Performance Analysis of Work-Conserving Schedulers for Minimizing Total Flow-Time with Phase Precedence

Yousi Zheng[1], Prasun Sinha[2], Ness B. Shroff[1],[2]
[1]Department of Electrical and Computer Engineering, [2]Department of Computer Science and Engineering
The Ohio State University, Columbus, OH 43210

***Abstract*— We consider the problem of minimizing the total flow-time of multiple jobs in a pool of multiple homogeneous machines, where the jobs arrive over time and have to be served with phase precedence. This is a common occurrence in job scheduling for the increasingly popular data center oriented systems, where jobs need to be processed through Map and Reduce procedures before leaving the system. For this problem, one can construct an arrival pattern such that no scheduler can achieve a constant competitive ratio. However, what we find is that by using a slightly weaker performance metric, which we call the efficiency ratio, we can provide bounds on the performance. We say that a scheduler achieves an efficiency ratio of γ when the flow-time incurred by that scheduler divided by the minimum flow-time achieved over all possible schedulers is less than or equal to γ almost surely, when the time slots or job arrivals go to infinity. Under some weak assumptions, we show a surprising property that all work-conserving schedulers for the flow-time problem with phase precedence have a constant efficiency ratio in both preemptive and non-preemptive scenarios. We provide numerical results to support our analysis.**

I. Introduction

In many widely used systems, the workload of arriving jobs can be divided into several phases. Some phases need to finish before other phases can be started. One common example of such a system is the MapReduce framework, which is designed to process massive amounts of data in a cluster of machines [1]. It is widely used for applications such as search indexing, distributed searching, web statistics generation, and data mining. MapReduce has two elemental phases: Map and Reduce. A critical consideration for the design of the scheduler is the *precedence* between the Map and Reduce tasks. For each job, the Map tasks need to be finished before starting any of its Reduce tasks [1], [2].

To minimize the total flow-time, it is well known that the Shortest-Remaining-Processing-Time (SRPT) scheduler is optimal in one machine case [3] or in multiple machines with work preserving malleable tasks[4]. The related scenarios are studied: scheduling malleable tasks [5], scheduling chain-structured tasks [6], and scheduling with release time constraint [7]. However, they focus on the complexity analysis, and they cannot be directly applied to the multi-phase setting. For the MapReduce framework, some scheduling solutions have also been proposed [8], [9], [10], [11], [2], but analytical bounds on performance have been derived only in some of these works [10], [11], [2]. However, rather than focusing directly on the flow-time, for deriving performance bounds, [10], [11] have considered a slightly different problem of minimizing total completion time and [2] has assumed speed-up of the machines.

In this paper, we directly analyze the performance of total delay (flow-time) in the system. In an attempt to minimize this, we introduce a new metric to analyze the performance of schedulers called *efficiency ratio*. Based on this new metric, we analyze and design schedulers that can provide performance guarantees.

The contributions of this paper are as follows:

- To directly analyze the total delay in the system, we propose a new metric to measure the performance of schedulers, which we call the *efficiency ratio*. (Section II)
- For bounded workload of Phase 2, we then show a surprising property that for the flow-time problem any work-conserving scheduler has a constant efficiency ratio in the preemptive as well as in the non-preemptive scenario (precise definitions provided in Section II). (Sections III and IV)
- We relax the assumption of Phase 2 from bounded workload to light-tailed distributed workload, and get the same properties in both preemptive and non-preemptive scenarios. (Section V)
- For the typical work-conserving schedulers, we show that the convergence of efficiency ratios through simulations. (Section VI)

II. System Model and Efficiency Ratio

A. System Model

Consider a pool of N machines. There are n jobs arriving into the system, and each machine can only process one job at a time. Time is slotted and each machine can run one unit of workload in each time slot. We assume that the distribution of job arrivals in each time slot is i.i.d., and the arrival rate is λ. Each job i contains multiple tasks, which can be classified into two phases: Phase 1 and Phase 2. Its tasks of Phase 1 need to be finished before starting any of its tasks of Phase 2. For each job i, Phase 1 brings M_i units of workload and Phase 2 brings R_i units of workload. Each

We gratefully acknowledge Dr. Zizhan Zheng for his helpful suggestions. The work has been partially supported through a grant from the Army Research Office MURI award W911NF-08-1-0238.

task of Phase 1 has 1 unit of workload[2], however, each task of Phase 2 can have multiple units of workload. Assume that $\{M_i\}$ are i.i.d. with expectation $\overline{M}$, and $\{R_i\}$ are i.i.d. with expectation $\overline{R}$. We assume that the traffic intensity $\rho < 1$, i.e., $\lambda < \frac{N}{\overline{M}+\overline{R}}$. Assume the moment generating function of workload of arriving jobs in a time slot has finite value in some neighborhood of 0. In time slot t for job i, $m_{i,t}$ and $r_{i,t}$ machines are scheduled for the tasks of Phase 1 and 2, respectively. We assume that job i contains K_i tasks of Phase 2, and the workload of the Phase 2 task k of job i is $R_i^{(k)}$. Thus, for any job i, $\sum_{k=1}^{K_i} R_i^{(k)} = R_i$. In time slot t for job i, $r_{i,t}^{(k)}$ machines are scheduled for the task k of Phase 2. As each task of Phase 2 may consist of multiple units of workload, it can be processed in either preemptive or non-preemptive fashion based on the type of scheduler.

Definition 1: A scheduler is called **preemptive** if the tasks of Phase 2 belonging to the same job can run in parallel on multiple machines, can be interrupted by any other task, and can be rescheduled to different machines in different time slots.

A scheduler is called **non-preemptive** if each task of Phase 2 can only be scheduled on one machine and, once started, it must keep running without any interruption.

The flow-time F_i of job i is equal to $f_i^{(r)} - a_i + 1$, where $f_i^{(r)}$ is the finish time of the reduce tasks and a_i is the arrival time of job i. The objective of the scheduler is to determine the assignment of jobs in each time slot, such that the cost of delaying the jobs or equivalently the flow-time is minimized.

