JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A power and response time management mechanism for an M/M/k queue system with setup costs

[1]Abdellah OUAMMOU, [1]Hibat eallah MOHTADI, [1,2]Adnane EL HANJRI,[1]Mohamed HANINI

[1] Laboratory of Computer, Networks, Mobility and Modeling, Faculty of Sciences and Technologies, Hassan 1st University, Settat, Morocco,

[2] Laboratory of Intelligent System and Applications (LSIA), Moroccan School of Engineering Sciences Tangier, Morocco.

Abstract: Efficiently managing server quantities in contemporary server farms to meet demand while conserving energy poses a critical challenge. Turning off idle servers saves energy, but the associated setup costs during reactivation can impact performance. This study delves into the M/M/k queuing system with setup costs (M/M/k/Setup), exploring strategic server shutdowns and their implications. The analysis addresses the pivotal question of how many servers to activate, deactivate, or maintain in a sleep state, introducing a decomposition property that equates the response time of configured and non-reserved M/M/k systems. In addition, the research suggests a power management approach for server farms, specifically focusing on an M/M/k queue system with setup costs. It analyzes system states, response time distribution, and average energy consumption. Rigorous scrutiny and simulations confirm the proposed mechanism's effectiveness in optimizing energy efficiency. This work significantly contributes to practical and theoretical aspects of server farm management, offering insights for improved resource allocation and performance in IT and cloud computing.

Index Terms **- M/M/k/setup, Mathematical analysis, Cloud computing, Power management, Resource allocation, Setup cost.**

I. INTRODUCTION

Cloud computing has become a prominent issue in the field of information technology, enabling users to access and utilize IT resources online. Various commercial offerings are available to customers, both residential and professional, facilitated by expansive data centers such as Amazon and Azure[1]. However, the energy consumption of these servers now exceeds 2% of the total electricity consumption in the United States, resulting in a staggering cost of approximately 6 billion dollars. With the continuous expansion of data centers, the energy expenditure is projected to increase further [2]. Interestingly, on average, only 20 to 30% of the server's total energy storage capacity is utilized, with the primary cause of this waste being the inactive machines within the excessively large server farms.

Efficient allocation of servers to effectively manage unpredictable demand patterns is a critical challenge in modern server farm system architecture. Achieving optimal performance without wasting energy requires a strategic approach to determine the ideal number of servers to allocate [3]. The primary objective is to address the issue of server inefficiency caused by idle machines and minimize the energy consumption associated with unnecessary server operations.

While it is essential to shut down inactive servers to conserve energy, restarting servers incurs high configuration costs in terms of setup time and energy consumption, which can negatively impact performance. Moreover, contemporary servers offer various sleep or standby states that allow the setup cost to be balanced by the power consumed while the server is in sleep mode. Research has shown that remote servers consume approximately 60% of their maximum power[4]. Hence, substantial energy savings can be achieved by powering off inactive servers when they are not needed [5], [6]. However, the act of turning servers on and off also incurs significant costs. The setup cost, which includes a time delay, is incurred when transitioning a server from the off state to the idle state. In this work, we propose an approach that involves initially activating a minimum number of servers and placing the remaining servers in a reserve state, turning them on one by one when the demand necessitates and when they are in the sleep state. The setup cost for servers in the idle state is lower than that for servers in the off state, despite the former consuming more power than the latter [7]. This study aims to address the critical challenge of optimizing server farm management for energy efficiency and cost-effectiveness. By introducing a strategy that activates servers incrementally and utilizes a reserve state, we aim to strike a balance between minimizing setup costs and conserving energy. The decision to place servers in the idle state, despite higher power consumption, is rooted in the cost-effectiveness of their lower setup costs compared to servers in the off state. Ultimately, this research contributes valuable insights to enhance resource allocation, improve performance, and reduce operational expenses in server farms, benefiting both information technology and cloud computing sectors.

This paper focuses on analysing an M/M/k queue system with setup costs, specifically the M/M/k/Setup model. Servers are categorized into two types: those in the run mode and those in sleep mode as reserves for idle periods. When the system requires additional active servers, one server is selected from the reserves in sleep mode, incurring setup costs. The setup cost manifests as a delay. Due to the high energy consumption associated with reserve servers, the number of servers available for reservation at any given time is often limited. In the proposed configuration model, only one server can be configured at a time. While previous literature has analysed M/M/k systems with exponentially distributed setup times [3], no closed-form solutions have been obtained. Thus, in this work, we introduce the concept of reserves and provide primary analytical expressions for the limiting distribution of system states, response time distribution, and average energy consumption.

The subsequent sections of this paper are organized as follows: Section 3 presents the proposed system model, while Section 4 formulates the model. Performance measures are discussed in Section 5, followed by the presentation of numerical results in Section 6. Finally, Section 7 concludes the paper. Furthermore, the paper includes appendices at the end, providing proofs for the theorems and corollaries referenced in the main text.

II. Related Work

Setup times play a vital role in the analysis of computer systems and manufacturing systems within the realm of queueing theory. In manufacturing systems, it is common for a job to wait for an idle server to "warm up" before service initiation. Similarly, in retail and hospital settings, the arrival of customers may necessitate the involvement of additional human servers, leading to setup times as servers are brought in. In the context of computer systems, setup times have gained significant attention once again due to their crucial role in dynamic capacity provisioning for data centers. In data centers, the optimization goal is to power off idle servers or reallocate them to conserve energy. Idle servers consume power at a significant percentage ($60\%\ to\ 70\%$) of their peak rate, making it wasteful to leave them on and idle [8]. However, many companies are reluctant to power off idle servers due to the high setup time required to restart them. The setup times for servers are considerably longer, typically measured in hundreds of seconds, while job service requirements are usually less than a second [9], [10]. Beyond the lengthy setup period, power is consumed at the peak rate, even when the server is non-functional. Consequently, the benefits of powering off idle servers might not be immediately evident. Numerous ideas have been proposed to minimize the frequency of server setups in data centers. One prominent area of research focuses on load prediction techniques [9], [11], [12], [13]. In scenarios with unpredictable load, policies such as delayed off have been explored, where the shutdown of an idle server is delayed for a fixed period, anticipating new arrivals [14], [15], [16]. Another avenue of research involves reducing setup times through the development of low-power sleep modes [15], [17].

