Optimal Rate-Matrix Pruning For Heterogeneous Systems

Zhisheng Zhao
Georgia Tech, Atlanta, USA

Debankur Mukherjee
Georgia Tech, Atlanta, USA

ABSTRACT

We consider large-scale load balancing systems where processing time distribution of tasks depend on both task and server types. We analyze the system in the asymptotic regime where the number of task and server types tend to infinity proportionally to each other. In such heterogeneous setting, popular policies like Join Fastest Idle Queue (JFIQ), Join Fastest Shortest Queue (JFSQ) are known to perform poorly and they even shrink the stability region. Moreover, to the best of our knowledge, in this setup, finding a scalable policy with provable performance guarantee has been an open question prior to this work. In this paper, we propose and analyze two asymptotically delay-optimal dynamic load balancing approaches: (a) one that efficiently reserves the processing capacity of each server for "good" tasks and route tasks under the Join Idle Queue policy; and (b) a speed-priority policy that increases the probability of servers processing tasks at a high speed. Introducing a novel analytical framework and using the mean-field method and stochastic coupling arguments, we prove that both policies above achieve asymptotic zero queueing, whereby the probability that a typical task is assigned to an idle server tends to 1 as the system scales.

1. INTRODUCTION

Advanced cloud computing platforms, such as AWS, Azure, and Google Cloud, handle millions of requests per second. Efficiently assigning tasks across servers using a load balancing algorithm is critical in such environments. While previous theoretical works have mostly focused on homogeneous load balancing models, where parallel servers process only one type of task at the same rate, real-world cloud computing platforms receive requests containing multiple classes of tasks with varying characteristics, such as accessing websites, training machine learning models, or backing up data. Additionally, with the expansion of these platforms, servers can be of different types (multi-skilled), as evident from AWS's website, which lists at least 9 server types with varying memory and bandwidth. Moreover, due to the storage capacity limitation at servers (a.k.a. *data locality*), a server can only have required resource files to execute only a (small) subset of tasks. Thus, it is natural to model such large-scale data center networks as heterogeneous parallel-server systems, where the time to process a task in a server depends on *both the type of the task and that of the server*.

For such general heterogeneous setting, popular routing policies, like Join Shortest Queue (JSQ), Join Idle Queue (JIQ), Join Fastest Shortest Queue (JFSQ) and the Join Fastest Idle Queue (JFIQ) are known to perform poorly. One reason is that they prioritize servers with the shortest or idle queue and might assign tasks to servers that cannot process at a relatively high speed with high probability, leading to inefficient server utilization.

In the seminal work [3], Stolyar proposed the MinDrift policy, which can be understood as the Gcμ-rule ([2, Section 4]) in the (output-queued) load balancing setup. It has been shown that MinDrift asymptotically minimizes the server workload in the conventional heavy traffic regime. However, implementing the MinDrift policy requires the dispatcher to know the total expected workload and service rate of every compatible server for the new task, which could result in a prohibitive communication burden when dealing with a large number N servers.

Model description. Consider a heterogeneous parallel-server system denoted by $G^N = (\mathcal{W}^N, \mathcal{V}^N, \boldsymbol{\lambda}^N, \mathcal{U}^N)$. In this system, $\mathcal{W}^N = \{1, ..., W(N)\}$ represents the set of dispatchers, where each dispatcher $i \in \mathcal{W}^N$ can only handle one type of task. Hence, the terms 'task-type' and 'dispatcher' will be used interchangeably. $\mathcal{V}^N = \{1, ..., N\}$ denotes the set of servers, where each server $j \in \mathcal{V}^N$ has a dedicated queue with infinite buffer capacity, and tasks are scheduled using the FCFS policy. The arrival process of tasks at the dispatcher $i \in \mathcal{W}^N$ is a Poisson process with rate $\lambda_i^N \in \boldsymbol{\lambda}^N = (\lambda_1^N, ..., \lambda_{W(N)}^N)$, independently of other processes. $\mathcal{U}^N = (\mu_{i,j}^N, i \in \mathcal{W}^N, j \in \mathcal{V}^N) \in \mathbb{R}_+^{W(N)\times N}$ represents a matrix of service rates, where the service time of a type $i \in \mathcal{W}^N$ task at server $j \in \mathcal{V}^N$ is exponentially distributed with mean $1/\mu_{i,j}^N$, if $\mu_{i,j}^N > 0$. Otherwise (i.e., when $\mu_{i,j}^N = 0$), by convention, the server j cannot process type i tasks. A server $j \in \mathcal{V}^N$ is considered '*compatible*' for type $i \in \mathcal{W}^N$ tasks if $\mu_{i,j}^N > 0$. It is assumed that tasks arriving at a dispatcher must be instantaneously and irrevocably assigned to one of the compatible servers. A schematic diagram of the system is shown in Figure 1

