Analysis of Fork-Join Scheduling on Heterogeneous Parallel Servers

Moonmoon Mohanty, *Student Member, IEEE*, Gaurav Gautam, Vaneet Aggarwal, *Senior Member, IEEE*, and Parimal Parag, *Senior Member, IEEE*

***Abstract*— This paper investigates the (k,k) fork-join scheduling scheme on a system of n parallel servers comprising both slow and fast servers. Tasks arriving in the system are divided into k sub-tasks and assigned to a random set of k servers, where each task can be assigned independently to a distinct slow or fast server with selection probability p_s or $1-p_s$, respectively. Our analysis demonstrates that the joint distribution of the stationary workload across any set of k queues becomes asymptotically independent as the number of servers n grows, with k scaling as $o(n^{\frac{1}{4}})$. Under asymptotic independence, the limiting mean task completion time can be expressed as an integral. However, it is analytically challenging to compute the optimal selection probability p_s^* that minimizes this integral. To address this, we provide an upper bound on the limiting mean task completion time and identify the selection probability $\hat{p}_s$ that minimizes this bound. We validate that this selection probability $\hat{p}_s$ yields a near-optimal performance through numerical experiments.**

***Index Terms*— Heterogeneous servers, fork-join scheduling, asymptotic independence, completion time.**

I. Introduction

In recent years, there has been a significant shift towards horizontal scaling of resources in distributed computing, driven by the need for improved performance and scalability. In distributed computing systems, tasks are typically divided into smaller sub-tasks and distributed across multiple servers to leverage parallel processing capabilities. However, the overall task completion time is inherently limited by the slowest server in the system. This limitation becomes particularly challenging in practical scenarios where servers exhibit heterogeneity, with some servers being faster and others slower in terms of processing power. Treating all servers equally in such heterogeneous environments can lead to an imbalanced utilization of resources, with some servers becoming congested while others remain underutilized. Consequently, the mean task completion time increases, resulting in potential revenue loss for the service provider and a degradation of overall system performance. Addressing this issue and optimizing task completion time in heterogeneous distributed computing environments is paramount to maximize efficiency and resource utilization.

In this paper, we focus on optimizing task completion time in distributed computing systems comprising two distinct classes of servers: slow servers and fast servers. When a task arrives, it is divided into k sub-tasks and assigned to a set of k out of n servers. This choice of k servers is referred to as scheduling. Completing all k sub-tasks signifies the departure of the task from the system. Such a system, where a task is divided (forked) into k sub-tasks and all k completed sub-tasks are aggregated (joined) to complete the task, is called a (k,k) fork-join system.

The (k,k) fork-join system is a critical building block in the job processing workflow of many data center services including web search [2] and big data analytics [3], which constitutes a significant part of job processing time and hardware cost, e.g., more than two-thirds of the total processing time and 90 percent hardware cost for a Web search engine [4], [5], [6], [7]. For example, in large-scale data processing frameworks like MapReduce, jobs are split into multiple tasks during the map phase and assigned to different servers. The reduce phase follows when these tasks are completed, imposing synchronization constraints on task finishing times. Each arriving job is divided into k map tasks, simultaneously sent to k servers. Each task requires a random service time, reflecting varying execution times on different servers during the map phase. A job exits the fork-join system only when all its tasks are served, ensuring that the reduce phase commences after all map tasks are completed. Further, fork-join systems have applications in distributed erasure-coded storage, where the content can be requested from k of the servers [8], [9], [10], [11], [12], [13].

Our primary objective is to identify a scheduling policy that minimizes the mean completion time of incoming tasks.

Manuscript received 12 July 2023; revised 19 January 2024, 18 April 2024, and 16 July 2024; accepted 16 July 2024; approved by IEEE/ACM Transactions on Networking Editor B. Ji. Date of publication 29 July 2024; date of current version 19 December 2024. The work of Vaneet Aggarwal was supported in part by Cisco Systems Inc., and Office of Naval Research under Award N00014-23-1-2532. The work of Parimal Parag was supported in part by the Qualcomm Inc., under Qualcomm University Relations 6G India; in part by IBM Research Lab India under IBM-IISc Hybrid Cloud Lab (IIHCL) open research collaboration; in part by the Science and Engineering Research Board (SERB) under Grant CRG/2023/008854; in part by U.K.-India Education and Research Initiative (UKIERI) under Grant SPARC/2024/3927; and in part by the Centre for Networked Intelligence [a Cisco Corporate Social Responsibility (CSR) Initiative], IISc. An earlier version of this work was presented in part at the poster session of ACM SIGMETRICS 2022 [DOI: 10.1145/3595244.3595248]. *(Corresponding author: Parimal Parag.)*

Moonmoon Mohanty and Parimal Parag are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru, Karnataka 560012, India (e-mail: moonmoonm@iisc.ac.in; parimal@iisc.ac.in).

Gaurav Gautam was with the Centre for Networked Intelligence, Indian Institute of Science, Bengaluru, Karnataka 560012, India. He is now with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: gauta044@umn.edu).

Vaneet Aggarwal is with the School of Industrial Engineering, the School of Electrical and Computer Engineering, and the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA (e-mail: vaneet@purdue.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109//TNET.2024.3432183, provided by the authors.

Digital Object Identifier 10.1109/TNET.2024.3432183

However, due to the heterogeneous nature of the servers, determining the optimal set of k servers for each incoming task becomes challenging. Achieving optimal mean task completion time performance necessitates considering the following key parameters: the arrival rate of tasks, the number of sub-tasks k, the number of slow and fast servers, and the absolute speeds of the servers. We introduce a novel probabilistic policy for task scheduling, which involves assigning sub-tasks to either slow or fast servers based on a selection probability p_s. Specifically, a sub-task is sent to a slow server with probability p_s and a fast server with probability $1-p_s$. Within each class, the selection is made uniformly at random without replacement. By utilizing this proposed policy, we aim to determine the optimal selection probability p_s that minimizes the mean task completion time. In essence, finding the optimal policy reduces to identifying the selection probability that yields the most efficient distribution of sub-tasks among the servers, ultimately minimizing the overall mean task completion time.

A. Related Work

Numerous load balancing strategies have been proposed to minimize the mean task completion time in distributed computing systems. The join shortest queue (JSQ) policy [14], join smallest work (JSW) policy [15], [16], [17], and water filling policy [18] are among the commonly studied approaches.

We note that classical load balancing policies such as JSQ/JSW, for homogeneous parallel server systems, require queue/workload information from all queues at all arrival instants. The information overhead in JSQ/JSW can be reduced by "power-of-d" variants [17], [19], [20], [21], [22], [23] of these policies, where only d queues are queried.

These variants involve sampling a random subset of d servers and assigning the job to a server based only on the state of the queried servers, e.g., the server with the shortest queue length or workload. Other efficient dispatching policies for parallel server systems include the size interval task assignment policy [24], Redundant-to-Idle queue [25], and load balancing with timed replicas [26]. Power-of-d variants without subdivision of tasks are akin to $(d, 1)$ fork-join queue [17], [19], [20], and with subdivision of tasks they are akin to (d, k) fork-join queues [9], [11], [13]. However, it is important to note that these policies are primarily designed for homogeneous server systems, and a direct adaptation of these strategies to a heterogeneous system may not yield optimal performance. Please see a detailed discussion in Appendix C on the classical load balancing policies adapted to our setting.

In [27], a comprehensive comparison of various load balancing algorithms designed explicitly for heterogeneous systems is presented. More recent studies have proposed load balancing strategies tailored for heterogeneous parallel server environments where join the shortest queue type strategy is studied in [28] and "power-of-d" type strategies are studied [29], [30], both without task subdivision. Further, the load balancing problem of selecting a single server has been studied in [31]. However, it is worth noting that all these studies focus on load balancing in the context of heterogeneous servers without explicitly considering the subdivision of tasks into multiple sub-tasks, which is the primary focus of our research.

Analysis of power-of-d type strategies involves showing the statistical independence of marginal stationary workload distribution of a finite set of queues in the limit of a large number of queues. This is referred to as *asymptotic independence*, and has been shown to hold under various conditions in [18], [19], [20], [21], [22], and [23] in the homogeneous server settings. We note that establishing asymptotic independence for the setting of heterogeneous queues requires non-trivial adaptation of the existing proof techniques.

B. Our Contributions

The key contributions of this paper can be summarized as follows.

1) We demonstrate the asymptotic independence of the stationary workload distribution in a heterogeneous server system with two classes of heterogeneity. This result is achieved by implementing a probabilistic policy and considering a general service distribution for the two server classes, along with Poisson arrivals. Specifically, we establish asymptotic independence for k out of n queues, as long as $k = o(n^{\frac{1}{4}})$.
2) Leveraging the asymptotic independence of the stationary workload distribution, we analytically calculate the limiting mean task completion time for systems with an arbitrarily large number of servers.
3) The analytical determination of the optimal selection probability p_s^*, which minimizes the limiting mean task completion time, poses significant computational challenges. Consequently, we derive an upper bound on the limiting mean task completion time and identify the selection probability $\hat{p}_s$ that minimizes this bound. Although this obtained selection probability approximates the optimal selection probability, we empirically demonstrate its accuracy through numerical studies.
4) We adapt classical load balancing policies such as JSQ/JSW and their power-of-d variants to the setting of heterogeneous servers with subdivision of tasks in Appendix C, and compare their performance with the proposed policy. We note that modified JSQ/JSW has a large sampling overhead compared to their power-of-d variants, whereas the proposed policy requires no sampling of the queues. We observe that the modified JSQ/JSW outperforms the proposed policy. However, the proposed policy outperforms power-of-d variants, even when the number of queried servers d is slightly larger than the number of sub-tasks k.

Notation: We denote the set of first k positive integers by $[k] \triangleq \{1, \dots, k\}$, the set of first $k+1$ non-negative integers by $[k]_0 \triangleq \{0, \dots, k\}$, the set of all positive integers by $\mathbb{N}$, the set of all non-negative integers by $\mathbb{Z}_+$, the set of all non-negative reals by $\mathbb{R}_+$, and the set of all vectors of length k taking values in a set A by A^k. The set of all probability measures on a countable set $\mathcal{X}$ is defined by $\mathcal{M}(\mathcal{X}) \triangleq \left\{\nu \in [0,1]^{\mathcal{X}} : \sum_{x\in\mathcal{X}} \nu_x = 1\right\}$.