For the preemptive scenario, the problem definition is as follows:

$$\begin{aligned}
\min_{m_{i,t},r_{i,t}} \quad & \sum_{i=1}^{n} \left(f_i^{(r)} - a_i + 1 \right) \\
s.t. \quad & \sum_{i=1}^{n} (m_{i,t} + r_{i,t}) \le N,\ r_{i,t} \ge 0,\ m_{i,t} \ge 0,\ \forall t, \\
& \sum_{t=a_i}^{f_i^{(m)}} m_{i,t} = M_i,\ \sum_{t=f_i^{(m)}+1}^{f_i^{(r)}} r_{i,t} = R_i,\ \forall i \in \{1,...,n\}.
\end{aligned} \tag{1}$$

In the non-preemptive scenario, the tasks of Phase 2 cannot be interrupted by other jobs. Once a task of Phase 2 begins execution on a machine, it has to keep running on that machine without interruption until all its workload is finished. Also, the optimization problem in this scenario is similar to Eq. (1), with additional constraints representing the non-preemptive nature, as shown below:

[2]For example, in the MapReduce framework as the Map tasks are independent and have small workload [9], such an assumption is valid.

$$\begin{aligned}
\min_{m_{i,t},r_{i,t}^{(k)}} \quad & \sum_{i=1}^{n} \left(f_i^{(r)} - a_i + 1 \right) \\
s.t. \quad & \sum_{i=1}^{n} (m_{i,t} + \sum_{k=1}^{K_i} r_{i,t}^{(k)}) \le N,\ \forall t, \\
& \sum_{t=a_i}^{f_i^{(m)}} m_{i,t} = M_i,\ m_{i,t} \ge 0,\ \forall i \in \{1,...,n\}, \\
& \sum_{t=f_i^{(m)}+1}^{f_i^{(r)}} r_{i,t}^{(k)} = R_i^{(k)},\ \forall i \in \{1,...,n\}, \forall k \in \{1,...,K_i\}, \\
& r_{i,t}^{(k)} = 0 \ or\ 1,\ r_{i,t}^{(k)} = 1 \text{ if } 0 < \sum_{s=0}^{t-1} r_{i,s}^{(k)} < R_i^{(k)}.
\end{aligned}$$

The proof of the following theorem is in the technical report [12].

Theorem 1: The scheduling problem (both preemptive and non-preemptive) is NP-complete in the strong sense.

B. *Efficiency Ratio*

The competitive ratio is often used as a measure of performance in a wide variety of scheduling problems. For our problem, the scheduling algorithm S has a competitive ratio of c, if for any total time T, any number of arrivals n in the time T, any arrival time a_i of each job i, any workload M_i and R_i of the Phase 1 and 2 tasks with respect to each arrival job i, the total flow-time $F^S(T, n, \{a_i, M_i, R_i; i = 1...n\})$ of scheduling algorithm S satisfies the following:

$$\frac{F^S(T, n, \{a_i, M_i, R_i; i = 1...n\})}{F^*(T, n, \{a_i, M_i, R_i; i = 1...n\})} \le c,$$

where

$$\begin{aligned}
& F^*(T, n, \{a_i, M_i, R_i; i = 1...n\}) \\
&= \min_S F^S(T, n, \{a_i, M_i, R_i; i = 1...n\}),
\end{aligned}$$

is the minimum flow-time of an optimal off-line algorithm.

For our problem, we can construct an arrival pattern such that no scheduler can achieve a constant competitive ratio c. The quick example is given in technical report [12]. We now introduce a slightly weaker notion of performance, called the **efficiency ratio**.

Definition 2: We say that the scheduling algorithm S has an efficiency ratio γ, if the total flow-time $F^S(T, n, \{a_i, M_i, R_i; i = 1...n\})$ of scheduling algorithm S satisfies the following:

$$\lim_{T\to\infty} \frac{F^S(T, n, \{a_i, M_i, R_i; i = 1...n\})}{F^*(T, n, \{a_i, M_i, R_i; i = 1...n\})} \le \gamma, \text{ w.p.1}$$

Later, we will show that for the quick example, a constant efficiency ratio γ can still exist (e.g., the non-preemptive scenario with light-tailed distributed Reduce workload in Section IV).

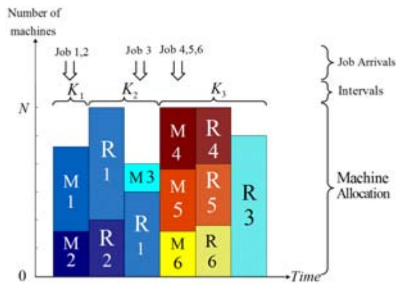


Fig. 1. A example schedule of a work-conserving scheduler

III. Work-Conserving Scheduler in the Preemptive Scenario

In this section, we analyze the performance of work-conserving scheduler in the preemptive scenario. In this section, we study the case in which the workload of Phase 2 of each job is bounded by a constant, i.e., there exists a constant R_{max}, s.t., $R_i \leq R_{max},\ \forall i$.

Consider the total scheduled number of machines over all the time slots. If all the N machines are scheduled in a time slot, we call this time slot a "complete" time slot; otherwise, we call this time slot a "incomplete" time slot. We define the j^{th} "interval" to be the interval between the $(j-1)^{st}$ incomplete time slot and the j^{th} incomplete time slot. (We define the first interval as the interval from the first time slot to the first incomplete time slot.) Thus, the last time slot of each interval is the only incomplete time slot in the interval. Let K_j be the length of the j^{th} interval, as shown in Fig. 1.

Firstly, we show that different moments of the length K_j of interval j is bounded by different constants, as shown in Lemma 1.

Lemma 1: If the scheduler is work-conserving, then for any given number H, there exists a constant B_H, such that $E[K_j^H] < B_H, \forall j$.

Proof: Since K_j is a positive random variable, then $E[K_j^H] \leq 1$ if $H \leq 0$. Thus, we only focus on the case that $H > 0$.

