Asymptotic Performance of Energy-Aware Multiserver Queueing Systems with Setup Times*

Vincent J. Maccio[1] and Douglas G. Down[1]

***Abstract*— Energy demands of modern datacentres are an immense concern. An intuitive solution is to turn servers off to incur less costs. However, the control problem of when to turn a specific server off, and when to then turn that server back on, is far from trivial. 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. We analyse this well-established model under the asymptotic regime where the number of servers approaches infinity while the load per server remains fixed and show that not only are many of the control policies in the literature equivalent under this regime, but they are also optimal under any cost function which is non-decreasing in the expected energy cost and response time.**

I. 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, control policy implemented, model details, etc., different conclusions can be drawn from similar underlying problems. One consequence of this variety is that it is difficult to confidently draw conclusions which are overarching across the problem domain. To address this issue, we consider an asymptotic regime, such that when the system parameters are appropriately scaled up, a general class of policies is optimal, under all reasonable cost functions. This asymptotic regime is useful for practical applications, in our case datacentres with a large number of servers.

To the best of our knowledge, Chen et al. [1] and Sledger et al. [2] were the first to apply queueing models in the context of energy-aware datacentres. This work introduced a 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 decrease cost). This introduced a control problem: when should servers be turned on, and when should they be turned off (if at all)? 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 both 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 [7] studied this model where a reserved set of servers are brought into setup when the number of jobs in the system exceeds a threshold, and then shuts those servers off once the number of jobs drops below another threshold. This policy was further studied in [8]. Xu and Tian [9] examined the model where e servers turn off when d servers idle. Variations on this theme of employing threshold policies have been studied by a number of additional authors, including [10]–[13].

The body of work discussed in the previous paragraph takes a common approach: propose a policy, choose a cost function which one wishes to minimize, then evaluate the performance of the resulting stochastic model. An immediate observation that can be made about all of the policies considered is that they all have similar form, in this case a threshold policy, where thresholds are used to determine when servers should be turned on or off. In our experience, we observed that several policies of this form exhibited similar mean response time performance and similar energy consumption, when the systems increased in size. Moreover, it appeared that both the mean response time and energy consumption were approaching their minimum possible values. Our goal in this paper is to make this observation precise. In particular, we identify structural properties of the optimal policy that lead to a wide range of policies (including all of those in the previous paragraph) being optimal in an asymptotic sense. The asymptotic regime studied in this work is a fixed-utilization, many-server regime, i.e. one where the utilization (load per server) remains fixed ($\rho < 1$), while the number of servers approaches infinity ($C \to \infty$).

This work makes the following contributions:

1) We provide an overview of the control problem and summarize structural properties of the optimal policy (the latter is a summary of previous work).
2) It is shown that in the fixed-utilization, many-server regime, any policy satisfying the derived structural properties minimizes both the expected response time and expected energy cost, and therefore is optimal under all cost functions which are non-decreasing in those metrics.
3) Numerical experiments are conducted to determine how quickly the asymptotic behaviour is reached, and it is shown that particular choices of policy parameters may be used to induce faster convergence to the minimum values.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada

[1]Vincent J. Maccio and Douglas G. Down are with the Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada macciov@mcmaster.ca, downd@mcmaster.ca

Finally, we note that a similar asymptotic approach has been employed concurrently to our work in Mukherjee et al. [14]. 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. They identify a combined routing and server control policy that is optimal in the fixed-utilization, many-server regime. Here, 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 [14] provides a complete picture of the control problem (at least asymptotically) for both central and parallel queue architectures.

II. Model

The model under study is an $M/M/C$ queue where each server can be switched on and off, and where turn-offs are instantaneous, but turn-ons take an exponentially distributed setup time. Jobs arrive to a central queue following a Poisson process with rate λ, are processed on a first come first served basis, and have processing times (job sizes) that are exponentially distributed with rate μ. There are C homogeneous servers present, each of which can be in one of four energy states: *off, setup, idle*, or *busy*. A server is *idle* if it is on and not processing a job. The server becomes *busy* when it is processing a job. 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 exponentially distributed with rate γ, after which it will become *busy* if there is a waiting job; otherwise it becomes idle. A system which meets the criteria of the above model is said to be an *energy-aware system*.

