On Optimal Policies for Energy-Aware Servers

Vincent J. Maccio
Department of Computing and Software
McMaster University
Hamilton, Ontario
Email: macciov@mcmaster.ca

Douglas G. Down
Department of Computing and Software
McMaster University
Hamilton, Ontario
Email downd@mcmaster.ca

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

I. Introduction

The relative as well as absolute energy consumed by servers have been steadily increasing in North America over the past several years. As systems grow and expand, energy concerns become a major factor for server farm managers from both environmental and economic viewpoints. However, the task of creating feasible optimal or near optimal policies is a daunting problem due to the sheer complexity these systems exhibit. Even for single server systems, when energy is a factor, optimal policies remain unknown for a number of metrics considered in the literature. We focus on developing a model that allows one to determine an optimal policy for a single server system under a broad range of metrics that consider the expected response time of a job in the system ($\mathbb{E}[R]$), the expected energy consumed by the system ($\mathbb{E}[E]$), and the expected rate that the server switches between two energy states, i.e. turning off and on ($\mathbb{E}[Sw]$).

The typical approach to developing energy-aware policies focuses on a particular metric. In [1]–[3] for example, simple optimal policies were determined with respect to the energy response product (ERP) metric, $\mathbb{E}[R]\mathbb{E}[E]$. However, many simple metrics still have unknown optimal policies. For example, the optimal policy for the simple weighted sum used in [4], [5], [8], [14] of $\mathbb{E}[R]+\beta_1\mathbb{E}[E]+\beta_2\mathbb{E}[Sw]$ is unknown. Other work has been done in the analysis of vacation models [11]–[13], which in many cases can be leveraged to fit energy-aware server systems. However, to the best of our knowledge, no known vacation model can describe all optimal policies under these different metrics. Here, we are able to determine optimal policies in great generality, both in terms of the cost function and the assumptions on underlying distributions.

II. Model

We wish to capture the behaviour of a single server system, where the server can be dynamically set to a low or high energy state. Furthermore, we wish to add the restriction that jobs may only be processed when the server is in its higher energy state. Such a system is modelled as being in one of four system states: *LOW, SETUP, BUSY,* or *IDLE*. Each of these states has a corresponding energy value E_{Low}, E_{Setup}, E_{Busy}, and E_{Idle}, respectively. For simplicity of analysis and understanding, if $E_{Low} = 0$, we rename *LOW* to *OFF*. We will see that optimal policies typically depend on the ratio of the energy values rather than the values themselves. We take these ratios with respect to E_{Busy}, and denote them as r_{Low}, r_{Setup}, and r_{Idle} (in practice, r_{Idle} is typically between 0.6 and 0.8 [6], [7]). For the remainder of this paper we will often refer to moving to a higher or lower energy state as turning the server on or off, respectively.

Jobs arrive to the system according to a Poisson process of rate λ and are put in a FIFO queue. If the system is in state *LOW/OFF* when a job arrives, it checks how many jobs are currently waiting in the queue. If the number in the queue plus the arriving job is equal to a given threshold k, the system moves into state *SETUP*. This corresponds to the server moving from its lower to higher energy state. The time it takes to make this transition is exponentially distributed with rate γ. Once the server has completed making its transition from its low to high energy state, it leaves state *SETUP* and enters state *BUSY*. Once in state *BUSY*, the server begins to process the accumulated jobs. The job processing times are exponentially distributed with rate μ. When a job is completed and no jobs remain in the queue, the system moves to state *IDLE*. Upon reaching state *IDLE*, the system begins to wait. If no job arrives to the system once the server has waited a given amount of time, the system moves to state *LOW/OFF*. If a job arrives while the system is in *IDLE*, it moves to state *BUSY* and the waiting time is remembered for the next time the system becomes *IDLE*. The amount of time in which the system waits is referred to as the idling time and is exponentially distributed with rate α. It is important to note that the time between moving from *IDLE* to *OFF*, is not the time which it takes for a server to turn off. For the purpose of our model, we assume that the time taken for the system to move from its high to low energy state is negligible, i.e. the transition occurs

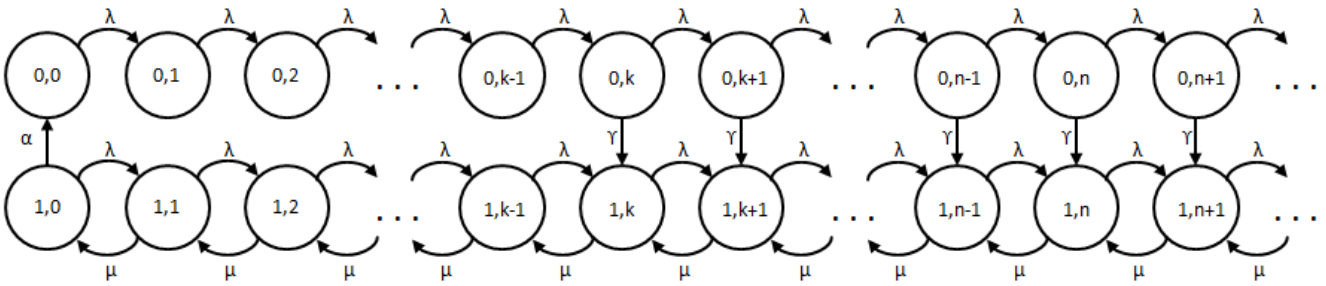


Fig. 1: $M/M/1 \circ \{M, M, k\}$ queue Markov Chain

instantaneously.

Due to the exponential assumptions, this system can be modelled as a continuous time Markov chain and is depicted in Figure 1, where the state (n_1, n_2) means that the server is off when $n_1 = 0$, on when $n_1 = 1$, and there are n_2 jobs in the system. To denote these systems we use a composition of two sets of parameters i.e. $\{\} \circ \{\}$. The first set of parameters is given in classical Kendall notation to describe the non-energy-aware portions of the system. The set of parameters listed after the composition symbol are all parameters which are incorporated due to energy concerns. The first of these parameters is the turn on time distribution of the server, the second is the idling time distribution, and the last is the number of jobs allowed to accumulate before the server begins to turn on. For example, the queue in Figure 1 is an $M/M/1 \circ \{M, M, k\}$ system while if the job processing times along with the server turn on times follow general distributions, the system would be an $M/G/1 \circ \{G, M, k\}$. The reason for denoting the systems in this way, as we will show later, is that their metrics can often be written as a decomposition where one of the terms will be the corresponding metric of the non-energy-aware counterpart (the first set of parameters).

A. Assumptions Justification and Parameter Summary

The model includes several assumptions in order to be tractable. Firstly, arrival times, set-up times, processing times, and idling times are initially all assumed to be exponentially distributed. The assumptions on the arrival and processing times are quite standard for approximating systems of this kind. It will be shown later on that the exponential assumption for the idling times is dampened by properties of the optimal policies. However, in general, the assumption that the turn on times of the servers as well as the job processing times are exponentially distributed is typically not a good approximation. Later in our analysis we relax these assumptions on the distributions and analyse the system under general settings.

