Multiple Server SRPT With Speed Scaling Is Competitive

Rahul Vaze, *Member, IEEE*, and Jayakrishnan Nair

***Abstract*— Can the popular shortest remaining processing time (SRPT) algorithm achieve a constant competitive ratio on multiple servers when server speeds are adjustable (speed scaling) with respect to the flow time plus energy consumption metric? This question has remained open for a while, where a negative result in the absence of speed scaling is well known. The main result of this paper is to show that multi-server SRPT with speed scaling can be constant competitive, with a competitive ratio that only depends on the power-usage function of the servers, but not on the number of jobs/servers or the job sizes (unlike when speed scaling is not allowed). When all job sizes are unity, we show that round-robin routing is optimal and can achieve the same competitive ratio as the best known algorithm for the single server problem. Finally, we show that a class of greedy dispatch policies, including policies that route to the least loaded or the shortest queue, do not admit a constant competitive ratio. When job arrivals are stochastic, with Poisson arrivals and i.i.d. job sizes, we show that random routing and a simple gated-static speed scaling algorithm achieves a constant competitive ratio.**

***Index Terms*— Speed scaling, Shortest Remaining Processing Time (SRPT), parallel servers, competitive ratio.**

I. Introduction

How to route and schedule jobs are two of the fundamental problems in multi-processor/multi-server settings, e.g. microprocessors with multiple cores. Microprocessors also have the flexibility of variable speed of operation, called *speed scaling*, where to operate at speed s, the power utilization is $P(s)$; typically, $P(s) = s^\alpha$, with $2 \leq \alpha \leq 3$. Speed scaling is also available in modern queuing systems where servers can operate at variable service rates with an appropriate cost function $P(.)$.

Increasing the speed of the server reduces the response times (completion minus arrival time) but incurs a larger energy cost. Thus, there is a natural tradeoff between the the *flow time* (defined as the sum of the response times across all jobs) and the total energy cost, and a natural objective is to minimize a linear combination of the flow time and total energy, called *flow time plus energy*.

In this paper, we consider the *online* problem of routing, scheduling, and speed scaling in a multi-server setting to minimize the flow time plus energy, where jobs arrive (are released) over time and decisions have to be made causally. On the arrival of a new job, a centralized controller needs to make a causal decision about which jobs to process on which server and at what speed, where preemption and migration is allowed. By migration, we mean that a job can be preempted on one server and restarted on another server later. The model, however, does not allow job splitting, i.e., a job can only be processed on a single server at any time.

For this problem, both the stochastic and worst case analysis is of interest. In the stochastic model, the input (job sizes and arrival instants) is assumed to follow a distribution, and performance guarantees in expectation are derived. In the worst case analysis, the input can be generated by an adversary, and the performance metric is the competitive ratio, that is defined as the maximum of the ratio of the cost of the online algorithm and the optimal offline algorithm (that knows the entire input sequence ahead of time).

A. Prior Work

1) Single Server: For a single server, it is known that Shortest Remaining Processing Time (SRPT) is an optimal scheduling policy, and the only decision with speed scaling is the optimal dynamic speed choice. There is a large body of work on speed scaling in the single server setting [1]–[8] both in the stochastic as well as worst case settings, where mostly $P(s) = s^\alpha$ is used, under various assumptions, e.g. bounded speed $s \in [0, S]$ [9], with and without deadlines [10]–[12], etc.

In the stochastic model, [6] showed that a simple fixed speed policy (called gated static) that depends only on the load/utilization and is independent of the current number of unfinished jobs/sizes has a constant multiplicative gap from the 'unknown' optimal policy. Further work in this direction can be found in [13], [14], where [14] derived the mean response time under the SRPT algorithm. For the worst case, there are many results [1]–[5], [7]–[12], [15]. A key result in this space was proved in [15], where an SRPT-based speed scaling algorithm is proved to be $(3+\epsilon)$-competitive algorithm for an arbitrary power function $P(\cdot)$. In [8], using essentially the same ideas as in [15], but with a more careful analysis, a slightly modified SRPT-based speed-scaling algorithm is shown to be $(2+\epsilon)$-competitive algorithm, also for an arbitrary power function.

In the worst case setting, when considering speed scaling, two classes of problems are studied: (i) unweighted and (ii) weighted, where in (i) the delay that each

Manuscript received February 25, 2019; revised December 14, 2019; accepted April 30, 2020; approved by IEEE/ACM Transactions on Networking Editor M. Andrews. Date of publication May 28, 2020; date of current version August 18, 2020. *(Corresponding author: Jayakrishnan Nair.)*

Rahul Vaze is with the School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai 400005, India.

Jayakrishnan Nair is with the Department of Electrical Engineering, Indian Institute of Technology (IIT) Bombay, Mumbai 400076, India (e-mail: jayakrishnan.nair@ee.iitb.ac.in).

Digital Object Identifier 10.1109/TNET.2020.2993142