R_{3,n}^{A}]$ and $\mathbb{E}[R_{3,n}^{I}]$ denote the expected response times given that a job is served by a server from set $A_{3,n}$ or $I_{3,n}$, respectively. Due to the homogeneity of the servers, it is assumed that if there are c servers currently on and $i\leq c$ jobs in the system, then servers 1 through i will be the servers that are busy. From this it can be noted that always busy servers, i.e. servers from set $A_{3,n}$, have serving priority over the others when jobs arrive. That is, it is known that the probability of a particular job being served by an always busy server is greater than or equal to choosing it randomly and uniformly from the entire pool. In other words,

$$P_{A_{3,n}}\geq\frac{|A_{3,n}|}{C_{3,n}}=\frac{|A_{3,n}|}{\lambda_n+\lambda_n^{0.5+\epsilon}}\,.$$

Furthermore, from the definition of $A_{3,n}$, it is known that $|A_{3,n}|=B_{3,n}$ which implies,

$$\lim_{n\to\infty}P_{A_{3,n}}\geq\lim_{n\to\infty}\frac{B_{3,n}}{\lambda_n+\lambda_n^{0.5+\epsilon}}=\lim_{n\to\infty}\frac{B_{3,n}}{\lambda_n}=1\,.$$

Noting $P_{I_{3,n}}=1-P_{A_{3,n}}$ alongside (5) and (7) it becomes clear that,

$$\lim_{n\to\infty}\mathbb{E}[R_{3,n}]=\lim_{n\to\infty}\mathbb{E}[R_{3,n}^{A}]=1.$$

With the limit of $\mathbb{E}[R_{3,n}^{A}]$ explicitly determined, the proof proceeds with a sample path argument involving S_1 and S_3. As $A_{3,n}$ and $I_{3,n}$ denote the sets of always busy and sometimes idle servers respectively for $S_{3,n}$, let $A_{1,n}$ and $I_{1,n}$ denote the corresponding always busy and sometimes idle sets for $S_{1,n}$. By sometimes idle, we mean not always busy.

It is worth noting that while calling $I_{3,n}$ a set of sometimes idle servers is apt, applying the same notion to $I_{1,n}$ is somewhat inappropriate. $I_{1,n}$ is the set of servers which are not always busy. These servers potentially could spend zero time idling, i.e. they immediately switch off when they complete a job and the queue is empty. However, for the purposes of the proof it is only required that these servers spend some portion of time not busy.

Continuing with the sample path argument, consider systems $S_{1,n}$ and $S_{3,n}$, as $n \to \infty$, with identical arrival processes. Jobs are viewed as being marked to be served by a server belonging to the sets $A_{1,n}$ and $A_{3,n}$, or $I_{1,n}$ and $I_{3,n}$, without loss of generality. If a job arrives to $S_{3,n}$ and is served from the set $A_{3,n}$, then the corresponding job in $S_{1,n}$ will almost surely be served from the set $A_{1,n}$. This is the case by noting from (6) and Lemma 1 that

$$\lim_{n\to\infty} \frac{B_{1,n}}{B_{3,n}} = 1. \tag{8}$$

Moreover, if a job arrives to $S_{3,n}$ and is served from the set $I_{3,n}$ then the corresponding job in $S_{1,n}$ will almost surely be served from the set $I_{1,n}$. It must be shown that $I_{1,n}$ has the capacity to almost surely serve the load which $I_{3,n}$ has the capacity to serve. This follows immediately by noting that

$$\lim_{n\to\infty} \frac{|I_{3,n}|}{\lambda_n} = 0, \quad \text{while} \quad \lim_{n\to\infty} \frac{|I_{1,n}|}{C_{1,n}} = (1-\rho)$$

which implies,

$$\lim_{n\to\infty} \frac{|I_{3,n}|}{|I_{1,n}|} = 0.$$

As was done with $S_{3,n}$, one can decompose the expected response time of $S_{1,n}$ into two distinct components as follows,

$$\mathbb{E}[R_{1,n}] = P_{A_{1,n}} \mathbb{E}[R^A_{1,n}] + P_{I_{1,n}} \mathbb{E}[R^I_{1,n}].$$