The system is defined by the 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 utilization (load per server) ρ is defined as $\rho = \lambda/(C\mu)$. Moreover, the well-known $M/M/C$ queue is referred to by a three-tuple (C, λ, μ), with the traditional interpretation of those parameters. In this work a specific control policy for turning on and off servers is denoted by π and our control problem is to determine a control policy that minimizes an appropriate cost function.

The cost function that we consider captures the tradeoff between efficacy (response time performance) and efficiency (energy costs). The expected response time, denoted by $\mathbb{E}[R]$, is employed to evaluate efficacy, while 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*) has a corresponding energy consumption rate. Let these rates be denoted by E_{Off}, E_{Idle}, E_{Busy}, and E_{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 (in steady-state). Then

$$\begin{aligned}\mathbb{E}[E] = E_{\text{Off}}\mathbb{E}[C_{\text{Off}}] + E_{\text{Idle}}\mathbb{E}[C_{\text{Idle}}] \\ + E_{\text{Busy}}\mathbb{E}[C_{\text{Busy}}] + E_{\text{Setup}}\mathbb{E}[C_{\text{Setup}}].\end{aligned} \quad (1)$$

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

Common cost functions considered are $\mathbb{E}[R]\mathbb{E}[E]$ and $\mathbb{E}[R] + \beta\mathbb{E}[E]$ for some parameter $\beta > 0$. The former takes the viewpoint that a decrease by a proportion p in either term is of equal value, while the second allows one to weight the terms as desired in a linear cost function. We would like to make more general observations, so consider the following family of cost functions.

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

In Section III, we first provide several key structural properties that optimal server control policies satisfy - these are taken from [15], [16], but are repeated here, to provide the reader with a self-contained overview of our approach to the control problem. We then proceed in Section IV to show that control policies satisfying these structural results (plus some extra conditions), are asymptotically optimal for all well-formed cost functions, in a fixed-utilization, many-server asymptotic regime.

III. Structural Results

Three key structural results are given in this section. Further details are provided in [15], [16], where these results (and several others) are proved within a Markov Decision Process (MDP) framework under the assumption that the cost function is linear, i.e. is of the form $\mathbb{E}[R] + \beta\mathbb{E}[E]$.

We define the state of the system to be (n_1, n_2, n_3), where n_1 denotes the number of jobs in the system, n_2 denotes the number of servers either *idle* or *busy* (the number of servers on), and n_3 denotes the number of servers in *setup*. The first result shows that the decision to turn on a server follows a threshold policy, i.e. if in state (n_1, n_2, n_3) it is optimal to begin turning on an additional i servers, then in state $(n_1 + 1, n_2, n_3)$ it is optimal to turn on at least i additional servers.

Theorem 1: [16, Theorem 1] The decision to turn on a specific server follows a threshold policy based on the number of jobs in the system.

Note that Theorem 1 of [16] actually shows more than this - the decision to turn a server off also follows a threshold policy. However, this is not required for our asymptotic results.

The next result states that it is suboptimal to turn off servers in an anticipatory manner. While one could not classify this result as entirely counterintuitive, one may have thought that if the load on a system were sufficiently small and the current number of jobs in the system were also

sufficiently small, then it may be advantageous to begin turning off servers, even if these servers could process jobs.

Theorem 2: [16, Theorem 2] Suppose that $\beta\lambda E_{Idle} < 1$. If the number of jobs in the system is greater than or equal to the number of servers currently turned on, it is suboptimal to turn a server off.

The final result of this section demonstrates that the decision to turn a server on should also not be made in an anticipatory manner. Again, this result is not completely intuitive. If the server setups are instantaneous, then this result is obvious. However, one could think that if setup times were sufficiently long, that it might be optimal to begin turning servers on in anticipation of the expected state of the system once the server finishes its setup. The following theorem shows that this is not the case.

Theorem 3: [15, Theorem 3] If in state $(n_1, n_2, n_3 - 1)$ it is optimal to begin turning on the $(n_3 + n_2)^{th}$ server, then in state $(n_1, n_2 + 1, n_3 - 1)$ it is suboptimal to turn off the $(n_2 + 1)^{th}$ server.

We now turn to leveraging these structural results to study how server control policies perform in a fixed-utilization, many-server asymptotic regime.