In the j^{th} interval, consider the t^{th} time slot in this interval. Assume that the total workload (including Map and Reduce tasks) of the arrivals in the t^{th} time slot of this interval is $W_{j,t}$. Also assume that the unavailable workload of Phase 2 at the end of the t^{th} time slot in this interval is $R_{j,t}$, i.e., the workload of Phase 2 whose corresponding tasks of Phase 1 are finished in the t^{th} time slot in this interval is $R_{j,t}$. The remaining workload of the Reduce function left by the previous interval is $R_{j,0}$. Then, the distribution of K_j will be as follows:

$$\begin{aligned} P(K_j = k) =& P(W_{j,1} + R_{j,0} - R_{j,1} \geq N, ... \\ & \sum_{t=1}^{k-1} W_{j,t} + R_{j,0} - R_{j,k-1} \geq (k-1)N, \\ & \sum_{t=1}^{k} W_{j,t} + R_{j,0} - R_{j,k} < kN). \end{aligned} \tag{2}$$

For each job i, $R_i \leq R_{max}$. And in the last slot of the previous interval, the maximum number of finished jobs of Phase 1 is $N-1$. Thus, $R_{j,0} \leq (N-1)R_{max}$. Now, let's consider the case of $k > 1$.

$$\begin{aligned} P(K_j = k) &\leq P\left(\sum_{t=1}^{k-1} W_{j,t} + R_{j,0} - R_{j,k-1} \geq (k-1)N\right) \\ &\leq P\left(\sum_{t=1}^{k-1} W_{j,t} + (N-1)R_{max} \geq (k-1)N\right) \\ &= P\left(\frac{\sum\limits_{t=1}^{k-1} W_{j,t}}{k-1} \geq N - \frac{(N-1)R_{max}}{k-1}\right). \end{aligned} \tag{3}$$

By the assumptions, the total (both Phase 1 and 2) arriving workload $W_{j,t}$ in each time slot is i.i.d., and the distribution does not depend on the interval index j or the index of time slot t in this interval. Then, for any interval j, we know that

$$P(K_j = k) \leq P\left(\frac{\sum\limits_{s=1}^{k-1} W_s}{k-1} \geq N - \frac{(N-1)R_{max}}{k-1}\right),$$

where $\{W_s\}$ is a sequence of i.i.d. random variables with the same distribution of $W_{j,t}$.

Since the traffic intensity $\rho < 1$, $E[W_s] = \lambda(\overline{M}+\overline{R}) < N$.

For any $\epsilon > 0$, there exists a k_0, such that for any $k > k_0$, $\frac{(N-1)R_{max}}{k-1} \leq \epsilon$. Then for any $k > k_0$, we have

$$P(K_j = k) \leq P\left(\frac{\sum\limits_{s=1}^{k-1} W_s}{k-1} \geq N - \epsilon\right),$$

We take $\epsilon < N - E[W_s]$ and corresponding k_0. For any $k > k_0 = 1 + \lceil \frac{(N-1)R_{max}}{\epsilon} \rceil$, by Cramer-Chernoff Theorem [13], we can achieve that

$$P\left(\frac{\sum\limits_{s=1}^{k-1} W_s}{k-1} \geq N - \epsilon\right) \leq e^{-kl(N-\epsilon)},$$

where the rate function $l(a)$ is shown as below:

$$l(a) = \sup_{\theta \geq 0} \left(\theta a - \log(E[e^{\theta W_s}])\right). \tag{4}$$

Since the moment generating function of workload in a time slot has finite value in some neighborhood around 0, for any $0 < \epsilon < N - \lambda(\overline{M} + \overline{R})$, $l(N-\epsilon) > 0$. Thus,

$$\begin{aligned}
E[K_j^H] &= \sum_{k=0}^{\infty} k^H P(K_j = k) \\
&= \sum_{k=0}^{k_0} k^H P(K_j = k) + \sum_{k=k_0+1}^{\infty} k^H P(K_j = k) \\
&\le k_0^H + \sum_{k=2}^{\infty} k^H e^{-kl(N-\epsilon)} \\
&= \left(1 + \lceil \frac{(N-1)R_{max}}{\epsilon} \rceil\right)^H + \sum_{k=2}^{\infty} k^H e^{-kl(N-\epsilon)}.
\end{aligned}$$

Since $l(N-\epsilon) > 0$, $\sum_{k=2}^{\infty} k^H e^{-kl(N-\epsilon)}$ is bounded.

Thus, given H, $E[K_j^H]$ is bounded by a constant B_H, for any j, where B_H is given as below.

$$B_H \triangleq \min_{\epsilon \in (0, N - \lambda(\overline{M}+\overline{R}))} \left\{ \begin{array}{l} \left(1 + \lceil \frac{(N-1)R_{max}}{\epsilon} \rceil\right)^H \\ + \sum_{k=2}^{\infty} k^H e^{-kl(N-\epsilon)} \end{array} \right\}. \quad (5)$$

■

Now, we directly obtain the expression for first and second moments of K_j.

Corollary 1: $E[K_j]$ is bounded by a constant B_1, $E[K_j^2]$ is bounded by a constant B_2, for any j. The expressions of B_1 and B_2 are shown as below, where the rate function $l(a)$ is defined in Eq. (4).

$$B_1 = \min_{\epsilon \in (0, N - \lambda(\overline{M}+\overline{R}))} \left\{ \begin{array}{l} 1 + \lceil \frac{(N-1)R_{max}}{\epsilon} \rceil \\ + \frac{2e^{l(N-\epsilon)} - 1}{e^{l(N-\epsilon)}(e^{l(N-\epsilon)} - 1)^2} \end{array} \right\},$$

$$B_2 = \min_{\epsilon \in (0, N - \lambda(\overline{M}+\overline{R}))} \left\{ \begin{array}{l} \left(1 + \lceil \frac{(N-1)R_{max}}{\epsilon} \rceil\right)^2 \\ + \frac{4e^{2l(N-\epsilon)} - 3e^{l(N-\epsilon)} + 1}{e^{l(N-\epsilon)}(e^{l(N-\epsilon)} - 1)^3} \end{array} \right\}.$$

Proof: By substituting $H = 2$ and $H = 1$ in Eq. (5), we directly achieve the corollary. ■

We add some dummy workload of Phase 2 in the beginning time slot of each interval, such that the remaining workload of the previous interval is equal to $(N-1)R_{max}$. The added workload are only scheduled when no real job can be scheduled. Then, in the new artificial system, the total flow-time of the real jobs doesn't change. However, the total number of intervals may decrease, because the added workload may merge some adjacent intervals.