From here two important observations are made. Firstly because $S_{1,n}$ is a stable system $\mathbb{E}[R^I_{1,n}]$ is known to be finite. Secondly, because $S_{1,n}$ and $S_{3,n}$ have identical arrival processes, it can be said

$$\lim_{n\to\infty} P_{A_{1,n}} = \lim_{n\to\infty} P_{A_{3,n}} = 1$$

which alongside (8) implies,

$$\lim_{n\to\infty} \mathbb{E}[R^A_{1,n}] = \lim_{n\to\infty} \mathbb{E}[R^A_{3,n}] = 1.$$

Therefore,

$$\lim_{n\to\infty}\mathbb{E}[R_{1,n}]=\lim_{n\to\infty}\mathbb{E}[R_{1,n}^{A}]=1.$$

4.2. Proof of Theorem 2

For simplicity of navigation, Theorem 2 is restated.

Theorem 2. *All policies in Π_B are asymptotically optimal with regards to expected energy cost. In other words, given a general energy-aware system, for any $\pi_b\in\Pi_B$, as $\lambda,C\to\infty$ while μ is held constant, and $\lambda/\mu C$ is fixed to be ρ where $0<\rho<1$, $\mathbb{E}[E^{\pi_b}]\to\mathcal{P}_{\text{Busy}}/\mu$.*

A high-level description of the proof is as follows. The notion of the energy cost contributed by a single job is examined from a different angle. That is, the energy cost of a single job does not take an *even share* of the system's power demands, but is instead deliberately determined based on what consequences its arrival induces. Lemma 2 gives an exact value of the energy cost of a single job from this viewpoint, assuming it was served by an always busy server. Moreover, this is a minimum value. On the other hand, Lemma 3 shows if a job is not served by an always busy server, the energy cost is finite. From there, similar to the procedure in the proof of Theorem 1, it becomes clear that the total expected energy cost is dominated by jobs which are served by always busy servers, and therefore is minimized.

Definition 4. *E^J* **:** *Let E^J be a random variable which denotes the energy cost contributed by a randomly chosen job J under the following interpretation. There are four contributing factors to consider when determining E^J for some job J.*

1. *Each job J is responsible for the energy required to process it.*
2. *If a job J is the first job which server s serves after completing its setup process and s regularly turns on and off, i.e. $0<\lim_{t\to\infty}\mathbb{E}[X_{\text{Setup}}(s,t)]<1$, then it is said that J is responsible for contributing the entire cost of the setup process of s and any other servers which cancel their setups due to s completing its setup, as well as any idling costs of s until the next time s is switched off.*
3. *If a job J is responsible for causing a server setup which is canceled due to that job entering service before the setup process completes, then J is also responsible for the energy cost incurred by that setup as well as the cost of all other setups which may be canceled due to s completing its setup.*
4. *The idling cost of servers which never turn off is divided evenly among all jobs that pass through the system. Furthermore, a special case is added to this factor. If some of the servers are always busy servers, i.e. some of the system parameters are approaching infinity, then the aforementioned idling cost is evenly distributed only across jobs which are served by always busy servers, rather than across all jobs*

which pass through the system.

As mentioned in Section 2 and from the definition of E^J, it is clear that

$$\mathbb{E}[E^J] = \lambda \mathbb{E}[\mathcal{P}] = \mathbb{E}[E]. \tag{9}$$

Therefore, although E^J and E may be different random variables (depending on one's interpretation of E), their expectations are equal.

Lemma 2. *In an energy-aware queueing system employing a Class B policy, if a job* J *is served by an always busy server, then* $\mathbb{E}[E^J] = \mathcal{P}_{\text{Busy}} / \mu$.

