Minimal-variance distributed scheduling under strict demands and deadlines

Yorie Nakahira†, Andres Ferragut‡, Adam Wierman†
†California Institute of Technology, E-mail: {ynakahir,adamw} @caltech.edu
‡Universidad ORT Uruguay, E-mail: ferragut@ort.edu.uy

ABSTRACT

Many modern schedulers can dynamically adjust their service capacity to match the incoming workload. At the same time, however, variability in service capacity often incurs operational and infrastructure costs. In this abstract, we characterize an optimal distributed algorithm that minimizes service capacity variability when scheduling jobs with deadlines. Specifically, we show that *Exact Scheduling* minimizes service capacity variance subject to strict demand and deadline requirements under stationary Poisson arrivals. Moreover, we show how close the performance of the optimal distributed algorithm is to that of the optimal centralized algorithm by deriving a competitive-ratio-like bound.

Keywords

Deadline scheduling, Service capacity control, Exact Scheduling, Online distributed algorithm

1. INTRODUCTION

Traditionally, the scheduling literature has assumed a static (fixed) service capacity. However, it is increasingly common for modern applications to have the ability to dynamically adjust their service capacity in order to match the current demand. For example, when using cloud computing services, one can modify the total computing capacity by changing the number of computing instances and their speeds. Power distribution networks can also adapt the energy supply to match the energy demand as it changes over time.

The ability to adapt service capacity dynamically gives rise to challenging new design questions. In particular, how to maintain *predictability* and *stability* of service capacity is of great importance in such applications since peaks and fluctuations often come with significant costs [1]. This trend is especially true for the examples of cloud computing and power distribution networks mentioned above. Cloud content providers prefer stable and predictable service capacity because on-demand contracts for compute instances (e.g., Amazon EC2 and Microsoft Azure) are typically more expensive than long-term contracts. Additionally, large fluctuations in service capacity induce unnecessary power consumption and infrastructure strain for computing equipment. The emerging load from electric vehicle charging stations also leads to similar challenges in power distribution networks. Charging stations require stability in power consumption because fluctuations and large peaks in power use may strain the grid infrastructure and result in a high peak charge for the station operators. The stations also prefer predictable power consumption because purchasing power in real time is typically more expensive than purchasing in advance.

Thus, in situations where service capacity can be dynamically adjusted, an important design goal is to minimize the costs associated with variability in the service capacity while maintaining high quality of service. In this paper, we study this problem by minimizing the *variance* of the service capacity in systems where jobs arrive with demand and deadline requests. Our focus on service capacity variance is motivated by applications such as cloud computing and power distribution networks, where contracts often explicitly depend on service capacity variability, e.g., if a charging station participates in the regulation market, then costs/payments depend explicitly on the variance of the total capacity [2, 3].

Although the literature on deadline scheduling is large and varied (see [4] and references therein), optimal algorithms are only known for certain niche cases. In particular, the problem of designing an *optimal* algorithm that minimizes service capacity variability while satisfying service quality constraints, i.e., meeting demands and deadlines, has remained open. Solving this problem is a challenging task due to the heterogeneous constraints (diversity in service requests) and the size of the state and decision space (the number of possible remaining job profile configurations and feasible control policies).

The goal of this work is to characterize the distributed scheduling algorithm that minimizes the variance of service capacity subject to service quality constraints, e.g., meeting job deadlines and satisfying job demands. Our focus is on *distributed* algorithms since implementing centralized algorithms is likely to be prohibitively slow and costly in large-scale service systems today. From cloud computing to power distribution networks, such systems are unlikely to be able to access global information about every job and server in the system when deciding the service rate of each job/server. Therefore, distributed algorithms are a necessity to enable large-scale implementation.