Interestingly, despite the importance of setup times, their analysis remains relatively understudied. In 1964, Welch analyzed the M/G/1 queueing system with setup times [18]. However, the analysis of the M/M/k system with setup times, referred to as M/M/k/setup, has proven to be challenging, primarily due to the complexity of the underlying Markov chain. In 2010, several analytical approximations for M/M/k/Setup were proposed in [19]. These approximations perform well under low load or reduced setup times. The $M/M/\infty$/Setup system was also analyzed in [19] , revealing a product form. However, no significant progress has been made regarding the M/M/k/Setup system. Additionally, the M/M/k/setup/delayed off, where idle servers delay their shut down for a finite period, and the M/M/k/setup/sleep, where idle servers can be either turned off (high setup time, zero power) or put to sleep (lower setup time, low power), remain largely unexplored. Regarding the M/M/k/setup/delayed off system, only iterative matrix-analytic approaches have been employed [15]. Currently, no analysis exists for the M/M/k/setup/sleep system. In the following discussion, we will explore papers that have examined repeating Markov chains and proposed techniques for their solution, considering how these techniques may or may not be applicable to the M/M/k/Setup system. Matrix-analytic based approaches have been widely used for the analysis of Markov chains with repeating structures. These numerical methods involve iterative processes to determine the rate matrix, denoted as R. While they may not provide closed-form solutions or intuitive insights, they are valuable for evaluating chains under different parameters. In some cases, it is possible to explicitly express the R matrix using a combinatorial interpretation [20]. In an attempt to extend the applicability of the combinatorial interpretation, [21] explores a broader class of chains. However, the M/M/k/Setup system, with its inherent complexity arising from transition rates dependent on the number of jobs in the system, is not included in their findings. To enhance the efficiency of matrix-analytic methods, significant research efforts have been dedicated to improving iterative procedures. For example, the authors of [22] provide an explicit solution for the rate matrix involving infinite sums. Generating function-based approaches have also been utilized to solve chains with repeating structures. However, similar to matrix-analytic methods, these approaches lack intuitive insights. The process of employing generating functions involves making an educated guess about the solution form and subsequently determining the coefficients of the guess, often resulting in lengthy computations. In theory, generating function approaches hold promise for solving highly general chains [16]. The author initially attempted to apply a generating function approach to the M/M/2/setup system. However, faced with considerable complexity and a lack of intuitive understanding, the author sought a simpler and more intuitive approach for the analysis.

The M/M/k system with vacations has received considerable attention in numerous research papers [23], [24], [25], [26], [27]. Although the Markov chain for the M/M/k with vacations shares some similarities with the M/M/k/Setup system, the dynamics of these two systems differ significantly. In the M/M/k with vacations, a server takes a vacation as soon as it becomes idle and there are no jobs in the queue. On the other hand, setup times in the M/M/k/Setup system are initiated by the arrival of jobs in the queue. Most studies on vacation models impose stringent restrictions, allowing only a fixed group of servers to go on vacation simultaneously. This is in contrast to our system, where any number of servers can be set up at any given time. The model presented in [26] comes closest to our system, but the authors utilize generating functions and assume that all idle servers are on vacation, rather than having one server in setup for each job in the queue. This assumption renders the transitions in their chain independent of the number of jobs. Some prior works have considered highly restricted versions of the M/M/k/Setup system. For instance, a few papers [19], [28], [29] examine cases where at most one server can be in setup at a time. Additionally, other studies [30], [31] investigate an M/M/k system in which a fixed subset of servers can be turned on and off based on the load. The underlying Markov chains in these restricted systems are amenable to analytical tractability, leading to straightforward closed-form expressions due to

the fixed rate at which servers are turned on. In contrast, our M/M/k/Setup system is more general, allowing for the setup of any number of servers simultaneously. Consequently, this generality adds a significant level of complexity to the problem at hand.

III. Model Architecture

In our study, we utilize an M/M/k queuing system to model a server with setup time. The model assumes a Poisson arrival process with a rate of λ and exponentially distributed service time with a mean of μ. The system load, denoted by $\rho = \frac{\lambda}{\mu}$, satisfies the condition $\rho < k$ The server operates in three states: On, in setup, or reserve.

When no work is present, servers can be shut down to reduce costs, but reactivating a shut-down server incurs setup costs, including delays and power penalties. The setup process is limited to one server at a time. The ON state signifies an actively serving server, which turns off when no jobs are pending, consuming no power. In the reserve state, the server consumes less power (P_s) than in the ON state (P_{on}). Activating a server from reserve incurs setup time modelled as an exponentially distributed random variable I with a rate of $\alpha = \frac{E[I]}{1}$.

To summarize, the M/M/k queuing system captures key aspects of a server with setup time, considering arrival processes, service times, system load, and server states. Incorporating setup costs and power dynamics allows analysis for optimizing energy consumption and operational efficiency.

In the M/M/k/Setup model, jobs adhere to a First Come First Serve (FCFS) discipline but encounter added complexity due to setup costs. Jobs arriving with a server in setup wait in the queue. If no server is in setup, a job randomly selects a server from the reserve set, incurring setup time. Completed jobs move the first queue item to the server, regardless of prior setup states. This model introduces uncertainty in job allocation.

This approach provides realistic insights into queuing systems with setup costs, aiding decisions for efficiency and resource allocation. The M/M/k/Setup model reflects real-world scenarios where servers undergo setup before service, helping evaluate system performance and design strategies for activation and job assignment.

IV. Model Analysis

In this section, our emphasis is on deriving the limit probabilities of the system states for an M/M/k/Setup model (Theorem 1). Due to space limitations, we provide concise proofs here, while more detailed proofs can be found in the appendix A.