IV. Fixed-Utilization, Many-Server Asymptotics

The previous section gave structural results that partially describe the optimal policy for well-formed cost functions. One possibility at this point would be to explore if this gives sufficient reduction in the possible control actions to explicitly derive the optimal policy. Unfortunately, to this point such a derivation has not been possible. One avenue that we have explored is whether the determined structure is sufficient to calculate efficiently the performance of policies. One of our previous works [16] does exactly this - it provides an efficient algorithmic approach to computing the performance of systems following the structure described here, providing insight into several design questions. Upon performing the work in [16], we became cognizant of the fact that larger systems appeared to have very similar performance under different control policies that all conformed to the structure described in Section III. What follows is a formalization of that observation.

We first formally define the policies which we analyze.

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 with finite thresholds for all servers.
2) A server will never turn off if there is a job which it could be processing.

Note that these two properties correspond to Theorems 1 and 2, respectively.

Before the second class of policies is given, another definition must first be introduced. Let $X_{\mathcal{E}}(s,t)$ be an indicator function 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 $\mathcal{E} \in \{$*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 $t \to \infty$, $P(X_{\mathit{off}}(s,t) = 1) \to 1$. 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) For all n_1 and n_2, if a server s turns off when there are n_1 jobs in the system and n_2 servers on, then while there are at least n_2 servers on, s will not begin its setup until there are at least $n_1 + 1$ jobs in the system.

The second condition for Class B policies states that the number of servers that are always idle cannot be of the same order as the total number of servers. The third condition corresponds to Theorem 3. It is worth noting that most policies studied in the literature are Class B policies, e.g., the policies in [3]–[6], [9]–[13], [16] are all Class B policies. The sets of Class A and Class B policies are denoted by Π_A and Π_B respectively, and 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.

We consider a fixed-utilization, many-server asymptotic regime. That is, for an 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.

Theorem 4: All policies in Π_A are asymptotically optimal with regards to expected response time. In other words, given an energy-aware system, for any $\pi_a \in \Pi_A$, as $\lambda, C \to \infty$ and $\lambda/(\mu C)$ is fixed to be $0 < \rho < 1$, $\mathbb{E}[R^{\pi_a}] \to 1/\mu$.

While perhaps surprising at first, such a result does have an intuitive explanation. 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 processing 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 4 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 only a sufficient condition for minimizing the expected response time. With optimal policies now known for $\mathbb{E}[R]$, our focus shifts to the second cost metric, $\mathbb{E}[E]$. Note that the appropriate quantity to examine in this case is the energy cost per job, $\mathbb{E}[E]/\lambda$, as $\mathbb{E}[E]$ diverges as $C \to \infty$.

Theorem 5: All policies in Π_B are asymptotically optimal with regards to expected energy cost. In other words, given an energy-aware system, for any $\pi_b \in \Pi_B$, as $\lambda, C \to \infty$ and $\lambda/(\mu C)$ is fixed to $0 < \rho < 1$, $\mathbb{E}[E^{\pi_b}]/\lambda \to E_{Busy}/\mu$.

Complete details of the proof of Theorem 5 are in [17],

but it is instructive to include an outline at this point. The key to the proof is to assign all of the system energy costs to individual jobs (rather than servers) in an intelligent manner. The energy cost assigned to a job consists of the following four components.

1) The energy required to process it.
2) The entire cost of the setup process for the server on which it is served, if it is the first job served following the setup process.
3) The cost of cancelled server setups, if the setups are cancelled due to the job entering service.
4) The idling cost of servers that never turn off is evenly divided across jobs which are served by always busy servers.

One can then argue that the expected cost for a job served by an always busy server is asymptotically determined by the first component only, which is the minimum possible energy consumption (Lemma 2 of [17]). If a job is not served by an always busy server, the expected cost can be shown to be finite (Lemma 3 of [17]). From there, similar to the procedure in the proof of Theorem 4, it becomes clear that the total expected energy cost is dominated by jobs which are served by always busy servers, and therefore is minimized.

Combining Theorem 4 and 5 immediately yields asymptotic optimality.

Corollary 1: All Class B policies are asymptotically optimal under all well-formed cost functions.

The most significant implication of Corollary 1 is that under the asymptotic regime there is no significant trade-off between $\mathbb{E}[R]$ and $\mathbb{E}[E]$. 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 asymptotic regime? We address this question by performing several numerical experiments.

