Journal of the Operations Research Society of Japan
Vol. 59, No. 2, April 2016, pp. 174–194

PERFORMANCE ANALYSIS OF TASK REPLICATION IN LARGE-SCALE PARALLEL-DISTRIBUTED PROCESSING: AN EXTREME VALUE THEORY APPROACH

Tsuguhito Hirai Hiroyuki Masuyama
Kyoto University

Shoji Kasahara
Nara Institute of Science and Technology

Yutaka Takahashi
Kyoto University

(Received September 11, 2014; Revised September 29, 2015)

Abstract In cloud computing, a large-scale parallel-distributed processing service is provided in which a huge task is split into a number of subtasks, which are processed independently on a cluster of machines referred to as workers. Those workers that take longer to process their assigned subtasks result in the processing delay of the task (the issue of stragglers). An efficient way to address this issue is for other workers to execute the troubled subtasks for backup purposes (task replication). In this paper, we evaluate the efficiency of task replication from a theoretical point of view. The mean value and standard deviation of the task-processing time are derived approximately using extreme value theory, while the mean total processing time is evaluated exactly, for cases in which the worker-processing time follows a hyper-exponential, Weibull, or Pareto distribution. The numerical results reveal that the efficiency of task replication depends significantly on the tail of the worker-processing time distribution. In addition, the optimal number of replications which achieves the shortest task-processing time mainly depends on the coefficient of variation of the worker-processing time. Furthermore, three replications are effective to guarantee a low variance of the task-processing time, regardless of the tail.

Keywords: Mathematical modeling, parallel-distributed processing, task scheduling, task replication, extreme value theory, performance analysis

1. Introduction

Recently, cloud computing has attracted considerable attention due to the emergence of huge computing resources and its significant improvement in usage fee. In [3], cloud computing is defined as the sum of the existing concepts, software as a service (SaaS) and utility computing. More precisely, cloud computing is the combined concept of providing a computer-processing service only as needed via the Internet (SaaS) and using server resources in a data center only as needed (utility computing). A remarkable feature of cloud computing is that data centers providing cloud computing services have a huge number of computing resources, and this number is still increasing. For example, Google aims to have several million machines in their data centers [7].

In addition, volunteer computing is becoming popular due to the spread of computing resources with an Internet connection. In volunteer computing, distributed computing resources are donated by individuals as well as organizations, and, for some projects, the number of hosts is in the hundreds of thousands [14]. Therefore, in both cloud and volunteer computing, efficient use of an extremely large number of computing resources is a critical issue.

With the increase in the capacity of hard-disks, computing tasks must handle a greater volume of data, and an enormous amount of time is required if a task is carried out by an individual computing resource. In cloud and volunteer computing, an enormous amount of data is handled by a huge number of computing resources in parallel-distributed processing fashion [4, 6, 22]. This scheme is used for data mining, document processing, and machine learning and is used by numerous companies and organizations for processing large-scale data [21]. In the following, we refer to this processing mechanism as *large-scale parallel-distributed processing.*

In large-scale parallel-distributed processing, a huge task is split into a number of subtasks and those are processed independently in parallel on a cluster of machines referred to as workers. The huge task completes when all the subtasks have finished. Therefore, workers that take longer to process their assigned subtasks result in delay in the processing of the task (the issue of stragglers) [6]. One of the reasons causing slow workers is frequent machine failure because data centers consist of a huge number of commodity machines for reducing hardware cost [4, 7]. Moreover, it is reported in [25] that virtualization technology can cause resource competition, and as a result heterogeneity occurs in processing speed of workers. In the following, we refer to the time to complete a task (resp. subtask) as the *task-processing* (resp. *subtask-processing*) *time.*

In order to alleviate the issue of stragglers, there exist two scheduling schemes: load balancing [8] and task replication [5]. In load balancing, the subtask size for a worker is determined according to its processing speed. In other words, small subtasks are allocated to slow workers, while large subtasks are performed by fast workers. This scheduling makes the variance of the subtask-processing times significantly small, although the load-balancing scheduler must know each worker's subtask-processing time a priori.

In task-replication scheduling, on the other hand, backup executions of the remaining in-progress subtask are conducted when the elapsed time of subtask processing is greater than a pre-specified threshold. Then, the processing of the subtask ends when either the original subtask or backup execution is completed. One advantage of this scheduling is that the task-replication scheduler activates backup executions for a worker according to the elapsed time of subtask processing, i.e., no a priori information about the subtask-processing time is needed.

In this paper, we evaluate the effect of task-replication scheduling on two performance measures: the task-processing time and the total amount of execution times of workers for the processing of a task. The latter is referred to as the *total processing time* hereafter. Note that the former indicates how the performance is improved by task replication, whereas the latter characterizes the cost resulting from task replication. We consider the task-replication scheduling policy in which a task entering the service facility is split into subtasks of equal size, and the task service ends when all of the subtasks are completed. Note that the assumption of equally sized subtasks becomes reasonable when a huge amount of input data is split into data pieces of approximately equal size [6]. Moreover, each subtask is processed not only by its own worker but also by alternative distinct workers, and the subtask service ends when one of the relevant workers' processes is completed. In the following, we assume that the times to complete subtask by its own worker and alternative distinct workers are independent and identically distributed (i.i.d.), and refer to these times as the *worker-processing time*s. Note here that it is reported in [4–6] that most large distributed systems are heterogeneous and dynamic due to many reasons. For example, a machine with an ill-conditioned disk may suffer from a long disk-read time. The machine-/cluster-level task scheduler may schedule the other tasks before subtasks. Software failure also causes a long worker-processing time. It is also reported that hardware faults and the complexity of software process make the system behavior unpredictable even when the system is operated in a centralized manner. Based on this unpredictability, we assumed that the worker-processing time is i.i.d. even when the duplicated subtasks are the same as the original one. Note also that the subtask-processing time is given by the minimum of some worker-processing times.

For this system, we propose an approach based on extreme value theory for approximately deriving the mean value and standard deviation of the task-processing time. Moreover, we exactly derive the mean total processing time. It is reported in [14, 15] that the time between worker failures has a heavy-tailed property. Therefore, in order to investigate how the tail of the distribution affects performance measures, we consider cases in which the worker-processing time follows a hyper-exponential, Weibull, or Pareto distribution. These distributions are also used for modeling the worker-processing time in the literature (see, for example, [1, 10, 24]). In numerical examples, we investigate the accuracy of the approximations derived with extreme value theory in comparison with exact analyses. We then determine the optimal number of alternative workers which achieves the shortest task-processing time, and consider the effect of task replication on the performance measures. Finally, we discuss the effect of the starting time of task replication on performance measures through Monte Carlo simulation because we assumed in the analytical model that alternative subtasks are simultaneously executed from the beginning of the processing of the task in order to simplify the analysis.

The remainder of this paper is organized as follows. We describe previous studies on task replication and point out the differences between these studies and the present study in Section 2. The analytical model for large-scale parallel-distributed processing with task replication is described in Section 3. For this model, we approximately derive the mean value and standard deviation of the task-processing time using extreme value theory and exactly derive the mean total processing time in Section 4. Section 5 presents numerical examples of the derived performance measures. Finally, we conclude the paper in Section 6.

2. Related Work

A number of studies have investigated the performance of task replication. From the approach based on real-data measurements, Dean et al. [6] implement a task-replication scheme referred to as backup-task scheduling for MapReduce framework and report that backup mechanisms can significantly reduce the processing time of a task, increasing computational resource consumption by no more than a few percent. Zaharia et al. [25] focus on a speculative task assignment mechanism, which is a kind of task-replication scheme, implemented on Hadoop [23]. They propose a new task-selection algorithm to improve the accuracy of speculation and confirm through measurement-based evaluation that their algorithm works significantly better than Hadoop's algorithm in heterogeneous environments.

On the other hand, from the viewpoint of simulation experiments, Anglano et al. [2] investigate a scenario in which several users submit multiple sets of tasks to a scheduler simultaneously and propose several set selection strategies while using task replication to process individual sets of tasks. They compare these strategies through discrete-event simulation and confirm the effectiveness of task replication. Cirne et al. [5] investigate the effectiveness of several job-replication schedulers by simulation in comparison with traditional information-based schedulers. In [8], Dobber et al. investigate the effectiveness of dynamic load balancing (DLB) and job replication (JR) by trace-driven simulation experiments and propose a hybrid scheduling scheme of DLB and JR. Nóbrega et al. [17] propose replication schedulers that use any available information about applications and resources and evaluate these schedulers through simulations. They demonstrate that the use of partial information (e.g., the size of the tasks and the speed of the workers) on replication schedulers can greatly decrease resource wastage without affecting the processing time of a task.

As mentioned above, many researches investigate the performance of task replication through simulations or measurements. However, these approaches require excessive time or computing resources for evaluating extremely large-scale systems. Moreover, it is difficult to see how the system characteristics, such as the number of workers and the heterogeneity of the worker-processing time, affect performance measures.

To overcome these challenges, we have investigated the performance of task replication from a theoretical point of view. In [12, 13], we model the task-scheduling server of parallel-distributed processing as a single-server queue and explicitly derive task-processing time distributions when the worker-processing time obeys a Weibull or Pareto distribution. We then compare the mean response time under task-replication scheduling with the mean response time obtained under normal scheduling and demonstrate that the effect of task-replication depends significantly on the workers' processing time distribution.