There are several constraints imposed on the model to ensure stability and that the model is non-trivial:

$$0 < \lambda < \mu, \quad 0 < \gamma, \quad 0 \leq \alpha, \quad 1 \leq k.$$

The parameters of the model are summarized in Table I.

TABLE I: Parameter Summary

Parameter(s)	Explanation
E_{Low}, E_{Setup}, E_{Busy}, E_{Idle}	The energy values associated with the different system states.
r_{Low}, r_{Setup}, r_{Idle}	The ratios between the system states energy values and E_{Busy}.
λ	The arrival rate of jobs to the system.
μ	The server's processing rate.
γ	The rate at which the server moves to its higher energy state from the lower.
α	The rate at which a server waits while idle before moving to its lower energy state.
k	The number of jobs the system allows to accumulate in the queue before beginning to move to the higher energy state.

III. Analysis

The goal of our analysis is to arrive at closed form expressions for a range of system metrics. Namely we wish to solve for the expected number of jobs in the system, the expected response time of a job, the expected energy used by the system, and the expected rate of switching to the system's lower energy state from its higher energy state. In our analysis, we denote these quantities as $\mathbb{E}[N]$, $\mathbb{E}[R]$, $\mathbb{E}[E]$, and $\mathbb{E}[Sw]$, respectively. Once we derive these expressions, we can solve for optimal values of the parameters which the system manager has control over, in particular α and k.

A. Set of Optimal Policies

Before we begin to analyse our model, we must first define what we mean by an optimal policy. We define our cost to be a function of M weighted terms each containing $\mathbb{E}[R]$, $\mathbb{E}[E]$, $\mathbb{E}[Sw]$, each raised to given powers. We leave out the system metric of $\mathbb{E}[N]$ since we can always obtain it by weighting $\mathbb{E}[R]$ by $\frac{1}{\lambda}$ via Little's Law. Formally, our cost function $f(\beta, w)$ is,

$$f(\beta, w) = \sum_{i=1}^{M} \beta_i \mathbb{E}[R]^{w_{R,i}} \mathbb{E}[E]^{w_{E,i}} \mathbb{E}[Sw]^{w_{Sw,i}}, \qquad (1)$$

where $\forall i . 0 \leq \beta_i, w_{R,i}, w_{E,i}, w_{Sw,i}$ are of the appropriate units. Our model makes two assumptions about the optimal policies.

$$\pi_{0,n} = \pi_{0,0} \qquad (0 \le n < k) \quad (2)$$

$$\pi_{0,n} = \pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \qquad (k \le n) \quad (3)$$

$$\pi_{1,n} = \pi_{0,0}\left(\frac{\lambda}{\alpha}\rho^n + \frac{\lambda}{\mu-\lambda}(1-\rho^n)\right) \qquad (0 \le n < k) \quad (4)$$

$$\pi_{1,n} = \pi_{0,0}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^n + \frac{1}{\mu-\lambda-\gamma}\left((\lambda+\gamma)\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} - \frac{\gamma}{1-\rho}\rho^{n-(k-1)}\right)\right] \qquad (k \le n) \quad (5)$$

$$\pi_{0,0} = (1-\rho)\frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \quad (6)$$

- The decision to start transitioning between lower and higher energy states is made at the moment a job arrives to the system.
- If there are jobs in the system and the server is in its higher energy state, the server will never move to its lower energy state.

The first assumption is made without loss of generality due to the memoryless property of the arrival stream (the same decision would be made at any point in time between arrivals). The second assumption is a property of the optimal policy due to the nature of the cost function. If the system were to turn the server off while a job(s) remains in the system, $\mathbb{E}[R]$ will increase, since the job(s) that was in the system when it turned off must now wait until the system turns on before it can be completed. At the same time, the system does not gain any benefit with respect to the $\mathbb{E}[E]$ component since it will still have to expend energy to complete the job(s) in the system at some point in the future. So, as the weights in the cost function are positive we know that in the optimal policy the server will only be turned off while the server is idling. Similar assumptions are made in the model used in [2]. Knowing that these two assumptions are valid, we know that any optimal policy can be instantiated using the model we have described, under the model's assumptions.

Similar to the argument made to justify the servers beginning to turn on only when an arrival occurs to the system, the decision to turn a server off or keep it on is made when a job departs the system and leaves it idle. This would imply that in our model, in any policy which minimizes the cost, $\alpha = 0$ or $\alpha \to \infty$. We leave α as part of our model for several reasons. Firstly, it gives us insight on how scaling between these two extremes affects the system. Secondly, it allows us to easily determine where in the parameter space the optimal policy switches between $\alpha = 0$, and $\alpha \to \infty$. Thirdly, it allows for easier extensions of the model where this property may not necessarily hold. For example, this property does not hold when the arrivals do not follow a Poisson process, or in a multi-server setting. Lastly when optimizing under different conditions, i.e. minimizing a linear function of $\mathbb{E}[E]$ with a constraint on $\mathbb{E}[R]$, the optimal α could lie anywhere on the positive real line.

B. Steady State

To analyse our model, we begin by solving the steady state probabilities for the Markov chain in Figure 1. Each row of the Markov chain is partitioned into two sections according to $n < k$, or $n \ge k$. The balance equations used to solve for the four different sections of the Markov chain are:

$$\begin{aligned}
\pi_{0,n} &= \pi_{0,0} && (n < k)\\
(\lambda+\gamma)\pi_{0,n} &= \lambda\pi_{0,n-1} && (n \ge k)\\
\mu\pi_{1,n} &= \lambda\pi_{1,n-1} + \lambda\pi_{0,n-1} && (0 < n < k)\\
(\mu+\lambda)\pi_{1,n} &= \lambda\pi_{0,n-1} + \gamma\pi_{0,n} + \mu\pi_{1,n+1} && (n \ge k)
\end{aligned}$$

where π_{n_1,n_2} denotes the steady state probability of being in state (n_1, n_2). We also have the boundary and normalization conditions:

$$\pi_{1,0} = \frac{\lambda}{\alpha}\pi_{0,0} \quad \text{and} \quad \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}\pi_{n_1,n_2} = 1$$

While the first three balance equations can be solved with respect to $\pi_{0,0}$ via simple recursions, the fourth equation takes more effort to solve. However, using similar methods to those used in [2], we are able to arrive at a closed form solution. For $n > k$, we fit the steady state distribution to be of the form,

$$\pi_{1,n} = A\rho^{n-(k-1)} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}$$

where with the use of the boundary equations we find that,

$$B = \pi_{0,0}\frac{\lambda+\gamma}{\mu-\lambda-\gamma}$$

and,

$$A = \pi_{0,0}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^{k-1} - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)}\right].$$

With the balance equations solved we use some algebra to yield the steady state distribution for our system model.