Our Contributions. In this work, we characterize the optimal distributed policy by using tools from optimization and control theory. Specifically, we show that *Exact Scheduling* is the optimal distributed algorithm, i.e., it minimizes the stationary variance of the service capacity among all distributed policies that strictly satisfy the service require-

Yorie Nakahira (ynakahir@caltech.edu) and Adam Wierman (adamw@caltech.edu) are with California Institute of Technology. Andres Ferragut (ferragut@ort.edu.uy) is with Universidad ORT Uruguay.

ments. Exact Scheduling is a classical algorithm that works by finishing job service *exactly* at their deadlines using a constant service rate [1, 4, 5]. Given that our results focus on distributed algorithms, we also study how the optimal distributed algorithm performs compared with the optimal centralized algorithm, which may provide better performance in theory but requires prohibitively expensive computation to find in practice. To answer this question, we derive a closed-form bound on the performance degradation due to using a distributed algorithm. The bound suggests that, when the sojourn time is a deterministic variable, Exact Scheduling attains the optimal trade-off between service capacity variance and total remaining demand variance achievable by any centralized algorithms.

2. PROBLEM FORMULATION

We consider a setting in which a service system dynamically adjusts its capacity in order to serve jobs that arrive randomly with heterogeneous service requirements. We use a continuous time model and use $t \in \mathbb{R}_+$ to denote a point in time. Each job, indexed by $k \in \mathcal{V} = \{1, 2, \ldots\}$, is characterized by a random arrival time a_k, a random service demand σ_k, and a random sojourn time $\tau_k (\geq \sigma_k)$.[1] In order to formalize the scheduler design procedure into an optimization problem, we introduce below the arrival profiles, the service profiles, the system dynamics, and the design objectives in detail.

Arrival profiles. We represent the set of jobs as a marked point process $\{(a_k; \sigma_k, \tau_k)\}_{k\in\mathcal{V}}$ in $\mathbb{R}_+ \times S$, where the arrival times $a_k \in \mathbb{R}_+$ are the set of points, and the service requirements $(\sigma_k, \tau_k) \in S$ are the set of marks. We assume that the marked point process is a stationary independently marked Poisson Point Process, which is defined by a locally finite non-null intensity measure Λ on $\mathbb{R}_+$ and a mark density measure $f(\sigma, \tau)$ on S [6]. This also implies that $\{(a_k; \sigma_k, \tau_k)\}_{k\in\mathcal{V}}$ is a Poisson Point Process on $\mathbb{R}_+ \times S$ with an intensity measure $f(\sigma, \tau)\Lambda$. Intuitively, $\int_A f(\sigma, \tau)\Lambda d\sigma d\tau$ is the average rate at which jobs with service requirement $(\sigma, \tau) \in A \subset S$ arrive. We additionally assume that S is bounded, and $S \subset \{(\sigma, \tau) : \tau \geq \sigma \geq 0\}$.[1]

Service profiles. The service system works on each job $k \in \mathcal{V}$ with a *service rate* $r_k(t) \geq 0$. The service rate can take non-zero values only when the job sojourns in the system, *i.e.*, $r_k(t) = 0$ for any $t \notin [a_k, a_k + \tau_k]$. To meet the service demand of job k, its service rate must satisfy

$$\int_{a_k}^{a_k+\tau_k} r_k(t)dt = \sigma_k, \qquad k \in \mathcal{V}. \tag{1}$$

Without loss of generality, $r_k(t) = 1$ is assumed to be the maximum rate: that is, $r_k(t)$ can take any values in $[0, 1]$, and $r_k < 1$ corresponds to throttling down service speed at the expense of prolonging job completion times. The above sojourn time and maximum rate constraints can be jointly written as

$$0 \leq r_k(t) \leq \mathbf{1}_{\{t\in[a_k,a_k+\tau_k)\}}, \tag{2}$$

where $\mathbf{1}_A$ denotes the indicator function for an event A. The resource consumption is defined by the (active) service capacity

$$P(t) = \sum_{k\in\mathcal{V}} r_k(t).$$

System dynamics. At each time $t \in \mathbb{R}_+$, job k has a remaining demand $x_k(t) = \sigma_k - \int_{a_k}^{t} r_k(h)dh$ and a remaining time $y_k(t) = a_k + \tau_k - t$. The set of remaining jobs in the system can be considered as a Point Process $\{(x_k(t), y_k(t))\}_k$ in $\mathbb{R}^2$, where x-axis is the remaining demand and y-axis is the remaining time (see Fig 1). At time t, each point(job) moves with velocity $-r_k(t)$ in x-axis and velocity -1 in y-axis.