II. System Model

We consider a system $\mathcal{S}$ of n heterogeneous servers with two types of heterogeneity. The sets of slow and fast servers are denoted by $E_s \subseteq [n]$ and $E_f = [n] \setminus E_s$, respectively. We denote the number of slow and fast servers by $n_s \triangleq |E_s|$ and $n_f \triangleq |E_f| = n - n_s$, respectively. The fraction of slow and fast servers are denoted by $f_s \triangleq \frac{n_s}{n}$ and $\bar{f}_s \triangleq 1 - f_s = \frac{n_f}{n}$ respectively.

A. Task Arrival and Completion

Each arriving task is subdivided into k sub-tasks and dispatched to k distinct servers selected out of n. The task is assumed to be completed when all k sub-tasks are completed, and it leaves the system. We assume that each sub-task is served in a first-come-first-served (FCFS) manner at each server. For this system, we assume a Poisson arrival of tasks with homogeneous rate $\Lambda \triangleq \frac{n\lambda}{k}$. We assume that $k < \frac{n_s \wedge n_f}{2}$.

B. Sub-Task Service Time

The sub-task service time for task i at server j is denoted by a random variable $X_{i,j}$. We assume that $(X_{i,j} : i \in \mathbb{N}, j \in [n])$ is independent across servers $[n]$ and across tasks $i \in \mathbb{N}$. The sub-task service time distribution at server j is denoted by $G_{X_j} : \mathbb{R}_+ \to [0,1]$. We assume this distribution is identical for servers within each class and has bounded first and second moments.

Definition 1: The sub-task service time distribution at slow and fast servers is denoted by G_s and G_f, respectively. That is,

$$G_{X_j} = G_s \mathbb{1}_{\{j \in E_s\}} + G_f \mathbb{1}_{\{j \in E_f\}}.$$

The service rates of slow and fast servers are denoted by μ_s and μ_f, respectively, where $\mu_s < \mu_f$. That is,

$$\mathbb{E}X_{i,j} = \frac{1}{\mu_s} \mathbb{1}_{\{j \in E_s\}} + \frac{1}{\mu_f} \mathbb{1}_{\{j \in E_f\}}.$$

The second moments of service distributions for slow and fast servers are denoted by $g_{s,2}$ and $g_{f,2}$, respectively. That is,

$$\mathbb{E}X_{i,j}^2 = g_{s,2} \mathbb{1}_{\{j \in E_s\}} + g_{f,2} \mathbb{1}_{\{j \in E_f\}}.$$

C. Server Selection for Sub-Task Completion

We consider a probabilistic selection of k servers out of n. Servers are selected sequentially to be either slow or fast with probabilities $(p_s, 1 - p_s)$ respectively. Once the server is selected to be slow or fast, it is chosen to be one of the slow or fast servers uniformly at random without replacement from the respective pool of servers.

Definition 2: For task i, let I^i be the k-set of probabilistically selected servers, then we denote the random set of selected slow and fast servers by $I_s^i \triangleq I^i \cap E_s$ and $I_f^i \triangleq I^i \cap E_f$ respectively and denote the random number of slow and fast servers as $K_s^i \triangleq \left|I_s^i\right|$ and $K_f^i \triangleq k - K_s^i$ respectively.

Denoting $p_f \triangleq 1 - p_s$, we can write the probability of selecting k_s slow servers for task i, as

$$q(k_s) \triangleq P\left\{K_s^i = k_s\right\} = \binom{k}{k_s} p_s^{k_s} p_f^{k - k_s}. \tag{1}$$

D. Sub-Task Arrival Rate

We can compute the probability that a slow server $j \in E_s$ is selected by the dispatcher for an incoming task, as

$$\sum_{k_s=1}^{k} q(k_s) \frac{\binom{n_s - 1}{k_s - 1}}{\binom{n_s}{k_s}} = \frac{1}{n_s} \sum_{k_s=1}^{k} k_s q(k_s) = \frac{kp_s}{nf_s}.$$

This probability is independent of the incoming task, and hence the arrival at each slow server is a thinned Poisson process with an arrival rate

$$\lambda_s \triangleq \frac{\lambda n}{k}\Big(\frac{kp_s}{nf_s}\Big) = \frac{\lambda p_s}{f_s}. \tag{2}$$

Analogously, we can compute the probability that a server $j \in E_f$ is selected by the dispatcher for an incoming task as $\frac{kp_f}{n\bar{f}_s}$ independent of the task. Consequently, the arrival process at each fast server is a thinned Poisson process with arrival rate

$$\lambda_f \triangleq \frac{\lambda p_f}{\bar{f}_s}. \tag{3}$$

III. Performance Metrics

We denote the marginal workload at server j seen by ith incoming task by $W_{i,j}$, and its limiting distribution by $F_{W_j} : \mathbb{R}_+ \to [0,1]$ such that

$$F_{W_j}(x) \triangleq \lim_{i \to \infty} P\left\{W_{i,j} \leqslant x\right\}. \tag{4}$$

Due to symmetry in the system, the marginal workload distribution is identical at all slow servers and all fast servers. The limiting distribution of the marginal workload at a slow and a fast server is denoted by F_s and F_f, respectively. That is,

$$F_{W_j}(x) = F_s(x) \mathbb{1}_{\{j \in E_s\}} + F_f(x) \mathbb{1}_{\{j \in E_f\}}. \tag{5}$$

If one of the k sub-tasks for the ith task is dispatched to a server $j \in I^i$, then the sub-task completion time at this server is denoted by $T_{i,j} \triangleq W_{i,j} + X_{i,j}$. Since the sub-task service times are *i.i.d.* at each server, $W_{i,j}$ and $X_{i,j}$ are independent and for any $x \in \mathbb{R}_+$

$$\begin{aligned} F_{T_{i,j}}(x) &\triangleq P\left\{T_{i,j} \leqslant x\right\} \\ &= \int_{y \leqslant x} P\left\{W_{i,j} \leqslant x - y\right\} dG_{X_j}(y). \end{aligned}$$

We denote the limiting distribution of sub-task completion time at any server j as $L_{T_j} : \mathbb{R}_+ \to [0,1]$, which can be written for any $x \in \mathbb{R}_+$, as

$$L_{T_j}(x) \triangleq \lim_{i \to \infty} P\left\{T_{i,j} \leqslant x\right\} = \int_{y \in \mathbb{R}_+} F_{W_j}(x - y) dG_{X_j}(y).$$

The above equality follows from the dominated convergence theorem since the integrand is positive and bounded by unity. It follows that the limiting distribution of sub-task completion times are identical for slow and fast servers, and we denote them by L_s and L_f, respectively. That is,

$$L_{T_j}(x) = L_s(x) \mathbb{1}_{\{j \in E_s\}} + L_f(x) \mathbb{1}_{\{j \in E_f\}}. \tag{6}$$

The completion time for task i is denoted by T_i, which is the maximum of the sub-task completion times at the selected I^i servers and written as

$$T_i \triangleq \max_{j \in I^i} T_{i,j}.$$

The equilibrium distribution of task completion times for n server system is denoted by $H_n : \mathbb{R}_+ \to [0,1]$, and defined for all $x \in \mathbb{R}_+$ as

$$H_n(x) \triangleq \lim_{i\to\infty} P\left\{T_i \leqslant x\right\}.$$

When the number of servers increases, the asymptotic equilibrium distribution of task completion times for n server system is denoted by $H_n : \mathbb{R}_+ \to [0,1]$, and defined for all $x \in \mathbb{R}_+$ as

$$H(x) \triangleq \lim_{n\to\infty} H_n(x).$$

Remark 1: Consider a system of n homogeneous and independent servers with *i.i.d.* service times having the following identical bimodal distribution for each server $j \in [n]$

$$F_{X_{i,j}} \triangleq G_s \Xi_{i,j} + G_f(1-\Xi_{i,j}),$$

where $\Xi_{i,j}$ indicates slow service for the sub-task corresponding to task i at server j, and $\Xi : \Omega \to \{0,1\}^{\mathbb{N}\times[[n]]}$ is an *i.i.d.* Bernoulli random sequence with $\mathbb{E}\Xi_{i,j} = p_s$. The asymptotic independence for this system follows in a straightforward manner from [18] and [23]. However, we note that this is a different system than the one under consideration. For example, we compare this system to the one we are considering in our system model with the following coupling. Arrival instants in both systems are identical, and the k sub-tasks are sent to the same set of servers. We notice that the slow servers remain slow for all sub-tasks in our system. However, each server can be slow or fast in this system for different sub-tasks. Consequently, the marginal distribution at each server in this system is identical. Contrastingly, the marginal distribution at each server is identical only within a class for our system and is completely different between the two classes. Even though the proposed system model is a bit more difficult to analyze, it is a better fit for practical systems.

IV. Asymptotic Independence

We will consider the system of servers where the number of sub-tasks k scales with n. However, for ease of exposition, we will not explicitly mention dependence on the number of servers n. Apart from the system under consideration $\mathcal{S}$, we consider two related systems $\tilde{\mathcal{S}}$ and $\hat{\mathcal{S}}$. We assume all three systems start empty at time 0 and focus on the joint distribution of queues at the set of first k servers in all three systems. The set of slow and fast servers in the first k servers are denoted by $I_s \triangleq [k] \cap E_s$ and $I_f \triangleq [k] \cap E_f$ respectively, such that $I_s \cup I_f = [k]$. The number of slow and fast servers in the first k servers is denoted by $i_s \triangleq |I_s|$ and $i_f \triangleq |I_f| = k - i_s$ respectively.

Definition 3 (Independent system): System $\hat{\mathcal{S}}$ consists of n independent $M/G/1$ queues partitioned into two disjoint sets of slow and fast servers denoted by E_s and E_f, respectively. Each server gets an independent Poisson arrival with rates λ_s and λ_f for slow and fast servers respectively, where the arrival rates λ_s and λ_f are defined in (2) and (3) respectively. We denote the marginal workload at server j in the system $\hat{\mathcal{S}}$ at time t by $\hat{W}_j(t)$, and as seen by ith incoming task by $\hat{W}_{i,j}$.

Definition 4 (Coupled system): Recall that I^i is the set of servers where sub-tasks are dispatched for each arrival i in the system $\mathcal{S}$. We couple systems $\tilde{\mathcal{S}}$ and $\mathcal{S}$ in the following way for each arrival i. The sub-tasks are dispatched to the set of servers $I^i \cap ([n] \setminus [k])$ in $\tilde{\mathcal{S}}$ with sub-task service times identical to the corresponding sub-tasks in $\mathcal{S}$. The sub-tasks are dispatched to the set of servers $I^i \cap [k]$ in the first k servers of $\mathcal{S}$. If $I^i \cap [k] = \emptyset$, then there are no sub-tasks dispatched to the first k servers in both $\mathcal{S}$ and $\tilde{\mathcal{S}}$. If $I^i \cap [k] \neq \emptyset$, we pick exactly one server for sub-task dispatch in the first k servers of $\tilde{\mathcal{S}}$, and the rest of the sub-tasks are dropped. We define the number of selected slow and fast servers as $J_s^i \triangleq \left|I^i \cap E_s \cap [k]\right|$ and $J_f^i \triangleq \left|I^i \cap E_f \cap [k]\right|$ respectively, for sub-task dispatch to the first k servers in $\mathcal{S}$. In the following three cases, we describe the selection criterion for single sub-task dispatch among the first k servers of $\tilde{\mathcal{S}}$.