Theorem 1. *The steady state distribution for an $M/M/1 \circ \{M, M, k\}$ queue, depicted by the Markov chain in Figure 1 is given by the set of equations (2)-(6).*

C. System Metrics

With the steady state distribution of our model now solved, we wish to arrive at closed form expressions for the system metrics, namely $\mathbb{E}[N]$, $\mathbb{E}[R]$, $\mathbb{E}[E]$, and $\mathbb{E}[Sw]$. Determining these expectations will allow us to build expressions for our cost function and in turn allow us to arrive at optimal values for α and k.

The simplest expression to solve for is $\mathbb{E}[Sw]$, the expected rate at which the server turns off. The only state from which the server turns off is the IDLE state, $\pi_{1,0}$. Therefore the expected switching rate is just the rate out of IDLE going to OFF.

$$\mathbb{E}[Sw] = \alpha\pi_{1,0} = (1-\rho)\frac{\alpha\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{7}$$

Here we see some things we would expect. Firstly, the direct relationship to $(1-\rho)$ is quite intuitive as a heavily loaded system would rarely switch off. Secondly, k only appears in the denominator, giving $\mathbb{E}[Sw]$ an inverse relationship to k. This is also expected as allowing k jobs to build up slows down the turn on rate of the server as k increases, and the expected turn on rate is equal to the expected turn off rate.

We solve $\mathbb{E}[E]$ by viewing it as a sum of being in states *OFF*, *IDLE*, *SETUP*, and *BUSY* weighted by the corresponding energy values. We sum the states using equations (2)-(6), and exploit our assumption that $E_{Low} = 0$ in state *OFF*.

$$\begin{aligned}
\mathbb{E}[E] &= E_{Busy}\sum_{n=1}^{\infty}\pi_{1,n} + E_{Setup}\sum_{n=k}^{\infty}\pi_{0,n} + E_{Idle}\pi_{1,0} \\
&= E_{Busy}\left[\rho + \frac{(1-\rho)\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(r_{Idle}\gamma + r_{Setup}\alpha)\right] \\
&= \mathbb{E}[E_{M/M/1}] \\
&\quad + E_{Busy}\frac{(1-\rho)\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{setup} - (\lambda + k\gamma)r_{idle})
\end{aligned}$$

This gives us the true expected energy of the system, however since in our cost function we weigh $\mathbb{E}[E]$ by a constant β, we can absorb the constant E_{Busy}, and derive a new metric normalized by this weight:

$$\begin{aligned}
\mathbb{E}[E^N] &= \frac{\mathbb{E}[E]}{E_{Busy}} \\
&= \mathbb{E}[E^N_{M/M/1}] + \frac{(1-\rho)\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{setup} - (\lambda + k\gamma)r_{idle}).
\end{aligned} \tag{8}$$

We arrive again at a decomposition, the terms here are scaled by ρ or $(1-\rho)$. As we would expect there is an $E_{Busy}\rho$ term present, since based on our model assumptions $E_{Busy}\rho$ is a lower bound to the expected energy consumed by the system. We also note that letting $\alpha = 0$, simply leaves us with $\mathbb{E}[E^N_{M/M/1}]$, the expected normalized energy consumption in an M/M/1 queue, as anticipated. How the rest of the terms arise is at this point not intuitively clear, but in the next section we give a different point of view on $\mathbb{E}[E]$ which allows us to gain much more intuition.

To solve for $\mathbb{E}[R]$, we use the traditional method of solving first for $\mathbb{E}[N]$ by weighting the steady state distribution and then applying Little's Law.

$$\mathbb{E}[N] = \sum_{n=0}^{k-1} n\pi_{0,n} + \sum_{n=k}^{\infty} n\pi_{0,n} + \sum_{n=0}^{k-1} n\pi_{1,n} + \sum_{n=k}^{\infty} n\pi_{1,n}$$

After quite a bit of algebra we are able to write:

$$\begin{aligned}
\mathbb{E}[N] = \mathbb{E}[N_{M/M/1}] &+ \frac{\alpha\lambda(\lambda + k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)} \\
&+ \frac{k\alpha\gamma(k-1)}{2(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}.
\end{aligned} \tag{9}$$

Applying Little's Law gives us:

$$\begin{aligned}
\mathbb{E}[R] = \mathbb{E}[R_{M/M/1}] &+ \frac{1}{\gamma}\frac{\alpha(\lambda + k\gamma)}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \\
&+ \frac{1}{2\lambda}\frac{k\alpha\gamma(k-1)}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}.
\end{aligned} \tag{10}$$

Both terms yield convenient decompositions. We would expect to find some form of the $M/M/1$ queue embedded within the $M/M/1 \circ \{M, M, k\}$ queue since many of its metrics are optimized when their behaviours are equivalent ($\alpha = 0$). For the terms $\mathbb{E}[N]$ and $\mathbb{E}[R]$, we obtain exact $M/M/1$ expressions when we let $\alpha = 0$. $\mathbb{E}[N_{M/M/1}]$ and $\mathbb{E}[R_{M/M/1}]$ are lower bounds to $\mathbb{E}[N]$ and $\mathbb{E}[R]$, respectively.

To analyse the second term of (10), it is easier to first allow $k = 1$ which will eliminate the third term. With $k = 1$ and letting α approach ∞ our system reduces to that of the system described in [2], where $\mathbb{E}[R] = \mathbb{E}[R_{M/M/1}] + \frac{1}{\gamma}$. We can see from (10) that this is also the case here. So the expected response time of a job is bounded below by $\mathbb{E}[R_{M/M/1}]$ and bounded above by $\mathbb{E}[R_{M/M/1}] + \frac{1}{\gamma}$ when $k = 1$. Moving α along the positive real line scales $\mathbb{E}[R]$ between these two bounds.

The third and last term of (10) is the effect imposed on the response time when k jobs are allowed to accumulate. As k increases, we see a linear increase in the response time in the third term. In the second term k appears in both the numerator and denominator scaled by the same coefficient so the effect which is present is dampened, and in fact as k approaches ∞ the second term still approaches $\frac{1}{\gamma}$.

Viewing equations (7), (8), and (10) together, one begins to understand the mathematical difficulty of optimization of a given metric with respect to choosing α and k. Each individual metric prefers α and k to be either set to their respective upper or lower bounds, but unfortunately they pull in different directions, as seen in Table II.

TABLE II: Optimal Parameters of Metrics

	Optimal Values of	
Metric	α	k
$\mathbb{E}[R]$	0	1
$\mathbb{E}[E]$	0 or $\to\infty$	$\to\infty$
$\mathbb{E}[Sw]$	0	$\to\infty$

Note that to minimize $\mathbb{E}[E]$, the system will let $\alpha = 0$ when

$$r_{idle} < \frac{\lambda}{k\gamma + \lambda} r_{setup}$$

and will let $\alpha \to \infty$ otherwise.