Corollary 2: In the artificial system, the length of new intervals $\{\widetilde{K_j},\ j = 1, 2, ...\}$ also satisfies the same results of Lemma 1 and Corollary 1.

Proof: The proof is similar to the proofs of Lemma 1 and Corollary 1. ■

Remark 1: In the artificial system constructed in Corollary 2, the length of each interval has no effect on the length of other intervals, and it is only dependent on the job arrivals and the workload of each job in this interval. Based on our assumptions, the job arrivals in each time slot are i.i.d.. Also, the workload of jobs are also i.i.d.. Thus, the random variables $\{\widetilde{K_j},\ j = 1, 2, ...\}$ are i.i.d.

Theorem 2: For any work-conserving scheduling policy, the efficiency ratio is not greater than $\frac{B_2 + B_1^2}{\max\left\{2, \frac{1-p_0}{\lambda}, \frac{\rho}{\lambda}\right\} \max\left\{1, \frac{1}{N(1-\rho)}\right\}}$ w.p.1, when n goes to infinity. B_1 and B_2 can be achieved by Corollary 1, and p_0 is the probability that no job arrives in a time slot.

Proof: Consider the artificial system constructed in Corollary 2. Since we have more dummy workload of Phase 2 in the artificial system, the total flow-time $\widetilde{F}$ of the artificial system is not less than the total flow-time F of the work-conserving scheduling policy.

Note that, in each interval, all the job arrivals in this interval must finish Phase 1 in this interval, and all their workload of Phase 2 will finish before the end of the next interval. In other words, for all the arrivals in the j^{th} interval, Phase 1 is finished in K_j time slots, and Phase 2 is finished in $K_j + K_{j+1}$ time slots, as shown in Fig. 1. This is also true for the artificial system with dummy workload of Phase 2, i.e., all the arrivals in the $\widetilde{j^{th}}$ interval of the artificial system can be finished in $\widetilde{K_j} + \widetilde{K_{j+1}}$ time slots.

Let A_j be the number of arrivals in the j^{th} interval. The total flow-time F_j of the work-conserving scheduler for all the arrivals in the j^{th} interval is not greater than $A_j(K_j + K_{j+1})$. Similarly, the total flow-time $\widetilde{F_j}$ for all the arrivals in the j^{th} interval of the artificial system is not greater than $\widetilde{A_j}(\widetilde{K_j} + \widetilde{K_{j+1}})$, where $\widetilde{A_j}$ is the number of arrivals in the j^{th} interval of the artificial system.

Let F be the total flow-time of any work-conserving scheduler, and F^* be the minimized total flow-time. Assume that there are a total of m intervals for the n arrivals. Correspondingly, there are a total of $\widetilde{m}$ intervals in the artificial system. Also, based on the construction of the artificial system, $m \ge \widetilde{m}$ and $F = \widetilde{F} = \sum_{j=1}^{\widetilde{m}} \widetilde{F_j}$. $T \triangleq \sum_{j=1}^{m} K_j$ is equal to the finishing time of all the n jobs. For any scheduler, the total flow-time is at least $\hat{T}$, which is the number of time slots in which the number of scheduled machines is at least 1. Thus, the total flow-time F_j^* of the optimal scheduler should be at least $\hat{T}$.

For any scheduling policy, each job needs at least 1 time slot to schedule its tasks of Phase 1 and 1 time slot to schedule its tasks of Phase 2. Thus, the lower bound of the each job's flow-time is 2, assuming that the Map and Reduce workload of any job is at least 1 unit each. Thus, for the optimal scheduling method, its total flow-time F_j^* should not be less than $2A_j$.

Since $E(M_i + R_i) > 0$, $\rho < 1$, then $\lim_{n \to \infty} m = \infty$ and

$\lim_{n\to\infty} \widetilde{m} = \infty$.

$$\lim_{n\to\infty} \frac{F}{F^*} = \lim_{n\to\infty} \frac{\widetilde{F}}{F^*} \le \lim_{\widetilde{m}\to\infty} \frac{1}{\hat{T}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}(\widetilde{K_j} + \widetilde{K_{j+1}})$$

$$\le \frac{\lim_{\widetilde{m}\to\infty} \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}\widetilde{K_j} + \lim_{\widetilde{m}\to\infty} \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}\widetilde{K_{j+1}}}{\lim_{m\to\infty} \hat{T}/m}.$$

Since $\{\widetilde{K_j},\ j = 1, 2, ...\}$ are i.i.d. distributed random variables, then $\{\widetilde{A_j}\widetilde{K_j},\ j = 1, 2, ...\}$ are also i.i.d. Based on Strong Law of Large Number (SLLN), we can achieve that

$$\lim_{\widetilde{m}\to\infty} \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}\widetilde{K_j} \overset{\text{w.p.1}}{=\!=} E\left[\widetilde{A_j}\widetilde{K_j}\right]. \tag{6}$$

For $\{\widetilde{A_j}\widetilde{K_{j+1}},\ j = 1, 2, ...\}$, they are identically distributed, but not independent. However, all the odd term are independent, and all the even term are independent. Based on SLLN, we can achieve that

$$\lim_{\widetilde{m}\to\infty} \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}\widetilde{K_{j+1}} \overset{\text{w.p.1}}{=\!=} E[\widetilde{A_j}\widetilde{K_{j+1}}]. \tag{7}$$

Based on Eqs. 6 and 7, we can achieve that

$$\begin{aligned} &\lim_{\widetilde{m}\to\infty} \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}\widetilde{K_j} + \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}\widetilde{K_{j+1}} \\ &\overset{\text{w.p.1}}{=\!=} E[\widetilde{A_j}\widetilde{K_j}] + E[\widetilde{A_j}\widetilde{K_{j+1}}] \\ &= E[E[\widetilde{A_j}\widetilde{K_j}|\widetilde{K_j}]] + E[\widetilde{A_j}]E[\widetilde{K_{j+1}}] \\ &= E[\widetilde{K_j}E[\widetilde{A_j}|\widetilde{K_j}]] + E[E[\widetilde{A_j}|\widetilde{K_j}]]E[\widetilde{K_{j+1}}] \\ &= \lambda(E[\widetilde{K_j}^2] + E[\widetilde{K_j}]^2). \end{aligned}$$