Hashimoto et al. [11] consider the effect of backup tasks on performance of systems with parallel-distributed processing, in which one replicated subtask is activated if an original subtask is not completed by a pre-specified time referred to as the deadline time. They derive approximate formulas for the task-processing time and total processing time by extreme value theory, investigating how the deadline time affects the performance measures for three cases of the worker-processing time distribution: hyper-exponential, Weibull and Pareto distributions. In [11], the number of replications is one, and a primary concern is the effect of deadline time on system performance. In the present paper, on the other hand, we focus on how the number of replications affects performance measures.

3. Analytical Model

We make the following assumptions to construct a stochastic model of the large-scale parallel-distributed processing system.

(a) When a task is accepted by the server, the task is divided into N subtasks, each of which is duplicated $R-1$ times.
(b) The system has a server consisting of M $(:= NR)$ workers.
(c) The M subtasks (N original subtasks and their $N(R-1)$ copies) are assigned to the M workers on a one-to-one basis (see Figure 1).
(d) The processing of a group consisting of one original subtask and its $R-1$ copies is terminated when one of the R subtasks is processed completely*. We define the subtask-processing time as the processing time of such a group.
(e) The task-processing time is equal to the maximum of its N subtask-processing times.

As for the time to make replications, the number of replications R is small in general. For example, the default setting of R in Hadoop is three [23]. In such a case, the time to make replications is negligible.

In what follows, we describe the assumption on the subtask-processing time.

*There exists some delay for terminating all unfinished subtasks. This delay can be included as a part of the subtask-processing time, but is not taken into consideration in our model due to analytical tractability. Note that this delay can be negligible if the subtask-processing time is large compared to the overhead of this termination process.

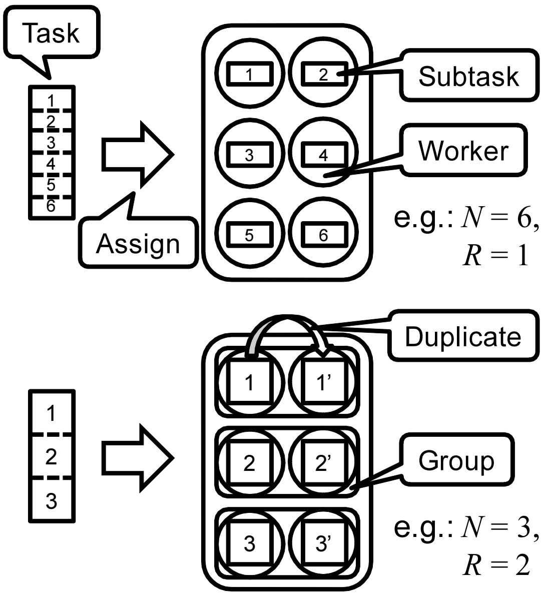


Figure 1: Examples of assigning subtasks

Let $U_0^{(i)}$ $(i = 1, 2, \dots, N)$ denote the worker-processing time of the i-th original subtask generated from a task. Moreover, for each $i = 1, 2, \dots, N$, let $U_j^{(i)}$ $(j = 1, 2, \dots, R-1)$ denote the worker-processing time of the j-th copy of the i-th original subtask. We now make the following assumption.

(f) The $U_j^{(i)}$'s $(i = 1, 2, \dots, N,\ j = 0, 1, \dots, R-1)$ are i.i.d. random variables which follow a common distribution function H_N with positive mean b/N and positive left endpoint $a/N := \inf\{t \in \mathbb{R}; H_N(t) > 0\}$, where $b > a \ge 0$. Note here that the sizes of a task and its subtasks are deterministic whereas the processing times of them vary stochastically due to the unpredictability of workers' ability. The distribution H_N is called a *worker-processing time distribution.*

We define S_i $(i = 1, 2, \dots, N)$ as the i-th subtask-processing time. It follows from assumption (d) that

$$S_i = \min_{0 \le j \le R-1} U_j^{(i)}, \qquad i = 1, 2, \dots, N.$$

Clearly, the S_i's are i.i.d. random variables and

$$F_{N,R}(t) := \mathsf{P}(S_i \le t) = 1 - \{1 - H_N(t)\}^R, \qquad t \in \mathbb{R}, \tag{1}$$

for all $i = 1, 2, \dots, N$. We also define $T_{N,R}$ as the task-processing time. From assumption (e), we have

$$T_{N,R} = \max_{1 \le i \le N} S_i,$$

and thus

$$G_{N,R}(t) := \mathsf{P}(T_{N,R} \le t) = \{F_{N,R}(t)\}^N, \qquad t \in \mathbb{R}.$$

4. Analysis of Performance Measures

We evaluate the stochastic model described in the previous section in terms of three performance measures.

To this end, we introduce the performance measures. Let $A_{N,R}$ and $D_{N,R}$ denote the mean value and standard deviation of the task-processing time, respectively, i.e.,

$$A_{N,R} = \mathsf{E}[T_{N,R}], \qquad D_{N,R} = \sqrt{\mathsf{Var}[T_{N,R}]}.$$

Thus we have

$$A_{N,R} = g_{N,R}^{(1)}, \qquad D_{N,R} = \sqrt{g_{N,R}^{(2)} - \left(g_{N,R}^{(1)}\right)^2}, \tag{2}$$

where $g_{N,R}^{(k)} = \int_0^\infty t^k \mathrm{d}G_{N,R}(t)$ $(k = 1, 2)$. Furthermore, let $P_{N,R}$ denote the mean total processing time. Since the total processing time is defined as the total amount of execution times of M workers for the processing of a task, and the S_i's are i.i.d., we have

$$P_{N,R} = \mathsf{E}\left[\sum_{i=1}^{N} RS_i\right] = NRf_{N,R}^{(1)}, \tag{3}$$

where $f_{N,R}^{(1)} = \int_0^\infty t \mathrm{d}F_{N,R}(t)$.

In what follows, we discuss three types of worker-processing time distributions with mean b/N and left endpoint a/N $(b > a \geq 0)^\dagger$:

(a) Hyper-exponential distribution‡

$$H_N(t) = \begin{cases} 1 - \sigma \exp\{-\sigma(t - a/N)/\nu_N\} \\ \qquad - \widetilde{\sigma} \exp\{-\widetilde{\sigma}(t - a/N)/\nu_N\}, & t \geq a/N, \\ 0, & t < a/N, \end{cases} \tag{4}$$

with $\nu_N = (b - a)/(2N)$ and $\widetilde{\sigma} = 1 - \sigma$ $(0 < \sigma \leq 1/2)$;

(b) Weibull distribution

$$H_N(t) = \begin{cases} 1 - \exp\{-\{(t - a/N)/\eta_N\}^\alpha\}, & t \geq a/N, \\ 0, & t < a/N, \end{cases} \tag{5}$$

with $\eta_N = (b - a)/\{\Gamma(1 + 1/\alpha)N\}$ $(\alpha > 0)$; and

(c) Pareto distribution

$$H_N(t) = \begin{cases} 1 - \{\mu_N/(t + \mu_N - a/N)\}^\beta, & t \geq a/N, \\ 0, & t < a/N, \end{cases} \tag{6}$$

with $\mu_N = (b - a)(\beta - 1)/N$ $(\beta > 2)$.

The coefficients of variation (i.e., the ratio of the standard deviation to the mean) of these distributions are as follows:

(a) Hyper-exponential distribution

$$\frac{b - a}{b}\sqrt{\frac{1}{2\sigma(1 - \sigma)} - 1}; \tag{7}$$

[†] The aim of introducing left endpoints to the three distributions is to compute performance measures under the same mean and same coefficient of variation for the worker-processing time. This enables us to investigate how the heavy-tailedness of distributions affects the performance measures.

[‡] Strictly speaking, this is a two-phase balanced hyper-exponential distribution.

(b) Weibull distribution

$$\frac{b-a}{b}\sqrt{\frac{\Gamma(1+2/\alpha)}{\{\Gamma(1+1/\alpha)\}^2}-1}; \tag{8}$$

(c) Pareto distribution

$$\frac{b-a}{b}\sqrt{\frac{\beta}{\beta-2}}. \tag{9}$$

4.1. Exact expressions for the first and second moments of the task-processing time

A straightforward calculation yields the following expressions for $g_{N,R}^{(1)}$ and $g_{N,R}^{(2)}$:
(a) Hyper-exponential distribution

$$\begin{aligned}
g_{N,R}^{(1)} &= \frac{1}{N}\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{k}\sum_{l=0}^{Rk}\binom{Rk}{l}\sigma^l(1-\sigma)^{Rk-l} \\
&\quad\times\left[\frac{b-a}{2\{\sigma l+(1-\sigma)(Rk-l)\}}+a\right], \\
g_{N,R}^{(2)} &= \frac{1}{N^2}\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{k}\sum_{l=0}^{Rk}\binom{Rk}{l}\sigma^l(1-\sigma)^{Rk-l} \\
&\quad\times\left[\frac{(b-a)^2}{2\{\sigma l+(1-\sigma)(Rk-l)\}^2}+\frac{a(b-a)}{\sigma l+(1-\sigma)(Rk-l)}+a^2\right];
\end{aligned}$$