D. Work-cycle Analysis

Here we approach the analysis of our system from a different angle. This method allows us to relax several of our assumptions while still arriving at closed form expressions, as well as allowing us to gain deeper insight and intuition into the equations we derived in the previous section.

We view the system using the rate at which "Work-cycles" complete. Let $S_{0,0}$ denote the state of the system where the server is off and there are 0 jobs in the system, and let $P_{0,0}$ denote the proportion of time the system spends in $S_{0,0}$ in steady state. We define a Work-cycle to start at state $S_{0,0}$, moving through the energy state *OFF* into state *SETUP*. Once the server has turned on, it continues to move between states *BUSY* and *IDLE* a number of times before it lastly moves from *IDLE* back to $S_{0,0}$. In our model, during a Work-cycle the total time spent idling is remembered, and not reset each time the server switches between *BUSY* and *IDLE*. Once the system moves back to state $S_{0,0}$ however, the idling time is reset and all knowledge of the past is forgotten. Since for every Work-cycle the system visits state $S_{0,0}$ exactly once, the rate at which Work-cycles occur in the system is the rate out of state $S_{0,0}$. The rate out of state $S_{0,0}$ is simply the arrival rate to the system, λ. Therefore in steady state the rate of Work-cycles is $\lambda P_{0,0}$.

We also make the observation that the expected proportion of time which the system spends in states *OFF*, *SETUP*, *BUSY* and *IDLE* (denoted P_{Off}, P_{Setup}, P_{Busy}, and P_{Idle} respectively) over just one of its Work-cycles, is equal to the proportion of time the system spends in those states, in steady state. For each Work-cycle the server turns on a single time, therefore P_{Setup} equals the product of the Work-cycle rate and the expected turn on time of the server i.e. $\frac{\lambda}{\gamma} P_{0,0}$. This same argument can be used for P_{Off}, which is the time it takes for k jobs to arrive to the system multiplied with the Work-cycle rate, $k\frac{1}{\lambda}\lambda P_{0,0} = kP_{0,0}$. We know that the rate into state $S_{0,0}$ must equal the rate out which implies $P_{1,0} = \frac{\lambda}{\alpha} P_{0,0}$. However, once again this is also just the product of the expected time waited before turning off and the Work-cycle rate of the system. Finally we get the proportion of time the system spends in state *BUSY* for free since we know it must be ρ. Putting it all together we have:

$$1 = \rho + \frac{\lambda}{\alpha} P_{0,0} + \frac{\lambda}{\gamma} P_{0,0} + kP_{0,0}.$$

This analysis has been done without imposing assumptions on any of the distributions, expect for the arrival stream. This assumption was made when we assumed the rate out of state $S_{0,0}$, is $\lambda P_{0,0}$. Isolating and solving for $P_{0,0}$ we find it is equal to $\pi_{0,0}$ from our previous analysis, and the same can be said for the expected energy used by the system.

Theorem 2. *The proportion of time spent in the energy states of an $M/G/1 \circ \{G, G, k\}$ queue, is completely insensitive to the distributions themselves, giving general expressions for $\mathbb{E}[E^N]$ and $\mathbb{E}[Sw]$. That is,*

$$\mathbb{E}[E^N] = \rho + \frac{(1-\rho)\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(r_{Idle}\gamma + r_{Setup}\alpha)$$

$$\mathbb{E}[Sw] = (1-\rho)\frac{\alpha\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

and for any single server system if $r_{idle} < \frac{\lambda}{k\gamma+\lambda} r_{setup}$ it is always optimal, to leave the server on.

It may be counter intuitive for the proportion of time spent in different energy states to be independent of the underlying distributions of the processing and setup times. However, this is similar to the fact that an $M/G/1$ queue is in state *BUSY* and *IDLE* with probabilities ρ and $(1-\rho)$ respectively.

While the energy and switching metrics can be solved in almost complete generality, the response time is harder to arrive at. We therefore again impose exponential assumptions upon the idling times of the system, but still allow for general distributions on the the processing and setup times. Some generality is lost, but we argue that the exponential assumptions on the idling and interarrival times are not nearly as limiting. For many applications, modelling the arrivals as a Poisson process is a reasonable assumption, while as we have stated before, having the server turn on times and job processing times being exponentially distributed can be problematic. We also know that if the arrivals do follow a Poisson process then α is either 0 or approaches ∞, meaning the actual distribution has little impact. With this in mind, we analyse the $M/G/1 \circ \{G, M, k\}$ queue with the goal of determining $\mathbb{E}[R]$.

We tackle the problem similar to the way one would traditionally solve an $M/G/1$ queue. We define N_n to be a random variable denoting the number of jobs left in the system as the n^{th} job departs. As in the $M/G/1$ analysis,

$$N_{n+1} = \begin{cases} N_n + A_{n+1} - 1 & N_n \geq 1 \\ A_{n+1} & N_n = 0 \end{cases}$$

where A_{n+1} denotes the number of arrivals which occurred between the departure of the n^{th} and $(n+1)^{th}$ jobs, not counting the $(n+1)^{th}$ if it arrived during that period. For our model, we have to condition A_{n+1} on N_n,

$$A_{n+1} = \begin{cases} A_{S,n} & N_n \geq 1 \\ A_{S,n} + X_{Off,n}(k - 1 + A_{\Gamma,n}) & N_n = 0 \end{cases}$$

where $A_{S,n}$ is a random variable denoting the number of jobs which arrive while the n^{th} job is being processed. $A_{\Gamma,n}$ is a random variable denoting the number of jobs which arrive to the system during the server's setup time, given the $(n+1)^{th}$ job is the first to arrive once the server has switched off. $X_{Off,n}$ is an indicator variable that is 1 when the system is in state *IDLE* and the next state it moves to is *OFF* or 0 if the next state it moves to is *BUSY*, given that the n^{th} job to depart, leaves behind an empty system. We note that the distributions

for all three of these random variables are independent of n, and from here on refer to them simply as A_S, A_Γ, and X_{Off}. We can now rewrite the expressions for N_{n+1} and A_{n+1} with the use of the Heaviside step function.

$$\begin{aligned} N_{n+1} &= N_n - \mathcal{U}(N_n) + A_{n+1} \\ A_{n+1} &= A_S + (1-\mathcal{U}(N_n))X_{Off}(k-1+A_\Gamma) \\ \Rightarrow N_{n+1} &= N_n - \mathcal{U}(N_n) + A_S \\ &\quad + (1-\mathcal{U}(N_n))X_{Off}(k-1+A_\Gamma) \end{aligned} \tag{11}$$