Figure 1 illustrates the Markov chain representation for the M/M/k/Setup model. The states of the Markov chain are denoted as (a, b), where a represents the number of powered-on servers, and b indicates the number of tasks in the system. The Markov chain consists of $(k + 1)$ rows, where the first row corresponds to states with all main servers powered on, the second row represents states with exactly one server in setup from the reserves, and so on. It should be noted that only one server can be in the setup state at any given time. Hence, the transition rate from state (i, j) to $(i + 1, j)$ is a constant α for all $0 \leq i < k$ and $i < j$. The Markov chain representation for the M/M/k/Setup model illustrates states as (a, b), where a is the number of powered-on servers, and b is the number of tasks in the system. There are (k+1) rows in the Markov chain. The first row corresponds to all main servers powered on, the second to one server in setup from reserves, and so forth. Notably, only one server can be in setup at a time. The transition rate from state (i, j) to $(i + 1, j)$ is a constant α for all $0 \leq i < k$ and $i < j$.

It's important to highlight that $k = m - n$, where k is the number of servers as reserves, m is the total number of servers in the system, and n is the number of main servers. This distinction provides a clear understanding of the system's configuration, reinforcing the representation's accuracy and relevance to the M/M/k/Setup model. To determine the limiting probabilities for each state in the Markov chain depicted in Figure 1, we follow a step-by-step approach. Firstly, we calculate the limit probabilities for the states in the first row, expressed in terms of $\pi_{0,0}$. Then, we proceed to find the limiting probabilities for the states in the second row, considering the first-row solution already established in terms of $\pi_{0,0}$. By continuing this process, we can obtain the limiting probabilities for all states in the Markov chain using $\pi_{0,0}$ as a reference.

By solving the Markov chain and obtaining the limit probabilities for each state, we gain valuable insights into the steady-state behaviour of the M/M/k/Setup model. These limit probabilities allow us to assess various performance measures, such as the probability of having a certain number of servers powered on or the probability of having a specific number of tasks in the system. Furthermore, the derived limit probabilities serve as the foundation for further analyses and decision-making in optimizing system performance and resource allocation.

For detailed proofs and a more comprehensive understanding, we refer readers to the appendix A where the complete derivations and explanations are provided.

Theorem 1:

The limit probabilities of an M/M/k/Setup system can be expressed as follows:

$$\pi_{i,j} = \frac{\rho^n}{(n+1)!}\frac{\pi_{0,0}\cdot\gamma^i}{i!}\beta^j \quad \text{for } 0 \leq i < k \text{ and } j > n + i - 1$$

$$\pi_{k,j} = \frac{\rho^n}{(n+1)!}\left[\frac{\pi_{0,0}\gamma^k k\mu\beta^j}{k!\cdot(k\mu-(\lambda+\alpha))} - \frac{\pi_{0,0}k^k(\lambda+\alpha)\left(\frac{\rho}{k}\right)^j}{k!\cdot(k\mu-(\lambda+\alpha))}\right] \quad \text{for } j > k - 1$$

$$\pi_{0,0} = \left[\sum_{j=0}^{n}\frac{\rho^j}{j!} + \frac{\rho^n}{(n+1)!}\cdot\frac{1}{(1-\beta)}\cdot\left[\sum_{0\le i<k}\frac{\rho^i}{i!} + \frac{\rho^k\mu}{(k-1)!\cdot(k\mu-\lambda)}\right]\right]^{-1}$$

Where

$$\alpha = \frac{1}{E[I]}, \beta = \frac{\lambda}{\lambda+\alpha} \text{ and } \gamma = \frac{\lambda+\alpha}{\mu}$$

The detailed proof of this Theorem is provided in the appendix C of this paper.