V. Numerical Experiments

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 $\{\lfloor\{j - C^*\}^+/k\rfloor - i\}^+$, where k is the threshold parameter of the policy, $C^* + i$ 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 [18]. 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 of [3]. All experiments have energy costs $E_{Busy} = E_{Setup} = 1.0$ and $E_{Idle} = 0.7$.

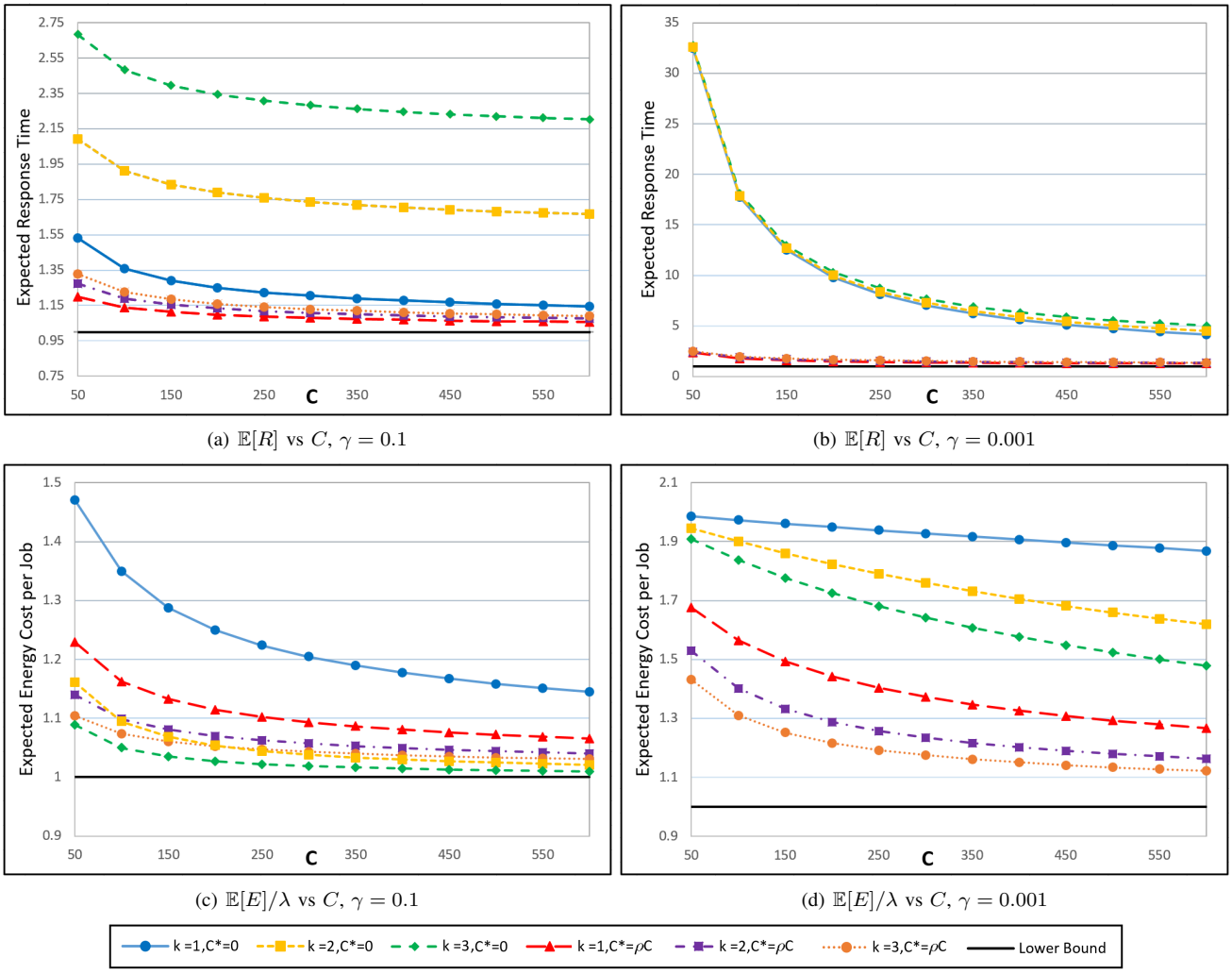
This policy leads to a CTMC model of the system, which has the form of a quasi birth-death process. This can be analysed using any of a number of well-understood methods. We chose to analyse the CTMCs using the RRR technique described in [3]. Therefore, all numerical results are exact and were evaluated using standard Matlab libraries. The source code for the numerical analysis can be found at [19]. 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 the rate of convergence to the asymptotic regime.

