Distributed Speed Scaling in Large-Scale Service Systems

Daan Rutten
Georgia Institute of Technology
Atlanta, GA, USA

Martin Zubeldia
University of Minnesota
Minneapolis, MN, USA

Debankur Mukherjee
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

We consider a large-scale parallel-server loss system with an unknown arrival rate, where each server is able to adjust its processing speed. The objective is to minimize the system cost, which consists of a power cost to maintain the servers' processing speeds and a quality of service cost depending on the tasks' processing times, among others. We draw on ideas from stochastic approximation to design a novel speed scaling algorithm and prove that the servers' processing speeds converge to the globally asymptotically optimum value. Curiously, the algorithm is fully distributed and does not require *any* communication between servers. Apart from the algorithm design, a key contribution of our approach lies in demonstrating how concepts from the stochastic approximation literature can be leveraged to effectively tackle learning problems in large-scale, distributed systems. En route, we also analyze the performance of a fully heterogeneous parallel-server loss system, where each server has a distinct processing speed, which might be of independent interest.

CCS CONCEPTS

• **Theory of computation → Stochastic control and optimization**; *Distributed algorithms*; *Online learning algorithms*; • **Mathematics of computing → Markov processes**.

KEYWORDS

Load balancing; Rate-scaling; Distributed optimization

ACM Reference Format:
Daan Rutten, Martin Zubeldia, and Debankur Mukherjee. 2024. Distributed Speed Scaling in Large-Scale Service Systems. In *Abstracts of the 2024 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE Abstracts '24), June 10–14, 2024, Venice, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3652963.3655053

1 INTRODUCTION

Massive power consumption by data centers is a growing concern in recent years. If left unchecked, the electricity demand of data centers is predicted to grow up to 8% by 2030 [2]. This issue has sparked a renewed interest to not only consider performance metrics such as user-perceived sojourn time, but also power consumption [1, 4–6]. Processing tasks at a higher speed naturally reduces the user-perceived sojourn time, but it comes at a higher power consumption cost, which is typically cubic in the processing speed. Thus, the servers must strike a delicate balance in this tradeoff. The above can be formulated as an optimization problem, where a server must choose its service rate to minimize certain global objective function.

Consider a system consisting of $n \in \mathbb{N}$ parallel servers with unit buffers, where server $v \in [n]$ has processing rate $\mu_v^n > 0$. Tasks arrive as a Poisson process with rate λn, and have i.i.d. exponential sizes with unit mean. Upon arrival, tasks are either routed to an idle server chosen uniformly at random, if there are any, or dropped otherwise. The goal is for the servers to run at processing rates that minimize the cost function

$$r^n(\boldsymbol{\mu}^n) := \lambda g\left(\frac{1}{\mathbb{E}_{\boldsymbol{\mu}^n}[S^n]}\right) + \frac{1}{n}\sum_{v\in[n]} h(\mu_v^n) + q\Big(\mathbb{P}_{\boldsymbol{\mu}^n}\left(Q^n = n\right)\Big), \quad (1.1)$$

where S^n is the processing time of a non-dropped task in steady-state and $Q^n \in \{0, \dots, n\}$ is the number of busy servers in steady-state. Here, $\mathbb{E}_{\boldsymbol{\mu}^n}$ and $\mathbb{P}_{\boldsymbol{\mu}^n}$ denote expectation and probability, respectively, when the processing rates equal $\boldsymbol{\mu}^n$. Moreover, $g : \mathbb{R}_+ \to \mathbb{R}_+$ represents the cost due to the processing time of tasks, $h : \mathbb{R}_+ \to \mathbb{R}_+$ represents the cost of maintaining a specific processing speed and $q : [0, 1] \to \mathbb{R}_+$ represents the cost of dropping tasks.

Key challenges. The first challenge arises from the fact that an expression for the processing time for heterogeneous systems (when all servers have different service rates) is required to be able to characterize the cost function. To the best of our knowledge, such expressions under policies like the join-idle-queue are unknown till date. Next, in terms of the algorithm design, the 'gradient descent' cannot be blindly applied to our current setup for several reasons: *First,* as mentioned before, the lack of a simple expression for the processing time in a heterogeneous system. *Second,* Even if such expression is available, it would depend on the service rates of all servers in the system and hence cannot be computed at a single server. Similarly, *unbiased noisy estimates* of the processing time cannot be obtained. Another challenge is created by the distributed nature of the optimization problem. Since we do not allow any explicit communication between servers, we need to effectively use the sparse and implicit hints given by the actions of the load balancing policy to change a server's service rate. For example, under the join-the-idle-queue policy, if a server is idle and received a new task very fast, then it is likely that the system is overloaded and the server should increase its service rate.