Case 1: $J_s^i + J_f^i = 1$. The corresponding server is selected.

Case 2: $J_s^i J_f^i = 0$ and $J_s^i + J_f^i \geqslant 2$. If $J_f^i = 0$, then a slow server is selected uniformly at random from $I^i \cap E_s \cap [k]$. If $J_s^i = 0$, then a fast server is selected uniformly at random from $I^i \cap E_f \cap [k]$.

Case 3: $J_s^i J_f^i \geqslant 1$ and $J_s^i + J_f^i \geqslant 2$. A slow or a fast server is randomly selected with probability p_s and p_f, respectively. If a slow server is chosen for selection, then a server is selected uniformly at random from $I^i \cap E_s \cap [k]$. If a fast server is chosen for selection, then a server is selected uniformly at random from $I^i \cap E_f \cap [k]$.

We dispatch the corresponding sub-task to the selected server in $\tilde{\mathcal{S}}$, with sub-task service time identical to the corresponding sub-task in $\mathcal{S}$. We drop the remaining $J_s^i + J_f^i - 1$ sub-tasks in $\tilde{\mathcal{S}}$. We denote the marginal workload at server j in the system $\tilde{\mathcal{S}}$ at time t by $\tilde{W}_j(t)$, and as seen by ith incoming task by $\tilde{W}_{i,j}$.

Lemma 1: Consider the system $\mathcal{S}$, where the first k servers have i_s slow and $i_f = k - i_s$ fast servers. Any arrival i selects K_s^i out of n_s slow servers and $K_f^i = k - K_s^i$ out of n_f slow servers, for scheduling k sub-tasks on these servers. Further, this arrival selects J_s^i out of i_s slow and J_f^i out of i_f fast servers, among the first k servers. We can write this joint probability as

$$P\left\{(J_s^i, J_f^i) = (j_s, j_f)\right\} = \sum_{k_s=0}^{k} q(k_s) r_s(k_s, j_s) r_f(k - k_s, j_f),$$

where we define the selection probability of j_s out of i_s slow servers given k_s out of n_s slow servers were selected, as

$$r_s(k_s, j_s) \triangleq P(J_s^i = j_s \mid K_s^i = k_s) = \frac{\binom{i_s}{j_s}\binom{n_s - i_s}{k_s - j_s}}{\binom{n_s}{k_s}}, \tag{7}$$

and the selection probability of j_f out of i_f fast servers given k_f out of n_f slow servers were selected, as

$$r_f(k_f, j_f) \triangleq P(J_f^i = j_f \mid K_f^i = k_f) = \frac{\binom{i_f}{j_f}\binom{n_f - i_f}{k_f - j_f}}{\binom{n_f}{k_f}}. \tag{8}$$

Proof: Please refer to Appendix A-A. ∎

Remark 2: We will frequently use the following identity in the subsequent results given by

$$\binom{n}{k} = \sum_{j=0}^{i\wedge k} \binom{i}{j}\binom{n-i}{k-j}. \tag{9}$$

Recall that $\binom{n}{k}$ is the coefficient of x^k in the polynomial $(1+x)^n$. Since $(1+x)^n = (1+x)^i(1+x)^{n-i}$ for any $i \leqslant n$, the coefficient of x^k in the product would be the sum of the products of the coefficients of x^j and x^{k-j} in the first and the second polynomial, summed over all $j \leqslant i \wedge k$. In particular, this remark implies that

$$\sum_{j_s=0}^{k_s \wedge i_s} r_s(k_s, j_s) = 1, \qquad \sum_{j_f=0}^{k_f \wedge i_f} r_f(k_f, j_f) = 1.$$

Lemma 2: The workload distribution at first k servers of the coupled system $\tilde{\mathcal{S}}$ defined in Definition 4 are mutually independent. Each of them is an $M/G/1$ queue with independent Poisson arrivals to slow and fast servers in the first k servers, having homogeneous rates given by

$$\tilde{\lambda}_s \triangleq \Lambda \frac{kp_s}{n_s}(p_s + p_f r_f(k_f, 0)), \tag{10}$$

$$\tilde{\lambda}_f \triangleq \Lambda \frac{kp_f}{n_f}(p_f + p_s r_s(k_s, 0)), \tag{11}$$

where rate $\Lambda = \frac{n\lambda}{k}$, probabilities r_s, r_f are defined in (7) and (8) respectively, and $k_s + k_f = i_s + i_f = k$.

Proof: Please refer to Appendix A-B. ■

Lemma 3: Consider the arrival rates $\tilde{\lambda}_s$ defined in (10) for slow servers and $\tilde{\lambda}_f$ defined in (11) for fast servers, in the first k servers of the coupled system $\tilde{\mathcal{S}}$. Then, we have

(a) $\tilde{\lambda}_s \leqslant \lambda_s$ and $\tilde{\lambda}_f \leqslant \lambda_f$ and

(b) $\lambda_s - \tilde{\lambda}_s = O\Big(\frac{k^2}{n}\Big)$, and $\lambda_f - \tilde{\lambda}_f = O\Big(\frac{k^2}{n}\Big)$.

Proof: Please refer to Appendix A-C. ■

Definition 5: For $w \in \mathbb{R}_+^k$, we define the joint distribution of workload at first k servers in systems $\mathcal{S}, \tilde{\mathcal{S}}, \hat{\mathcal{S}}$ at time t by

$$\pi_t^k(w) \triangleq P\Big(\cap_{j=1}^k \{W_j(t) \leqslant w_j\}\Big),$$
$$\tilde{\pi}_t^k(w) \triangleq P\Big(\cap_{j=1}^k \Big\{\tilde{W}_j(t) \leqslant w_j\Big\}\Big),$$
$$\hat{\pi}_t^k(w) \triangleq P\Big(\cap_{j=1}^k \Big\{\hat{W}_j(t) \leqslant w_j\Big\}\Big).$$

The corresponding equilibrium distributions are denoted by $\pi^k, \tilde{\pi}^k, \hat{\pi}^k$ respectively.

Definition 6: Consider two distributions $\pi, \nu : \mathcal{B}(\mathcal{X}) \to [0,1]$. Then, the total variation distance is defined as

$$d_{\mathrm{TV}}(\pi, \nu) \triangleq \sup_{A \in \mathcal{B}(\mathcal{X})} |\pi(A) - \nu(A)|\,.$$

Remark 3: If π, ν are distributions for random variables $W, V : \Omega \to \mathcal{X}$, then $d_{\mathrm{TV}}(\pi, \nu) \leqslant P\{W \neq V\}$. To see this, we observe that for all events $A \in \mathcal{B}(\mathcal{X})$, we have

$$\pi(A) - \nu(A) = P\{W \in A, W \neq V\} - P\{V \in A, W \neq V\}$$
$$\leqslant P\{W \neq V\}\,.$$

Lemma 4: If $\tau = O\Big(\frac{\sqrt{n}}{k}\Big)$, then $d_{\mathrm{TV}}(\pi_\tau^k, \tilde{\pi}_\tau^k) = O\Big(\frac{k^2}{\sqrt{n}}\Big)$.

Proof: Please refer to Appendix A-D. ■

Lemma 5: If time $\tau = O(\frac{\sqrt{n}}{k})$, then we have

$$d_{\mathrm{TV}}(\pi_\tau^k, \pi^k) = O\Big(\frac{k^2}{\sqrt{n}}\Big), \quad d_{\mathrm{TV}}(\tilde{\pi}_\tau^k, \tilde{\pi}^k) = O\Big(\frac{k^2}{\sqrt{n}}\Big),$$
$$d_{\mathrm{TV}}(\hat{\pi}_\tau^k, \hat{\pi}^k) = O\Big(\frac{k^2}{\sqrt{n}}\Big).$$

Proof: Please refer to Appendix A-E. ■

Lemma 6: The total variation distance between the equilibrium distribution of workloads in the first k servers of systems $\tilde{\mathcal{S}}$ and $\hat{\mathcal{S}}$ is $d_{\mathrm{TV}}(\tilde{\pi}^k, \hat{\pi}^k) = O(\frac{k^2}{\sqrt{n}})$.

Proof: Please refer to Appendix A-F. ■

Theorem 1 (Asymptotic independence): Consider the equilibrium distributions $\pi^k, \hat{\pi}^k$ for workloads in the first k servers of systems $\mathcal{S}$ and $\hat{\mathcal{S}}$, respectively. Then, the total variance distance $d_{\mathrm{TV}}(\pi^k, \hat{\pi}^k) = O\Big(\frac{k^2}{\sqrt{n}}\Big)$. In particular, if $k = o(n^{\frac{1}{4}})$, then

$$\lim_{n\to\infty} d_{\mathrm{TV}}(\pi^k, \hat{\pi}^k) = 0.$$

Proof: Let $\tau = O(\frac{\sqrt{n}}{k})$. Using triangular inequality for the total variation distance, we can write

$$d_{\mathrm{TV}}(\pi^k, \hat{\pi}^k) \leqslant d_{\mathrm{TV}}(\pi^k, \pi_\tau^k) + d_{\mathrm{TV}}(\pi_\tau^k, \tilde{\pi}_\tau^k) + d_{\mathrm{TV}}(\tilde{\pi}_\tau^k, \tilde{\pi}^k)$$
$$+ d_{\mathrm{TV}}(\tilde{\pi}^k, \hat{\pi}^k).$$

The result follows from Lemma 4, Lemma 5, and Lemma 6. ■

Remark 4: We have shown asymptotic independence for the first k out of n queues, so long as $k = o(n^{\frac{1}{4}})$. Without any loss of generality, the asymptotic independence holds for any set $A \subseteq [n]$ of size $|A| = o(n^{\frac{1}{4}})$ out of n queues.