Figures 1 (a) and (b) show the behaviour of $\mathbb{E}[R]$ as the system is scaled up. (Note that the largest values of C displayed are significantly fewer than the number of servers in large datacentres, which can be in the tens of thousands.) One observation is that for the curves where $C^* = 0$, the corresponding values of $\mathbb{E}[R]$ 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 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. The curves that have a number of servers which are forced to be on, i.e. $C^* = \rho C$, get 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.

The appealing choice of forcing $\lambda/\mu = \rho C$ servers to always remain on is interesting, since as will be seen in Section VI, 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 $\mathbb{E}[R]$ are closer to values which would be seen under the asymptotic regime.

Shifting focus to the expected energy cost per job and Figures 1 (c) and (d), a similar trend regarding the choice of C^* is seen. That is, when C^* is forced to take on the value of the number of always busy servers under the asymptotic regime, $\mathbb{E}[E]/\lambda$ approaches its optimal value.

Another aspect of these systems which warrants attention is how sensitive the convergence rate is to the utilization. This is seen in Figure 2. Examining the effect of utilization 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 utilization. Furthermore, the most sensitive parts of the curves are when the utilization is low or high, especially in Figure 2 (b) where the setup times are longer. This makes some intuitive sense, since when the utilizations are at either extreme, the system is



(a) $\mathbb{E}[R]$ vs C, $\gamma = 0.1$

(b) $\mathbb{E}[R]$ vs C, $\gamma = 0.001$

(c) $\mathbb{E}[E]/\lambda$ vs C, $\gamma = 0.1$

(d) $\mathbb{E}[E]/\lambda$ vs C, $\gamma = 0.001$

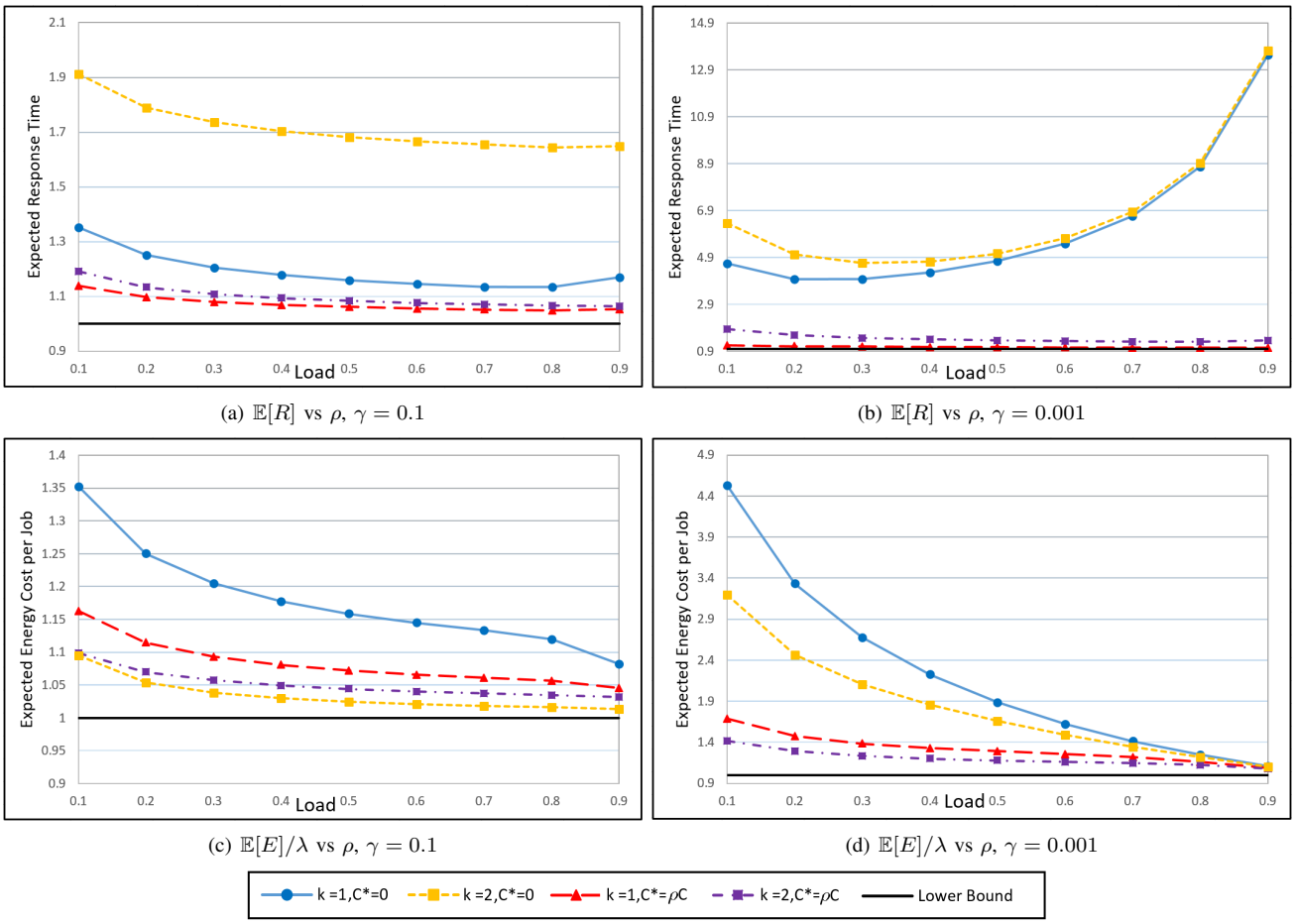
Fig. 1: Expected response time and Expected energy cost per job vs C for $\lambda = C/2$, $\mu = 1$