For these n jobs, there are m incomplete time slots before the time T. For each incomplete time slot, there are at most $N-1$ machines are assigned. Then, $N\sum_{j=1}^{m}(K_j - 1) + (N-1)m \ge \sum_{s\in\{\text{Arrivals before } T\}} W_s$. Thus,

$$\left(N - \frac{\sum_{s\in\{\text{Arrivals before } T\}} W_s}{T}\right) \frac{\sum_{j=1}^{m} K_j}{m} \ge 1. \tag{8}$$

Since the workload W_s of each job is i.i.d., by SLLN, we have $\left(N - \lambda(\overline{M} + \overline{R})\right) \lim_{m\to\infty} \frac{\sum_{j=1}^{m} K_j}{m} \ge 1$ w.p.1, i.e., $\lim_{m\to\infty} \frac{\sum_{j=1}^{m} K_j}{m} \ge \frac{1}{N(1-\rho)}$ w.p.1. Thus,

$$\lim_{m\to\infty} \frac{T}{m} = \lim_{m\to\infty} \frac{\sum_{j=1}^{m} K_j}{m} \ge \max\left\{1, \frac{1}{N(1-\rho)}\right\} \text{ w.p.1.}$$

For any work-conserving scheduler, $\hat{T}$ is not less than the total number of time slots with at least 1 arrival, we have

$$\lim_{m\to\infty} \frac{\hat{T}}{m} \ge (1 - p_0) \lim_{m\to\infty} \frac{T}{m} \text{ w. p. 1.}$$

At the same time, for each time slot in which the number of scheduled machines is greater than 0, there are at most N machines are scheduled. Then, we can get $N\hat{T} \ge \sum_{i=1}^{n}(M_i + R_i)$. Then, $\lim_{m\to\infty} \frac{\hat{T}}{m} \ge \frac{\lambda(\overline{M}+\overline{R})}{N} \lim_{m\to\infty} \frac{T}{m}$ w.p.1. $= \rho \lim_{m\to\infty} \frac{T}{m}$. Thus,

$$\begin{aligned} &\lim_{n\to\infty} \frac{F}{F^*} \le \frac{\lambda(E[\widetilde{K_j}^2] + E[\widetilde{K_j}]^2)}{\max\{\rho, 1-p_0\} \max\left\{1, \frac{1}{N(1-\rho)}\right\}} \text{ w.p.1.} \\ &\le \frac{\lambda(B_2 + B_1^2)}{\max\{\rho, 1-p_0\} \max\left\{1, \frac{1}{N(1-\rho)}\right\}} \text{ w.p.1.} \end{aligned} \tag{9}$$

Similarly,

$$\lim_{n\to\infty} \frac{F}{F^*} = \lim_{n\to\infty} \left(\sum_{j=1}^{m} F_j \Big/ \sum_{j=1}^{m} F_j^*\right)$$

$$\le \frac{\lim_{\widetilde{m}\to\infty} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}(\widetilde{K_j} + \widetilde{K_{j+1}})}{\lim_{m\to\infty} \sum_{j=1}^{m} 2A_j} \le \frac{\lim_{\widetilde{m}\to\infty} \frac{1}{\widetilde{m}} \sum_{j=1}^{\widetilde{m}} \widetilde{A_j}(\widetilde{K_j} + \widetilde{K_{j+1}})}{\lim_{m\to\infty} \frac{1}{m} \sum_{j=1}^{m} 2A_j}$$

With Strong Law of Large Number (SLLN), we can get

$$\lim_{m\to\infty} \frac{\sum_{j=1}^{m} A_j}{m} = \lim_{m\to\infty} \left(\sum_{j=1}^{m} A_j / \sum_{j=1}^{m} K_j\right) \lim_{m\to\infty} \left(\frac{1}{m} \sum_{j=1}^{m} K_j\right)$$

$$= \lambda \lim_{m\to\infty} \left(\frac{1}{m} \sum_{j=1}^{m} K_j\right) \text{ w. p. 1.} \ge \lambda \max\left\{1, \frac{1}{N(1-\rho)}\right\}$$

Thus,

$$\lim_{n\to\infty} \frac{F}{F^*} \le \frac{\lambda(B_2 + B_1^2)}{2\lambda \max\left\{1, \frac{1}{N(1-\rho)}\right\}} \text{ w.p.1.} \tag{10}$$

Based on the Eqs. 9 and 10, we can achieve that the efficiency ratio of any work-conserving scheduling policy is $\frac{B_2 + B_1^2}{\max\left\{2, \frac{1-p_0}{\lambda}, \frac{\rho}{\lambda}\right\} \max\left\{1, \frac{1}{N(1-\rho)}\right\}}$. ■

IV. Work-Conserving Scheduler in the Non-Preemptive Scenario

The definitions of B_1, B_2, p_0, F, $\widetilde{F}$, F^*, F_j, $\widetilde{F_j}$, K_j, $\widetilde{K_j}$, A_j, $\widetilde{A_j}$, $W_{j,t}$, and $R_{j,t}$ in this section are same as Section III.

Lemma 2: Lemma 1, Corollary 1, Corollary 2 in Section III are also valid in the non-preemptive scenario.

Proof: Different from preemptive scenario in Section III, the constitution of $R_{j,t}$ are different, because it contains two parts. First, it contains the unavailable workload of Phase 2 which are just released by Phase 1 which are finished in the last time slot of each interval. Second, it also

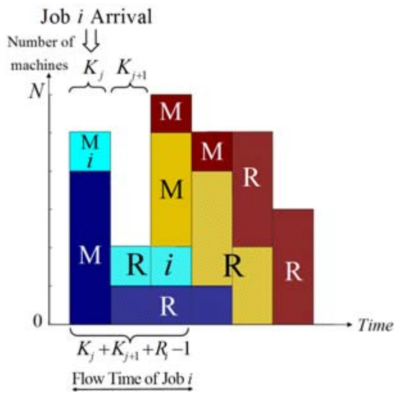


Fig. 2. The arrivals in the j^{th} interval finish Phase 1 functions in K_j time slots and Phase 2 in $K_j + K_{j+1} + R_{max} - 1$ time slots

contains the workload of Phase 2 which cannot be processed in this time slot, even if there are idle machines, because the tasks of Phase 2 are non-preemptive.