Remark 5: Denoting the equilibrium distribution for the workload at servers in a subset $A \subseteq [n]$ as $\pi^A : \mathcal{B}(\mathbb{R}_+^A) \to [0,1]$, defined for all $x \in \mathbb{R}_+^A$ as

$$\pi^A(x) \triangleq \lim_{t\to\infty} P\Big(\cap_{j\in A}\{W_j(t) \leqslant x_j\}\Big).$$

Since the system $\mathcal{S}$ has Poisson arrivals, it follows from PASTA property [32] that for all $x \in \mathbb{R}_+^A$

$$\pi^A(x) = \lim_{i\to\infty} P\Big(\cap_{j\in A}\{W_{i,j} \leqslant x_j\}\Big).$$

From the definition of limiting marginal workload distribution in (4), the asymptotic independence of the workload distribution for any finite set of servers in Theorem 1, the definition of total variation distance, and the fact that the limiting marginal workload distribution is identical within a class as shown in (5), we get for all $x \in \mathbb{R}_+^A$

$$\pi^A(x) = \prod_{j\in A\cap E_s} F_s(x_j) \prod_{j\in A\cap E_f} F_f(x_j) + O\Big(\frac{|A|^2}{\sqrt{n}}\Big). \tag{12}$$

V. Mean Task Completion Time

Recall that task completion time is the maximum of all k sub-task completion times. From the asymptotic independence of limiting sub-task completion times in Theorem 1, we can compute the limiting mean task completion time as the number of servers grows larger. This allows us to analytically compute the limiting mean task completion time as an integral. This is shown in Section V-A for a general sub-task service time distribution and specifically computed for the exponential distribution. One can numerically evaluate this integral to find the optimal selection probability p_s^* that minimizes the limiting mean task completion time. We next propose an upper bound on the limiting mean task completion time in Section V-B for a general sub-task service time distribution. We analytically compute the selection probability $\hat{p}_s$ that minimizes this upper bound for exponential distribution in Section V-B.1 and for shifted exponential in Section V-B.2. The shifted exponential distribution is a generalization of the exponential distribution and is shown to be a better model for service in realistic cloud computing systems such as Amazon S3 and Tahoe [8], [33], [34]. The probability $\hat{p}_s$ serves as an approximation for the optimal selection probability p_s^*.

A. Exact Computation

Theorem 2: Consider the system $\mathcal{S}$ with $k = o(n^{\frac{1}{4}})$ with the slow server selection probability p_s, and the limiting distribution of sub-task completion times $L_s, L_f : \mathbb{R}_+ \to [0,1]$ at slow and fast servers respectively. The asymptotic equilibrium distribution $H : \mathbb{R}_+ \to [0,1]$ of task completion time is

$$H(x) = (p_s L_s(x) + p_f L_f(x))^k, \quad x \in \mathbb{R}_+. \tag{13}$$

Proof: Recall that for ith arriving task, the sub-task completion time at server j is $T_{i,j} = W_{i,j} + X_{i,j}$, and the task completion time $T_i = \max_{j\in I^i} T_{i,j}$. Using the tower property of conditional expectation, we can write the probability of task completion time being less than equal to a threshold x, as

$$P\{T_i \leqslant x\} = \mathbb{E}[\mathbb{E}[\prod_{j\in I^i} \mathbb{1}_{\{W_{i,j}\leqslant x - X_{i,j}\}} \mid (X_{i,j}, j \in I^i), I^i]].$$

Since I^i takes finitely many values, we can write the conditional expectation

$$\begin{aligned}&\mathbb{E}[\prod_{j\in I^i} \mathbb{1}_{\{W_{i,j}\leqslant x - X_{i,j}\}} \mid (X_{i,j}, j \in I^i), I^i]\\ &= \sum_{A\subseteq[n]:|A|=k} \mathbb{1}_{\{I^i = A\}} \mathbb{E}[\prod_{j\in A} \mathbb{1}_{\{W_{i,j}\leqslant x - X_{i,j}\}} \mid (X_{i,j}, j \in A)].\end{aligned}$$

Taking time equilibrium limit $i \to \infty$, exchanging limit and expectation using the monotone convergence theorem for non-negative random variables, exchanging finite sum and limits, and independence of selection set I^i and service-time $X_{i,j}$ for each task-arrival $i \in \mathbb{N}$, we get

$$\begin{aligned}H_n(x) &= \lim_{i\to\infty} \mathbb{E}[\prod_{j\in I^i} \mathbb{1}_{\{W_{i,j}\leqslant x - X_{i,j}\}}]\\ &= \sum_{A\subseteq[n]:|A|=k} P\{I^\infty = A\}\, \mathbb{E}\pi^A(x - X_{\infty,j} : j \in A).\end{aligned}$$

From (12) for joint equilibrium workload distribution on servers A, the fact that $L_{T_j}(x) = \mathbb{E}F_{W_j}(x - X_{\infty,j})$ for all $x \in \mathbb{R}_+$, the definition of distribution $q \in \mathcal{M}([k]_0)$ in (1), and (6) for marginal sub-task completion distribution being identical within a class, we get

$$H_n(x) = \sum_{k_s=0}^{k} q(k_s) L_s(x)^{k_s} L_f(x)^{k-k_s} + O\Big(\frac{k^2}{\sqrt{n}}\Big).$$

Result follows from taking limit $n \to \infty$ on both sides for $k = o(n^{\frac{1}{4}})$, the binomial form of $q(k_s) = \binom{k}{k_s} p_s^{k_s}(1-p_s)^{k-k_s}$, and the binomial expansion of $(a+b)^k$. ■

Corollary 1: The mean task completion time for the heterogeneous system under consideration is given by

$$\lim_{i\to\infty} \mathbb{E}[T_i] = \int_{x\in\mathbb{R}_+} [1 - (p_s L_s(x) + p_f L_f(x))^k] dx.$$

If the sub-task completion times are exponentially distributed, each queue observed in isolation is an $M/M/1$ queue, and we get the following proposition.

Proposition 1: Consider the case when sub-task service times at slow and fast servers are distributed exponentially with rates μ_s and μ_f respectively, such that slow server loads $\rho_s \triangleq \frac{\lambda_s}{\mu_s} < 1$ and fast server loads $\rho_f \triangleq \frac{\lambda_f}{\mu_f} < 1$. Then, the limiting marginal workload distribution at slow and fast servers for $w \in \mathbb{R}_+$ are

$$F_s(w) = 1 - \rho_s e^{-(\mu_s - \lambda_s)w}, \quad F_f(w) = 1 - \rho_f e^{-(\mu_f - \lambda_f)w}.$$

Further, the limiting sub-task completion times for slow and fast servers are

$$L_s(x) = 1 - e^{-(\mu_s - \lambda_s)x}, \quad L_f(x) = 1 - e^{-(\mu_f - \lambda_f)x}.$$

Remark 6: We observe that the slow and fast server queues are unstable for $\rho_s \geqslant 1$ and $\rho_f \geqslant 1$, respectively. It follows that the stability conditions for all queues in the system are

$$\lambda p_s \leqslant \mu_s f_s, \quad \lambda p_f \leqslant \mu_f \bar{f}_s, \quad \lambda < \mu_s f_s + \mu_f \bar{f}_s. \tag{14}$$

We have normalized the arrival rates to be independent of the system size n and the number of forked sub-tasks k, such that the stability region only depends on service rates μ_s, μ_f and the fraction of slow servers f_s. In particular, we observe that the stability region for normalized arrival rate λ is a convex sum of the fast and slow service rates and reduces linearly with increased fraction f_s of slow servers.

Corollary 2: For stable $M/M/1$ queues in Proposition 1, the limiting mean task completion time is

$$\int_{x\in\mathbb{R}_+} [1 - (1 - p_s e^{-(\mu_s-\lambda_s)x} - p_f e^{-(\mu_f-\lambda_f)x})^k] dx.$$

Remark 7: For stable $M/M/1$ queues in Proposition 1, the limiting mean task completion time as

$$\begin{aligned}&\sum_{i=1}^{k}\sum_{k_s=0}^{i} (-1)^{i-1}\binom{k}{i}\binom{i}{k_s}\\ &\quad\times \frac{p_s^{k_s}(1-p_s)^{i-k_s}}{k_s(\mu_s-\lambda_s) + (i-k_s)(\mu_f-\lambda_f)}.\end{aligned}$$

Even for exponentially distributed sub-task completion times, analytical computation of the optimal selection probability p_s^* that minimizes the limiting mean task completion time seems intractable for $k > 1$. However, one can numerically evaluate the optimal selection probability.

B. Upper and Lower Bound

For an $M/G/1$ queue with Poisson arrivals of rate λ and *i.i.d.* service time sequence X, the system load is $\rho \triangleq \lambda\mathbb{E}X_1$. Using Pollaczek-Khintchine formula [35], we can write the limiting mean sojourn time as $\mathbb{E}X_1 + \frac{\lambda\mathbb{E}X_1^2}{2(1-\rho)}$, for load $\rho < 1$. In this section, we will provide an upper and lower bound on the mean task completion time for the heterogeneous system $\mathcal{S}$ with Poisson arrivals and general *i.i.d.* service times.

Theorem 3: The mean task completion time for the heterogeneous system $\mathcal{S}$ with Poisson arrivals and general i.i.d. *service times is upper and lower bounded as*

$$h(p_s) \leqslant \lim_{i\to\infty} \mathbb{E}T_i \leqslant kh(p_s),$$

where the mapping $h : [0,1] \to \mathbb{R}_+$ is defined for $p \in (1 - \frac{\bar{f}_s}{\lambda\mathbb{E}X_f}, \frac{f_s}{\lambda\mathbb{E}X_s})$ as

$$h(p) \triangleq p\Big(\mathbb{E}X_s + \frac{\mathbb{E}X_s^2}{2(\frac{f_s}{\lambda p} - \mathbb{E}X_s)}\Big) + \bar{p}\Big(\mathbb{E}X_f + \frac{\mathbb{E}X_f^2}{2(\frac{\bar{f}_s}{\lambda\bar{p}} - \mathbb{E}X_f)}\Big).$$

Proof: The maximum of k random variables is upper bounded by their sum and lower bounded by their average. Therefore, we can upper bound the completion time of task i by the sum of sub-task completion times at k selected servers I^i and lower bound it by their average. That is,

$$\frac{1}{k}\sum_{j\in I^i} T_{i,j} \leqslant T_i = \max_{j\in I^i} T_{i,j} \leqslant \sum_{j\in I^i} T_{i,j}.$$

Since the marginal sub-task completion times at all slow and fast servers are identical, we get

$$\mathbb{E}\sum_{j\in I^i} T_{i,j} = \sum_{k_s=0}^{k} q(k_s)\Big(k_s\mathbb{E}[T_{i,j}]\mathbb{1}_{\{j\in E_s\}} + (k-k_s)\mathbb{E}[T_{i,j}]\mathbb{1}_{\{j\in E_f\}}\Big).$$

The result follows by taking limit $i \to \infty$ on both sides, applying the Pollaczek-Khintchine formula for stable $M/G/1$ queues, the definition of λ_s in (2) and λ_f in (3), and the fact that $\sum_{k_s=0}^{k} k_s q(k_s) = kp_s$. ■

Remark 8: Recall that the optimal slow server selection probability $p_s^* = \arg\min_{p_s} \lim_{i\to\infty} \mathbb{E}T_i$ and we can define $\hat{p}_s \triangleq \arg\min_p h(p)$. Since the function h is independent of k, we observe that the $\hat{p}_s$ minimizes both the lower and the upper bound on the limiting mean task completion time. Even though the lower and the upper bound differ by a factor of k, they have the same minimizer $\hat{p}_s$. We take this minimizing probability as an approximation for the optimal slow server selection probability p_s^*.