more likely to exhibit behaviour that is not as well described by the asymptotic regime. When the utilization is low, 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 utilization 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 many servers regularly thrashing, or having more jobs in the system than servers are two characteristics which are not properties of the asymptotic regime. Therefore, it is intuitive that a system under high or low utilizations would be slower to exhibit asymptotic behaviour than under medium utilization.

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 2 (c). In this case the system is approaching the minimum value 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, but is only achieved at the price of a large value of $\mathbb{E}[R]$, and therefore this configuration would not be suggested.

VI. Proof of Key Lemma

The proofs of Theorems 4 and 5 are presented in detail in [17]. Here, we present the key lemma that underlies the main results. This shows that the number of servers that are always busy is no less than the proportion required by the utilization (obviously it can be no more). Once one has this result in hand, one can show that the mean response time is minimized by noting that the system is asymptotically equivalent to an $M/M/C$ system with load per server less than one and C approaching infinity and hence the expected time waiting to be processed is asymptotically equal to zero. In terms of the expected energy costs, the fact that the proportion of always busy servers is precisely equal to the load per server means that almost every job is served by an always busy server and hence almost every job uses an amount of energy equal to



(a) $\mathbb{E}[R]$ vs ρ, $\gamma = 0.1$

(b) $\mathbb{E}[R]$ vs ρ, $\gamma = 0.001$

(c) $\mathbb{E}[E]/\lambda$ vs ρ, $\gamma = 0.1$

(d) $\mathbb{E}[E]/\lambda$ vs ρ, $\gamma = 0.001$

Fig. 2: Expected response time and Expected energy cost per job vs ρ for $C = 500$, $\mu = 1$

the minimum required (E_{Busy}/μ).

Before presenting this lemma, some preliminary material must first be presented. Consider the following two sequences of systems with $0 < \rho < 1$, where for each we assume that as $n \to \infty$, $\lambda_n/n \to \rho$. S_1 is the sequence of interest (a sequence of energy-aware systems), while S_2 is an auxiliary sequence of regular $M/M/C$ systems (all of the servers are on all of the time).

1) Let S_1 be a sequence of energy-aware queueing systems, where the nth energy-aware queueing system is given by $S_{1,n} = (n, \lambda_n, 1, \gamma_{1,n})$, which employ some policy $\pi_n \in \Pi_A$.
2) Let S_2 be a sequence of $M/M/C$ queues, where the nth $M/M/C$ queue is denoted by $S_{2,n} = (n, \lambda_n, 1)$.