However, we can also achieve that the remaining workload of Phase 2 from previous interval satisfies $R_{j,0} \leq (N-1)R_{max}$, because the total number of remaining unfinished tasks of Phase 2 and new available tasks of Phase 2 in this time slot is less than N. All the following steps are same. ■

Theorem 3: In the non-preemptive scenario, any work-conserving scheduling policy has an efficiency ratio of $\frac{B_2+B_1^2+B_1(\overline{R}-1)}{\max\left\{2,\frac{1-p_0}{\lambda},\frac{\rho}{\lambda}\right\}\max\left\{1,\frac{1}{N(1-\rho)}\right\}}$.

Proof: In the non-preemptive scenario, in each interval, all the workload of Phase 1 of job arrivals in this interval is also finished in this interval, and all their tasks of Phase 2 will be started before the end of the next interval. In other words, for the job i in the j^{th} interval, the tasks of Phase 1 are finished in K_j time slots, and the tasks of Phase 2 are finished in $K_j+K_{j+1}+R_i-1$ time slots, as shown in Fig. 2. And this is also true for the artificial system with dummy workload of Phase 2, i.e., all the arrivals in the j^{th} interval of the artificial system can be finished in $\widetilde{K_j}+\widetilde{K_{j+1}}+R_i-1$. So, F_i in the j^{th} interval is less than $A_j(K_j+K_{j+1}+R_i-1)$. Similarly, $\widetilde{F_i}$ in the j^{th} interval of the artificial system is less than $\widetilde{A_j}(\widetilde{K_j}+\widetilde{K_{j+1}}+R_i-1)$. Using the same technique as in the proof of Theorem 2 we can show that,

$$\lim_{n\to\infty}\frac{F}{F^*} \leq \lim_{m\to\infty}\frac{\sum_{j=1}^{m}\widetilde{A_j}(\widetilde{K_j}+\widetilde{K_{j+1}})+\sum_{i=1}^{n}(R_i-1)}{\sum_{j=1}^{m}(K_j-S_j)}$$
$$=\frac{\lim_{m\to\infty}\frac{1}{m}\sum_{j=1}^{m}\widetilde{A_j}(\widetilde{K_j}+\widetilde{K_{j+1}})+\lim_{m\to\infty}\frac{n}{m}\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}(R_i-1)}{\lim_{m\to\infty}\frac{1}{m}\sum_{j=1}^{m}(K_j-S_j)}$$
$$\leq\frac{\lambda[B_2+B_1^2+B_1(\overline{R}-1)]}{\max\{\rho,1-p_0\}\max\left\{1,\frac{1}{N(1-\rho)}\right\}}\text{ w.p.1,} \tag{11}$$

and

$$\lim_{n\to\infty}\frac{F}{F^*}\leq\frac{\lambda[B_2+B_1^2+B_1(\overline{R}-1)]}{2\lambda\max\left\{1,\frac{1}{N(1-\rho)}\right\}}\text{ w.p.1.} \tag{12}$$

Thus, based on the Eqs. 11 and 12, we obtain that any work-conserving scheduler has the efficiency ratio of $\frac{B_2+B_1^2+B_1(\overline{R}-1)}{\max\left\{2,\frac{1-p_0}{\lambda},\frac{\rho}{\lambda}\right\}\max\left\{1,\frac{1}{N(1-\rho)}\right\}}$. ■

V. Light-tailed Distributed Workload of Phase 2

In this part, we assume the workload R of Phase 2 are light-tailed distributed, i.e., $\exists r_0$, such that

$$P(R\geq r)\leq\alpha\exp(-\beta r),\ \forall r\geq r_0,$$

where $\alpha,\ \beta>0$ are two constants. We use the same definitions of $W_{j,t}$, $R_{j,t}$ and K_j as Section III.

Lemma 3: If the scheduling algorithm is work-conserving, then for any given number H, there exists a constant B_H, such that $E[K_j^H]<B_H$, $\forall j$.

Proof: Similarly to Eqs. 2 and 3, we can achieve

$$P(K_j=k)\leq P\left(\sum_{t=1}^{k-1}W_{j,t}+R_{j,0}\geq(k-1)N\right).$$

Let R' be the maximal remaining workload of jobs in the previous interval, then

$$P(K_j=k)\leq P\left(\sum_{t=1}^{k-1}W_{j,t}+(N-1)R'\geq(k-1)N\right)$$
$$=\sum_{r=0}^{\infty}\left[P\left(\sum_{t=1}^{k-1}W_{j,t}+(N-1)R'\geq(k-1)N\middle|R'=r\right)\right.$$
$$\left.P(R'=r)\right]$$
$$=\sum_{r=0}^{\infty}\left[P\left(\frac{\sum_{t=1}^{k-1}W_{j,t}}{k-1}\geq N-\frac{(N-1)r}{k-1}\right)P(R'=r)\right]. \tag{13}$$

Thus, we can achieve

$$E[K_j^H]=\sum_{k=0}^{\infty}k^HP(K_j=k)$$
$$=\sum_{k=0}^{\infty}\left\{k^H\sum_{r=0}^{\infty}\left[P(\frac{\sum_{t=1}^{k-1}W_{j,t}}{k-1}\geq N-\frac{(N-1)r}{k-1})P(R'=r)\right]\right\}$$
$$=\sum_{r=0}^{\infty}\left\{P(R'=r)\sum_{k=0}^{\infty}\left[k^HP(\frac{\sum_{t=1}^{k-1}W_{j,t}}{k-1}\geq N-\frac{(N-1)r}{k-1})\right]\right\}.$$