If we let $n \to \infty$ and then take the expectation of both sides, the N_n and N_{n+1} terms cancel out. We also exploit the fact that X_{Off} is independent from A_Γ, since A_Γ is dependent only on the interarrival and turn on times. After some algebra we are left with an expression for $\mathbb{E}[\mathcal{U}(N)]$.

$$\mathbb{E}[\mathcal{U}(N)] = \frac{\mathbb{E}[A_S] + \mathbb{E}[X_{Off}](k-1+\mathbb{E}[A_\Gamma])}{1+\mathbb{E}[X_{Off}](k-1+\mathbb{E}[A_\Gamma])}$$

This should not give us any new information about the system, as an $M/G/1$ queue this would yield $\mathbb{E}[\mathcal{U}(N)] = \rho$. Of course the interpretation of $\mathbb{E}[\mathcal{U}(N)]$ is the steady state probability there is at least one job in the system. From our previous analysis we know this to be:

$$1 - P_{0,0} - P_{1,0} = \rho + (1-\rho)\alpha\frac{\gamma(k-1)+\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}.$$

As a sanity check this is what $\mathbb{E}[\mathcal{U}(N)]$ evaluates to when,

$$\mathbb{E}[A_S] = \rho, \quad \mathbb{E}[X_{Off}] = \frac{\alpha}{\lambda+\alpha}, \quad \text{and} \quad \mathbb{E}[A_\Gamma] = \frac{\lambda}{\gamma}.$$

To arrive at $\mathbb{E}[N]$, we use the usual approach: square both sides of (11), let $n \to \infty$, take expectations and exploit the following equalities.

$$\begin{aligned} \mathcal{U}^2(N) &= \mathcal{U}(N) \\ N\mathcal{U}(N) &= N \\ N(1-\mathcal{U}(N)) &= 0 \\ \mathcal{U}(N)(1-\mathcal{U}(N)) &= 0 \\ \mathbb{E}[X_{Off}A_S] &= \mathbb{E}[X_{Off}]\mathbb{E}[A_S] \\ \mathbb{E}[X_{Off}A_\Gamma] &= \mathbb{E}[X_{Off}]\mathbb{E}[A_\Gamma] \end{aligned}$$

Substituting those equations into (10) after squaring both sides yields,

$$\begin{aligned} &2(1-\mathbb{E}[A_S])\mathbb{E}[N] = \\ &\quad \mathbb{E}[\mathcal{U}(N)][1+2\mathbb{E}[A_S](1-\mathbb{E}[X_{Off}](k-1+\mathbb{E}[A_\Gamma]))] \\ &\quad + \mathbb{E}[A_S^2]\mathbb{E}[X_{Off}](k-1+\mathbb{E}[A_S]) \\ &\quad + (1-\mathbb{E}[\mathcal{U}(N)])\mathbb{E}[X_{Off}] \\ &\quad\quad ((k-1)^2 + 2(k-1)\mathbb{E}[A_\Gamma] + \mathbb{E}[A_\Gamma^2]). \end{aligned}$$

After some algebra, we are able to arrive at a relatively clean expression for the expected number of jobs in the system.

Theorem 3. *For an $M/G/1 \circ \{G, M, k\}$ queue, the expected number of jobs in the system and the expected response time for a job are given by:*

$$\begin{aligned} \mathbb{E}[N] &= \mathbb{E}[N_{M/G/1}] \\ &\quad + \alpha\frac{\gamma(k-1)+\lambda}{k\alpha\gamma+\alpha\lambda+\lambda\gamma}\left[\frac{1}{2} - \rho\right. \\ &\quad - \rho\frac{\alpha}{\alpha+\lambda}\left(\frac{\gamma(k-1)+\lambda}{\gamma}\right) - \frac{1}{2}\frac{\alpha}{\alpha+\lambda}\Gamma\Big] \\ &\quad + \rho\frac{\alpha}{\alpha+\lambda}\left(\frac{\gamma(k-1)+\lambda}{\gamma}\right) + \frac{1}{2}\frac{\alpha}{\alpha+\lambda}\Gamma \end{aligned}$$

where letting σ^2_{setup} denote the variance of the setup time distribution,

$$\Gamma = (k-1)^2 + (2k-1)\frac{\lambda}{\gamma} + \lambda^2\sigma^2_{setup}$$

and,

$$\mathbb{E}[R] = \frac{\mathbb{E}[N]}{\lambda}.$$

Again we see this recurring decomposition of the energy-aware system into its classical queue counterpart plus additional terms. We would expect to see this result for the same reasons discussed when we solved the $M/M/1 \circ \{M, M, k\}$ queue. Combining Theorem 2 and Theorem 3 now gives us the tools to optimize $M/G/1 \circ \{G, M, k\}$ systems under any metric defined by (1).

IV. Applications

In this section, we derive optimal values for the parameters under popular optimization criteria, as well as how these results can be used in other settings. We revert back to our model with exponential assumptions for simplification of calculations, however all methods used are still applicable in the general setting.

A. Weighted Sum Cost Function

One of the more popular metrics used is a weighted sum of the three system metrics, $\mathbb{E}[R] + \beta_1\mathbb{E}[E] + \beta_2\mathbb{E}[Sw]$. Often $\mathbb{E}[Sw]$ is ignored ($\beta_2 = 0$) and the weights β_1 and β_2 convert the units of the overall function to be dollars. This means of course that $\mathbb{E}[R]$ must be scaled by a unit constant of dollars/time. We take the partial derivatives first with respect to α.

$$\begin{aligned} \frac{\partial}{\partial\alpha}\mathbb{E}[R] &= \frac{\lambda(\lambda+k\gamma)}{(k\alpha\gamma+\alpha\lambda+\lambda\gamma)^2} + \frac{\gamma^2k(k-1)}{(k\alpha\gamma+\alpha\lambda+\lambda\gamma)^2} \\ \frac{\partial}{\partial\alpha}\mathbb{E}[E^N] &= (1-\rho)\lambda\gamma\frac{r_{setup}\lambda - r_{idle}(\lambda+k\gamma)}{(k\alpha\gamma+\alpha\lambda+\lambda\gamma)^2} \\ \frac{\partial}{\partial\alpha}\mathbb{E}[Sw] &= (1-\rho)\frac{\lambda^2\gamma^2}{(k\alpha\gamma+\alpha\lambda+\lambda\gamma)^2} \end{aligned}$$

As expected, α only appears in the denominators. This means that when we take the weighted sum of the derivatives, there is no value of α to make the sum evaluate to 0. In other words, the optimal value of α occurs at one of its bounds, $\alpha = 0$ or $\alpha \to \infty$ (which we knew from our previous analysis). What this yields that we did not have before is the point where the preference of α switches. From our cost function we can

see that when the following inequality holds, the optimal value is to have $\alpha \to \infty$, while it is 0 otherwise.