Note that the choice of $\mu_n = 1$ is without loss of generality, as time can be rescaled in a corresponding manner. Let $B_{1,i}$ and $B_{2,i}$ denote the number of always busy servers in the ith system of sequence S_1 and S_2, respectively.

Lemma 1: For the sequence of energy-aware systems S_1

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

Proof: The proof is via a sample path argument comparing 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 servers off or in setup, while $S_{2,n}$ has all of its 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 server 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 being 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, a server will never turn off if there is work to do and a server will eventually turn on from the threshold scheme. It then 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,

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

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 the system $S_{2,n}$ at time t, and let

a corresponding diffusion scaling be denoted by $\hat{N}_{2,n}(t)$, where

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

From Theorem 4.1 in [20], $\hat{N}_{2,n}(t)$ weakly converges to an Ornstein-Uhlenbeck process. After some elementary algebra,

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

As $n \to \infty$, $\hat{N}_{2,n}(t)$ is normally distributed with finite mean and variance, so $\lim_{n\to\infty} \hat{N}_{2,n}(t)/\sqrt{\lambda_n} = 0$. Thus,

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

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

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

We can now examine S_1. Specifically, from (2),

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

Lemma 1 follows as the reverse inequality is obvious. ■

VII. Future Work

An important issue to address is how well these policies perform under a time varying arrival rate. It is our intuition that if the arrival rate varied on a relatively long time scale, with regards 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 theorems presented here regarding the optimality of all Class B policies under all well-formed cost functions allow one to confidently make powerful statements and conclusions which are overarching across the problem domain.

Acknowledgment: This research was funded by the Natural Sciences and Engineering Research Council of Canada.

References

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

[2] J. Slegers, N. Thomas, and I. Mitrani, "Dynamic server allocation for power and performance," in *SPEC International Workshop on Performance Evaluation: Metrics, Models and Benchmarks*, pp. 247–261, 2008.

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

[4] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

[5] A. Gandhi and M. Harchol-Balter, "M/M/k with exponential setup." Technical Report, Carnegie Mellon University, 2010.

[6] A. Gandhi, M. Harchol-Balter, and I. Adan, "Server farms with setup costs," *Performance Evaluation*, vol. 67, no. 11, pp. 1123–1138, 2010.

[7] I. Mitrani, "Managing performance and power consumption in a server farm," *Annals of Operations Research*, vol. 202, no. 1, pp. 121–134, 2013.

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

[9] X. Xu and N. Tian, "The M/M/c queue with (e, d) setup time," *Journal of Systems Science and Complexity*, vol. 21, no. 3, pp. 446–455, 2008.

[10] P. J. Kuehn and M. E. Mashaly, "Automatic energy efficiency management of data center resources by load-dependent server activation and sleep modes," *Ad Hoc Networks*, vol. 25, no. 2, pp. 497–504, 2015.

[11] T. Phung-Duc, "Exact solutions for M/M/c/setup queues," *Telecommunication Systems*, pp. 1–16, 2016.

[12] T. Phung-Duc and K. Kawanishi, "Energy-aware data centers with s-staggered setup and abandonment," in *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pp. 269–283, Springer, 2016.

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

[14] D. Mukherjee, S. Dhara, S. Borst, and J. Leeuwaarden, "Optimal service elasticity in large-scale distributed systems," in *ACM SIGMETRICS 2017*, 2017.

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

[16] V. Maccio and D. Down, "Structural properties and exact analysis of energy-aware multiserver queueing systems with setup times," *Performance Evaluation*. Accepted.

[17] V. J. Maccio and D. G. Down, "Asymptotic performance of energy-aware multiserver queueing systems with setup times." Technical Report, `http://www.cas.mcmaster.ca/cas/0reports/CAS-16-05-DD.pdf`, 2016.

[18] V. Maccio and D. Down, "Exact analysis of energy-aware multiserver queueing systems with setup times," in *MASCOTS 2016*, 2016.

[19] "Source code." `http://www.cas.mcmaster.ca/~macciov/publications.html`. Accessed: 2016-10-10.

[20] D. L. Iglehart, "Limiting diffusion approximations for the many server queue and the repairman problem," *Journal of Applied Probability*, vol. 2, no. 2, pp. 429–441, 1965.