Our contribution. In this paper, we design a (simple) distributed algorithm that updates the service rate of each server based only on the current service rate and idle times of the local server. In an appropriate asymptotic regime, we show that the cost of the server rates under our algorithm converge to the globally optimal cost. The keys to obtaining such a result are to characterize the processing time of an heterogeneous loss system, and to show that

SIGMETRICS/PERFORMANCE Abstracts '24, June 10–14, 2024, Venice, Italy.

ACM ISBN 979-8-4007-0624-0/24/06.
https://doi.org/10.1145/3652963.3655053

the expected service times become equal as the size of the system increases. A full version of the paper can be found in [7].

2 MAIN RESULTS

In the formulation of Equation (1.1), we assume that the functions g, h, and q satisfy the following assumptions.

Assumption 2.1. *There exist $0 < \mu_- < \lambda < \mu_+$ such that:*

(i) $g(x)$ and $h(x)$ are twice continuously differentiable for all $x \in \mathbb{R}_+$, $g''(x) \geq 0$ and $h''(x) \geq 0$ for all $x \in \mathbb{R}_+$ and $g''(x) \geq \sigma_g > 0$ and $h''(x) \geq \sigma_h > 0$ for all $x \in [\mu_-, \mu_+]$;
(ii) $g(x)$ is non-increasing and $h'(x)/x$ is non-decreasing for all $x \in \mathbb{R}_+$;
(iii) $\lambda g'(\lambda) + h'(\lambda) < 0$;
(iv) for all $0 \leq y \leq 1/\lambda$, $g'(\mu_-) + h'(\mu_-)(y + 1/\mu_-) \leq 0$ and $g'(\mu_+) + h'(\mu_+)(y + 1/\mu_+) \geq 0$;
(v) $\lim_{x\to 0} q(x) = q(0) = 0$.

Theorem 2.2. *Under Assumption 2.1, we have*

$$\lim_{n\to\infty} \inf_{\boldsymbol{\mu}^n \in \mathbb{R}_+^n} r^n(\boldsymbol{\mu}^n) = \lambda g(\mu^*) + h(\mu^*),$$

where $\mu^* = \arg\min_{\mu\geq 0}\{\lambda g(\mu) + h(\mu)\} > \lambda$.

To establish this result, we show that, for any average rate, a homogeneous rate vector minimizes the cost as $n \to \infty$. While the convexity of h easily implies that homogeneous processing rates minimize the second term in equation (1.1) for any given average rate, the fact that this also minimizes the first term crucially relies on the fact that the expected idle times of servers become equal.

2.1 Adaptive algorithm

Since servers do not know the arrival rate λ and cannot communicate with each other, we design an adaptive algorithm that learns the optimal processing rates in a completely distributed way. The algorithm is designed around the key insight that *the average idle time of each server becomes equal as $n \to \infty$, and thus servers may use their idle times as 'signals' to gauge how its own service rate compares to the average service rate, and adjust accordingly.* Under our algorithm, the service rate of each server is updated as follows:

$$\begin{aligned}\mu_v^{n,m}(t) = \mu_v^{n,m}(0) - \frac{1}{m}\int_0^t & g'(\mu_v^{n,m}(s)) \\ & + h'(\mu_v^{n,m}(s))\left(I_v^{n,m}(s) + \frac{1}{\mu_v^{n,m}(s)}\right) \mathrm{d}s,\end{aligned} \tag{2.1}$$

where $m > 0$ is a tunable hyperparameter, and $I_v^{n,m}(t)$ denotes the idle time of server $v \in [n]$. That is, if server v is idle at time t, then $I_v^{n,m}(t)$ is the length of time since the last service completion, and if server v is busy at time t, then $I_v^{n,m}(t)$ is the length of the last idle period.

Note that Equation (2.1) resembles a continuous-time version of a stochastic approximation algorithm with constant step size $1/m$. In order to analyze it, we consider the asymptotic regime where $m \to \infty$. As m gets larger, the service rates are updated at a slower pace, but with less randomness. Thus, if we accelerate time by a factor of m, we obtain a deterministic limiting trajectory with a constant learning rate. This is formalized in the following theorem.