2. MAIN RESULTS

In order to perform asymptotic analysis of the sequence $\{G^N = (\mathcal{W}^N, \mathcal{V}^N, \boldsymbol{\lambda}^N, \mathcal{U}^N)\}_{N\in\mathbb{N}}$, we need to define the above

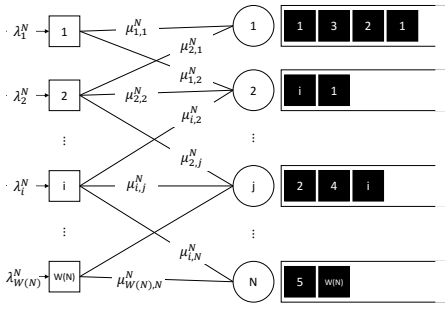

Figure 1: Heterogeneous Load Balancing System G^N

sequence consistently across different values of N. For that, we assume that $\{G^N\}_{N\in\mathbb{N}}$ has a nested structure, that is, for all $N \in \mathbb{N}$, $\mathcal{W}^N \subseteq \mathcal{W}^{N+1}$, $\mathcal{V}^N \subseteq \mathcal{V}^{N+1}$, and $\mu^N_{i,j} = \mu^{N+1}_{i,j}$, $\forall (i,j) \in \mathcal{W}^N \times \mathcal{V}^N$. Inspired by the concept of *graphon* [1, Chapter 7], we define two membership mapping functions $\phi_i : \mathbb{N} \to [0,1)$, $i = 1,2$, for dispatchers and servers, respectively. Next, we model the heterogeneity of the processing rates using these mapping functions and a function $f : [0,1)^2 \to \mathbb{R}_+$ such that for $(i,j) \in \mathcal{W}^N \times \mathcal{V}^N$, $\mu^N_{i,j} = f(\phi_1(i), \phi_2(j))$. With the function f, we can model the service rates between dispatchers and servers for all systems in the sequence in a consistent way, instead of writing a matrix $\mathcal{U}^N$ whose dimension explodes as $N \to \infty$. Hence, we formally introduce what we call an 'f-sequence' and use it in the rest of the analysis.

Definition 2.1 (f-sequence). *Given a function $f : [0,1]^2 \to \mathbb{R}_+$. A sequence $\{G^N\}_N = (\mathcal{W}^N, \mathcal{V}^N, \mathcal{U}^N, \boldsymbol{\lambda}^N)$ is an f-sequence, if for each $N \in \mathbb{N}$, $\mu^N_{i,j} = f(\phi_1(i), \phi_2(j))$, $\forall (i,j) \in \mathcal{W}^N \times \mathcal{V}^N$.*

The f-sequence is a general notion that encompasses the majority of classic queueing systems: (i) *homogeneous systems*, if f is a constant function; (ii) *multiclass many-server systems*, if $f(x,y) = f(x,0)$, $\forall y \in [0,1)$; (iii) *heterogeneous-server systems*, if $f(x,y) = f(0,y)$, $\forall x \in [0,1)$; (iv) *multiclass multiserver systems*, if f is a stepwise function. Furthermore, we make the following assumptions.

Assumption 2.2. (i) *(Arrival rate function) There exists an integrable function $\lambda : [0,1) \to \mathbb{R}_+$ with $\int_0^1 \lambda(x)dx = a > 0$ such that $\lambda(\phi_1(i)) = \lambda^N_i$, $\forall i \in \mathcal{W}^N, N \in \mathbb{N}$.*

(ii) *(Service rate function) The function f has finitely many discontinuity points on $[0,1)^2$, and there exists $\mu^o > 0$ such that for all $x \in [0,1)$, $|\{y \in [0,1) : f(x,y) \geq \mu^o\}| > 0$, where $|\cdot|$ is the Lebesgue measure.*

(iii) *(Regularity of membership map) For any subinterval $E \subseteq [0,1)$,*

$$\lim_{N\to\infty} \sum_{i\in\mathcal{W}^N} \frac{\mathbb{1}_{(\phi_1(i)\in E)}}{W(N)} = \lim_{N\to\infty} \sum_{j\in\mathcal{V}^N} \frac{\mathbb{1}_{(\phi_2(j)\in E)}}{N} = |E|.$$

(iv) $\lim_{N\to\infty} \frac{W(N)}{N} = \xi > 0$, *where ξ is a constant.*

For the asymptotic analysis, we consider the f-sequence in a subcritical regime defined as follows.