(b) Weibull distribution

$$\begin{aligned}
g_{N,R}^{(1)} &= \frac{1}{N}\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{k}\left(\frac{b-a}{R^{1/\alpha}k^{1/\alpha}}+a\right), \\
g_{N,R}^{(2)} &= \frac{1}{N^2}\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{k}\left[\frac{\Gamma(1+2/\alpha)}{\{\Gamma(1+1/\alpha)\}^2}\frac{(b-a)^2}{R^{2/\alpha}k^{2/\alpha}}+\frac{2a(b-a)}{R^{1/\alpha}k^{1/\alpha}}+a^2\right];
\end{aligned}$$

(c) Pareto distribution

$$\begin{aligned}
g_{N,R}^{(1)} &= \frac{1}{N}\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{k}\left\{\frac{(b-a)(\beta-1)}{\beta Rk-1}+a\right\}, \\
g_{N,R}^{(2)} &= \frac{1}{N^2}\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{k}\left\{\frac{2(b-a)^2(\beta-1)^2}{(\beta Rk-2)(\beta Rk-1)}+\frac{2a(b-a)(\beta-1)}{\beta Rk-1}+a^2\right\}.
\end{aligned}$$

Combining (2) with the above equations, we can obtain *exact* expressions for $A_{N,R}$ and $D_{N,R}$. However, these expressions are not suitable for computing with high accuracy because a number of subtractions in the above equations can cause loss of significant digits. Moreover, it is difficult to make further insights for these expressions against the parameters such as N and R. Therefore, using extreme value theory, we derive asymptotic formulas for $g_{N,R}^{(k)}$ $(k=1,2)$, which can serve as approximations when N is large. The accuracy of the approximations is numerically investigated in Subsection 5.1.

4.2. Asymptotic formulas for the first and second moments of the task-processing time

Using extreme value theory, we derive asymptotic formulas for $g_{N,R}^{(1)}$ and $g_{N,R}^{(2)}$ for three cases. Note that the preliminary results of extreme value theory used in this subsection are summarized in appendix.

4.2.1. Case of a hyper-exponential worker-processing time

Let $F_R(t) = F_{1,R}(t)$. From (1), we then have

$$F_R(t) = 1 - \{1 - H_1(t)\}^R, \qquad t \in \mathbb{R}. \tag{10}$$

Substituting (4) into (10), we have

$$F_R(t) = \begin{cases} 1 - [\sigma \exp\{-\sigma(t-a)/\nu_1\} - \widetilde{\sigma} \exp\{-\widetilde{\sigma}(t-a)/\nu_1\}]^R, & t \ge a, \\ 0, & t < a. \end{cases}$$

Note that F_R is independent of N and

$$F_R(t) = F_{N,R}\left(\frac{t}{N}\right) = \mathsf{P}(NS_i \le t), \qquad t \in \mathbb{R}, \tag{11}$$

which implies that $\{NS_i; i = 1, 2, \dots, N\}$ is a sequence of i.i.d. random variables with distribution function F_R. We can confirm that F_R can be expressed as (20), where

$$x_0 = a, \qquad c(x) = 1, \qquad g(x) = 1,$$

$$\begin{aligned} a(x) = {} & \left[\sigma \exp\left\{-\frac{\sigma(x-a)}{\nu_1}\right\} + \widetilde{\sigma} \exp\left\{-\frac{\widetilde{\sigma}(x-a)}{\nu_1}\right\}\right] \\ & \times R^{-1} \left[\frac{\sigma^2}{\nu_1} \exp\left\{-\frac{\sigma(x-a)}{\nu_1}\right\} + \frac{\widetilde{\sigma}^2}{\nu_1} \exp\left\{-\frac{\widetilde{\sigma}(x-a)}{\nu_1}\right\}\right]^{-1}. \end{aligned}$$

Thus it follows from Proposition A.1 that $F_R \in \mathrm{MDA}(\Lambda)$ and $c_n = a(d_n)$, where d_n is given as the solution of the equation $F_R(d_n) = 1 - 1/n$, which can be solved using a numerical method (e.g., Newton's method).

Recall here that $NT_{N,R} = \max_{1 \le i \le N} NS_i$ and that the i.i.d. random variables NS_i's ($i = 1, 2, \dots, N$) follow distribution function F_R (see (11)). Therefore proposition A.2 implies that

$$\lim_{N \to \infty} \mathsf{E}\left[\left\{\frac{NT_{N,R} - d_N}{c_N}\right\}^k\right] = (-1)^k \Gamma^{(k)}(1), \qquad k = 1, 2, \tag{12}$$

where $\Gamma^{(k)}(1)$ ($k = 1, 2$) is given by (see Subsection 5.4 (ii) in [19])

$$\Gamma^{(1)}(1) = \lim_{x \to 1} \frac{\mathrm{d}}{\mathrm{d}x} \Gamma(x) = -\gamma \ (\gamma : \text{Euler constant}), \qquad \Gamma^{(2)}(1) = \lim_{x \to 1} \frac{\mathrm{d}^2}{\mathrm{d}x^2} \Gamma(x) = \gamma^2 + \frac{\pi^2}{6}.$$

As a result, from (12) and $g_{N,R}^{(k)} = \mathsf{E}\left[T_{N,R}^k\right]$ ($k = 1, 2$), we have

$$g_{N,R}^{(1)} \overset{N}{\sim} \gamma \frac{c_N}{N} + \frac{d_N}{N}, \qquad g_{N,R}^{(2)} \overset{N}{\sim} \left(\gamma^2 + \frac{\pi^2}{6}\right) \frac{c_N^2}{N^2} + 2\gamma \frac{c_N d_N}{N^2} + \frac{d_N^2}{N^2},$$

where $f(x) \overset{x}{\sim} g(x)$ represents $\lim_{x \to \infty} f(x)/g(x) = 1$. Substituting these asymptotic formulas into (2), we obtain approximate formulas for $A_{N,R}$ and $D_{N,R}$.

4.2.2. Case of a Weibull worker-processing time

Substituting (5) into (10), we have

$$F_R(t) = \begin{cases} 1 - \exp\{-R\{(t-a)/\eta_1\}^\alpha\}, & t \ge a, \\ 0, & t < a. \end{cases}$$

The distribution function F_R has the representation (20) with

$$x_0 = a, \qquad c(x) = 1, \qquad g(x) = 1, \qquad a(x) = \frac{\eta_1^\alpha}{\alpha R (x-a)^{\alpha-1}}.$$

Therefore, according to Proposition A.1, $F_R \in \mathrm{MDA}(\Lambda)$ and the normalizing constants c_n and d_n are given by

$$c_n = \frac{\eta_1}{\alpha R}\left(\frac{\log n}{R}\right)^{1/\alpha - 1}, \qquad d_n = \eta_1 \left(\frac{\log n}{R}\right)^{1/\alpha} + a.$$

As a result, Proposition A.2 yields

$$\begin{aligned} g_{N,R}^{(1)} &\overset{N}{\sim} -\Gamma^{(1)}(1)\frac{c_N}{N} + \frac{d_N}{N} \\ &= \frac{\gamma(b-a)}{\Gamma(1+1/\alpha)\alpha N R}\left(\frac{\log N}{R}\right)^{1/\alpha-1} + \frac{(b-a)}{\Gamma(1+1/\alpha)N}\left(\frac{\log N}{R}\right)^{1/\alpha} + \frac{a}{N}, \qquad (13) \\ g_{N,R}^{(2)} &\overset{N}{\sim} \Gamma^{(2)}(1)\frac{c_N^2}{N^2} - 2\Gamma^{(1)}(1)\frac{c_N d_N}{N^2} + \frac{d_N^2}{N^2} \\ &= \left(\gamma^2 + \frac{\pi^2}{6}\right)\left\{\frac{(b-a)}{\Gamma(1+1/\alpha)\alpha N}\left(\frac{\log N}{R}\right)^{1/\alpha-1}\right\}^2 \\ &\quad + \frac{2\gamma(b-a)}{\Gamma(1+1/\alpha)\alpha N R}\left(\frac{\log N}{R}\right)^{1/\alpha-1}\left\{\frac{(b-a)}{\Gamma(1+1/\alpha)N}\left(\frac{\log N}{R}\right)^{1/\alpha} + \frac{a}{N}\right\} \\ &\quad + \left\{\frac{(b-a)}{\Gamma(1+1/\alpha)N}\left(\frac{\log N}{R}\right)^{1/\alpha} + \frac{a}{N}\right\}^2. \end{aligned}$$

4.2.3. Case of a Pareto worker-processing time

From (6) and (10), we have

$$F_R(t) = \begin{cases} 1 - \{\mu_1/(t+\mu_1-a)\}^{\beta R}, & t \ge a, \\ 0, & t < a. \end{cases}$$

Note that $1 - F_R$ is regularly varying with index $-\beta R$, and thus (22) holds. Therefore, Proposition A.3 implies that $F_R \in \mathrm{MDA}(\Phi_{\beta R})$, and the normalizing constant c_n is given by

$$c_n = \mu_1\left\{n^{1/(\beta R)} - 1\right\} + a.$$

From Proposition A.4, we have

$$\lim_{N\to\infty} \mathsf{E}\left[\left\{\frac{NT_{N,R}}{c_N}\right\}^k\right] = \Gamma\left(1 - \frac{k}{\beta R}\right), \qquad k = 1, 2.$$