Proof. The proof of this Lemma is argued after several key observations. To prove the Lemma, it is equivalent to show that if J is served by an always busy server, then $\lim_{n\to\infty} \mathbb{E}[E^J_{1,n}] = \mathcal{P}_{\text{Busy}}$. Moreover, for there to be any always busy servers present in the system, it is required that $n \to \infty$. Therefore, it will be argued that

$$\lim_{n\to\infty} \mathbb{E}[E^J_{1,n} \mid J \text{ was served by an always busy server}] = \mathcal{P}_{\text{Busy}}.$$

To show the above equality, the four contributing factors to $\mathbb{E}[E^J_{1,n}]$, from Definition 4 are addressed individually and summed.

1. It is known that each job J will eventually be served, by an always busy server or otherwise, and therefore J incurs an expected cost of $\mathcal{P}_{\text{Busy}} / \mu_{1,n} = \mathcal{P}_{\text{Busy}}$.
2. If it is known that J is served by an always busy server then it is trivially known that it is not the first job to be processed after a server completes its setup process. Therefore it can be said that J incurs no cost from this contributing factor.
3. The third contributing factor takes a little more care but can be shown to almost surely incur no cost. Consider the two cases of J being served by an always busy server, where 1) J does not wait in the queue to be served, and 2) J waits some amount of time in queue before it is served. Starting with case 1), having an arriving job become immediately served by an always busy server implies that in the same instant a job completes its service at an always busy server. Otherwise, that server is not always busy. Therefore, J cannot be the job to initiate a setup due to the threshold nature of the policy alongside the total number of jobs not increasing by its arrival. So if J is served by an always busy server there is no contributed energy cost (from the factor in question). Now consider case 2), where J waits in queue before being served. Here J may start a finite number of setups (as allowed by Class B policies), which are canceled due to J being served before their setups can complete. The probability of this case occurring is zero because J queueing with a non-zero probability would contradict Theorem 1. Therefore, case 2) has a zero probability of occurring and therefore, J almost surely incurs no cost from this

contributing factor.

4. Because $n \to \infty$ the special case of this factor is invoked, i.e. jobs that are served by always busy servers are responsible for the full cost of the always idle servers. From the definition of Class B policies it is known that the number of always idle servers is less than $(1-\lambda_{1,n}/C_{1,n})C_{1,n}^{\alpha}$, where $0 \le \alpha < 1$. Therefore, costs from these idle servers are incurred at some rate less than $(1-\rho)C_{1,n}^{\alpha}\mathcal{P}_{\text{Idle}}$. Moreover, from the proof of Theorem 1, it is known that as $n \to \infty$, the probability of being served by an always busy server approaches 1. It then follows that the rate at which jobs are served by always busy servers approaches the arrival rate, $\lambda_{1,n}$, as $n \to \infty$. Therefore, letting the expected contributing cost from these idle servers per job be denoted by $\mathbb{E}[E^{J,I}]$ and from the definition of S_1 knowing that $\lambda_{1,n} = C_{1,n}/\rho$:

$$\mathbb{E}[E_{1,n}^{J,I}] = \lim_{n\to\infty} \frac{(1-\rho)C_{1,n}^{\alpha}\mathcal{P}_{\text{Idle}}}{\lambda_{1,n}} = \lim_{n\to\infty} \rho(1-\rho)\mathcal{P}_{\text{Idle}} \frac{C_{1,n}^{\alpha}}{C_{1,n}} = \lim_{n\to\infty} \rho(1-\rho)\mathcal{P}_{\text{Idle}} \frac{1}{C_{1,n}^{(1-\alpha)}} = 0.$$

Therefore, the only contributing factor to $\mathbb{E}[E_{1,n}^{J}]$ given that J has been served by an always busy server is the cost of processing it, which implies

$$\lim_{n\to\infty} \mathbb{E}[E_{1,n}^{J} \mid J \text{ is served by an always busy server}] = \mathcal{P}_{\text{Busy}}.$$

Lemma 3. *In an energy-aware queueing system employing a Class B policy, if a job J is not served by an always busy server then $\mathbb{E}[E^{J}]$ is finite.*

Proof. Similar to the proof of Lemma 2, this proof iterates through the contributing factors of Definition 4 to show that the expectation of each factor is finite, and therefore the expectation of their sum is finite, i.e. $\mathbb{E}[E^{J}]$ is finite.