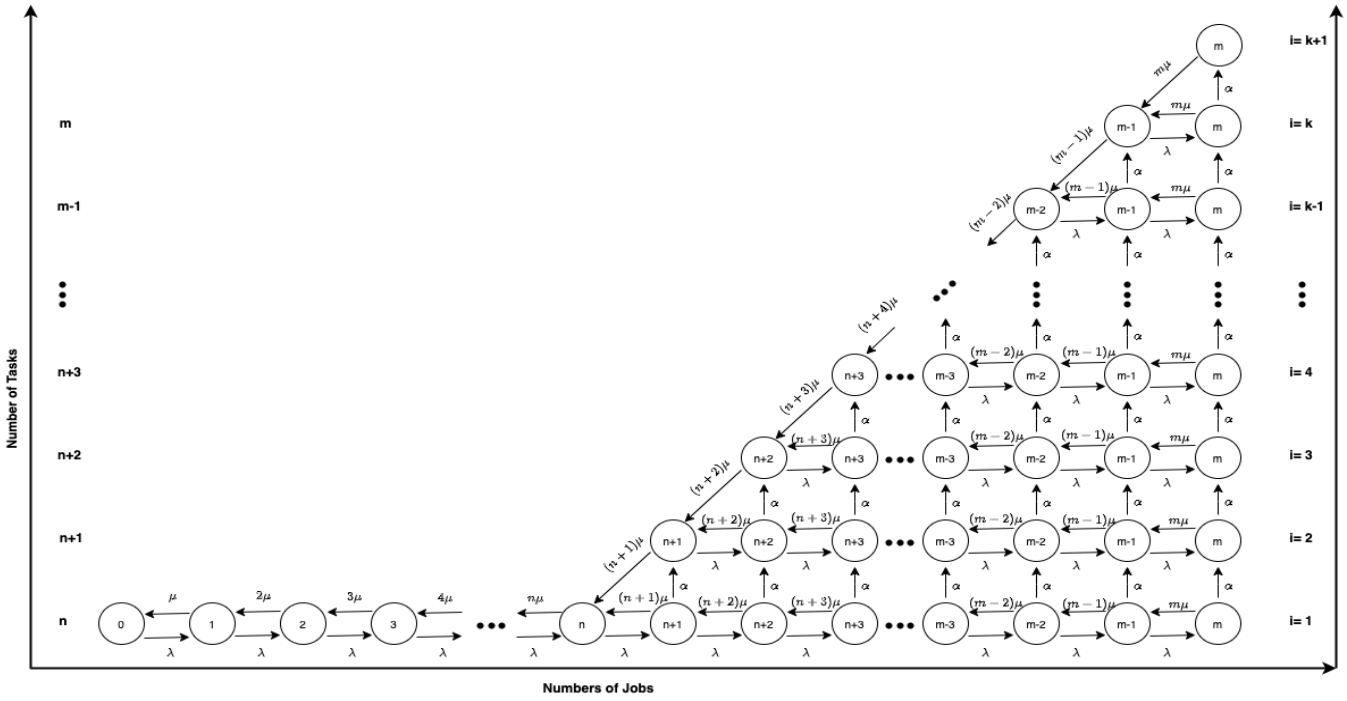


Figure 1 Markov chain for an M/M/k/Setup

V. PERFORMANCE MEASURES

In this section, we explore essential performance metrics to assess the efficiency and effectiveness of our proposed system, providing insights crucial for optimizing resource allocation and energy efficiency in server farm management.

Theorem 2:

The Mean number of jobs $E[N]$:

$$\mathbb{E}[N] = \pi_{0,0}\rho\left[\sum_{j=0}^{n-1}\frac{\rho^j}{j!} + \frac{\rho^n}{(n+1)!\cdot(1-\beta)}\sum_{j=0}^{k-1}\frac{\rho^j}{j!} + \frac{\rho^n}{(n+1)!\cdot(1-\beta)^2}\sum_{j=0}^{k}\frac{\rho^j}{j!} + \frac{\rho^{n+k-1}k\mu}{(n+1)!\,k!}\cdot\frac{(k-1)(1-\beta)(k\mu-\lambda)+k\mu-\lambda\beta}{(1-\beta)^2(k\mu-\lambda)^2}\right] \quad (1)$$

Also, the proof for this Theorem can be found in the appendix B.

Corollary *1:*

The Mean response time $E[T]$: $$\mathbb{E}[T] = \frac{\mathbb{E}[N]}{\mu} \quad (2)$$

3.1 EXPECTED NUMBER OF SERVERS IN NON-OFF STATE

In this scenario, servers can be categorized into three states: i) OFF, ii) ON, and iii) Setup. Our focus is on determining the expected number of servers that are not in the OFF state. Let's denote B_{busy} as the value assigned to each state in the $M/M/k$ Markov chain illustrated in the diagram Figure 1.

For instance, when the system is in the state $(0,0)$, the corresponding value of $B_{busy}(0,0)$ is 0. Similarly, in the state $(0,1)$, $B_{busy}(0,1)$ is equal to 1. In the case of state $(0,j)$, where $1 < j < n$, $B_{busy}(0,j)$ is calculated as j, as only one server can be in the setup mode at any given time. Generally, for state (i,j), the value of B_{busy} (i, j) can be determined as:

$$B_{busy}(i,j) = \begin{cases} j & if \quad i = 0, \quad 0 \le j \le n \\ n & if \quad i = 0, \quad j \ge n \\ i+n & if \quad i > 0, \quad j \ge n+1 \\ k & if \quad i = k, \quad j \ge n+k \end{cases}$$

Thus, the expected number of servers either ON or in Setup, is given by:

$$\mathbb{E}[B_{busy}] = \sum_{i,j} \pi_{i,j} B_{busy}(i,j)$$

Using Equations A1, A2, A3 and A4 defined in the appendix A, we get:

$$\mathbb{E}[B_{busy}] = \sum_{i,j} \pi_{i,j} B_{busy}(i,j) = \sum_{i=0}^{k-1} \sum_{j=0}^{+\infty} \pi_{i,j} B_{busy}(i,j) + \sum_{\substack{j=k+n \\ i=k}}^{+\infty} \pi_{k,j} B_{busy}(i,j)$$

$$= \sum_{i=0}^{n} \pi_{0,j} \cdot j + \sum_{j=n+1}^{+\infty} \pi_{0,j} \cdot n + \sum_{i=1}^{k-1} \sum_{j=n+1}^{+\infty} \pi_{i,j}(i+n) + \sum_{j=k+n}^{+\infty} \pi_{k,j} k$$

$$= \sum_{j=0}^{n} \pi_{0,0} \frac{\rho^j}{j!} \cdot j + \sum_{j=n+1}^{+\infty} \pi_{0,0} \frac{\rho^n}{n!} \beta^j \cdot n + \sum_{i=1}^{k-1} \sum_{j=n+1}^{+\infty} \pi_{0,0} \frac{\gamma^i}{i!} \frac{\rho^n}{(n+1)!} \beta^j (i+n) + k \sum_{j=k+n}^{+\infty} \pi_{k,j}$$

$$= \pi_{0,0}\rho \sum_{j=1}^{n} \frac{\rho^{j-1}}{(j-1)!} + \pi_{0,0} \frac{\rho^n}{(n-1)!} \sum_{j=n+1}^{+\infty} \beta^j + \pi_{0,0} \frac{\rho^n}{(n+1)!} \sum_{i=1}^{k-1} \frac{\gamma^i}{i!} (i+n) \sum_{j=n+1}^{+\infty} \beta^j$$
$$+ \frac{\rho^n}{(n+1)!} k \sum_{j=k+n}^{+\infty} \left[\frac{\pi_{0,0} \gamma^k k\mu}{k!\left(k\mu - (\lambda+\alpha)\right)} \cdot \beta^j - \frac{\pi_{0,0} k^k (\lambda+\alpha)}{k!\left(k\mu - (\lambda+\alpha)\right)} \cdot \left(\frac{\rho}{k}\right)^j \right]$$