We also obtain the following asymptotic formulas:

$$g_{N,R}^{(1)} \overset{N}{\sim} \Gamma\left(1-\frac{1}{\beta R}\right)\frac{c_N}{N} = \Gamma\left(1-\frac{1}{\beta R}\right)\left[\frac{(b-a)(\beta-1)}{N}\left\{N^{1/(\beta R)}-1\right\} + \frac{a}{N}\right], \qquad (14)$$

$$g_{N,R}^{(2)} \overset{N}{\sim} \Gamma\left(1-\frac{2}{\beta R}\right)\frac{c_N^2}{N^2} = \Gamma\left(1-\frac{2}{\beta R}\right)\left[\frac{(b-a)(\beta-1)}{N}\left\{N^{1/(\beta R)}-1\right\} + \frac{a}{N}\right]^2.$$

Table 1: Parameter set

Parameter		Value
b	[sec]	3.16×10^{10} (about 1,000 years)
a	[sec]	2.68×10^{10} (85% of b)
M		3×10^5, 3×10^6, 3×10^7
R		1, 2, 3, 4

4.3. Exact expressions for the mean total processing time

A straightforward calculation yields the following expressions for $f^{(1)}_{N,R}$:

(a) Hyper-exponential distribution

$$f^{(1)}_{N,R} = \frac{1}{N}\sum_{l=0}^{R}\binom{R}{l}\sigma^l(1-\sigma)^{R-l}\left[\frac{b-a}{2\{\sigma l + (1-\sigma)(R-l)\}} + a\right];$$

(b) Weibull distribution

$$f^{(1)}_{N,R} = \frac{1}{N}\left(\frac{b-a}{R^{1/\alpha}} + a\right);$$

(c) Pareto distribution

$$f^{(1)}_{N,R} = \frac{1}{N}\left\{\frac{(b-a)(\beta-1)}{\beta R - 1} + a\right\}.$$

Substituting these formulas into (3), we obtain exact expressions for $P_{N,R}$, as follows:

(a) Hyper-exponential distribution

$$P_{N,R} = R\sum_{l=0}^{R}\binom{R}{l}\sigma^l(1-\sigma)^{R-l}\left[\frac{b-a}{2\{\sigma l + (1-\sigma)(R-l)\}} + a\right];$$

(b) Weibull distribution

$$P_{N,R} = R\left(\frac{b-a}{R^{1/\alpha}} + a\right);$$

(c) Pareto distribution

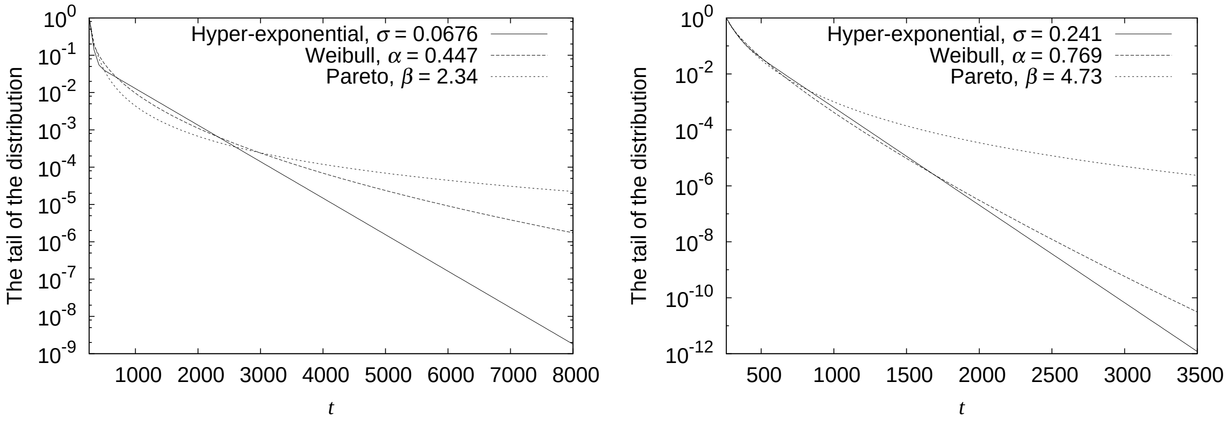
$$P_{N,R} = R\left\{\frac{(b-a)(\beta-1)}{\beta R - 1} + a\right\}.$$

5. Numerical Examples

In this section, we present some numerical examples. We first verify the proposed approximations of the mean value and standard deviation of the task-processing time by comparing the exact expressions and the approximate formulas derived by applying extreme value theory. We then calculate the optimal number of replications required to minimize the mean task-processing time using the derived approximate formulas. We compare the performance measures between the case of the optimal number of replications and that of no-replication and discuss the efficiency of task replication. Moreover, we consider the effect of task replication on reducing the standard deviation of the task-processing time. Finally, we discuss the effect of the starting time of task replication on performance measures through Monte Carlo simulation.

Table 2: Values of σ, α, and β

Coefficient of variation	0.40	0.30	0.20
σ	0.0676	0.115	0.241
α	0.447	0.548	0.769
β	2.34	2.69	4.73



(a) Case of a large coefficient of variation (b) Case of a small coefficient of variation

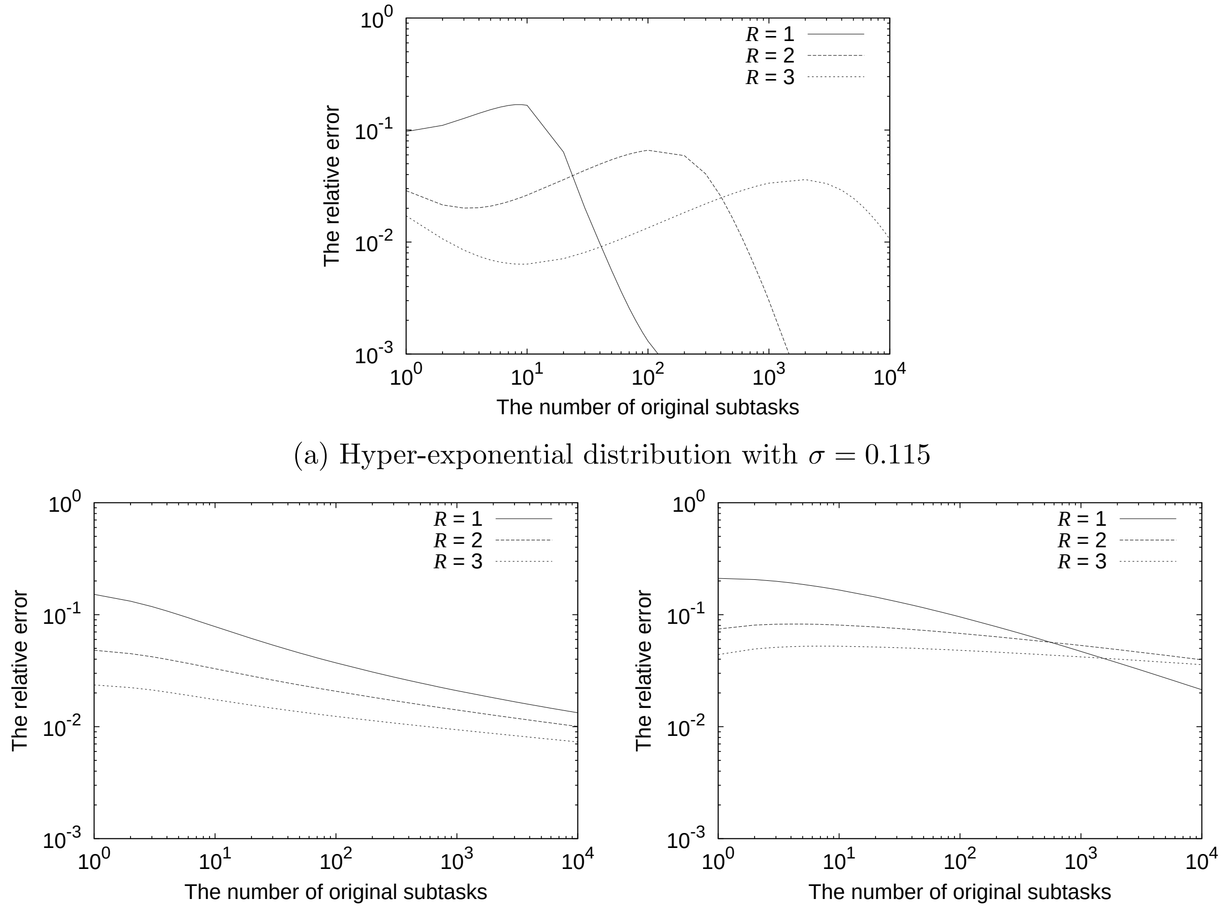
Figure 2: Tail of the worker-processing time distribution

Tables 1 and 2 show the parameter values used in the numerical experiments. We set these parameters according to the measurements and the settings of real systems [7, 18, 23]. The values of σ, α, and β are determined such that the coefficient of variation of the worker-processing time distribution takes the values shown in Table 2. Note that the coefficient of variation for a hyper-exponential (resp. Weibull and Pareto) distribution decreases with the increase in σ (resp. α and β). Note also that the coefficient of variation for each distribution depends on $b - a$, which is the difference between the mean and the left endpoint of the distribution (see (7) to (9)). When $\sigma = 0.500$ in (4) (resp. $\alpha = 1.00$ in (5)), the hyper-exponential (resp. Weibull) distribution is reduced to a shifted exponential distribution. Under the parameter setting of Table 1, the coefficient of variation of this shifted exponential distribution is equal to 0.15, which is smaller than one, the value of the coefficient of variation for the exponential distribution. Moreover, in the case of the same coefficient of variation, the tail of the hyper-exponential (resp. Pareto) distribution is the lightest (resp. heaviest) among the three distributions (see Figures 2a and 2b).