1. As before, it is known that each job J will eventually be served, by an always busy server or otherwise, and therefore J incurs an expected cost of $\mathcal{P}_{\text{Busy}}/\mu_{1,n} = \mathcal{P}_{\text{Busy}}$.
2. If it is known that a job J has been served by a server s which regularly completes its setup process, i.e. as $0 < \lim_{t\to\infty} \mathbb{E}[X_{\text{Setup}}(s,t)] < 1$, two cases must be considered due to Definition 3. The first and simpler case is that J is not the first job to be served by s after s completes a setup process. Here no costs will be incurred. The second and more interesting case is that J is the first job to be served by s following a setup process. Here J is responsible for the setup costs incurred by s alongside the idling costs incurred by s until the next time it shuts off. It is known s regularly spends some portion of time in setup, and therefore will have a finite amount of time until it is no longer idle, incurring a finite idling cost. So all that remains is to show the setup cost associated with s is also finite. The total setup cost of s can be split into two components, the cost directly incurred from the setup process of s, as well as the setup costs of servers which cancel their setups due to s turning on. Clearly,

the setup cost directly due to the setup of s is finite, and moreover, due to the definition of Class B policies, the number of setups canceled due to s turning on is finite. Hence the cost of those canceled setups is finite as well. Therefore, the expected energy cost incurred from this contributing factor is finite.

3. This is similar to the previous case of the server completing its setup process. A single job can only cause a finite number of servers to initiate their setups, and in turn, can only cause a finite number of servers to cancel their setups. Therefore, the cost associated with a job canceling setups by being processed beforehand is finite.
4. Because there are always busy servers present in the system, from the definition of E^J it is trivially known that zero costs are incurred by this factor.

All of the contributing terms of $\mathbb{E}[E^J]$ are finite, therefore $\mathbb{E}[E^J]$ is also finite.

With the proof of the Lemmas complete, it is relatively straightforward to prove Theorem 2.

Proof. To prove the theorem, it is equivalent to show $\lim_{n\to\infty}\mathbb{E}[E_{1,n}]/\lambda_n = \mathcal{P}_{\text{Busy}}$ under the assumption that the systems of S_1 are employing Class B policies. From the definition of $\mathbb{E}[E^J]$, it is known $\mathbb{E}[E]=\lambda\mathbb{E}[E^J]$. Therefore,

$$\lim_{n\to\infty}\mathbb{E}[E_{1,n}]/\lambda_n = \lim_{n\to\infty}\mathbb{E}[E^J_{1,n}].$$

Furthermore, using the notation introduced in the proof of Theorem 1 it is also known that

$$\lim_{n\to\infty}\mathbb{E}[E^J_{1,n}] = \lim_{n\to\infty} P_{A_{1,n}}[E^J_{1,n} \mid J \text{ is served from } A_{1,n}]$$
$$+ P_{I_{1,n}}[E^J_{1,n} \mid J \text{ is served from } I_{1,n}].$$

Leveraging past equalities allows one to simplify the above equation. From Lemma 2 it is known

$$\lim_{n\to\infty}\mathbb{E}[E^J_{1,n} \mid J \text{ is served from } A_{1,n}] = \mathcal{P}_{\text{Busy}}.$$

From Lemma 3, $\mathbb{E}[E^J_{1,n} \mid J \text{ is served from } I_{1,n}]$ is finite, i.e. for some $L_n > 0$

$$\lim_{n\to\infty}\mathbb{E}[E^J_{1,n} \mid J \text{ is served from } I_{1,n}] = L_n.$$