Definition 2.3 ($(\mathbf{w},\mathbf{v},\mathbf{p})$-Subcritical Regime). *The f-sequence $\{G^N\}_N$ is in the subcritical regime if the following is satisfied: There exist a pair of partitions $(\mathbf{w},\mathbf{v}) = (0 = w_0 < w_1 < \cdots < w_H = 1, 0 = v_0 < v_1 < \cdots < v_M = 1)$ of $[0,1]$ and a stochastic matrix $\mathbf{p} \in [0,1]^{H\times M}$ such that*

$$\rho_m(\mathbf{w},\mathbf{v},\mathbf{p}) \coloneqq \sum_{h\in[H]} \frac{p_{h,m}\lambda_h}{(v_m - v_{m-1})\mu^*_{h,m}} < 1, \quad m \in [M], \tag{2.1}$$

where, for each $h \in [H]$ and $m \in [M]$, $\lambda_h = \frac{1}{\xi}\int_{w_{h-1}}^{w_h} \lambda(x)dx$ and $\mu^*_{h,m} = \min_{(x,y)\in[w_{h-1},w_h)\times[v_{m-1},v_m)} f(x,y)$.

Having defined the above framework, we propose and analyze two asymptotically delay-optimal dynamic load balancing policies: Intelligent Capacity Reservation and Dispatching (ICRD) and Speed-Priority Dispatching (SPD).

ICRD. First, we prune the rate-matrix: (i) by the definition of $(\mathbf{w},\mathbf{v},\mathbf{p})$-subcritical regime, dividing dispatchers and servers into H classes and M types, respectively; (ii) reserving the capacity of a fraction $p_{h,m} + \varepsilon_{h,m}$ of type $m \in [M]$ servers for tasks from dispatchers of class $h \in [H]$, where $\varepsilon_{h,m}$ can be viewed as the capacity slack and should be determined carefully. After that, dispatchers assign tasks to servers according to the vanilla JIQ policy (that does not use processing rate information).

Theorem 2.4. *Consider the f-sequence $\{G^N\}_N$ in the subcritical regime. Through the ICRD approach, an arriving task will be assigned to an idle server with probability tending to 1 as $N \to \infty$.*

SPD. We divide dispatchers and servers into H classes and M types, respectively, based on the definition of $(\mathbf{w},\mathbf{v},\mathbf{p})$-subcritical regime. Suppose a task arrives at a dispatcher of class $h \in [H]$. It first selects a target server-type m^* with discrete distribution $\bar{p}_h = (p_{h,m})_{m\in[M]}$ and sends the new task to one of idle servers uniformly at random in $\mathcal{V}^N_{m^*}$, if any exist, and otherwise to one of the servers in $\mathcal{V}^N_{m^*}$, chosen uniformly at random.

Theorem 2.5. *Consider the f-sequence $\{G^N\}_N$ in the subcritical regime. Under the SPD approach and with the empty initial state, for any finite $T > 0$, all tasks are assigned to idle servers on $[0,T]$ with probability tending to 1 as $N \to \infty$.*

The key for implementing both approaches is to find the stochastic matrix $\mathbf{p}$, which can be done by solving the LP in (2.1). Also, both approaches can be implemented in a token-based fashion, inheriting scalability properties of the JIQ policy. The full version of the paper can be found in [4]

3. ACKNOWLEDGEMENTS

The work was supported by the NSF grant CIF-2113027.

4. REFERENCES

[1] L. Lovász. *Large Networks and Graph Limits.* Colloquium Publications, 2012.

[2] A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Oper. Res.*, 52(6):836–855, 2004.

[3] A. L. Stolyar. Optimal Routing in Output-Queued Flexible Server Systems. *Probab. Eng. Inf. Sci.*, 19:141–189, 2005.

[4] Z. Zhao and D. Mukherjee. Optimal rate-matrix pruning for large-scale heterogeneous systems. *arXiv:2306.00274*, 2023.