$$\begin{aligned}\beta_1(1-\rho)\lambda\gamma r_{idle}(k\gamma+\lambda) \leq \lambda(\lambda+k\gamma)+\gamma k(k-1) \\ +\beta_1(1-\rho)\lambda^2\gamma r_{setup} \\ +\beta_2(1-\rho)\lambda^2\gamma^2\end{aligned} \quad (12)$$

When solving for the optimal value of k, we can simplify our equations by initially having $\alpha \to \infty$ since we know that if $\alpha = 0$ an optimal k does not exist since the server never shuts off. Taking the partial derivatives of the metrics with $\alpha \to \infty$ gives us,

$$\begin{aligned}\frac{\partial}{\partial k}\mathbb{E}[R] &= \frac{\gamma}{2\lambda}\frac{k^2\gamma+2k\lambda-\lambda}{(\lambda+k\gamma)^2} \\ \frac{\partial}{\partial k}\mathbb{E}[E^N] &= -(1-\rho)\frac{\lambda\gamma r_{setup}}{(\lambda+k\gamma)^2} \\ \frac{\partial}{\partial k}\mathbb{E}[Sw] &= -(1-\rho)\frac{\lambda\gamma^2}{(\lambda+k\gamma)^2}.\end{aligned}$$

Setting the weighted sum of the above three terms equal to 0, we arrive at the following quadratic.

$$0 = \frac{\gamma^2}{2\lambda}k^2+\gamma k-(\lambda+(1-\rho)\lambda\gamma(\beta_1 r_{setup}+\beta_2\gamma)). \quad (13)$$

Solving (13) and substituting it into (12), one can determine the optimal values of the system parameters. If there exists a solution, k^*, for (13) on the constrained range of k, due to the convexity of our metrics with respect to k, one would just need to check both $\lceil k^* \rceil$ and $\lfloor k^* \rfloor$ to see which yields the best result.

B. Optimization with SLA Constraints

Here we consider a constrained optimization problem. We find that the optimal value of α is not necessarily at the bounds of its range. Imagine a server where for simplicity k is fixed at 1 and there is a service level agreement (SLA) that the expected response time for a job must be less than or equal to some constant T, where $\frac{1}{\mu-\lambda} \leq T \leq \frac{1}{\mu-\lambda}+\frac{1}{\gamma}$, and we wish to minimize the expected energy consumed by the system under the assumption that $r_{idle} < \frac{\lambda}{\lambda+\gamma}r_{setup}$. We set (10) equal to T and solve for α.

$$\alpha = \frac{\lambda\gamma^2}{\lambda+\gamma}\frac{T-\mathbb{E}[R_{M/M/1}]}{1-\gamma(T-\mathbb{E}[R_{M/M/1}])}$$

Using this value for α will minimize the expected energy used by the system. This value is optimal due to our assumption that implies $\mathbb{E}[E]$ decreases as α increases.

C. Sleep States

Modern servers usually have several different discrete sleep settings which they can be set to. While in these sleep states, the server consumes a lower amount of energy than being idle but it cannot process jobs. We define a class of policies $\mathcal{P}$, which exhibit very similar behaviour to the polices we have been considering. Policies of class $\mathcal{P}$ wait for k jobs to accumulate in the queue while in a lower energy state before beginning to turn on. Once turned on the system processes jobs until it becomes idle. If the system idles for a certain amount of time before a new job arrives, it moves to the same lower energy state that it started in, and repeats its behaviour. The key difference here is now that we have different lower energy states (the sleep states), and we allow the server to only use one of them. We show that our model can be used to find the optimal policy contained in $\mathcal{P}$.

We add the following variation to our previous model: the system now has I different sleep states it can be set to, where each of the i sleep states is denoted by ***SLEEP***$_i$. As stated before, jobs cannot be processed while the server is in state ***SLEEP***$_i$, $\forall i : 0 < i \leq I$. For each state ***SLEEP***$_i$, there is a corresponding energy cost, denoted $E_{Sleep,i}$ (along with an energy ratio with respect to E_{Busy}, $r_{Sleep,i}$), as well as a corresponding turn on rate, denoted γ_i. Typically, $\forall i : 0 < i < I . E_{Sleep,i} \leq E_{Sleep,i+1}$ and $\gamma_i \leq \gamma_{i+1}$.

Our original model can describe a system where instead of turning off after a given idling time, it instead transitions to some state ***SLEEP***$_i$. Here the steady state probabilities of $\pi^i_{0,0}$ to $\pi^i_{0,k-1}$ now correspond to the steady state probabilities of being in state ***SLEEP***$_i$ rather than ***OFF***, and γ is replaced with γ_i. To analyse this system, we must also replace each instance of γ in our equations for $\mathbb{E}[R]$, $\mathbb{E}[E^N]$, and $\mathbb{E}[Sw]$ to γ_i as well as make a slight addition to the expression for $\mathbb{E}[E^N]$, (8), to account for energy now being consumed in the sleep state.