From the proof of Theorem 1, it is known

$$\lim_{n\to\infty} P_{A_{1,n}} = 1 \quad \text{and} \quad \lim_{n\to\infty} P_{I_{1,n}} = 0.$$

Therefore,

$$\lim_{n\to\infty}\mathbb{E}[E^J_{1,n}] = \mathcal{P}_{\text{Busy}}. \tag{10}$$

It is worth noting that E_{Busy} is a lower bound for $\mathbb{E}[E^J_{1,n}]$ since for the system to be stable the job must be processed. In other words, as $n \to \infty$, $\mathbb{E}[E^J_{1,n}]$ approaches its minimum value. Returning to (9),

$$\lim_{n\to\infty} \mathbb{E}[E_{1,n}] = \lim_{n\to\infty} \mathbb{E}[E^J_{1,n}] .$$

It then follows from (10) that

$$\lim_{n\to\infty} \mathbb{E}[E_{1,n}] = \mathcal{P}_{\text{Busy}} .$$

Moreover, in system $S_{1,i}$, $\mathcal{P}_{\text{Busy}}$ is a trivial lower bound for the expected energy cost, implying that as $n \to \infty$, $\mathbb{E}[E_{1,n}]$ is minimized. That is, the policy which $S_{1,n}$ is employing, a Class B policy, minimizes the expected energy cost as $n \to \infty$.

5. Conclusions

This work examined an established multiserver queueing model, where each server can be turned on, which takes nonzero time, to improve performance, or turned off instantly to save on costs. How these servers are turned on and off define a policy, and this policy is evaluated under a cost function which takes performance and energy costs into account. A problem which arises is which policy should be employed for a given cost function, and furthermore, whether this policy will be robust enough to be a reasonable choice when evaluated under other cost functions. While the problem seems to be complex due to the lack of structure for the policies and cost functions, as the number of servers $C \to \infty$ with a fixed system load $\rho = \lambda / (C\mu)$, a large set of policies become equivalent, i.e. Class B policies. Therefore, under this asymptotic regime the choice of which Class B policy to employ is irrelevant. Furthermore, not only are these policies equivalent but they also simultaneously minimize $\mathbb{E}[R]$ and $\mathbb{E}[E]$. It then follows that all Class B policies will be optimal under all well-formed cost functions.

This work then numerically evaluated the staggered threshold policy to inspect how quickly these asymptotic behaviours are seen. These numerical results suggested that a finer-grained differentiation of Class B policies can be given such that the resulting subsets exhibit faster or slower convergence rates. In general, the rate of convergence to the asymptotic regime can be extremely slow. We have suggested that through careful choice of parameters the convergence can be significantly sped up. Specifically, it was determined that if a static provisioning of $C\rho$ servers is enforced within the policy, then as C increases the metrics approach their corresponding minimum values more quickly than if no static provisioning were present.

Looking forward to the future of this work, an issue to address is how well these policies do under a time varying arrival rate. It is our intuition that if the arrival rate varied on a relatively long time scale with respect to other system parameters such as the expected setup time, then the results given here may be reasonable to apply. However, questions remain which require a formal treatment. Can some of the asymptotic results be extended for a subset of Class B policies? If so, what new criteria must these policies adhere to? If not, what complication is the limiting factor in the analysis? While these questions are certainly deserving of attention, the results presented here regarding the optimality of all Class B policies under all well-formed cost functions allows one to confidently make overarching statements and conclusions across the problem domain.

Acknowledgments

This research was funded by the Discovery Grant Program of the Natural Sciences and Engineering Research Council of Canada.

References

[1] Allahverdi, A., Ng, C., Cheng, T., & Kovalyovc, M . Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187, 985–1032.

[2] Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., & Gautam, N. (2005). Managing server energy and operational costs in hosting centers. *SIGMETRICS Performance Evaluation Review*, 33(1):303–314, June 2005.