5.1. Verification of the proposed approximations

In this subsection, we investigate the approximation accuracy of the mean value and standard deviation of the task-processing time, while calculating the relative error between the approximations in Subsection 4.2 and exact analysis solutions in Subsection 4.1.

Figures 3a, 3b, and 3c show the relative error of the mean task-processing time for $R = 1$, 2, and 3 with respect to the number of original subtasks in a log-log plot. Here, the worker-processing time distribution is set to a hyper-exponential distribution with $\sigma = 0.115$ (resp. a Weibull distribution with $\alpha = 0.548$ and a Pareto distribution with $\beta = 2.69$) in Figure 3a (resp. Figures 3b and 3c). Figures 3a through 3c indicate that the relative error tends to decrease with the increase in N, the number of original subtasks. Note here that extreme value theory does not guarantee that the error between the approximations and



(a) Hyper-exponential distribution with $\sigma = 0.115$

(b) Weibull distribution with $\alpha = 0.548$

(c) Pareto distribution with $\beta = 2.69$

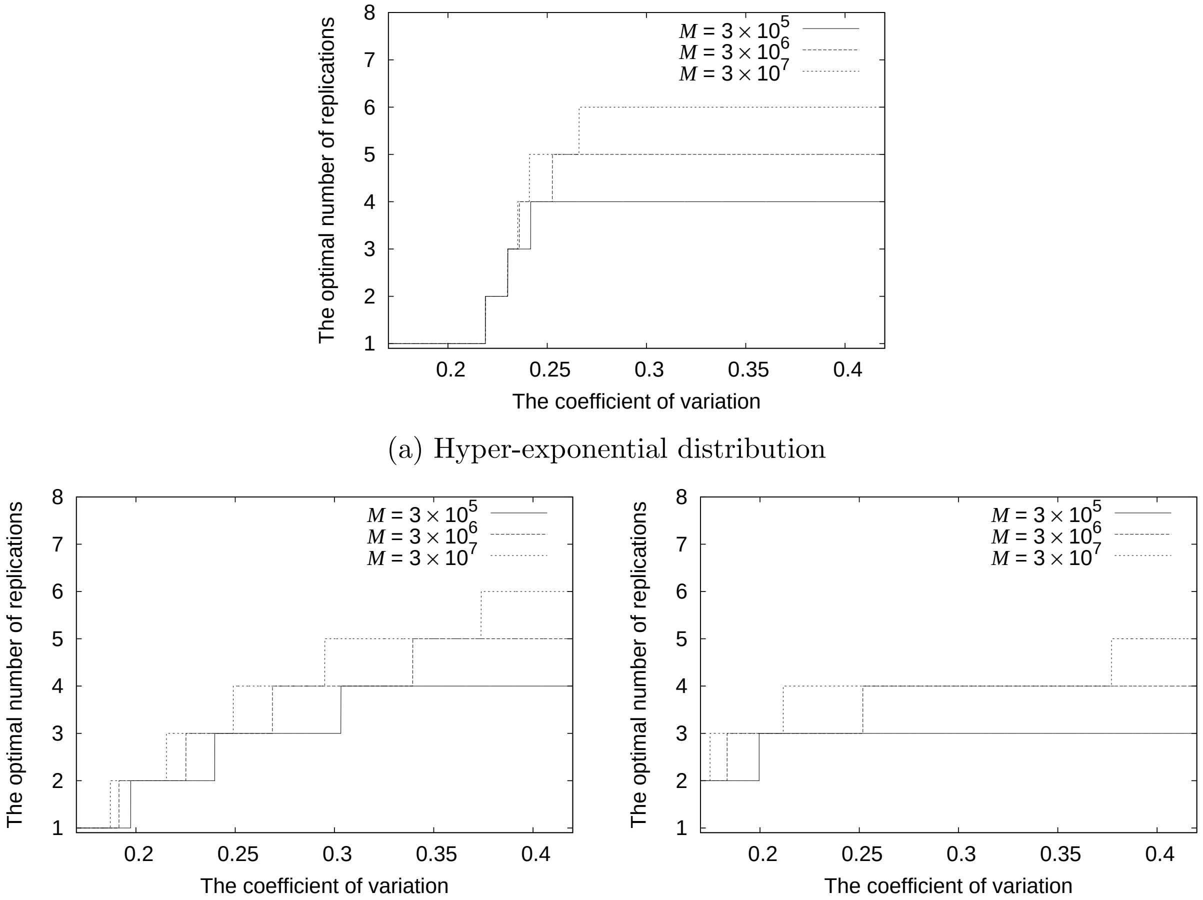
Figure 3: Relative error of the mean task-processing time

exact analysis solutions decreases monotonically when N increases. Therefore, as for the case with Figure 3a, the relative error can increase when the number of original subtasks is small. However, in particular, in Figure 3a (resp. Figures 3b and 3c), the relative error is approximately 1.1% (resp. 1.3% and 4.0%) and is sufficiently small when N is 10^4. This tendency is also observed for other parameter values of the mean value and standard deviation of the task-processing time. These results suggest that approximation formulas from extreme value theory are not accurate but have the relative error smaller than 0.1 when the number of original subtasks is greater than several tens of thousands.

5.2. Optimal number of replications

In this subsection, we investigate the optimal number of replications which achieves the shortest task-processing time.

Figure 4a (resp. Figures 4b and 4c) shows the optimal number of R for the mean task-processing time when the worker-processing time distribution follows a hyper-exponential (resp. Weibull and Pareto) distribution. The optimal number is calculated by applying the golden section search to the derived approximate formulas. The horizontal axis represents the coefficient of variation (i.e., σ, α, or β are varied). In these figures, the optimal number of replications increases with the increase in the coefficient of variation. At each value of the coefficient of variation, the optimal number of replications for $M = 3 \times 10^7$ achieves the largest value. However, the difference between optimal values of $M = 3 \times 10^5$ and 3×10^7



(a) Hyper-exponential distribution

(b) Weibull distribution

(c) Pareto distribution

Figure 4: Optimal number of replications for the mean task-processing time with respect to the coefficient of variation

is small. There is no significant difference in the optimal number of replications among the three worker-processing time distributions. This implies that the optimal number of replications depends significantly on the variance of the worker-processing time distribution, rather than the tail of distribution.

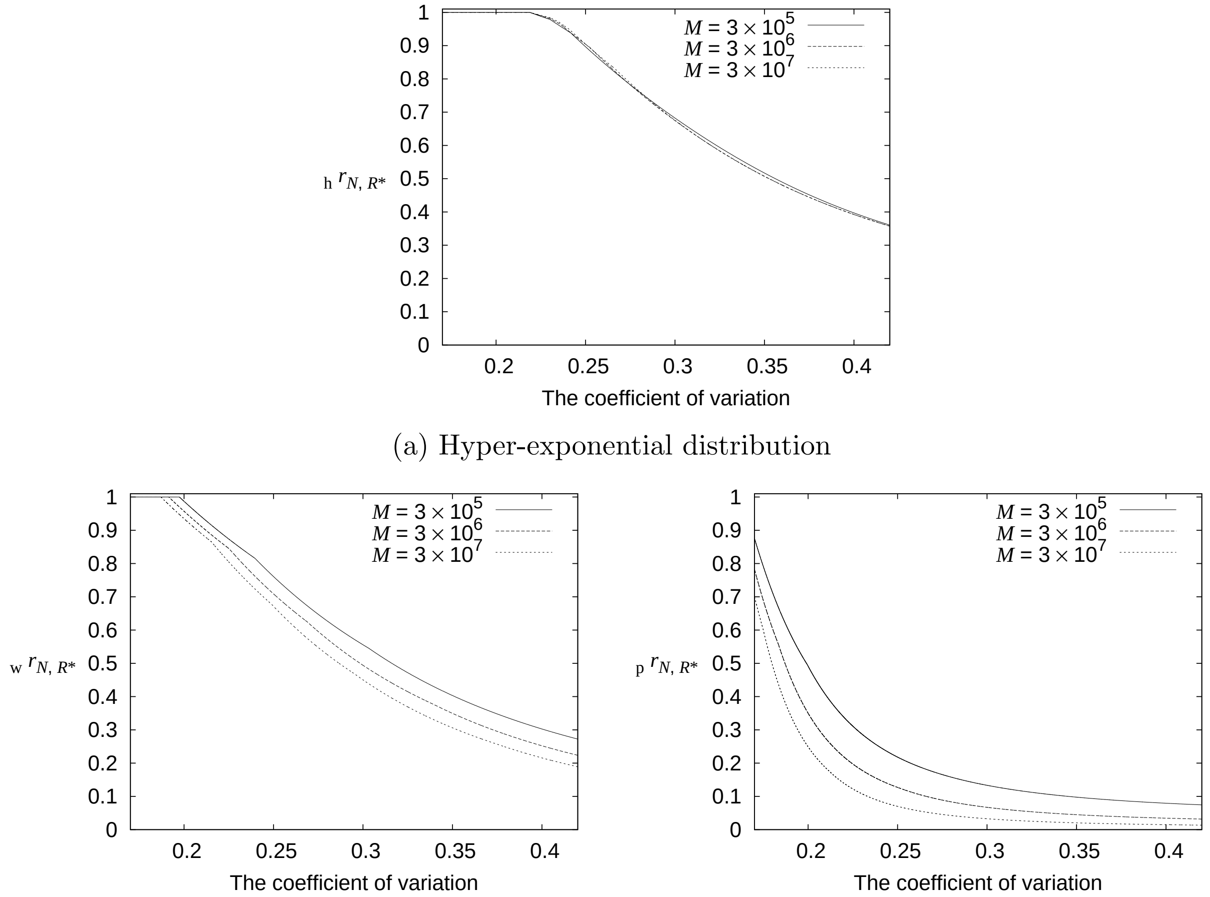
5.3. Efficiency of task replication

In this subsection, we discuss the efficiency of task replication by comparing the performance measures between the case of the optimal number of replications and that of no-replication.