The next steps are similar to the proof of Lemma 1, we skip the details and directly show the following result. For any $0<\epsilon<N-\lambda(\overline{M}+\overline{R})$, we know that $l(N-\epsilon)>0$. Thus, we can achieve

$$
\begin{aligned}
&E[K_j^H]\\
\le&\sum_{r=0}^{\infty}\Big\{P(R'=r)\Big[\big(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\big)^H+\sum_{k=2}^{\infty}k^He^{-kl(N-\epsilon)}\Big]\Big\}\\
=&\sum_{k=2}^{\infty}k^He^{-kl(N-\epsilon)}+\sum_{r=0}^{\infty}\Big\{P(R'=r)\big(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\big)^H\Big\}\\
\le&\sum_{k=2}^{\infty}k^He^{-kl(N-\epsilon)}+\sum_{r=0}^{\infty}\Big\{P(R'\ge r)\big(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\big)^H\Big\}\\
\le&\sum_{k=2}^{\infty}k^He^{-kl(N-\epsilon)}+\sum_{r=0}^{\infty}\Big\{P(R\ge r)\big(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\big)^H\Big\}\\
\le&\sum_{k=2}^{\infty}k^He^{-kl(N-\epsilon)}+\sum_{r=0}^{r_0-1}\Big\{\big(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\big)^H\Big\}\\
&+\sum_{r=r_0}^{\infty}\Big\{\alpha e^{-\beta r}\big(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\big)^H\Big\}.
\end{aligned}
$$

Thus, given H, $E[K_j^H]$ is bounded by a constant B_H, for any j, where B_H is given as below.

$$
B_H\triangleq\min_{\epsilon\in(0,N-\lambda(\overline{M}+\overline{R}))}\left\{\begin{array}{l}\sum_{k=2}^{\infty}k^He^{-kl(N-\epsilon)}+\\ \sum_{r=0}^{r_0-1}\left(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\right)^H+\\ \sum_{r=r_0}^{\infty}\alpha e^{-\beta r}\left(1+\lceil\frac{(N-1)r}{\epsilon}\rceil\right)^H\end{array}\right\}. \tag{14}
$$

■

Corollary 3: $E[K_j]$ is bounded by a constant B_1', $E[K_j^2]$ is bounded by a constant B_2', for any j. The expressions of B_1' and B_2' are shown as below, where the rate function $l(a)$ is defined in Eq. (4).

$$
B_1'=\min_{\epsilon\in(0,N-\lambda(\overline{M}+\overline{R}))}\left\{\begin{array}{l}2r_0+\frac{(N-1)r_0(r_0-1)}{2\epsilon}+\\ \frac{2\alpha}{e^\beta-1}+\frac{\alpha(N-1)}{4\epsilon}\text{csch}^2(\frac{\beta}{2})\\ +2\alpha+\frac{2e^{l(N-\epsilon)}-1}{e^{l(N-\epsilon)}(e^{l(N-\epsilon)}-1)^2}\end{array}\right\},
$$

$$
B_2'=\min_{\epsilon\in(0,N-\lambda(\overline{M}+\overline{R}))}\left\{\begin{array}{l}4r_0+\frac{2(N-1)r_0(r_0-1)}{\epsilon}+4\alpha\\ +\frac{(N-1)^2(r_0-1)r_0(2r_0-1)}{6\epsilon^2}\\ +\frac{\alpha(N-1)^2}{8\epsilon^2}\sinh(\beta)\text{csch}^4(\frac{\beta}{2})\\ +\frac{4\alpha}{e^\beta-1}+\frac{\alpha(N-1)}{\epsilon}\text{csch}^2(\frac{\beta}{2})\\ +\frac{4e^{2l(N-\epsilon)}-3e^{l(N-\epsilon)}+1}{e^{l(N-\epsilon)}(e^{l(N-\epsilon)}-1)^3}\end{array}\right\}.
$$

Proof: By substituting $H=2$ and $H=1$ in Eq. (14), we directly achieve the result by similar steps of Corollary 1. ■

Remark 2: Following the same proof steps, we can get the same results as Theorem 2 and Theorem 3. The difference is that we need use B_1' and B_2' instead of B_1 and B_2. Also, following the same steps, we can directly extend the results to more than 2 phases.

VI. Simulation Results

A. Simulation Setting

We evaluate the convergence of efficiency ratio for both preemptive and non-preemptive scenarios. We consider a data center with $N=100$ machines, and choose Poisson process with arrival rate $\lambda=2$ jobs per time slot as the job arrival process. We choose uniform distribution and exponential distribution as examples of bounded workload and light-tailed distributed workload, respectively. For short, we use $Exp(\mu)$ to represent an exponential distribution with mean μ, and use $U[a,b]$ to represent a uniform distribution on $\{a,a+1,...,b-1,b\}$. We choose the total time slots to be $T=800$, and the number of tasks in each job is up to 10.

We choose 4 typical schedulers to evaluate the convergence of efficiency ratio:

The **SRPT** scheduler: Jobs with smaller unfinished workload are always scheduled first. Since the exact SRPT scheduler is not always feasible in the problem with phase precedence, the SRPT scheduler here only schedules the available part of the workload.

The **FIFO** scheduler: It is the default scheduler in many systems, like Hadoop. All the jobs are scheduled in their order of arrival.

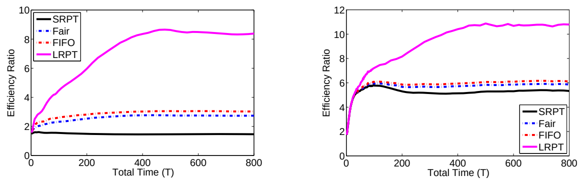
The **Fair** scheduler: It is also a widely used scheduler. The assignment of machines are scheduled to all the waiting jobs in a fair manner. However, if some jobs need fewer machines than others in each time slot, then the remaining machines are scheduled to the other jobs, to avoid resource wastage and to keep the scheduler work-conserving.

The **LRPT** scheduler: Jobs with larger unfinished workload are always scheduled first. Roughly speaking, the performance of this scheduler represents in a sense how poorly even some work-conserving schedulers can perform.

B. Convergence of Efficiency Ratio

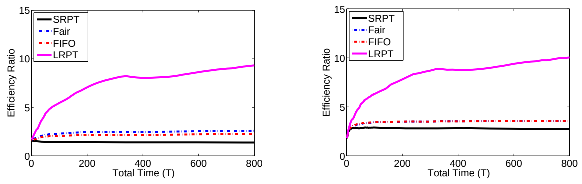
In the simulations, the efficiency ratio of a scheduler is obtained by the total flow-time of the scheduler over the lower bound of the total flow-time in T time slots. Thus, the real efficiency ratio should be smaller than the efficiency ratio given in the simulations.