$$= \pi_{0,0}\rho \sum_{j=0}^{n} \frac{\rho^j}{j!} + \pi_{0,0} \frac{\rho^n}{(n-1)!} \frac{\beta^{n+1}}{1-\beta} + \pi_{0,0} \frac{\rho^n}{(n+1)!} \sum_{i=1}^{k-1} \frac{\gamma^i}{i!} (i+n) \frac{\beta^{n+1}}{1-\beta}$$
$$+ \frac{\rho^n}{(n+1)!} k \left[\frac{\pi_{0,0} \gamma^k k\mu}{k!\left(k\mu - (\lambda+\alpha)\right)} \cdot \sum_{j=k+n}^{+\infty} \beta^j - \frac{\pi_{0,0} k^k (\lambda+\alpha)}{k!\left(k\mu - (\lambda+\alpha)\right)} \cdot \sum_{j=k+n}^{+\infty} \left(\frac{\rho}{k}\right)^j \right]$$

$$` = \pi_{0,0}\rho \sum_{j=0}^{n} \frac{\rho^j}{j!} + \pi_{0,0} \frac{\rho^n}{(n-1)!} \frac{\beta^{n+1}}{(1-\beta)} + \pi_{0,0} \frac{\rho^n}{(n+1)!} \cdot \frac{\beta^{n+1}}{(1-\beta)} \sum_{i=1}^{k-1} \frac{\gamma^i}{i!} (i+n)$$
$$+ \frac{\rho^n}{(n+1)!} \left[\frac{\pi_{0,0} \gamma^k k^2 \mu}{k!\left(k\mu - (\lambda+\alpha)\right)} \cdot \frac{\beta^{n+k}}{(1-\beta)} - k \frac{\pi_{0,0} k^k (\lambda+\alpha)}{k!\left(k\mu - (\lambda+\alpha)\right)} \cdot \frac{\left(\frac{\rho}{k}\right)^{n+k}}{\left(1 - \left(\frac{\rho}{k}\right)\right)} \right]$$

$$= \pi_{0,0}\rho \left[\sum_{j=0}^{n-1} \frac{\rho^j}{j!} + \frac{\rho^{n-1}}{(n-1)!} \frac{\beta^{n+1}}{(1-\beta)} + \frac{\rho^{n-1}}{(n+1)!} \cdot \frac{\beta^{n+1}}{(1-\beta)} \gamma \sum_{i=1}^{k-1} \frac{\gamma^{i-1}}{(i-1)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot \frac{\beta^{n+1}}{(1-\beta)} n \sum_{i=1}^{k-1} \frac{\gamma^i}{i!} \right]$$
$$+ \frac{\beta^n}{(n+1)!} \frac{\pi_{0,0} \rho^{k+n}}{k!\left(k\mu - (\lambda+\alpha)\right)} \left[\frac{k\mu}{\gamma^n (1-\beta)} - \frac{(\lambda+\alpha)}{\left(1 - \frac{\rho}{k}\right) k^n} \right]$$

$$\mathbb{E}[B_{busy}] = \pi_{0,0}\rho \left[\sum_{j=0}^{n-1} \frac{\rho^j}{j!} + \frac{\beta^{n+1}}{(1-\beta)} \left[\frac{\rho^{n-1}}{(n-1)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot \gamma \sum_{i=1}^{k-1} \frac{\gamma^{i-1}}{(i-1)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot n \sum_{i=1}^{k-1} \frac{\gamma^i}{i!} \right] \right]$$
$$+ \frac{\rho^n \cdot k \cdot \rho^{n+k} \pi_{0,0}}{(n+1)! \cdot k!\left(k\mu - (\lambda+\alpha)\right)} \left[\frac{k^n k\mu - \mu k^n \rho - \gamma^n \lambda - \gamma^n \alpha + \gamma^n \beta(\lambda+\alpha)}{\gamma^n (1-\beta)\left(1 - \frac{\rho}{k}\right) k^n} \right]$$

$$= \pi_{0,0}\rho \left[\sum_{j=0}^{n-1} \frac{\rho^j}{j!} + \frac{\beta^{n+1}}{(1-\beta)} \left[\frac{\rho^{n-1}}{(n-1)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot \gamma \sum_{i=1}^{k-1} \frac{\gamma^{i-1}}{(i-1)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot n \sum_{i=1}^{k-1} \frac{\gamma^i}{i!} \right] \right]$$
$$+ \frac{\beta^n \cdot k \cdot \rho^{n+k} \pi_{0,0}}{(n+1)! \cdot k!\left(k\mu - (\lambda+\alpha)\right)} \left[\frac{k^n (k\mu - \lambda) - \gamma^n \alpha}{(1-\beta)\left(1 - \frac{\rho}{k}\right) k^n} \right]$$

Since (⋆) from appendix B.

$$\mathbb{E}[B_{busy}] = \pi_{0,0}\rho \left[\frac{\exp(\rho) \cdot \Gamma(n,\rho)}{(n-1)!} + \frac{\beta^{n+1}}{(1-\beta)} \left[\frac{\rho^{n-1}}{(n-1)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot \gamma \frac{\exp(\gamma) \cdot \Gamma(k-1,\gamma)}{(k-2)!} + \frac{\rho^{n-1}}{(n+1)!} \cdot n \frac{\exp(\gamma) \cdot \Gamma(k,\gamma)}{(k-1)!} \right] \right]$$
$$+ \frac{\beta^n \cdot k \cdot \rho^{n+k} \pi_{0,0}}{(n+1)! \cdot k! \left(k\mu - (\lambda+\alpha)\right)} \left[\frac{k^n (k\mu - \lambda) - \gamma^n \alpha}{(1-\beta)\left(1 - \frac{\rho}{k}\right) k^n} \right]$$

3.2 The Mean Power Consumption and Optimization in Server Farms

Considering the significant impact of setup costs on both power usage and response time, it is crucial to account for them when optimizing the server farm's configuration.