Scheduling algorithms. An online scheduling algorithm decides the service rates in real-time without using the future job arrival information. For scalability, we additionally restrict our attention to the following form of *distributed* algorithms which decide the service rate of a job only using its own information:

$$r_k(t) = u(x_k(t), y_k(t)) \geq 0. \tag{3}$$

Here, $u : \mathbb{R}_+^2 \to \mathbb{R}_+$ is a deterministic function of the remaining demand $x_k(t)$ and the remaining time $y_k(t)$ of each job k at time t. Under any policy of the form (3), the set of jobs remaining in the system converges to a stationary distribution. This stationary distribution is a Spatial Poisson Point Process with intensity measure $\lambda(x, y)$ satisfying

$$0 = \frac{\partial}{\partial x}(\lambda(x, y)u(x, y)) + \frac{\partial}{\partial y}\lambda(x, y) + \Lambda f(x, y), \tag{4}$$

where x is the remaining demand and y is the remaining time. Because the remaining job distribution converges to a stationary distribution, $P(t)$ also converges to a stationary distribution. [2]

Design objectives. We consider minimizing service capacity variability under hard demand and deadline constraints:[2]

$$\underset{u:(1)(2)(3)(4)}{\text{minimize}} \qquad \mathrm{Var}(P), \tag{5}$$

where $\mathrm{Var}(P)$ is a functional of u and $\lambda(\sigma, \tau)$ satisfying (4). The optimization problem (5) has demand constraints as in (1) and service rate constraints as in (2), and the optimization variable u is constrained to have the form (3). It is worth noting that peaks in service rate amplifies the uncertainties in the future arrivals, which in turn produce large variance in $P(t) = \sum_{k\in\mathcal{V}} r_k(t) = \sum_{k\in\mathcal{V}} u(x_k(t), y_k(t))$.

3. OPTIMAL SCHEDULING

In order to minimize peaks, it is natural to use a flat service rate (Fig 1), which is achieved by the scheduling policy

$$u(x, y) = \begin{cases} \dfrac{x}{y}, & \text{if } y > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

This policy is known as Exact Scheduling and works by finishing all jobs *exactly* at their deadlines using a constant service rate. It is also highly scalable because it is distributed and asynchronous, and it does not require much

[1]The condition $\tau \geq \sigma$ requires a job to have a service demand σ that is no more than the maximum service that can be provided within its sojourn time τ.

[2]In this paper, we use the following notation: $\mathbb{E}[P]$ and $\mathrm{Var}(P)$ represent the stationary mean and variance of a stochastic process $\{P(t)\}_{t\in\mathbb{R}_+}$, while $\mathbb{E}[P(t)]$ and $\mathrm{Var}(P(t))$ represent the instantaneous mean and variance of $P(t)$ at time t.

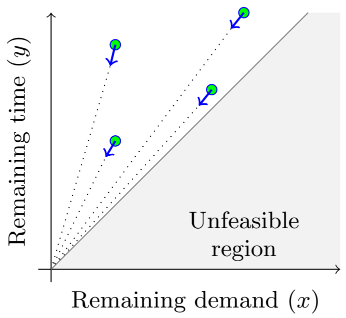

Figure 1: *Exact scheduling depicted in the space of remaining demand x and remaining time y.*

computation or memory use. Although existing literature has analyzed its performance in various settings [1, 4, 5, 7], it is not known whether and when the policy is optimal. Our main result is the following theorem, which shows that Exact Scheduling minimizes the variance of service capacity under stationary job arrivals and strict demand and deadline constraints among distributed algorithms.

THEOREM 1. *Exact Scheduling* (6) *is the optimal solution of* (5) *and achieves the optimal value*

$$\mathrm{Var}(P) = \Lambda \mathbb{E}\left[\frac{\sigma^2}{\tau}\right].$$

Theorem 1 shows the achievable performance improvement by performing distributed service capacity control. If no control is applied, *i.e.*, $r_k(t) = \mathbf{1}_{[a_k, a_k+\sigma_k]}(t)$, then $\mathbb{E}(P) = \mathrm{Var}(P) = \Lambda\mathbb{E}[\sigma]$. By performing a distributed service capacity control, the stationary variance can be reduced by $\Lambda\mathbb{E}\left[\sigma(\tau-\sigma)/\tau\right]$, where $\tau - \sigma$ is the slack time (the amount of time left at job completion if a job is served at its maximum service rate from its arrival time).