1) Exponential Sub-Task Service:

Corollary 3: Consider the stable heterogeneous system $\mathcal{S}$ with exponentially distributed sub-task service times having rates (μ_s, μ_f) for slow and fast servers. The limiting mean task completion time is upper and lower bounded as

$$g(p_s) \leqslant \lim_{i\to\infty} \mathbb{E}T_i \leqslant kg(p_s),$$

where $g : [0,1] \to \mathbb{R}_+$ is defined for $p \in (1 - \frac{\mu_f\bar{f}_s}{\lambda}, \frac{\mu_s f_s}{\lambda})$, as

$$g(p) \triangleq -\frac{1}{\lambda} + \frac{f_s}{\lambda(1-\frac{\lambda p}{\mu_s f_s})} + \frac{\bar{f}_s}{\lambda(1-\frac{\lambda(1-p)}{\mu_f\bar{f}_s})}. \tag{15}$$

Proof: For exponentially distributed service time with rates μ_s and μ_f for slow and fast servers, we have $\mathbb{E}X_s^2 = \frac{2}{\mu_s^2}$ and $\mathbb{E}X_f^2 = \frac{2}{\mu_f^2}$. ■

Remark 9: We observe that the upper bound on the mean task completion time has three terms. The second term is increasing and the third term is decreasing, both in slow server selection probability p_s. We can verify that $g(p_s)$ is convex in p_s and hence has a unique minimum. Accordingly, we define $\hat{p}_s \in [0,1]$ as the minimizing probability for the upper bound on the limiting mean task completion time for a stable system. We define $\alpha \triangleq \frac{\bar{f}_s}{f_s}\sqrt{\frac{\mu_f}{\mu_s}}$, and two thresholds τ_1, τ_2 on normalized arrival rates as

$$\tau_1 \triangleq \bar{f}_s(\mu_f - \sqrt{\mu_s\mu_f}), \qquad \tau_2 \triangleq \bar{f}_s(\mu_f + \sqrt{\mu_s\mu_f}). \tag{16}$$

We observe that $\tau_1 < \tau_2$, and verify that $\alpha \leqslant 1$ iff $\tau_2 \leqslant \mu_s f_s + \mu_f\bar{f}_s$.

Corollary 4: The upper bound on the limiting mean task completion time for exponentially distributed sub-task service times is minimized by the selection probability

$$\hat{p}_s = \begin{cases} 0, & \lambda \leqslant \tau_1, \\ \dfrac{1-\frac{\tau_1}{\lambda}}{1+\alpha}, & \tau_1 \leqslant \lambda < \mu_s f_s + \mu_f\bar{f}_s, \end{cases} \tag{17}$$

for α and τ_1 defined in Remark 9.

Proof: We take the derivative of the upper bound on the mean task completion time for memoryless sub-task completion times in Corollary 3 with respect to p_s and write it in terms of thresholds τ_1, τ_2 defined in (16) and constant α, as

$$g'(p_s) = \frac{(\tau_1 - \lambda + \lambda p_s(1+\alpha))(\tau_2 - \lambda + \lambda p_s(1-\alpha))}{\mu_s\mu_f^2\bar{f}_s^2(1-\frac{\lambda p_s}{\mu_s f_s})^2(1-\frac{\lambda p_f}{\mu_f\bar{f}_s})^2}.$$

We observe that the denominator is always positive, and the numerator is a product of two linear functions $f_1, f_2 : \mathbb{R} \to \mathbb{R}$, defined as $f_1(p) \triangleq \tau_1 - \lambda + \lambda p(1+\alpha)$ and $f_2(p) \triangleq \tau_2 - \lambda + \lambda p(1-\alpha)$. The roots of the two linear maps f_1, f_2 are respectively

$$p_1^* \triangleq \frac{\lambda - \tau_1}{\lambda(1+\alpha)}, \qquad p_2^* \triangleq \frac{\lambda - \tau_2}{\lambda(1-\alpha)}. \tag{18}$$

From the definition, it follows that $p_1^* \geqslant 0$ for $\lambda \geqslant \tau_1$ and $p_1^* \leqslant 1$. In addition, the condition $p_1^* < \frac{\mu_s f_s}{\lambda}$ stabilizes the slow server queues, and the condition $1 - p_1^* < \frac{\mu_f\bar{f}_s}{\lambda}$ stabilizes the fast server queues. Therefore, the condition on normalized arrival rate $\lambda < \mu_s f_s + \mu_f\bar{f}_s$ stabilizes all queues. We observe that $f_1 \leqslant 0$ iff $p \leqslant p_1^*$. We next observe that for stable queues, $p_2^* < p_1^*$ iff $\alpha < 1$. This is because the condition $p_2^* < p_1^*$ for $\alpha < 1$, is equivalent to the condition $p_1^* < p_2^*$ for $\alpha > 1$, which is equivalent to

$$\lambda < \frac{\tau_2}{2}\Big(\frac{1}{\alpha}+1\Big) - \frac{\tau_1}{2}\Big(\frac{1}{\alpha}-1\Big) = f_s\mu_s + \bar{f}_s\mu_f.$$

For normalized arrival rate $\lambda < \tau_1$, the upper bound g is always increasing and hence is minimized by $\hat{p}_s = 0$. We consider the following three cases for $\lambda > \tau_1$.

Case $\alpha > 1$**.** In this case, $f_2 \geqslant 0$ iff $p \leqslant p_2^*$. Thus $g'(p_s) \leqslant 0$ iff $p_s \in [0, p_1^*] \cup [p_2^*, 1]$. Hence, the upper bound g decreases in $[0, p_1^*]$, increases in $[p_1^*, p_2^*]$, and decreases thereafter. Therefore, g is minimized for $p_s \in \{p_1^*, 1\}$ for $\lambda \geqslant \tau_1$. We observe that p_1^* satisfies the stability conditions for slow server in the region $\lambda \in \mu_s f_s + [0, \mu_f \bar{f}_s)$ and for the fast server in the region $\lambda \in \mu_f \bar{f}_s + [0, \mu_s f_s)$. In addition, we observe that $g(p_1^*) < g(1)$ for $\lambda \in [\tau_1, \mu_s f_s)$. It follows that $\hat{p}_s = p_1^*$ for all $\lambda \in [\tau_1, \mu_s f_s + \mu_f \bar{f}_s)$.

Case $\alpha < 1$**.** In this case, $f_2 \leqslant 0$ iff $p \leqslant p_2^*$. Thus $g'(p_s) \geqslant 0$ iff $p_s \in [0, p_2^*] \cup [p_1^*, 1]$. Hence, the upper bound g increases in $[0, p_2^*]$, decreases in $[p_2^*, p_1^*]$, and increases thereafter. Therefore, g is minimized for $p_s \in \{0, p_1^*\}$ for $\lambda \geqslant \tau_1$. In addition, we observe that $g(p_1^*) < g(0)$ for all $\lambda \in [\tau_1, \mu_f \bar{f}_s)$. It follows that $\hat{p}_s = p_1^*$ for all $\lambda \in [\tau_1, \mu_s f_s + \mu_f \bar{f}_s)$.

Case $\alpha = 1$**.** In this case, $f_2 = \tau_2 - \lambda = \mu_s f_s + \mu_f \bar{f}_s - \lambda > 0$ in the stability region. Further, the upper bound g decreases in $[0, p_1^*]$ and increases in $[p_1^*, 1]$. Therefore, g is minimized for $p_s = p_1^*$ for $\lambda \geqslant \tau_1$. ■

Remark 10: From Corollary 4, we observe that the approximately optimal slow server selection probability $\hat{p}_s$ is a concave increasing function of normalized arrival rate λ. The probability $\hat{p}_s = 0$ until a threshold τ_1 and saturates to probability $\frac{\mu_s f_s}{\mu_s f_s + \mu_f \bar{f}_s}$ at the boundary of the stability region. In other words, it is best to schedule incoming jobs on fast servers for sufficiently low normalized arrival rates $\lambda < \tau_1$. As the load increases, incoming jobs need to be scheduled on slow servers, and the probability of selection of slow servers is a concave increasing function of the normalized arrival rate λ. We also observe that the threshold τ_1 is an affine decreasing function of the fraction of slow servers f_s. If there is a larger fraction of slow servers, then $\hat{p}_s$ quickly becomes non-zero.