In this section, we focus on a performance metric that combines the mean response time, denoted as $E[T]$ and the mean power consumption, denoted as $E\left[P^{M/M/k/\text{Setup}}\right]$. This metric, denoted as "Perf," is a weighted sum of these two factors, given by the equation:

$$\text{Perf} = E\left[P^{M/M/k/\text{Setup}}\right] + c \cdot E\left[T^{M/M/k/\text{Setup}}\right]$$

The weight parameter c represents the price required, in Watts, to reduce the mean response time of the server farm by one second. It has units of Watts/sec and allows us to balance the importance of power consumption and response time. A similar weighted linear combination has been used in previous literature [32], [33], [34] and to address similar optimization problems.

Let's consider the power consumption of servers in different states. When a server is in the OFF state, we assume its power consumption is 0. However, when the server is in the ON or SETUP states, we assume its power consumption is a value denoted as P_{max}. Based on this information, we can express the mean power consumption as:

$$E\left[P^{M/M/k/\text{Setup}}\right] = P_{max} \cdot E\left[B_{busy}\right]$$

Where $E\left[B_{busy}\right]$ represents the expected number of servers that are either ON or in SETUP states.

Considering both the mean power consumption and mean response time in our optimization strategy allows us to strike a balance between energy efficiency and performance in the server farm. By taking into account the setup costs and appropriately weighting the factors, we can develop strategies that optimize the configuration of the server farm, leading to improved power management and overall system performance.

In the next sections, we will explore various techniques and algorithms to analyse and optimize the M/M/k with setup costs model, enabling us to make informed decisions regarding power usage, response time, and the configuration of server farms. These approaches will help us achieve a fine-tuned balance between power efficiency and performance, ultimately enhancing the overall effectiveness and sustainability of server farm operations.

VI. NUMERICAL RESULTS

In this section, we provide detailed numerical results obtained from the proposed mechanism, aiming to illustrate and quantify the performance of the studied system across various loading conditions and system parameters. We conduct a validation process based on the presented numerical data to ensure accuracy and reliability. For our analysis, we consider an M/M/K/Setup model, where the maximum number of requests in the system is set to 18, and the number of reserves is configured as 8. These parameters are chosen to reflect a realistic scenario and allow us to assess the system's behaviour under different workloads. To evaluate the power consumption, we assign power weights to the principal server and the idle state. Specifically, we set the power weight for the principal server to 100 watts, representing the power consumption when the server is actively processing requests. On the other hand, the power weight for the idle state is set to 10 watts, indicating the power consumption when the server is not actively engaged in processing tasks. By considering these power weights, we can effectively measure and analyse the power consumption of the system at various stages, providing insights into the energy efficiency of the overall setup. This information is crucial for designing and optimizing server farms to achieve a balance between performance and power utilization. Through our numerical experiments, we gather data on factors such as response time, system utilization, and power consumption. These metrics enable us to assess the system's performance in terms of its ability to handle incoming requests, the efficiency of resource utilization, and the overall power consumption patterns. To validate our results, we compare them with theoretical models and conduct extensive simulations to ensure the accuracy and reliability of our findings. This validation process allows us to have confidence in the numerical results presented and reinforces the credibility of our proposed mechanism. The numerical results obtained from our analysis provide valuable insights into the behaviour and performance of the M/M/K/Setup system. They offer a quantitative understanding of how the system operates under different conditions, enabling us to make informed decisions regarding system design, resource allocation, and power management strategies.

In the rest of this section, we delve deeper into the numerical findings, discussing key observations, trends, and implications that emerge from the data. We analyse the performance metrics, highlight trade-offs, and explore potential optimizations to further enhance the efficiency and effectiveness of the studied system. By leveraging these numerical results, we aim to contribute to the body of knowledge in power management strategies for server farms. The insights gained can assist server farm operators and system designers in making informed decisions to achieve optimal performance, maximize resource utilization, and minimize power consumption, ultimately leading to more sustainable and efficient operations.

Figure 2 and 3 Show the system's power consumption behaviour while using the proposed mechanism, M/M/k/Setup, and compare it to the standard M/M/k system.

Table 1 : Simulation parameters and their corresponding values for Figure 2

Parameter	Description	Value
μ	Service rate	1(Jobs $/sec$)
K	Number of servers	18
n	Number of Main servers	8
a	Mean setup time	0.03(sec)

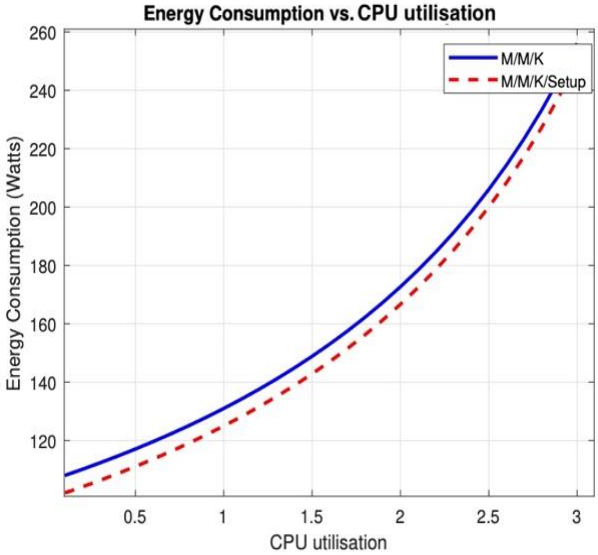


Figure 2 Energy consumption VS CPU utilization

In Figure 2, As the CPU utilisation increases, the energy consumption for both the M/M/K and M/M/K/Setup models increases. This is because higher arrival rates lead to a higher load on the system, resulting in more energy consumption to handle the increased workload. However, it is evident that the M/M/K system demonstrates the highest energy consumption out of the analysed scenarios. This indicates that when no setup mechanism or reserves are employed, the system operates at maximum energy usage. However, the introduction of reserves and implementation of the M/M/k/Setup configuration result in a noteworthy reduction in energy consumption. The utilization of reserves facilitates a more efficient allocation of resources, leading to significant power savings. As a result, the M/M/k/Setup configuration proves to be more energy-efficient when compared to the traditional M/M/k system. It is evident that the M/M/K system demonstrates the highest energy consumption out of the analysed scenarios. This indicates that when no setup mechanism or reserves are employed, the system operates at maximum energy usage. However, the introduction of reserves and implementation of the M/M/k/Setup configuration result in a noteworthy reduction in energy consumption. The utilization of reserves facilitates a more efficient allocation of resources, leading to significant power savings. As a result, the M/M/k/Setup configuration proves to be more energy-efficient when compared to the traditional M/M/k system.