First, we evaluate the exponentially distributed workload. We choose the workload distribution of Phase 1 for each job as $Exp(5)$ and the workload distribution of Phase 2 for each job as $Exp(40)$. The convergence of efficiency ratios of different schedulers are shown in Fig. 3. For different workload, we choose workload distribution of Phase 1 as $Exp(30)$ and the workload distribution of Phase 2 as



(a) Preemptive Scenario (b) Non-Preemptive Scenario

Fig. 3. Convergence of Efficiency Ratios (Exponential Distribution, Large Reduce)



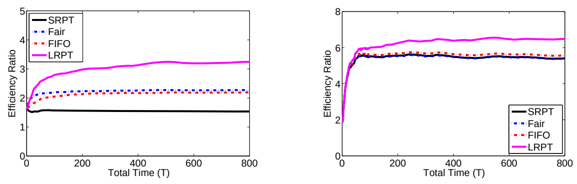
(a) Preemptive Scenario (b) Non-Preemptive Scenario

Fig. 4. Convergence of Efficiency Ratios (Exponential Distribution, Small Reduce)

$Exp(15)$. The efficiency ratios of schedulers are shown in Fig. 4.

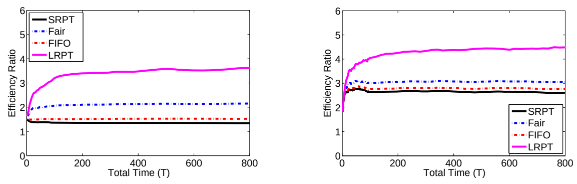
Then, we evaluate the uniformly distributed workload. We choose the workload distribution of Phase 1 for each job as $U[1, 9]$ and the workload distribution of Phase 2 for each job as $U[10, 70]$. The convergence of efficiency ratios of different schedulers are shown in Fig. 5. To evaluate for a smaller workload of Phase 2, we choose workload distribution of Phase 1 as $U[10, 50]$ and the workload distribution of Reduce as $U[10, 20]$. The convergence of efficiency ratios of different schedulers are shown in Fig. 6.

From Figs. 3-6, we observe that the efficiency ratios of these typical work-conserving schedulers are small constants.



(a) Preemptive Scenario (b) Non-Preemptive Scenario

Fig. 5. Convergence of Efficiency Ratios (Uniform Distribution, Large Reduce)



(a) Preemptive Scenario (b) Non-Preemptive Scenario

Fig. 6. Convergence of Efficiency Ratios (Uniform Distribution, Small Reduce)

VII. Conclusion

In this paper, we study the problem of minimizing the total flow-time of a sequence of jobs with phase precedence, where the jobs arrive over time and need to be processed through Phase 1 and Phase 2 before leaving the system. Since no on-line algorithm can achieve a constant competitive ratio in some scenarios, we define a weaker metric of performance called the efficiency ratio and propose a corresponding technique to analyze on-line schedulers. Under assumption of bounded workload of Phase 2, we then show a surprising property that for the flow-time problem any work-conserving scheduler has a constant efficiency ratio in both preemptive and non-preemptive scenarios. Furthermore, we relax the assumption to light-tailed distributed workload of Phase 2. The simulation results show that the efficiency ratios of typical work-conserving schedulers converge when time goes to infinity, for both preemptive and non-preemptive scenarios, and for both bounded and light-tailed distributed workload of Phase 2. We believe that the efficiency ratio metric and the proof technique presented here can be used for modeling and analyzing a wide range of online solutions.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *Proc. of Sixth Symposium on Operating System Design and Implementation, OSDI*, December 2004, pp. 137–150.

[2] B. Moseley, A. Dasgupta, R. Kumar, and T. Sarlos, "On scheduling in map-reduce and flow-shops," in *Proc. of the 23rd ACM symposium on Parallelism in algorithms and architectures, SPAA*, June 2011, pp. 289–298.

[3] K. R. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*. Hoboken, NJ, USA: John Wiley & Sons, 2009.

[4] edited by Joseph Y-T. Leung, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Boca Raton, FL, USA: Chapman and Hall/CRC, 2004.

[5] M. Caramia and M. Drozdowski, "Scheduling malleable tasks for mean flow time criterion," Institute of Computing Science, Poznan University of Technology, Tech. Rep., 2005. [Online]. Available: http://www.cs.put.poznan.pl/mdrozdowski/rapIIn/RA008-05.pdf

[6] J. Du, J. Y.-T. Leung, and G. H. Young, "Scheduling chain-structured tasks to minimizing makespan and mean flow time," *Information and Computation*, vol. 92, no. 2, pp. 219–236, June 1991.

[7] ——, "Minimizing mean flow time with release time constraint," *Theoretical Computer Science*, vol. 75, no. 3, pp. 347–355, October 1990.

[8] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Job sechduling for multi-user mapreduce clusters," University of Califonia, Berkley, Tech. Rep., April 2009.

[9] J. Tan, X. Meng, and L. Zhang, "Performance analysis of coupling scheduler for mapreduce/hadoop," in *Proc. of IEEE Infocom*, March 2012.

[10] H. Chang, M. Kodialam, R. R. Kompella, T. V. Lakshman, M. Lee, and S. Mukherjee, "Scheduling in mapreduce-like systems for fast completion time," in *Proc. of IEEE Infocom*, March 2011, pp. 3074–3082.

[11] F. Chen, M. Kodialam, and T. Lakshman, "Joint scheduling of processing and shuffle phases in mapreduce systems," in *Proc. of IEEE Infocom*, March 2012.

[12] Y. Zheng, P. Sinha, and N. Shroff, "Performance Analysis of Work-Conserving Scheduler in Minimizing Flowtime Problem with Two-Stages Precedence," Ohio State University, Tech. Rep., June 2012, http://www2.ece.ohio-state.edu/~zhengy/TR3.pdf. [Online]. Available: http://www2.ece.ohio-state.edu/~zhengy/TR3.pdf

[13] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis: Queues, Communication and Computing*. New York, NY, USA: Chapman and Hall, 1995.