Theorem 2.3. *If $\boldsymbol{\mu}^{n,m}(0) \to \boldsymbol{\mu}^n(0)$ weakly as $m \to \infty$, then the sequence of trajectories $\left\{(\boldsymbol{\mu}^{n,m}(mt))_{t\geq 0}\right\}_{m\in\mathbb{N}}$ is relatively compact with respect to the weak topology. Moreover, all limit points $(\boldsymbol{\mu}^n(t))_{t\geq 0}$ satisfy*

$$\begin{aligned}\mu_v^n(t) = \mu_v^n(0) - \int_0^t & \Big(g'(\mu_v^n(s)) \\ & + h'(\mu_v^n(s))\Big(\mathbb{E}_{\boldsymbol{\mu}^n(t)}\left[I_v^n(\infty)\right] + \frac{1}{\mu_v^n(s)}\Big)\Big) \mathrm{d}s,\end{aligned} \tag{2.2}$$

where $\mathbb{E}_{\boldsymbol{\mu}}[I_v^n(\infty)]$ denotes the expected idle time of server $v \in [n]$ in steady state in a system where the service rate vector is fixed at $\boldsymbol{\mu}$.

The proof of Theorem 2.3 is based on time-scale separation ideas introduced in [3]. Note that the limiting trajectories given by Equation (2.2) are roughly the expectation of the original trajectories given in Equation (2.1). In order to understand the limiting dynamics given by Equation (2.2), we need to get a handle on $\mathbb{E}_{\boldsymbol{\mu}^n}\left[I_v^n(\infty)\right]$, which is the expected idle time of a system where the service rates do not update and are fixed at μ_v for $v \in [n]$. As mentioned earlier, in this setting, we prove that the expected idle times of different servers become equal, as $n \to \infty$.

Theorem 2.4. *There exist constants $c^n(\boldsymbol{\mu}^n) \geq 0$ such that*

$$\max_{v\in[n]} \left|\mathbb{E}_{\boldsymbol{\mu}^n}\left[I_v^n(\infty)\right] - \frac{1 - c^n(\boldsymbol{\mu}^n)}{\lambda}\right| \to 0 \text{ as } n \to \infty. \tag{2.3}$$

Building upon this result, we obtain our main result that states that the processing rates indeed converge to the optimal, static processing rate μ^*.

Theorem 2.5. *Let μ^* be as in Theorem 2.2 and $\boldsymbol{\mu}^n(t)$ be as in Theorem 2.3. Then,*

$$\max_{v\in[n]} \left|\mu_v^n(t) - \mu^*\right| \to 0 \text{ as } n \to \infty \text{ and } t \to \infty, \tag{2.4}$$

where the limits are taken either jointly, or first as $n \to \infty$ and then as $t \to \infty$.

Theorem 2.5 states that $\mu^n(t)$, which is an asymptotic approximation of the original rate vector $\mu^{n,m}(t)$, converges to the optimal processing rate in time, when the number of servers n goes to infinity. This indicates that our algorithm indeed solves the optimization problem given in equation (1.1) when m and n are large.

3 ACKNOWLEDGEMENTS

This work was partially supported by the NSF grants CIF-2113027 and CPS-2240982.

REFERENCES

[1] 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 *Proc. SIGMETRICS'13*, pages 153–166, 2013.
[2] N. Jones. How to stop data centres from gobbling up the world's electricity. *Nature*, 561(7722):163–167, 2018.
[3] T. G. Kurtz. Averaging for martingale problems and stochastic approximation. In *Applied Stochastic Analysis*, pages 186–209. Springer, 1992.
[4] V. J. Maccio and D. G. Down. On optimal policies for energy-aware servers. *Perf. Eval.*, 90:36–52, 2015.
[5] D. Mukherjee, S. Dhara, S. C. Borst, and J. S. H. Van Leeuwaarden. Optimal service elasticity in large-scale distributed systems. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(1):25, 2017.
[6] D. Mukherjee and A. Stolyar. Join-Idle-Queue with service elasticity: Large-scale asymptotics of a non-monotone system. *Stoch. Syst.*, 9(4):338–358, 2019.
[7] D. Rutten, M. Zubeldia, D. Mukherjee. Distributed rate scaling in large-scale service systems. *arXiv:2306.02215*, 1–32, 2023.