Table 2 : Simulation parameters and their corresponding values for Figure 3

Parameter	Description	Value
μ	Service rate	1(jobs $/sec$)
K	Number of servers	44
n	Number of Main servers	34
a	Mean setup time	0.03(sec)

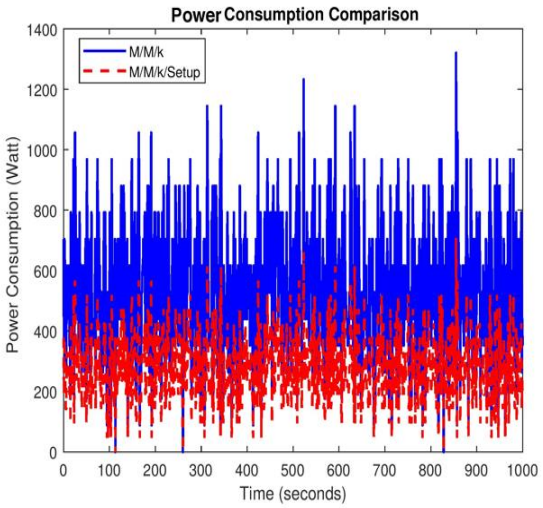


Figure 3 Power Consumption as function of time

In Figure 3, the plot illustrates the relationship between CPU utilization and power consumption for both the M/M/K/Setup and M/M/K systems, highlighting the differences in energy consumption between the two models. By comparing the two curves, we observe that the M/M/K/Setup system generally consumes less energy than the M/M/K system for the same level of CPU utilization. This is because the M/M/K/Setup system has additional servers dedicated to the setup mode, leading to less energy consumption.

Table 3 : Simulation parameters and their corresponding values for Figure 4

Parameter	Description	Value
μ	Service rate	2(jobs /sec)
K	Number of servers	*18*
n	Number of Main servers	*8*
a	Mean setup time	0.03(sec)

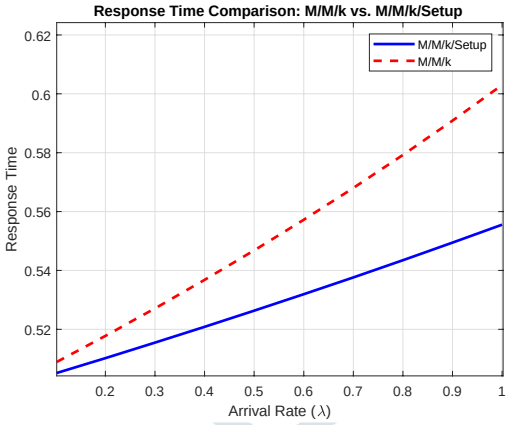


Figure 4 Response time as function of arrival rate

Figure 4 provides the response time curves for two systems M/M/k/Setup and M/M/k, with the same service rate. The M/M/k curve shows that as the arrival rate increases, the response time generally increases, indicating that higher arrival rates lead to longer response times in this system. On the other hand, the response time for the M/M/k/Setup also demonstrates an increasing trend with the arrival rate but generally exhibits lower response times compared to the M/M/k system for the same arrival rate. This suggests that the M/M/k/Setup system achieves shorter response times than the M/M/k system. Overall, the figure showcases the trade-off between setup time and the number of servers in determining the response time performance of these systems.

VII. Conclusion

In this paper, we have investigated the M/M/k queuing system with setup times, aiming to improve energy efficiency and response time in server farms. Through analytical analysis, we have derived closed-form expressions for the average response time and the limiting distribution of the number of tasks in the M/M/k/Setup system. Our proposed mechanism, which incorporates setup times, has been thoroughly analysed using mathematical modelling and evaluation of system parameters. We have also developed a model to assess the energy consumption within the suggested mechanisms. By considering both power usage and response time, we aim to strike a balance between energy efficiency and system performance. The findings of this study align with existing scientific articles that highlight the significance of optimizing power management in server farms.

In future work, we intend to delve deeper into our analysis and explore the determination of the optimal number of servers in a server farm with setup times. This optimization process will aim to minimize a weighted sum of the average power consumption and the average response time. By considering this weighted objective, we can further refine our mechanisms and achieve an even better balance between power efficiency and system performance. The insights gained from this study contribute to the growing body of research on power management strategies in server farms. By incorporating setup times and considering the interplay between power consumption and response time, we have made strides toward achieving more sustainable and efficient server farm operations. It is important to note that our study has limitations, such as the assumptions made in the modelling process. Future research should focus on refining these assumptions and conducting real-world experiments to validate the effectiveness of our proposed mechanisms. Additionally, considering dynamic workloads and varying system parameters can provide a more comprehensive understanding of the proposed mechanisms' performance in practical scenarios.

In conclusion, our analysis of the M/M/k queuing system with setup times has yielded promising results in terms of energy efficiency and response time optimization. By continuing to explore and refine these mechanisms, we can contribute to the development of effective power management strategies for server farms, leading to improved sustainability, reduced operational costs, and enhanced system performance.

VIII. REFERENCES

[1] A. Choudhary, S. Rana, and K. J. Matahai, "A critical analysis of energy efficient virtual machine placement techniques and its optimization in a cloud computing environment," *Procedia Comput Sci*, vol. 78, pp. 132–138, 2016.

[2] X. Zhang *et al.*, "Energy-aware virtual machine allocation for cloud with resource reservation," *Journal of Systems and Software*, vol. 147, pp. 147–161, 2019.