2) Shifted Exponential Sub-Task Service:

Corollary 5: Consider the heterogeneous system $\mathcal{S}$ for shifted exponentially distributed sub-task service times with parameters (c_s, μ_s) and (c_f, μ_f) for slow and fast servers, respectively, such that

$$\lambda p_s < \frac{f_s}{\mathbb{E}X_s}, \quad \lambda p_f < \frac{\bar{f}_s}{\mathbb{E}X_f}, \quad \lambda < \frac{f_s}{\mathbb{E}X_s} + \frac{\bar{f}_s}{\mathbb{E}X_f}. \tag{19}$$

Then, the mean task completion time is upper and lower bounded as

$$h(p_s) \leqslant \lim_{i \to \infty} \mathbb{E}T_i \leqslant k h(p_s),$$

where $h : [0,1] \to \mathbb{R}_+$ is defined for $p_s \in (1 - \frac{\bar{f}_s}{\lambda \mathbb{E}X_f}, \frac{f_s}{\lambda \mathbb{E}X_s})$, as

$$h(p_s) \triangleq p_s \Big(\frac{(1 - \lambda_s c_s)\mathbb{E}X_s + \frac{1}{2}c_s^2 \lambda_s}{(1 - \lambda_s \mathbb{E}X_s)} \Big) + p_f \Big(\frac{(1 - \lambda_f c_f)\mathbb{E}X_f + \frac{1}{2}c_f^2 \lambda_f}{(1 - \lambda_f \mathbb{E}X_f)} \Big).$$

Proof: For shifted exponentially distributed service time with parameters (c_s, μ_s) and (c_f, μ_f) for slow and fast servers respectively, we have means $\mathbb{E}X_s = c_s + \frac{1}{\mu_s}$ and $\mathbb{E}X_f = c_f + \frac{1}{\mu_f}$, and the second moments $\mathbb{E}X_s^2 = (\mathbb{E}X_s)^2 + \frac{1}{\mu_s^2}$ and $\mathbb{E}X_f^2 = (\mathbb{E}X_f)^2 + \frac{1}{\mu_f^2}$. ■

Corollary 6: Consider a shifted exponential distribution for sub-task completion times with parameters (c_s, μ_s) for slow servers and parameters (c_f, μ_f) for fast servers, such that $\lambda < f_s \mathbb{E}X_s \wedge \bar{f}_s \mathbb{E}X_f$. The upper bound on the limiting mean task completion time is minimized by the selection probability $\hat{p}_s$ that solves

$$\begin{aligned} c_s &+ \frac{\frac{1}{2}c_s^2 \lambda_s}{(1 - \lambda_s \mathbb{E}X_s)} + \frac{\frac{1}{\mu_s} + \frac{1}{2}c_s^2 \lambda_s}{(1 - \lambda_s \mathbb{E}X_s)^2} \\ &= c_f + \frac{\frac{1}{2}c_f^2 \lambda_f}{(1 - \lambda_f \mathbb{E}X_f)} + \frac{\frac{1}{\mu_f} + \frac{1}{2}c_f^2 \lambda_f}{(1 - \lambda_f \mathbb{E}X_f)^2}. \end{aligned} \tag{20}$$

VI. Comparison to Deterministic Selection

We compare the performance of the probabilistic selection of slow and fast servers to a deterministic selection. Consider the system $\mathcal{S}$ of n servers, partitioned by a set $E_s \subseteq [n]$ of slow servers, and remaining set $E_f \subseteq [n]$ of fast servers such that $n_s = |E_s|, n_f = |E_f|$, and a Poisson arrival of tasks with homogeneous rate $\Lambda = \frac{n\lambda}{k}$ where $n = n_s + n_f$. Service time at all slow servers is *i.i.d.* with distribution G_s, independent of the service time at all fast servers, which is *i.i.d.* with distribution G_f. Each incoming task i is scheduled on a set of slow servers $I_s^i \subseteq E_s$ and $I_f^i \subseteq E_f$, chosen uniformly at random within the class, where $\left|I_s^i\right| = k_s$ and $\left|I_f^i\right| = k_f$ are fixed. The task i is assumed to be completed when sub-tasks scheduled at all servers $I_s^i \cup I_f^i$ get completed.

Each server in this system is an $M/G/1$ queue, where one can verify that the arrivals to all servers are Poisson with homogeneous rates $\lambda_s \triangleq \frac{n\lambda k_s}{kn_s}$ for slow servers and $\lambda_f \triangleq \frac{n\lambda k_f}{kn_f}$ for fast servers. We denote the limiting marginal workload distribution at slow and fast servers by L_s and L_f, respectively. We observe that these arrival rates are identical to the ones defined in (2) and (3) for $p_s = \frac{k_s}{k}$ and $f_s = \frac{n_s}{n}$. Hence the limiting marginal workload distribution at slow and fast servers is identical to that of the system $\mathcal{S}$. Applying the techniques developed in [18] and [23], we can show that for any fixed server subset $A \subseteq [n]$, the joint stationary workload distribution at servers in A grows asymptotically independent as $n \to \infty$. Accordingly, the limiting distribution for task completion time for this deterministic setup when the number of servers n grows large can be written for all $x \in \mathbb{R}_+$ as

$$H_{k_s}^d(x) = L_s(x)^{k_s} L_f(x)^{k - k_s}. \tag{21}$$

The limiting mean task completion time can be written as an integral of limiting complementary distribution of task completion time. Hence, we define the optimal deterministic selection of the number of slow servers as

$$k_s^* \triangleq \min_{k_s \in [k]_0} \int_{x \in \mathbb{R}_+} (1 - H_{k_s}^d(x)) dx. \tag{22}$$

Proposition 2: Consider the heterogeneous system $\mathcal{S}$ with n servers, with constant fractions f_s and $\bar{f}_s$ of slow and fast servers, respectively, and (k,k) fork-join of tasks. Let k_s^* be the optimal deterministic selection of slow servers as defined *in (22). Then, the optimal selection probability of slow servers converges to*

$$\lim_{k\to\infty} p_s^* = \frac{k_s^*}{k}.$$

Further, the binomial probability $q^*(\ell)$ of choosing ℓ slow and $k-\ell$ fast servers with the optimal probability p_s^* of slow server selection, converges to

$$\lim_{k\to\infty} q^*(\ell) = \mathbb{1}_{\{\ell = kp_s^*\}}.$$

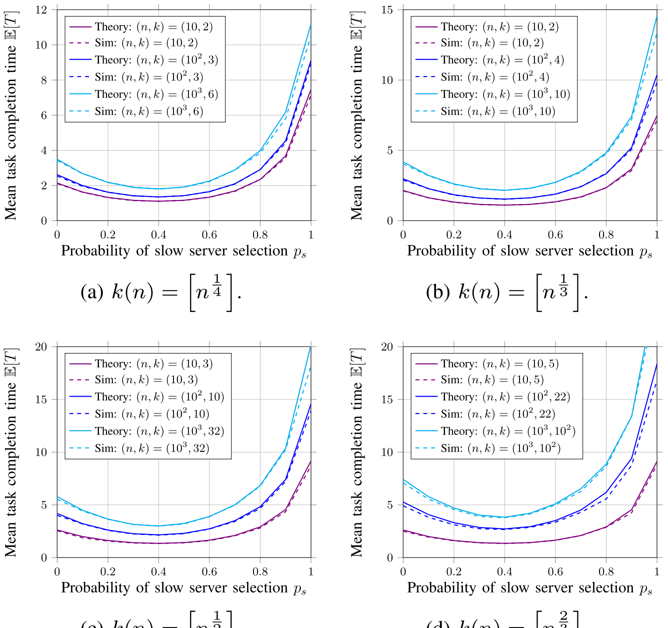
Proof: The proof is provided in Appendix B. ■

VII. Numerical Results

We have computed the mean task completion time under the regime of an asymptotically large number of servers n, which yields an asymptotic independence of marginal workload distribution at any arbitrary set of k servers. We observe in Section VII-A that this asymptotic independence seems to hold for $k = o(n^{\frac{2}{3}})$, even though theoretical guarantees only exist for $k = o(n^{\frac{1}{4}})$. We compare the numerically obtained optimal slow server selection probability p_s^* with its analytically obtained approximation $\hat{p}_s$ in Section VII-B as a function of normalized arrival rate in the stability region, varying the service rates (μ_s, μ_f), the number of sub-tasks k, and the fraction of slow servers f_s. We compare the performance of probabilistic and deterministic server selection in Section VII-C.

A. Asymptotic Independence

We have shown in Theorem 1 that the independence of marginal workload distribution at individual queues holds when the number of sub-tasks $k(n) = o(n^{\frac{1}{4}})$, for a large number of servers n. To verify the robustness of this condition, we conducted numerical and empirical studies to determine the limiting mean task completion time in n heterogeneous server systems under the proposed policy as a function of selection probability p_s for different scaling of the number of sub-tasks $k(n)$. We considered a finite number of servers $n \in \{10, 10^2, 10^3\}$, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks $k(n) = o(n^{\alpha})$, the normalized Poisson arrival rate of tasks to the system $\lambda = 0.9$, and exponentially distributed sub-task service times with rates $(\mu_s, \mu_f) = (2, 2.5)$, for the slow and the fast servers respectively. We have plotted the empirically obtained mean task completion time against the theoretically computed values from Corollary 2, as a function of increasing selection probability $p_s \in [0,1]$ for exponent $\alpha \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$ in Fig. 1a, Fig. 1b, Fig. 1c, and Fig. 1d respectively. Interestingly, we observe that the assumption of independence of marginal workloads remains robust even for small values of n. As expected, the accuracy of this independence assumption improves as n grows larger. Furthermore, we found that the mean task completion time is a convex function of the selection probability p_s, indicating that it possesses a unique minimum. Even though Theorem 1 demonstrated the asymptotic independence of marginal workloads for $k(n) = o(n^{\frac{1}{4}})$, the empirical observations suggest that this assumption continues to hold for a larger scaling.



(a) $k(n) = \left\lceil n^{\frac{1}{4}} \right\rceil$.

(b) $k(n) = \left\lceil n^{\frac{1}{3}} \right\rceil$.

(c) $k(n) = \left\lceil n^{\frac{1}{2}} \right\rceil$.

(d) $k(n) = \left\lceil n^{\frac{2}{3}} \right\rceil$.

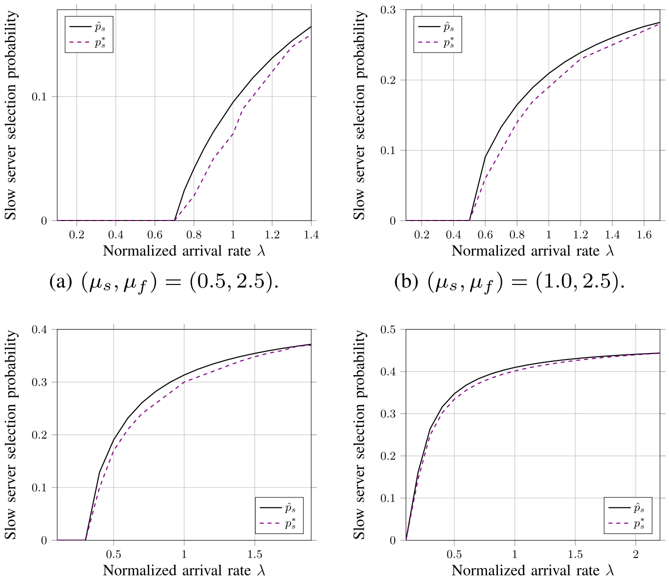
Fig. 1. Comparison of mean task completion time obtained theoretically and empirically as a function of slow server selection probability p_s for the fraction of slow servers $f_s = 0.5$, normalized Poisson task arrival rate $\lambda = 0.9$, exponential sub-task service times with rate $(\mu_s, \mu_f) = (2, 2.5)$ for the slow and the fast servers respectively, and the number of sub-tasks $k(n)$.