Let ${}_{\mathrm{h}}r_{N,R}$ (resp. ${}_{\mathrm{w}}r_{N,R}$ and ${}_{\mathrm{p}}r_{N,R}$) denote the ratio of the mean task-processing time in the case of $R \geq 2$ to that in the case of no-replication (which corresponds to $R = 1$). Note here that the pre-subscript "${}_{\mathrm{h}}$" (resp. "${}_{\mathrm{w}}$" and "${}_{\mathrm{p}}$") shows that the worker-processing time follows a hyper-exponential (resp. Weibull and Pareto) distribution. By definition,

$$ {}_{\mathrm{h}}r_{N,R} = \frac{{}_{\mathrm{h}}g_{N,R}^{(1)}}{{}_{\mathrm{h}}g_{N,1}^{(1)}}, \quad {}_{\mathrm{w}}r_{N,R} = \frac{{}_{\mathrm{w}}g_{N,R}^{(1)}}{{}_{\mathrm{w}}g_{N,1}^{(1)}}, \quad {}_{\mathrm{p}}r_{N,R} = \frac{{}_{\mathrm{p}}g_{N,R}^{(1)}}{{}_{\mathrm{p}}g_{N,1}^{(1)}}, \tag{15} $$

where ${}_{\mathrm{h}}g_{N,R}^{(1)}$, ${}_{\mathrm{w}}g_{N,R}^{(1)}$ and ${}_{\mathrm{p}}g_{N,R}^{(1)}$ denote the mean task-processing times in the cases of hyper-exponential, Weibull and Pareto distributions, respectively. Moreover, we define R^* as the optimal number of replications calculated in Subsection 5.2.



(a) Hyper-exponential distribution

(b) Weibull distribution

(c) Pareto distribution

Figure 5: Ratio of the mean task-processing time in the case of the optimal number of replications to that in the case of no-replication with respect to the coefficient of variation

Figure 5a shows that ${}_{\mathrm{h}}r_{N,R^*}$ gradually decreases with the increase in the coefficient of variation. On the other hand, Figures 5b and 5c show that ${}_{\mathrm{w}}r_{N,R^*}$ and ${}_{\mathrm{p}}r_{N,R^*}$ decrease more quickly. Moreover, the value of ${}_{\mathrm{h}}r_{N,R^*}$ for $M = 3 \times 10^5$ is almost the same as the one for 3×10^7 (see Figure 5a), whereas the difference between the values of ${}_{\mathrm{p}}r_{N,R^*}$ for $M = 3 \times 10^5$ and $M = 3 \times 10^7$ is comparatively large (see Figure 5c). These facts imply that task replication is highly effective in reducing the task-processing time when the variation of the worker-processing time is large and such effect depends significantly on the tail asymptotics of the worker-processing time distribution.

To verify the above observation from a theoretical point of view, we discuss the difference between ${}_{\mathrm{w}}r_{N,R}$ and ${}_{\mathrm{p}}r_{N,R}$ by using the asymptotic formulas (13) and (14). Removing the non-dominant terms from these formulas, we have

$$
{}_{\mathrm{w}}g^{(1)}_{N,R} \overset{N}{\sim} \frac{(b-a)}{\Gamma(1+1/\alpha)N} \left(\frac{\log N}{R}\right)^{1/\alpha}, \tag{16}
$$

$$
{}_{\mathrm{p}}g^{(1)}_{N,R} \overset{N}{\sim} \Gamma\left(1 - \frac{1}{\beta R}\right)(b-a)(\beta-1)N^{1/(\beta R)-1}. \tag{17}
$$

Substituting (16) and (17) into (15), we readily obtain

$$\lim_{N\to\infty} {}_{\mathrm{w}}r_{N,R} = \left(\frac{1}{R}\right)^{1/\alpha}, \qquad \lim_{N\to\infty} {}_{\mathrm{p}}r_{N,R} = 0, \qquad R = 2, 3, \dots,$$

which show that

$$\lim_{N\to\infty} \frac{{}_{\mathrm{p}}r_{N,R}}{{}_{\mathrm{w}}r_{N,R}} = 0. \tag{18}$$

Equation (18) implies that, as the number of original subtasks is larger, task replication is much more effective in the Pareto worker-processing time case, compared to the Weibull worker-processing time case. This result matches with the observation of Figures 5b and 5c.

Figure 6a (resp. Figures 6b and 6c) shows the ratio of the mean total processing time in the case of the optimal R to that in the case of $R = 1$ when the worker-processing time distribution follows a hyper-exponential (resp. Weibull and Pareto) distribution. The horizontal axis represents the coefficient of variation. These figures show that the ratios of the mean total processing time increase gradually with the increase in the coefficient of variation. Moreover, the ratios are approximately the same for the three values of the number of workers, which implies that the total processing time depends significantly on the variance of the worker-processing time distribution.

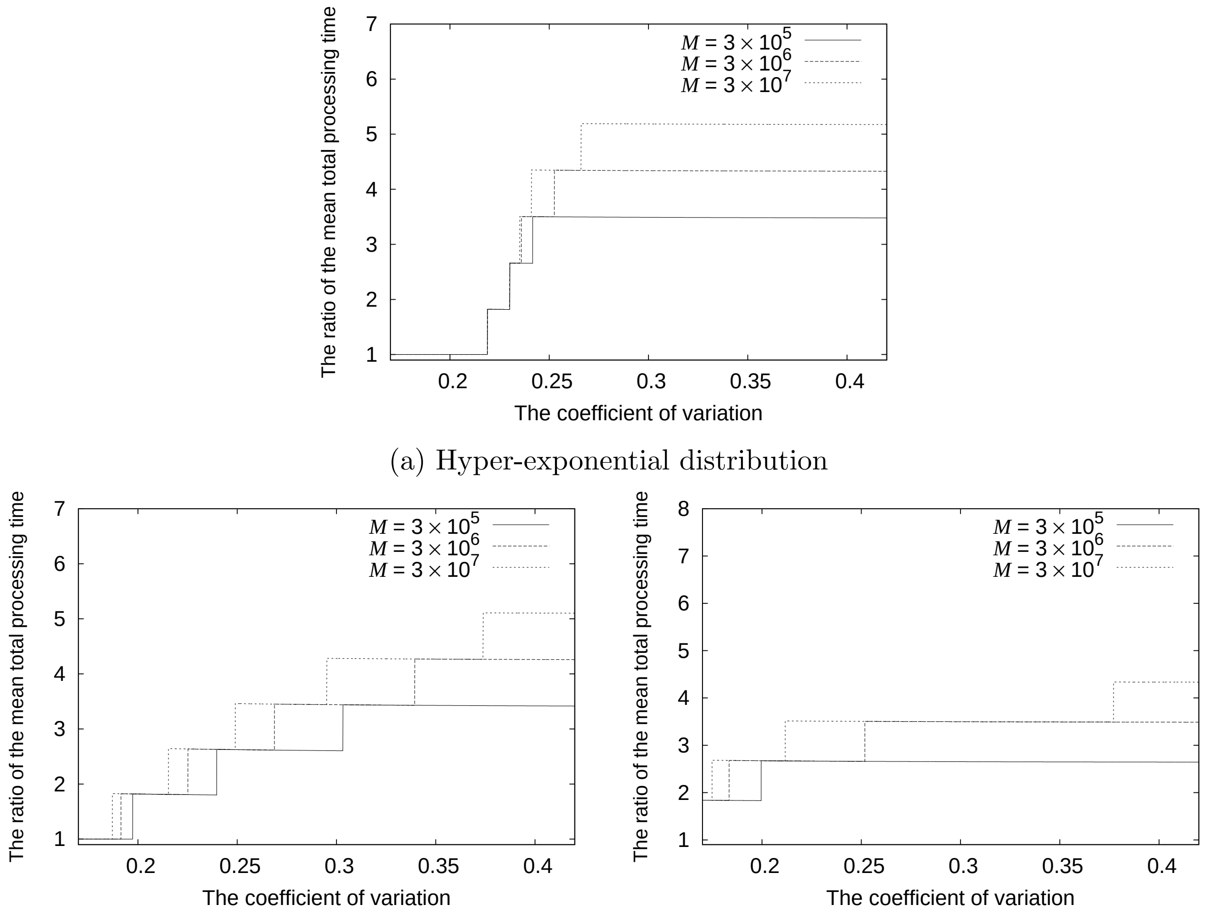
Finally, we discuss the effect of task replication on reducing the standard deviation of the task-processing time. Table 3 shows the standard deviation in the cases of hyper-exponential, Weibull, and Pareto distributions for $M = 3 \times 10^7$.