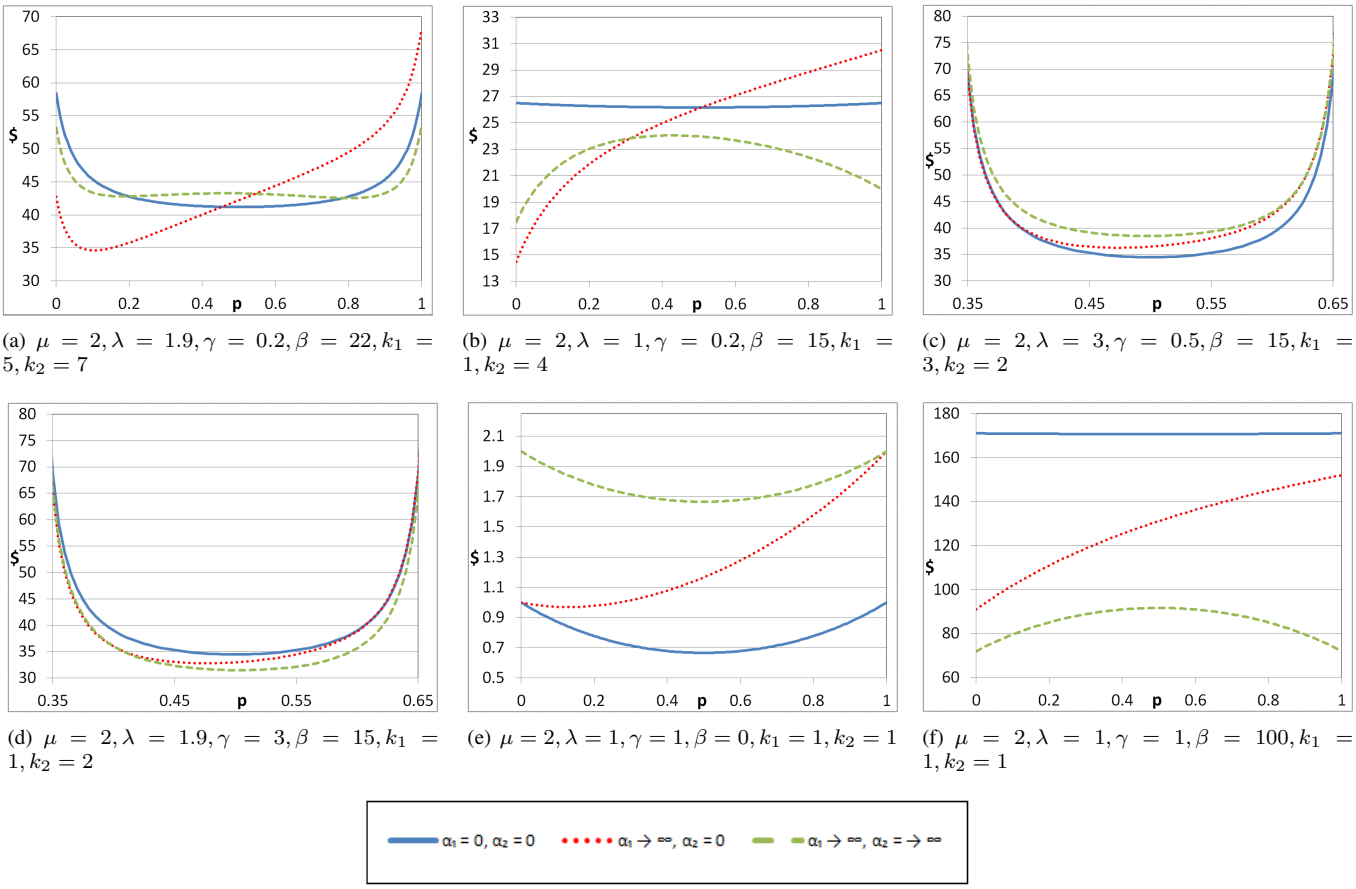
$$\mathbb{E}[E^N_{Sleep,i}] = \mathbb{E}[E^N] + (1-\rho)\frac{k\alpha\gamma_i}{k\alpha\gamma_i+\alpha\lambda+\lambda\gamma_i}r_{Sleep,i}$$

From here we can analyse the system, and obtain the optimal values of α and k. Substituting these values into our optimization metric gives us some value, denoted opt_i. Once we have these I optimal values as well as the optimal value for the server turning off, we can take the minimum of them and design our policy to always transition to the corresponding state ***OFF***, or ***SLEEP***$_i$.

Although accounting for the sleep states of the server allows us to determine improved policies than if we were to ignore them, we can no longer claim that our model can describe the optimal policy of the server, i.e. the optimal policy may not be contained in $\mathcal{P}$. This is due to the fact that the optimal policy may have the server be in some sleep state until k_1 jobs accumulate, then move to a higher sleep state where it waits for k_2 jobs to accumulate before turning on. However, when the optimal values of k are low for any individual sleep state under our analysis, we conjecture that the policy will be close to optimal.

V. Random Routing

Here we present an application of our model in a random routing setting, where we leverage our single server solutions. Imagine a system with two $M/M/1 \circ \{M, M, k\}$ queues. When a job arrives to the system, it is sent to the first queue with probability p and is sent to the second queue with probability $(1-p)$. If we wish to optimize for some metric, we now have five decision variables, α_1, α_2, k_1, k_2, and p,



(a) $\mu = 2, \lambda = 1.9, \gamma = 0.2, \beta = 22, k_1 = 5, k_2 = 7$

(b) $\mu = 2, \lambda = 1, \gamma = 0.2, \beta = 15, k_1 = 1, k_2 = 4$

(c) $\mu = 2, \lambda = 3, \gamma = 0.5, \beta = 15, k_1 = 3, k_2 = 2$

(d) $\mu = 2, \lambda = 1.9, \gamma = 3, \beta = 15, k_1 = 1, k_2 = 2$

(e) $\mu = 2, \lambda = 1, \gamma = 1, \beta = 0, k_1 = 1, k_2 = 1$

(f) $\mu = 2, \lambda = 1, \gamma = 1, \beta = 100, k_1 = 1, k_2 = 1$

Fig. 2: Random Routing: Optimization vs p

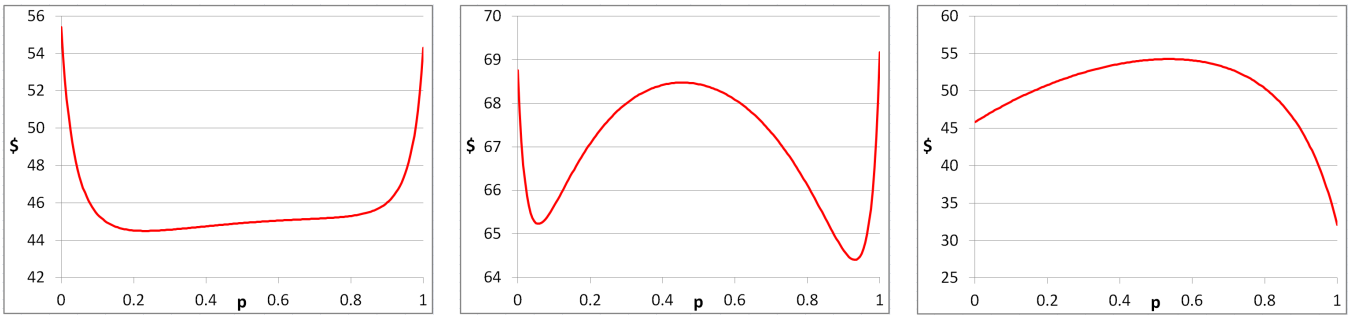
where the subscripts 1 and 2 denote the values for the first and second server, respectively. We know that the values for α_1 and α_2 will be either set to 0 or approach ∞, which breaks the problem set into three cases (due to symmetry) where we instead look to optimize against k_1, k_2 and p and then take the lowest value from among the three cases. We classify the cases as follows. The first is $\alpha_1 = \alpha_2 = 0$, the second is $\alpha \to \infty$ and $\alpha_2 = 0$, and the third is $\alpha_1 \to \infty$ and $\alpha_2 \to \infty$.

We wish to minimize $\mathbb{E}[N] + \beta\mathbb{E}[E]$. This falls within our class of cost functions, as $\mathbb{E}[N]$ can be scaled to give us $\mathbb{E}[R]$ and here it is in fact scaled by a unit constant of dollars/jobs. We know that for the first case since the servers will always be on and each server will be in *BUSY* for $\frac{p\lambda}{\mu}$ and $\frac{(1-p)\lambda}{\mu}$ proportion of time respectively, that the optimal configuration in that case is to set $p = 0.5$, i.e. balance the loads. As we will see, the other cases provide non-trivial optimal values for p.