B. Optimal Selection Probability p_s^* and Its Approximation $\hat{p}_s$

We observed that the limiting mean task completion time can be uniquely minimized by the optimal selection probability p_s^*. However, this optimal probability is difficult to compute analytically, even for the simplest case of exponential service. As such, we proposed an approximately optimal selection probability $\hat{p}_s$ that minimizes an upper and lower bound on the limiting mean task completion time. This approximately optimal selection probability $\hat{p}_s$ is analytically computable for many sub-task service time distributions. This subsection empirically evaluates the approximation error for (1) different service rate pairs (μ_s, μ_f), (2) different number of sub-tasks k, and (3) different fraction f_s of slow servers.

1) Varying Service Rate Pairs: We evaluate a heterogeneous system with the number of servers $n = 10^3$ and the fraction of slow servers $f_s = 0.5$. For the exponential distribution of sub-task service times with rates (μ_s, μ_f) for the slow and the fast servers, we numerically obtained the optimal selection probability p_s^* from Remark 7 and theoretically obtained the approximately optimal selection probability $\hat{p}_s$ from Corollary 4. We plotted the comparison of probability p_s^* and its approximation $\hat{p}_s$ as a function of normalized arrival rate λ for different service rate pairs in Fig. 2.

We repeated this comparison for the shifted exponential distribution for sub-task service times with rates (μ_s, μ_f) and shifts (c_s, c_f) for the slow and the fast servers. We empirically obtained the optimal selection probability p_s^* and numerically obtained the approximately optimal selection probability $\hat{p}_s$ from Corollary 6. We plotted the comparison of optimal selection probability p_s^* and its approximation $\hat{p}_s$ as a function



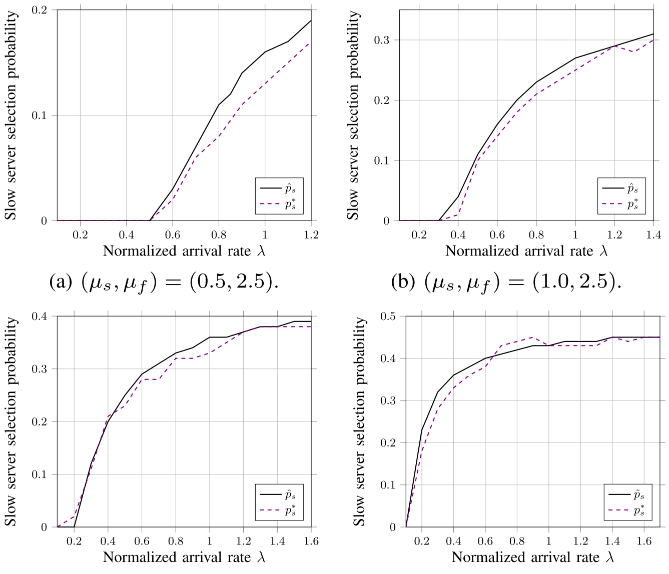
(a) $(\mu_s, \mu_f) = (0.5, 2.5)$.

(b) $(\mu_s, \mu_f) = (1.0, 2.5)$.

(c) $(\mu_s, \mu_f) = (1.5, 2.5)$.

(d) $(\mu_s, \mu_f) = (2.0, 2.5)$.

Fig. 2. Impact of difference in service rates on optimal selection probability p_s^* and its approximation $\hat{p}_s$ for exponential service. We plot p_s^* and $\hat{p}_s$ as a function of normalized Poisson arrival rate λ, for a heterogeneous system with the number of servers $n = 10^3$, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks $k = 10$, and exponential sub-task service times with rates (μ_s, μ_f) for the slow and the fast servers respectively.



(a) $(\mu_s, \mu_f) = (0.5, 2.5)$.

(b) $(\mu_s, \mu_f) = (1.0, 2.5)$.

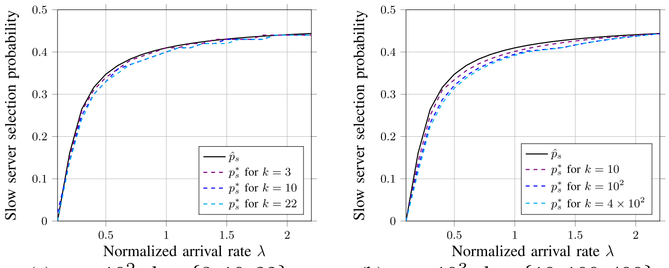
(c) $(\mu_s, \mu_f) = (1.5, 2.5)$.

(d) $(\mu_s, \mu_f) = (2.0, 2.5)$.

Fig. 3. Impact of difference in service rates on optimal selection probability p_s^* and its approximation $\hat{p}_s$ for shifted-exponential service. We plot p_s^* and $\hat{p}_s$ as a function of normalized Poisson arrival rate λ, for a heterogeneous system with the number of servers $n = 10^3$, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks $k = 10$, and shifted exponential sub-task service times with rates (μ_s, μ_f) and shifts $(c_s, c_f) = (0.1, 0.1)$ for the slow and the fast servers respectively.

of normalized arrival rate λ for a fixed shift pair and different service rate pairs in Fig. 3.

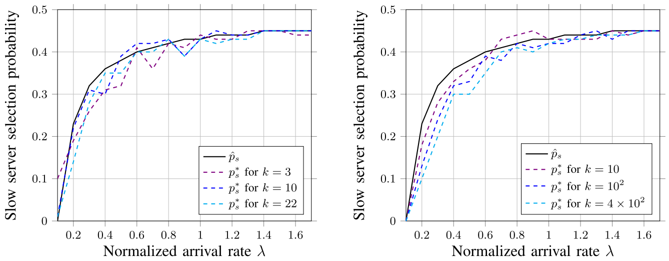
We observe that the approximately optimal selection probability $\hat{p}_s$ is close to the optimal selection probability p_s^* for all normalized arrival rates. In addition, we note that the optimal selection probability of slow servers is concave and increasing in the normalized arrival rate λ. This suggests that the system tries to reduce the use of slow servers at low loads to minimize the limiting mean task completion time. However, when the load increases, the system is forced to increase the usage of slow servers.



(a) $n = 10^2$, $k \in \{3, 10, 22\}$.

(b) $n = 10^3$, $k \in \{10, 100, 400\}$.

Fig. 4. Impact of changing the number of sub-tasks on optimal selection probability p_s^* and its approximation $\hat{p}_s$ for exponential service. We plot p_s^* and $\hat{p}_s$ as a function of normalized Poisson arrival rate λ, for a heterogeneous system with the number of servers n, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks k, and exponential sub-task service times with rates $(\mu_s, \mu_f) = (2, 2.5)$ for the slow and the fast servers respectively.



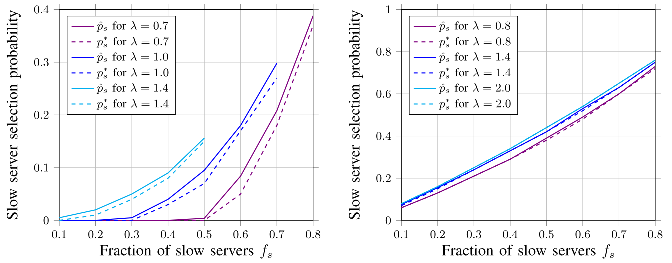
(a) $n = 10^2, k \in \{3, 10, 22\}$.

(b) $n = 10^3, k \in \{10, 100, 400\}$.

Fig. 5. Impact of changing the number of sub-tasks on optimal selection probability p_s^* and its approximation $\hat{p}_s$ for shifted-exponential service. We plot p_s^* and $\hat{p}_s$ as a function of normalized Poisson arrival rate λ, for a heterogeneous system with the number of servers n, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks k, and shifted exponential sub-task service times with rates $(\mu_s, \mu_f) = (2, 2.5)$ and shifts $(c_s, c_f) = (0.1, 0.1)$ for the slow and the fast servers respectively.

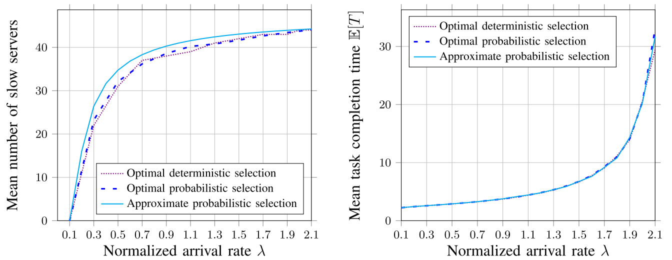
2) Varying the Number of Sub-Tasks: We plotted the optimal selection probability p_s^* and its approximation $\hat{p}_s$ as a function of normalized arrival rate λ for the number of servers $n \in \{10^2, 10^3\}$, different number of sub-tasks k, for exponentially and shifted exponentially distributed sub-task service times in Fig. 4 and Fig. 5 respectively. From Theorem 3 and Remark 8, we observe that the approximately optimal selection probability $\hat{p}_s$ is independent of the number of sub-tasks k. We observe that the optimal selection probability p_s^* weakly depends on k. However, the approximately optimal selection probability $\hat{p}_s$ remains close to the optimal selection probability p_s^* for all normalized arrival rates λ and the number of sub-tasks k.

3) Varying the Fraction of Slow Servers: We plotted the optimal selection probability p_s^* and its approximation $\hat{p}_s$ as a function of the fraction of slow servers f_s for a fixed number of servers and sub-tasks, different normalized task arrival rates, and exponentially distributed sub-task service times with two different service rate pairs in Fig. 6. We observe that the optimal slow server selection probability remains close to its approximation in both cases. The approximation is better



(a) $(\mu_s, \mu_f) = (0.5, 2.5)$. (b) $(\mu_s, \mu_f) = (2.0, 2.5)$.

Fig. 6. Impact of changing the slow server fraction f_s on optimal selection probability p_s^* and its approximation $\hat{p}_s$. We plot p_s^* and $\hat{p}_s$ as a function of the fraction of slow servers f_s for a heterogeneous system with the number of servers $n = 10^3$, the number of sub-tasks $k = 10$, different values of normalized Poisson task arrival rates λ, and exponential sub-task service times with rates (μ_s, μ_f) for the slow and the fast servers respectively.



(a) k_s^*, kp_s^*, and $k\hat{p}_s$. (b) Mean task completion time $\mathbb{E}T$.

