Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **20XX**

doi:10.3934/xx.xx.xx.xx
pp. **X–XX**

PERFORMANCE ANALYSIS OF LARGE-SCALE PARALLEL-DISTRIBUTED PROCESSING WITH BACKUP TASKS FOR CLOUD COMPUTING

Tsuguhito Hirai and Hiroyuki Masuyama

Graduate School of Informatics, Kyoto University
Yoshida-Hommachi, Sakyo-ku
Kyoto 606-8501, JAPAN

Shoji Kasahara

Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma
Nara 630-0192, JAPAN

Yutaka Takahashi

Graduate School of Informatics, Kyoto University
Yoshida-Hommachi, Sakyo-ku
Kyoto 606-8501, JAPAN

Abstract. In cloud computing, a large-scale parallel-distributed processing service is provided where a huge task is split into a number of subtasks and those subtasks are processed on a cluster of machines called workers. In such a processing service, a worker which takes a long time for processing a subtask makes the response time long (the issue of stragglers). One of efficient methods to alleviate this issue is to execute the same subtask by another worker in preparation for the slow worker (backup tasks). In this paper, we consider the efficiency of backup tasks. We model the task-scheduling server as a single-server queue, in which the server consists of a number of workers. When a task enters the server, the task is split into subtasks, and each subtask is served by its own worker and an alternative distinct worker. In this processing, we explicitly derive task processing time distributions for the two cases that the subtask processing time of a worker obeys Weibull or Pareto distribution. We compare the mean response time and the total processing time under backup-task scheduling with those under normal scheduling. Numerical examples show that the efficiency of backup-task scheduling significantly depends on workers' processing time distribution.

2010 *Mathematics Subject Classification.* Primary: 60K25, 60J10; Secondary: 68M20.

Key words and phrases. Cloud computing, Parallel-distributed processing, Task scheduling, Backup tasks, Queueing analysis.

The reviewing process of the paper was handled by Wuyi Yue as Guest Editor.

This is a pre-copy-editing, author-produced PDF of an article accepted for publication in Journal of Industrial and Management Optimization following peer review. The definitive publisher-authenticated version, 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*, vol. 10, no. 1, pp. 113–129, 2014, is available online at doi:10.3934/jimo.2014.10.113.

1. **Introduction.** Recently, cloud computing has attracted considerable attention due to the availability of huge computing resources and its significant cost efficiency. In [1], 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 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 are extremely large-sized.

With the increase in the capacity of hard-disk drives, the amount of data treated in computer processing also increases, and it is common that enormous time is needed in data processing with one worker machine. In cloud computing, enormous data is handled with a huge number of workers in parallel-distributed processing fashion [4, 2], and typical applications are data mining, document processing, and machine learning. In the following, we call this processing mechanism as a large-scale parallel-distributed processing. With a large-scale parallel-distributed processing, huge data is processed in relatively short time. Roughly speaking, data processing which needs 100 hours with one worker can process in one hour with 100 workers if overhead for parallel processing is relatively small.

In the large-scale parallel-distributed processing, a huge task is split into a number of subtasks and those subtasks are independently processed on a cluster of machines called workers. The huge task processing ends when all the subtasks are completed. Therefore, a worker machine which takes a long time for processing a subtask increases the response time (the issue of stragglers) [4]. The reasons causing slow workers are machine failure and resource competition because the system consists of a huge number of machines.

In order to alleviate the issue of stragglers, there exist two scheduling schemes: load balancing [6] and backup tasks [4]. In load balancing, the subtask size for a worker is determined according to the processing speed of the worker. That is, a small-sized subtask is allocated to a slow-processing worker, while a large-sized subtask is performed by a fast-processing worker. This scheduling makes the variance of the subtask-processing times of worker machines significantly small. However, the load-balancing scheduler must know each worker's subtask-processing time a priori.

In backup-task scheduling, on the other hand, backup executions of the remaining in-progress subtasks are scheduled. Then the process of a subtask ends when either original subtask or backup executions is completed. A strong point of the backup-task scheduling is that the backup-task scheduler activates backup tasks for a worker according to the elapsed subtask-processing time, i.e., no a priori information about the overall subtask-proceeding time of the worker is needed. It is reported in [4] that the backup tasks can significantly reduce the response time of a task.