Figure 2 shows several examples under different parameter configurations of the cost function versus p in the three different cases where the optimal k values are used, and r_{idle} and r_{setup} are both set to 0.8. In Figure 2(a), we see a medium loaded system where either server could take the full load of the arrival rate and still be stable. Here we can see that the optimal server configuration is to have a server which is always on which takes the majority of the system load (89.5%), while a server which turns off when it becomes idle takes a small portion of the system load (10.5%). This means that a lot of the time, the server that turns off will just remain off with up to four jobs waiting in the queue. This may seem unfair to the jobs which are "unlucky" enough to be put into this queue but this is an unfortunate side effect of energy concerns in this setting.

In Figure 2(b), we see a lightly loaded system and get a result that is not surprising. The optimal configuration is still one server that remains on and one that turns off. However, the server which turns on and off is completely ignored. In other words, the configuration which optimizes the random routing problem is simply an M/M/1 queue. This is somewhat expected since the load on the system is so light it is not advantageous to use the second server.

Figures 2(c) and 2(d) show the results for a heavily loaded system where both servers must be used or the system will be unstable. We can see the curves of the three cases here begin to converge to similar curvatures. In Figure 2(c), where the setup rate is relatively low ($\gamma = 0.5$), the classical load balancing approach gives us the best configuration with both servers always on and $p = 0.5$. We notice that as we increase the setup rate of the server ($\gamma = 3$), both servers being on



(a) $\mu = 2, \lambda = 1.9, \gamma = 0.1, \beta = 15, k_1 = 6, k_2 = 3$

(b) $\mu = 2, \lambda = 1.9, \gamma = 0.1, \beta = 30, k_1 = 3, k_2 = 4$

(c) $\mu = 2, \lambda = 0.5, \gamma = 0.1, \beta = 50, k_1 = 7, k_2 = 1$

Fig. 3: Random Routing: Single Case

becomes sub-optimal and the case of both servers turning on and off begins to dominate. In fact, the optimal value is $p = 0.505$ and not $p = 0.5$ as one might expect. This is as we would expect since the faster the server can turn on, the more appealing it is to shut it off.

As we see from Figure 2, simple load balancing is not sufficient to arrive at the optimal configuration as we have shown non-trivial values of p that optimize the system. Taking a more narrow look at the single case of having both servers able to turn off in Figure 3, shows a similar non-trivial result. Here the graphs also become asymmetric with respect to p, and even the optimal values of k_1 and k_2 are not equal. As in the case of having one server always on, and one server able to turn off, load balancing is not optimal. It is noted that if load balancing was used in Figure 3 (b), i.e $p = 0.5$, the result would be a disaster, as it is one of the worst configurations possible in this context. Adding energy concerns to these systems greatly impacts the complexity of the analysis as typical load balancing algorithms are no longer optimal. This also raises questions on the implications for other multi-server settings such as round robin routing or in an $M/M/c \circ \{M, M, k\}$ queue. Specifically, there is no reason why in general each server should be homogeneous with respect to the server's α and k values.

VI. Conclusion

As energy costs of servers as well as the relative energy consumed by servers increase, we must put greater emphasis on determining optimal policies. Here we gave a complete analysis of the single server systems $M/M/1 \circ \{M, M, k\}$ and $M/G/1 \circ \{G, M, k\}$, with respect to $\mathbb{E}[N]$, $\mathbb{E}[R]$, $\mathbb{E}[E]$, and $\mathbb{E}[Sw]$ as well as analysis for an $M/G/1 \circ \{G, G, k\}$ queue with respect to $\mathbb{E}[E]$ and $\mathbb{E}[Sw]$. This gave us an array of tools and equations to arrive at optimal policies for many single server energy-aware systems under general settings. We also leveraged our analysis in several other applications, such as SLA optimization, servers with sleep states, and a multi-server system with random routing. For the latter we showed that typical load balancing algorithms are not enough to arrive at an optimal configuration. Furthermore, this context gives a deeper insight into the analysis of these energy-aware multi-server system with other routing policies. In particular, heterogeneous servers may be desirable, in contrast to models where energy costs are not considered. Energy factors will always be present in these systems and it is important that we gain as much insight and understanding into these problems as possible.

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

References

[1] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. Kozuch. "Optimality Analysis of Energy-Performance Trade-off for Server Farm Management." *Performance Evaluation*, vol. 67, 2010, pp. 1155-1171.

[2] A. Gandhi and M. Harchol-Balter. "M/M/k with Exponential Setup." *CMU Technical Report CMU-CS-09-166*, 2010.

[3] A. Gandhi, M. Harchol-Balter, and I. Adan. "Server Farms with Setup Costs." *Performance Evaluation*, vol. 67, no. 11, 2010, pp. 1123-1138.

[4] A. Penttinen, E. Hyytia, and S. Aalto. "Energy-Aware Dispatching in Parallel Queues with On-Off Energy Consumption." *IEEE International Performance Computing and Communications Conference*, 2011.

[5] A. Wierman, L. L. H. Andrew, and A. Tang. "Power-Aware Speed Scaling in Processor Sharing Systems." *INFOCOM*, 2009.

[6] D. Meisner, B. T. Gold, T. F. Wenisch. "PowerNap: Eliminating Server Idle Power." *ACM Sigplan Notices*, 2009, pp. 205-216.

[7] L. A. Barroso and U. Holzle. "The Case for Energy-Proportional Computing." *Computer*, Dec 2007, pp. 33-37.

[8] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wand, and N. Gautam. "Managing Server Energy and Operational Costs in Hosting Centers." *ACM SIGMETRICS*, 2005, pp. 303-314.

[9] A. Wierman, L. L. H. Andrew, and M. Lin. "Speed Scaling: An Algorithmic Perspective." *Chapter in Handbook on Energy-Aware and Green Computing*, CRC Press, 2012, pp. 385-406.

[10] 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." *ACM SIGMETRICS*, 2013.

[11] X. Xu, and N. Tian. "The M/M/c Queue with (e, d) Setup Time." *Journal of Systems Science and Complexity 21* 2008, pp. 446-455.

[12] M. Hassan, and M. Atiquzzaman. "A Delayed Vacation Model of an M/G/1 Queue with Setup Time and Its Application to SVCC-Based ATM Networks." *IEICE TRANSACTIONS on Communications*, vol. E80-B, 1997, pp. 317-323.

[13] S. W. Fuhrmann, and R. B. Cooper. "Stochastic Decompositions in the M/G/1 Queue with Generalized Vacations." *Operations Research*, September/October 1985, vol. 33, pp. 1117-1129.

[14] J. Slegers, N Thomas, and I. Mitrani. "Dynamic Server Allocation for Power and Performance." *Performance Evaluation: Metrics, Models and Benchmarks Lecture Notes in Computer Science*, vol. 5119, 2008, pp. 247-261.