In Table 3, we observe that the standard deviation of the task-processing time increases with the increase in the coefficient of variation and is insensitive to the number of replications for the case of a hyper-exponential distribution. On the other hand, the standard deviation decreases with the number of replications for the cases of Weibull and Pareto distributions. In particular, the case of a Pareto distribution shows a significant decrease in the standard deviation. Note that the standard deviations for the cases of Weibull and Pareto distributions decrease gradually when the number of replications is greater than or equal to three. This implies that task replication is effective for decreasing the variance of the task-processing time. However, a large number of replications is less effective for reducing the variance. Based on numerical experiments, we confirmed that three replications are effective to reduce the variance of the task-processing time, even when the worker-processing time follows a heavy-tailed distribution.

These results indicate that task replication becomes more efficient when the coefficient of variation of the worker-processing time distribution increases. However, this efficiency is very different when the tail of the distribution is changed, and the difference of the mean task-processing time among distributions increases with the increase in the number of workers and the coefficient of variation. Therefore, we should consider the tail of the distribution as well as the first- and second-order statistics of the worker-processing time when we consider the efficiency of task replication for large-scale parallel-distributed processing. In addition, the results also show that three replications are effective to guarantee a low variance of the task-processing time regardless of the tail of the worker-processing time distribution.

5.4. Effect of the starting time of task replication

In this subsection, we discuss the effect of the starting time of task replication through Monte Carlo simulation. In the proposed model, we assumed for analytical simplicity that



(a) Hyper-exponential distribution

(b) Weibull distribution

(c) Pareto distribution

Figure 6: Ratio of the mean total processing time in the case of the optimal number of replications to that in the case of no-replication with respect to the coefficient of variation

alternative subtasks are simultaneously executed when the processing of a task starts. In some implementations of task replication [16], on the other hand, backup executions are activated when the elapsed time of subtask processing is greater than a pre-specified threshold. In order to investigate the usefulness of our analytical model, we conduct Monte Carlo simulation experiments. In the simulation setting, backup executions of a subtask start when the elapsed time of subtask processing exceeds $\xi b/M$ $(\xi \geq 0)$. Note that the case of $\xi = 0$ corresponds to the setting of the analytical model. We calculated the 95% confidence interval of the mean task-processing time and the mean total processing time.

Figure 7 shows the mean task-processing time in the case of a Pareto distribution ($\beta = 2.69$, $M = 3 \times 10^5$). The horizontal axis represents ξ. In Figure 7, the mean task-processing time increases more quickly according to ξ in the case of the larger number of replications. For each R, the analytical result is slightly greater than the simulation result when ξ is small. This discrepancy results from the approximation by extreme value theory. When ξ increases, the simulation result grows linearly, while the analytical remains constant.

Figure 8 shows the mean total processing time in the case of a Pareto distribution ($\beta = 2.69$, $M = 3 \times 10^5$). The horizontal axis indicates ξ. We observe in Figure 8 that the simulation result of the mean total processing time decreases with the increase in ξ, while the analytical result is the same for any ξ.

Table 3: Standard deviation of the task-processing time

Distribution	Coefficient of variation	The number of replications 1	2	3	4
hyper-exponential	0.20	4.26×10^2	4.26×10^2	4.25×10^2	4.14×10^2
	0.30	8.92×10^2	8.92×10^2	8.92×10^2	8.92×10^2
	0.40	1.52×10^3	1.52×10^3	1.52×10^3	1.52×10^3
Weibull	0.20	5.38×10^2	4.31×10^2	3.79×10^2	3.46×10^2
	0.30	2.29×10^3	1.25×10^3	8.75×10^2	6.80×10^2
	0.40	6.17×10^3	2.49×10^3	1.46×10^3	1.00×10^3
Pareto	0.20	9.15×10^3	1.19×10^3	6.47×10^2	4.89×10^2
	0.30	1.99×10^5	4.25×10^3	1.51×10^3	9.87×10^2
	0.40	6.75×10^5	6.46×10^3	1.94×10^3	1.19×10^3

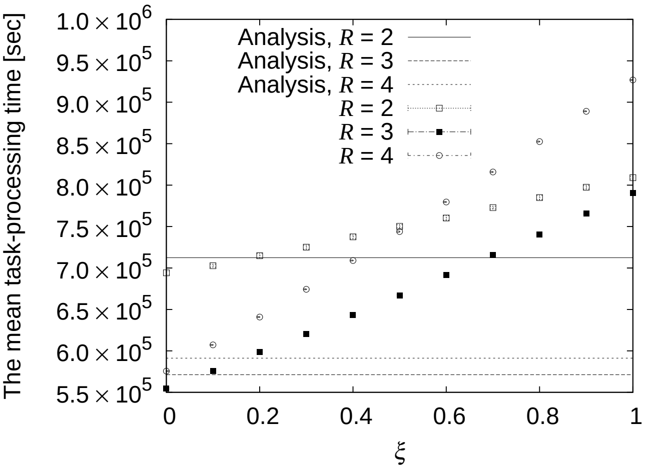


Figure 7: Mean task-processing time with simulation in the case of a Pareto distribution ($\beta = 2.69$, $M = 3 \times 10^5$)

In Figures 7 and 8, the discrepancy between analytical and simulation results is small when ξ is in $[0, 0.1]$. When ξ is greater than 0.1, however, we observe a large discrepancy between them. This indicates that the analytical model is not applicable when the starting time of task replication is not small. We need further study on formulation and analysis for such a kind of task replication.

6. Conclusion

In this paper, we evaluated the efficiency of task-replication scheduling in large-scale parallel-distributed processing from a theoretical point of view. To this end, the mean value and standard deviation of the task-processing time were derived approximately with extreme value theory, whereas the mean total processing time was evaluated exactly, for cases in which the worker-processing time obeys a hyper-exponential, Weibull, or Pareto distribution. Through numerical analysis, we verified the proposed approximations by comparing the exact expressions and the approximate formulas derived by applying extreme value theory. We then calculated the optimal number of replications which achieves the shortest task-processing time using the derived approximate formulas. We compared performance measures between the case of the optimal number of replications and that of no-replication. Finally, we considered the effect of task replication on reducing the standard deviation of

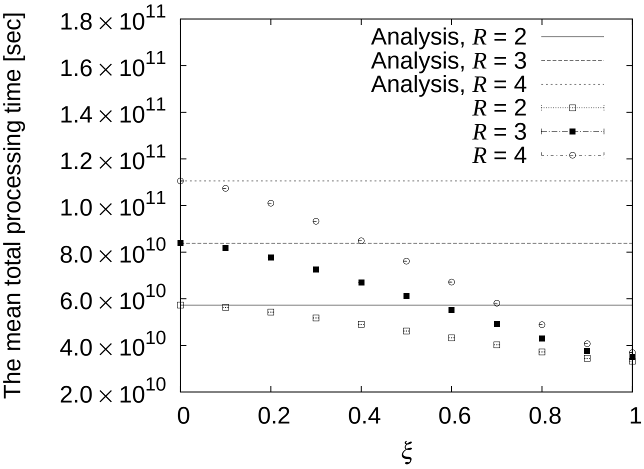


Figure 8: Mean total processing time with simulation in the case of a Pareto distribution ($\beta = 2.69$, $M = 3 \times 10^5$)

the task-processing time. We can claim that the efficiency of task-replication scheduling is improved significantly when the coefficient of variation of the worker-processing time increases. However, this efficiency depends significantly on the tail of the worker-processing time distribution even when the means and variances of the distributions are the same. In addition, we can also claim that the optimal number of replications which achieves the shortest task-processing time mainly depends on the coefficient of variation of the worker-processing time. Furthermore, three replications are effective to guarantee a low variance of the task-processing time, regardless of the tail of the worker-processing time distribution.

The numerical examples also showed that our analytical model is not useful for evaluating the backup-task scheduling for which backup executions are activated when the elapsed time of subtask processing is greater than a pre-specified threshold. This type of backup-task scheduling is implemented in real system such as Hadoop, and we need further development of our analytical model to treat this case. This is our future work.

References

[1] S. Ali, B. Eslamnour, and Z. Shah: A case for on-machine load balancing. *Journal of Parallel and Distributed Computing*, **71** (2011), 556–564.

[2] C. Anglano and M. Canonico: Scheduling algorithms for multiple bag-of-task applications on desktop grids: a knowledge-free approach. *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008)*, (2008), 1–8.

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia: A view of cloud computing. *Communications of the ACM*, **53** (2010), 50–58.

[4] L.A. Barroso and U. Hölzle: *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines* (Morgan & Claypool Publishers, San Rafael, CA, 2009).

[5] W. Cirne, D. Paranhos, F. Brasileiro, and L.F.W. Góes: On the efficacy, efficiency and emergent behavior of task replication in large distributed systems. *Parallel Computing*, **33** (2007), 213–234.

[6] J. Dean and S. Ghemawat: MapReduce: simplified data processing on large clus-

ters. *Communications of the ACM*, **51** (2008), 107–113.

[7] J. Dean: Designs, lessons and advice from building large distributed systems. *Keynote Presentation of the 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware (LADIS 2009)*, (2009).