[3] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti, "Pergamum: Replacing Tape with Energy Efficient, Reliable, Disk-Based Archival Storage.," in *Fast*, 2008, pp. 1–16.

[4] L. A. Barroso, U. Hölzle, and P. Ranganathan, "The datacenter as a computer: Designing warehouse-scale machines," *Synthesis Lectures on Computer Architecture*, vol. 13, no. 3, pp. i–189, 2018.

[5] A. Ouammou, A. BenTahar, M. Hanini, and S. El Kafhali, "Modeling and analysis of quality of service and energy consumption in cloud environment," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 10, pp. 98–106, Mar. 2018.

[6] S. El Kafhali and K. Salah, "Modeling and Analysis of Performance and Energy Consumption in Cloud Data Centers," *Arab J Sci Eng*, vol. 43, no. 12, pp. 7789–7802, Dec. 2018.

[7] A. Gandhi, M. Harchol-Balter, and I. Adan, "Decomposition results for an M/M/k with staggered setup," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 2, pp. 48–50, 2010.

[8] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer (Long Beach Calif)*, vol. 40, no. 12, pp. 33–37, 2007.

[9] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. H. Katz, "Napsac: Design and implementation of a power-proportional web cluster," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*, 2010, pp. 15–22.

[10] G. DeCandia *et al.*, "Dynamo: Amazon's highly available key-value store," *ACM SIGOPS operating systems review*, vol. 41, no. 6, pp. 205–220, 2007.

[11] W. Qin and Q. Wang, "Modeling and control design for performance management of web servers via an LPV approach," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 2, pp. 259–275, 2007.

[12] M. Castellanos, F. Casati, M.-C. Shan, and U. Dayal, "ibom: A platform for intelligent business operation management," in *21st International Conference on Data Engineering (ICDE'05)*, 2005, pp. 1084–1095.

[13] T. Horvath and K. Skadron, "Multi-mode energy management for multi-tier server clusters," in *2008 international conference on parallel architectures and compilation techniques (pact)*, 2008, pp. 270–279.

[14] J. Kim and T. S. Rosing, "Power-Aware Resource Management Techniques for Low-Power Embedded Systems.," in *of Design, Automation and Test in Europe (DATE'02)*, 2002, pp. 788–794.

[15] A. Gandhi and M. Harchol-Balter, "How data center size impacts the effectiveness of dynamic power management," in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2011, pp. 1164–1169.

[16] A. Ezzidani, A. Ouammou, M. Hanini, and A. Ben Tahar, "A SMDP Approach to Evaluate the Performance of a Vehicular Cloud Computing System with Prioritize Requests," *Mathematical Modelling of Engineering Problems*, vol. 8, no. 6, pp. 928–936, Dec. 2021, doi: 10.18280/mmep.080612.

[17] A. Ouammou, A. Zaaloul, M. Hanini, and A. Bentahar, "Modeling Decision Making to Control the Allocation of Virtual Machines in a Cloud Computing System with Reserve Machines."

[18] P. D. Welch, "On a generalized M/G/1 queuing process in which the first customer of each busy period receives exceptional service," *Oper Res*, vol. 12, no. 5, pp. 736–752, 1964.

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

[20] A. Ouammou, M. Hanini, A. Ben Tahar, and S. El Kafhali, "Analysis of a M/M/k system with exponential setup times and reserves servers," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Oct. 2019. doi: 10.1145/3372938.3372996.

[21] J. S. H. Van Leeuwaarden and E. M. M. Winands, "Quasi-birth-and-death processes with an explicit rate matrix," *Stoch Model*, vol. 22, no. 1, pp. 77–98, 2006.

[22] B. Van Houdt and J. S. H. van Leeuwaarden, "Triangular M/G/1-type and tree-like quasi-birth-death Markov chains," *INFORMS J Comput*, vol. 23, no. 1, pp. 165–171, 2011.

[23] Z. G. Zhang and N. Tian, "Analysis on queueing systems with synchronous vacations of partial servers," *Performance Evaluation*, vol. 52, no. 4, pp. 269–282, 2003.

[24] X. Xu and N. Tian, "The M/M/c Queue with (e, d) Setup Time," *J Syst Sci Complex*, vol. 21, no. 3, pp. 446–455, 2008.
[25] N. Tian, Q.-L. Li, and J. Gao, "Conditional stochastic decompositions in the M/M/c queue with server vacations," *Stoch Model*, vol. 15, no. 2, pp. 367–377, 1999.
[26] Y. Levy and U. Yechiali, "An M/M/s queue with servers' vacations," *INFOR: Information Systems and Operational Research*, vol. 14, no. 2, pp. 153–163, 1976.
[27] A. Ouammou, M. Hanini, A. Ben Tahar, and S. El Kafhali, "A dynamic programming approach to manage virtual machines allocation in cloud computing," 2018. [Online]. Available: www.sciencepubco.com/index.php/IJET
[28] I. J. B. F. Adan and J. der Wal, "Combining make to order and make to stock," *Operations-Research-Spektrum*, vol. 20, no. 2, pp. 73–81, 1998.
[29] J. R. Artalejo, A. Economou, and M. J. Lopez-Herrero, "Analysis of a multiserver queue with setup times," *Queueing Syst*, vol. 51, no. 1, pp. 53–76, 2005.
[30] I. Mitrani, "Managing performance and power consumption in a server farm," *Ann Oper Res*, vol. 202, no. 1, pp. 121–134, 2013.
[31] 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 *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, 2013, pp. 153–166.
[32] S. Albers and H. Fujiwara, "Energy-efficient algorithms for flow time minimization," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, pp. 49–es, 2007.
[33] N. Bansal, H.-L. Chan, and K. Pruhs, "Speed scaling with an arbitrary power function," *ACM Transactions on Algorithms (TALG)*, vol. 9, no. 2, pp. 1–14, 2013.
[34] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *IEEE INFOCOM 2009*, 2009, pp. 2007–2015.