[3] Gandhi, A., Doroudi, S., Harchol-Balter, M., & Scheller-Wolf, A. (2013). Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward. In *ACM SIGMETRICS Performance Evaluation Review*, 153–166.

[4] Gandhi, A., Gupta, V., Harchol-Balter, M., & Kozuch, M. A. (2010). Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67, 1155–1171.

[5] Gandhi, A., Harchol-Balter, M., & Adan, I. (2010). Server farms with setup costs. *Performance Evaluation*, 67, 1123–1138.

[6] Gandhi, A., Harchol-Balter, M. (2010). M/M/k with exponential setup. Technical report, Carnegie Mellon University, 2010.

[7] Harchol-Balter, M. (2013). *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press.

[8] Hu, J., & Phung-Duc, T. (2015). Power consumption analysis for data centers with independent setup times and threshold controls. In *AIP*.

[9] Iglehart, D. L. (1965). Limiting diffusion approximations for the many server queue and the repairman problem. *Journal of Applied Probability*, 2, 429–441.

[10] Iglehart, D. L. (1973). Weak convergence in queueing theory. *Advances in Applied Probability*, 5, 570–594.

[11] Iglehart, D. L. (1973). Weak convergence of compound stochastic process, i. *Stochastic Processes and Their Applications*, 1, 11 – 31.

[12] Kuehn, P. J., & Mashaly, M. E. (2015). Automatic energy efficiency management of data center resources by load-dependent server activation and sleep modes. *Ad Hoc Networks*, 25, 497–504.

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

[14] Maccio, V. J., & Down, D. G. (2016). Exact analysis of energy-aware multiserver queueing systems with setup times. In *IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 11–20.

[15] Maccio, V. J., & Down, D. G. (2018). Asymptotic performance of energy-aware multiserver queueing systems with setup times. In *Annual American Control Conference (ACC)*, 6266–6272.

[16] Maccio, V. J., & Down, D. G. (2018). Structural properties and exact analysis of energy-aware multiserver queueing systems with setup times. *Performance Evaluation*, 121-122, 48 – 66.

[17] Magazine, M. J. (1971). On optimal control of multi-channel service systems. *Naval Research Logistics Quarterly*, 18, 429–441.

[18] Mitrani, I. (2013). Managing performance and power consumption in a server farm. *Annals of Operations Research*, 202, 121–134.

[19] Mukherjee, D., Dhara, S., Borst, S., & Leeuwaarden, J. (2017). Optimal service elasticity in large-scale distributed systems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1, 1–28.

[20] Phung-Duc, T., & Kawanishi, K. (2016). Energy-aware data centers with s-staggered setup and abandonment. In *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, 269–283.

[21] Phung-Duc, T., & Kawanishi, K. (2019). Delay performance of data-center queue with setup policy and abandonment. *Annals of Operations Research,* https://doi.org/10.1007/s10479-019-03268-1.

[22] Phung-Duc, T. (2016). Exact solutions for M/M/c/setup queues. *Telecommunication Systems*, 1–16.

[23] Ren, Y., Phung-Duc, T., Yu, Z. W., & Chen, J. C. (2016). Design and analysis of dynamic auto scaling algorithm (DASA) for 5G mobile networks. *arXiv preprint arXiv:1604.05803*.

[24] Slegers, J., Thomas, N., & Mitrani, I. (2008). Dynamic server allocation for power and performance. In *SPEC International Workshop on Performance Evaluation: Metrics, Models and Benchmarks*, 247–261.

[25] Tian, N., & Zhang, Z. G. (2006). A two threshold vacation policy in multiserver queueing systems. *European Journal of Operational Research*, 168, 153–163.

[26] Xu, X., & Tian, N. (2008). The M/M/c queue with (e, d) setup time. *Journal of Systems Science and Complexity*, 21, 446–455.

[27] Source code (2018). http://www.cas.mcmaster.ca/macciov/publications.html.Accessed 2018-10-10.