[8] M. Dobber, R.V.D. Mei, and G. Koole: Dynamic load balancing and job replication in a global-scale grid environment: a comparison. *IEEE Transactions on Parallel and Distributed Systems*, **20** (2009), 207–218.

[9] P. Embrechts, C. Klüppelberg, and T. Mikosch: *Modelling Extremal Events for Insurance and Finance* (Springer, Berlin, 1997).

[10] C.P. Gomes, B. Selman, N. Crato, and H. Kautz: Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, **24** (2000), 67–100.

[11] K. Hashimoto, H. Masuyama, S. Kasahara, and Y. Takahashi: Performance analysis of backup-task scheduling with deadline time in cloud computing. *Journal of Industrial and Management Optimization*, **11** (2015), 867–886.

[12] T. Hirai, H. Masuyama, S. Kasahara, and Y. Takahashi: Effect of backup tasks on response time performance for cloud computing. *Proceedings of the 7th International Conference on Queueing Theory and Network Applications (QTNA 2012)*, (2012). Available: http://infosys.sys.i.kyoto-u.ac.jp/~tsugu/paper/thirai-qtna2012.pdf

[13] T. Hirai, H. Masuyama, S. Kasahara, and Y. Takahashi: Performance analysis of large-scale parallel-distributed processing with backup tasks for cloud computing. *Journal of Industrial and Management Optimization*, **10** (2014), 113–129.

[14] B. Javadi, D. Kondo, J.-M. Vincent, and D.P. Anderson: Discovering statistical models of availability in large distributed systems: an empirical study of SETI@home. *IEEE Transactions on Parallel and Distributed Systems*, **22** (2011), 1896–1903.

[15] D. Kondo, B. Javadi, A. Iosup, and D. Epema: The failure trace archive: enabling comparative analysis of failures in diverse distributed systems. *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CC-GRID 2010)*, (2010), 398–407.

[16] H. Lin, X. Ma, J. Archuleta, W. Feng, M. Gardner, and Z. Zhang: MOON: MapReduce on opportunistic environments. *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC 2010)*, (2010), 95–106.

[17] N. Nóbrega-Júnior, L. Assis, and F. Brasileiro: Scheduling CPU-intensive grid applications using partial information. *Proceedings of the 37th International Conference on Parallel Processing (ICPP 2008)*, (2008), 262–269.

[18] J. O'Loughlin and L. Gillam: Towards performance prediction for public infrastructure clouds: an EC2 case study. *Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2013)*, (2013), 475–480.

[19] F.W.J. Olver, D.W. Lozier, R.F. Boisvert, and C.W. Clark: *NIST Handbook of Mathematical Functions* (Cambridge University Press, Cambridge, 2010).

[20] S.I. Resnick: *Extreme Values, Regular Variation and Point Processes* (Springer, Berlin, 2008).

[21] S. Sakr, A. Liu, D.M. Batista, and M. Alomari: A survey of large scale data management approaches in cloud environments. *IEEE Communications Surveys & Tutorials*, **13** (2011), 311–336.

[22] M. Silberstein: Building an online domain-specific computing service over non-dedicated grid and cloud resources: the superlink-online experience. *Proceedings of the 11th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID 2011)*, (2011), 174–183.

[23] T. White: *Hadoop: The Definitive Guide 2nd edition* (O'Reilly Media, Sebastopol, CA, 2010).

[24] K. Wolter: Stochastic models for restart, rejuvenation and checkpointing. *Habilitation Thesis* (Institute of Computer Science, Humboldt University of Berlin, 2008).

[25] M. Zaharia, A. Konwinski, A.D. Joseph, R. Katz, and I. Stoica: Improving MapReduce performance in heterogeneous environments. *Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2008)*, (2008), 29–42.

A. Preliminary Analysis Results

This section summarizes the preliminary results on basic extreme value theory, which are used in Section 4.

Let $\{X, X_k; k = 1, 2, \dots, n\}$ denote a sequence of i.i.d. random variables with distribution function F, which is non-degenerative, i.e., $F(x_F) = 1$, where $x_F = \sup\{x \in \mathbb{R} := (-\infty, \infty); F(x) < 1\}$.

Let $\overline{X}_n = \max_{1\le k\le n} X_k$ for $n = 1, 2, \dots$. It follows from the fundamental Fisher–Tippett theorem (see, e.g., Theorem 3.2.3 in [9]) that if there exist some $c_n > 0$ and $d_n \in \mathbb{R}$ $(n = 1, 2, \dots)$ such that the distribution of $(\overline{X}_n - d_n)/c_n$ weakly converges to a non-degenerate distribution $\varTheta$, i.e.,

$$\lim_{n\to\infty} \mathsf{P}\left(\frac{\overline{X}_n - d_n}{c_n} \le x\right) = \varTheta(x), \tag{19}$$

for any $x \in \mathbb{R}$ such that $\varTheta$ is continuous, then $\varTheta$ must be one of the following three standard *extreme value distributions*:

$$\begin{array}{lll}
\text{Fréchet :} & \varPhi_\alpha(x) = \begin{cases} 0, & x \le 0, \\ \exp\{-x^{-\alpha}\}, & x > 0, \end{cases} & \alpha > 0, \\
\text{Weibull :} & \varPsi_\alpha(x) = \begin{cases} \exp\{-(-x)^{\alpha}\}, & x \le 0, \\ 1, & x > 0, \end{cases} & \alpha > 0, \\
\text{Gumbel :} & \varLambda(x) = \exp\{-\exp\{-x\}\}, \quad x \in \mathbb{R}. &
\end{array}$$

For simplicity, according to [9], we introduce the following notation.

Definition 1. *The random variable X and its distribution F are said to be in the maximum domain of attraction of the extreme value distribution $\varTheta$ (denoted by $X \in \mathrm{MDA}(\varTheta)$ and $F \in \mathrm{MDA}(\varTheta)$) if there exist some $c_n > 0$ and $d_n \in \mathbb{R}$ $(n = 1, 2, \dots)$ such that (19) holds.*

In Section 4, we use asymptotic results associated with the two classes $\mathrm{MDA}(\varLambda)$ and $\mathrm{MDA}(\varPhi_\alpha)$, which are described in Subsections A.1 and A.2.

A.1. Maximum domain of attraction of a Gumbel distribution

Proposition A.1 (Theorem 3.3.26 in [9]). *$F \in \mathrm{MDA}(\varLambda)$ if and only if there exists some $x_0 < x_F$ such that F has the following representation:*

$$1 - F(x) = c(x) \exp\left\{-\int_{x_0}^{x} \frac{g(t)}{a(t)} \mathrm{d}t\right\}, \qquad x_0 < x < x_F, \tag{20}$$

where c and g are measurable functions such that $\lim_{x\uparrow x_F} c(x) = c > 0$ and $\lim_{x\uparrow x_F} g(x) = 1$; and where $a(\cdot) > 0$ is an absolutely continuous function with respect to the Lebesgue measure and its density $a'(\cdot)$ satisfies $\lim_{x\uparrow x_F} a'(x) = 0$. In addition, we can choose the normalizing constants c_n and d_n in (19) as follows:

$$c_n = a(d_n), \qquad d_n = F^{-1}\left(1 - \frac{1}{n}\right), \tag{21}$$

where $F^{-1}(x) = \inf\{y; F(y) \geq x\}$.

Proposition A.2 (Proposition 2.1 (iii) in [20]). *If $F \in \mathrm{MDA}(\Lambda)$ and*

$$\int_{-\infty}^{0} |x|^k \mathrm{d}F(x) < \infty,$$

for some integer $k > 0$, then

$$\lim_{n\to\infty} \mathsf{E}\left[\left(\frac{\overline{X}_n - d_n}{c_n}\right)^k\right] = (-1)^k \lim_{x\to 1} \frac{\mathrm{d}^k}{\mathrm{d}x^k}\Gamma(x),$$

where Γ denotes the Gamma function and c_n and d_n are given by (21).

A.2. Maximum domain of attraction of a Fréchet distribution

Proposition A.3 (Theorem 3.3.7 in [9]). *$F \in \mathrm{MDA}(\Phi_\alpha)$ if and only if the tail distribution $1 - F$ is regularly varying with index $-\alpha$, i.e.,*

$$\lim_{x\to\infty} \frac{1 - F(xt)}{1 - F(x)} = t^{-\alpha}, \qquad t > 0. \tag{22}$$

The normalizing constants c_n and d_n can be chosen as

$$c_n = F^{-1}\left(1 - \frac{1}{n}\right), \qquad d_n = 0. \tag{23}$$

Proposition A.4 (Proposition 2.1 (i) in [20]). *If $F \in \mathrm{MDA}(\Phi_\alpha)$ and*

$$\int_{-\infty}^{0} |x|^k \mathrm{d}F(x) < \infty,$$

for some integer $0 < k < \alpha$, then

$$\lim_{n\to\infty} \mathsf{E}\left[\left(\frac{\overline{X}_n}{c_n}\right)^k\right] = \Gamma\left(1 - \frac{k}{\alpha}\right),$$

where c_n is given by (23).

Tsuguhito Hirai
Graduate School of Informatics
Kyoto University
Yoshida-hommachi, Sakyo-ku
Kyoto 606-8501, Japan
E-mail: tsugu@sys.i.kyoto-u.ac.jp