To enable large-scale implementation, we focus on distributed algorithms in this work. Given this focus, it is important to understand how much performance degradation is incurred due to restricting ourselves to distributed policies compared to centralized policies. Centralized policies are the class of algorithms of the form

$$r_k(t) = w(k, t, A_t), \qquad k \in \mathcal{V}, \tag{7}$$

where $A_t = \{(a_k, \sigma_k, x_k(t), y_k(t)) : a_k \le t\}$ is the set that contains the information of jobs arriving before t, and w is a deterministic mapping from (k, t, A_t) to $r_k(t)$.

LEMMA 1. *Under any centralized policy of the form* (7)*, the stationary variance of $P(t)$ is lower-bounded by*

$$\mathrm{Var}(P) \ge \frac{\Lambda^2\mathbb{E}[\sigma^2]^2}{4\mathrm{Var}(X)},$$

where $X(t)$ is the total amount of remaining service demand of jobs arriving before t.

Lemma 1 characterizes the trade-off between achieving a small variance of $X(t)$ and achieving a small variance of $P(t)$. An immediate consequence of Lemma 1 is a competitive-ratio-like bound that compares Exact Scheduling (6) and the best centralized algorithm having the same $\mathrm{Var}(X)$ as Exact Scheduling.

COROLLARY 1. *Let $\mathrm{Var}(P)$ be the stationary variance of $P(t)$ that is attained by Exact Scheduling* (6)*. Let $\mathrm{Var}(P^*)$ be the minimum stationary variance attainable by any centralized algorithm* (7) *with the same level of $\mathrm{Var}(X)$ as Exact Scheduling. Then, the following holds:*

$$\mathrm{Var}(P) \le \frac{\mathbb{E}[\sigma^2/\tau]\mathbb{E}[\sigma^2\tau]}{\mathbb{E}[\sigma^2]^2}\mathrm{Var}(P^*),$$

where all expectations on the right hand side are taken over the arrival distribution.

Corollary 1 bounds the ratio of $\mathrm{Var}(P)$ attained by Exact Scheduling (the optimal distributed algorithm) to $\mathrm{Var}(P^*)$ achievable by the best centralized algorithm having the same $\mathrm{Var}(X)$ as Exact Scheduling, so the rate of performance degradation due to restricting ourselves to distributed algorithms C_{ES} is bounded by:

$$C_{ES} \le \frac{\mathbb{E}[\sigma^2/\tau]\mathbb{E}[\sigma^2\tau]}{\mathbb{E}[\sigma^2]^2}, \tag{8}$$

which only depends on the characteristics of the incoming workload.

As an example, if the sojourn time τ is a deterministic random variable, (8) reduces to $C_{ES} = 1$ and therefore $\mathrm{Var}(P) = \mathrm{Var}(P^*)$, implying that Exact Scheduling achieves the optimal tradeoff, performing as well as the best centralized algorithm. One such case is when service demands and sojourn times are deterministic variables, and the service demand of each job equals its sojourn time (arrival times a are random). In this case, due to constraints (1) and (2), $r_k(t) = 1_{[a_k, a_k+\tau_k)}(t)$ is both the optimal centralized policy and the optimal distributed policy.

4. REFERENCES

[1] Giorgio C Buttazzo. *Hard real-time computing systems: predictable scheduling algorithms and applications*, volume 24. Springer Science & Business Media, 2011.

[2] Kathleen Spees and Lester B Lave. Demand response and electricity market efficiency. *The Electricity Journal*, 20(3):69–85, 2007.

[3] Mahdi Behrangrad. A review of demand side management business models in the electricity market. *Renewable and Sustainable Energy Reviews*, 47:270–283, 2015.

[4] Andres Ferragut, Fernando Paganini, and Adam Wierman. Controlling the variability of capacity allocations using service deferrals. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2(3):15, 2017.

[5] Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[6] François Baccelli and Bartlomiej Błaszczyszyn. *Stochastic Geometry and Wireless Networks, Volume I - Theory*. Now Publishers, 2009.

[7] John Lehoczky, Lui Sha, and Ye Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *Real Time Systems Symposium, 1989., Proceedings.*, pages 166–171. IEEE, 1989.