Fig. 7. Comparison of mean number of slow servers and mean task completion time for optimal deterministic choice of slow servers k_s^*, optimal probabilistic choice of slow servers, and approximately optimal probabilistic choice of slow servers as a function of normalized Poisson task arrival rate λ for a heterogeneous system with the number of servers $n = 10^3$, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks $k = 10^2$, and exponential sub-task service times with rates $(\mu_s, \mu_f) = (2, 2.5)$ for the slow and the fast servers respectively.

when the service rates are closer, and arrival rates are higher. From (17), we know that the approximately optimal slow server selection probability $\hat{p}_s$ is an increasing function of the slow server fraction f_s, for a fixed load. This property is empirically observed to hold for the optimal slow server selection probability p_s^*. This is due to the need for the utilization of slow servers to reduce the mean task completion time. We also observe that when the service rates are close, the optimal server selection probability depends weakly on the normalized arrival rate λ.

C. Deterministic Versus Probabilistic Selection

Finally, we compare deterministic and probabilistic selection of slow servers for scheduling k sub-tasks for each incoming task. To this end, we evaluated a heterogeneous system with $n = 10^3$ servers, the fraction of slow servers $f_s = 0.5$, the number of sub-tasks $k = 10^2$, and exponential distribution for the sub-task service times with rates $(\mu_s, \mu_f) = (2, 2.5)$ for the slow and the fast servers respectively. In Fig. 7a, we compare the optimal deterministic choice of the number of slow servers k_s^*, the mean number of optimally selected slow servers kp_s^* and its approximation $k\hat{p}_s$, as a function of normalized task arrival rate λ. In Fig. 7b, we compare the mean task completion time for the optimal deterministic choice of the number of slow servers k_s^*, with the mean task completion time for probabilistic selection with optimal slow server selection probability p_s^* and its approximation $\hat{p}_s$, all as a function of normalized task arrival rate λ. As expected from Proposition 2, we observe that $k_s^* \approx kp_s^* \approx k\hat{p}_s$ and the corresponding mean task completion times remain close for all arrival rates in the stability region.

VIII. Conclusion

In conclusion, this study investigates the (k, k) fork-join scheduling scheme in a system of parallel servers with two sets of heterogeneous servers, i.e., slow and fast servers. We show that the joint distribution of the stationary workload across k queues becomes asymptotically independent as the number of servers, n, grows and $k = o(n^{\frac{1}{4}})$. The limiting mean task completion time is analytically challenging to compute due to its integral expression. To address this, an upper bound on the limiting mean task completion time is derived, and the selection probability $\hat{p}_s$ that minimizes this bound is identified. Numerical experiments confirm that the selected probability provides near-optimal performance. These results offer valuable insights into workload distribution and performance optimization in heterogeneous server environments. Further research can explore additional system complexities and refine the proposed approach for enhanced performance.

Acknowledgment

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] R. Jinan, G. Gautam, P. Parag, and V. Aggarwal, "Asymptotic analysis of probabilistic scheduling for erasure-coded heterogeneous systems," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 50, no. 4, pp. 8–10, Apr. 2023.

[2] S. C. Leite and M. D. Fragoso, "Heavy traffic analysis of state-dependent fork-join queues with triggers," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst.*, 2008, pp. 488–494.

[3] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[4] M. Nguyen, S. Alesawi, N. Li, H. Che, and H. Jiang, "A black-box fork-join latency prediction model for data-intensive applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 1983–2000, Sep. 2020.

[5] M. Jeon et al., "Predictive parallelization: Taming tail latencies in web search," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2014, pp. 253–262.

[6] A. Badita, P. Parag, and V. Aggarwal, "Optimal server selection for straggler mitigation," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 709–721, Apr. 2020.

[7] A. Badita, P. Parag, and V. Aggarwal, "Single-forking of coded subtasks for straggler mitigation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 6, pp. 2413–2424, Dec. 2021.

[8] Y. Xiang, T. Lan, V. Aggarwal, and Y. R. Chen, "Joint latency and cost optimization for erasure-coded data center storage," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2443–2457, Aug. 2016.

[9] P. Parag, A. Bura, and J.-F. Chamberland, "Latency analysis for distributed storage," in *Proc. IEEE Conf. Comp. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[10] A. O. Al-Abbasi and V. Aggarwal, "Video streaming in distributed erasure-coded storage systems: Stall duration analysis," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1921–1932, Aug. 2018.

[11] A. Badita, P. Parag, and J.-F. Chamberland, "Latency analysis for distributed coded storage systems," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 4683–4698, Aug. 2019.

[12] V. Aggarwal and T. Lan, "Modeling and optimization of latency in erasure-coded storage systems," *Found. Trends Commun. Inf. Theory*, vol. 18, no. 3, pp. 380–525, 2021.

[13] R. Jinan, A. Badita, P. K. Sarvepalli, and P. Parag, "Latency optimal storage and scheduling of replicated fragments for memory constrained servers," *IEEE Trans. Inf. Theory*, vol. 68, no. 6, pp. 4135–4155, Jun. 2022.

[14] W. Winston, "Optimality of the shortest line discipline," *J. Appl. Probab.*, vol. 14, no. 1, pp. 181–189, Mar. 1977.

[15] R. R. Weber, "On the optimal assignment of customers to parallel servers," *J. Appl. Probab.*, vol. 15, no. 2, pp. 406–413, Jun. 1978.

[16] T. Hellemans and B. V. Houdt, "On the power-of-d-choices with least loaded server selection," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 2, no. 2, p. 27, Jun. 2018.

[17] U. Ayesta, T. Bodas, and I. M. Verloop, "On redundancy-d with cancel-on-start aka join-shortest-work (d)," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 2, pp. 24–26, Jan. 2019.

[18] S. Shneer and A. L. Stolyar, "Large-scale parallel server system with multi-component jobs," *Queueing Syst.*, vol. 98, nos. 1–2, pp. 21–48, Jun. 2021.

[19] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich, "Queueing system with selection of the shortest of two queues: An asymptotic approach," *Problemy Peredachi Informatsii*, vol. 32, no. 1, pp. 20–34, 1996.

[20] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, Oct. 2001.

[21] M. Bramson, Y. Lu, and B. Prabhakar, "Randomized load balancing with general service time distributions," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 275–286, Jun. 2010.

[22] M. Bramson, Y. Lu, and B. Prabhakar, "Asymptotic independence of queues under randomized load balancing," *Queueing Syst.*, vol. 71, no. 3, pp. 247–292, Jul. 2012.

[23] W. Wang, M. Harchol-Balter, H. Jiang, A. Scheller-Wolf, and R. Srikant, "Delay asymptotics and bounds for multitask parallel jobs," *Queueing Syst.*, vol. 91, nos. 3–4, pp. 207–239, Jan. 2019.

[24] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[25] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, M. Velednitsky, and S. Zbarsky, "Redundancy-d: The power of d choices for redundancy," *Oper. Res.*, vol. 65, no. 4, pp. 1078–1094, Aug. 2017.

[26] R. Jinan, A. Badita, T. Bodas, and P. Parag, "Load balancing policies without feedback using timed replicas," *Perform. Eval.*, vol. 162, Nov. 2023, Art. no. 102381.

[27] S. A. Banawan and N. M. Zeidat, "A comparative study of load sharing in heterogeneous multicomputer systems," in *Proc. 25th Annu. Simul. Symp.*, 1992, pp. 22–31.

[28] S. Bhambay and A. Mukhopadhyay, "Asymptotic optimality of speed-aware JSQ for heterogeneous service systems," *Perform. Eval.*, vols. 157–158, Oct. 2022, Art. no. 102320.

[29] K. Gardner, J. Abdul Jaleel, A. Wickeham, and S. Doroudi, "Scalable load balancing in the presence of heterogeneous servers," *Perform. Eval.*, vol. 145, Jan. 2021, Art. no. 102151.

[30] J. A. Jaleel, S. Doroudi, K. Gardner, and A. Wickeham, "A general 'power-of-d' dispatching framework for heterogeneous systems," *Queueing Syst.*, vol. 102, no. 3, pp. 431–480, Dec. 2022.

[31] M. van der Boor and C. Comte, "Load balancing in heterogeneous server clusters: Insights from a product-form queueing model," in *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQOS)*, Jun. 2021, pp. 1–10.

[32] L. Kleinrock, *Queueing Theory, Volume 1: Theory*. Hoboken, NJ, USA: Wiley, 1975.

[33] S. Chen et al., "When queueing meets coding: Optimal-latency data retrieving scheme in storage clouds," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2014, pp. 1042–1050.

[34] N. B. Shah, K. Lee, and K. Ramchandran, "When do redundant requests reduce latency?" *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 715–722, Feb. 2016.

[35] J. N. Daigle, *Queueing Theory With Applications to Packet Telecommunication*. Berlin, Germany: Springer, 2005.

Moonmoon Mohanty (Student Member, IEEE) received the B.Tech. degree in electronics and telecommunication engineering from the C. V. Raman College of Engineering, Odisha, in 2009, and the M.S. degree in electrical and computer engineering from Ajou University, South Korea, in 2017. She is currently pursuing the Ph.D. degree with the ECE Department, Indian Institute of Science, Bengaluru. Her research interests include stochastic modeling, scheduling in heterogeneous systems, energy sustainability and distributed computing, and data center protocols.

Gaurav Gautam received the B.Tech. degree in electronics and communication engineering from the Dronacharya College of Engineering, Gurugram, in 2015, and the M.Tech. degree in communication and networks from Indian Institute of Science, Bengaluru, in 2019. He is currently pursuing the Ph.D. degree in computer science with the University of Minnesota. His research interests lie in design and analysis of 5G networks and distributed systems.

Vaneet Aggarwal (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from Indian Institute of Technology, Kanpur, India, in 2005, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 2007 and 2010, respectively. He is currently an University Faculty Scholar and a Professor with Purdue University, West Lafayette, IN, USA, where he has been since January 2015. His current research interests are in foundations and applications of machine learning. Prior to this, he was a Senior Member of Technical Staff Research with the AT&T Laboratories-Research, NJ, USA (2010–2014). He received the Princeton University's Porter Ogden Jacobus Honorific Award in 2009 and Purdue University's Most Impactful Faculty Innovator Award in 2020. He received the IEEE Vehicular Technology Society Jack Neubauer Memorial Award in 2017 and the IEEE Communications Society William Bennett Prize in 2024.

Parimal Parag (Senior Member, IEEE) received the B.Tech. and M.Tech. degrees in electrical engineering from IIT Madras in 2004 and the Ph.D. degree in electrical engineering from Texas A&M University in 2011. He is currently an Associate Professor with the Electrical Communication Engineering Department, Indian Institute of Science, where he has been since December 2014. Prior to that, he was a Senior System Engineer in research and development with Assia Inc., Redwood (2011–2014). He is the co-author of the 2018 IEEE ISIT Student Best Paper and a recipient of the 2017 Early Career Award from the Science and Engineering Research Board. His research interests include the design and analysis of large scale distributed systems.