There is much literature on cloud computing, and most of works are concerned with service platform and cost efficiency from the economical point of view. There are a few studies for performance issues on cloud computing, and those are based on measurement-based analysis. (See, for example, [5, 9].) In terms of the theoretical approach to performance issues on cloud computing, Xiong et al. [8] consider a queueing network model which consists of a Web-server queue and a service-center queue. Focusing on the percentile of the response time as a performance measure of cloud computing, they approximately analyze the response time distribution. In their model, however, the service-center part is modeled as a single-server queue with a fixed service rate, and this model is too simple to describe a large-scale parallel-distributed processing of a task.

In [6], Dobber et al. investigate the effectiveness of dynamic load balancing (DLB) and job replication (JR) by trace-driven simulation experiments, proposing a hybrid scheduling scheme of DLB and JR. Cirne et al. [3] also investigate the effectiveness of several job-replication schedulers by simulation, comparing those with traditional information-based schedulers. Note that most of related works are concerned with the performance of task schedulers by simulation. To the best of the authors' knowledge, the effect of backup-task scheduling on the improvement of the task-processing time has not been fully studied yet.

In this paper, we consider the efficiency of backup-task scheduling on two performance measures: the response time of a task and the total processing time of workers. Note that the former indicates how the performance is improved by backup tasks, while the latter characterizes the cost resulting from backup tasks. We focus on a task-scheduling server in which tasks are processed in first-come, first-served (FCFS) order. We model the task-scheduling server as a single-server queue, in which the server consists of a number of workers. A task entering the service facility is split into subtasks of an equal size[1]. Then, the task service ends when all the subtasks are completed.

We consider two task-scheduling policies: normal scheduling and backup-task one. For normal scheduling, each subtask is served by its own worker. In backup-task scheduling, on the other hand, each subtask is processed not only by its own worker but also by an alternative distinct worker, and the subtask service ends when either of the two workers' processes is completed. In both scheduling policies, we explicitly derive task processing time distributions when the subtask processing time of a worker follows Weibull or Pareto distribution. Then, the maximum throughput, mean response time, and total processing time are derived. In numerical examples, we validate the analysis by Monte Carlo simulation. Then, we compare these performance measures under backup-task scheduling with that under normal scheduling, discussing the efficiency of backup-task scheduling.

This paper is organized as follows. In Section 2, analytical models for two scheduling policies are described. In Section 3, we derive performance measures. Section 4 shows numerical examples of derived performance measures. Finally, we conclude the paper in Section 5.

2. Analytical models for two scheduling policies.

We consider two analytical models for the large-scale parallel-distributed processing: *normal processing model* and *backup-task processing model* (called Models N and B, respectively, hereafter). In each model, the system consists of an infinite buffer and a server with workers. Tasks arrive at the system according to a Poisson process with rate λ, and they are processed on the FCFS basis.

The details of the two models are as follows:

(i) Normal processing model (Model N)
The server has $2M$ (which is a positive integer) workers, and a task is divided into $2M$ subtasks. Each subtask is processed by a worker, and its processing time follows a distribution function $F_{\rm N}$ with mean $b/(2M)$ ($b > 0$), independently of those of the other workers (Fig. 1). Further, the processing time of a

[1] The assumption of subtasks of an equal size becomes accurate when a huge-sized input data is almost equally split into data pieces [4].

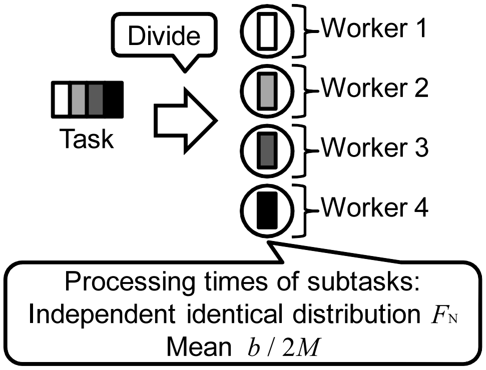
Divide
Task
Worker 1
Worker 2
Worker 3
Worker 4
Processing times of subtasks:
Independent identical distribution F_N
Mean $b/2M$

FIGURE 1. Normal processing model (Model N).

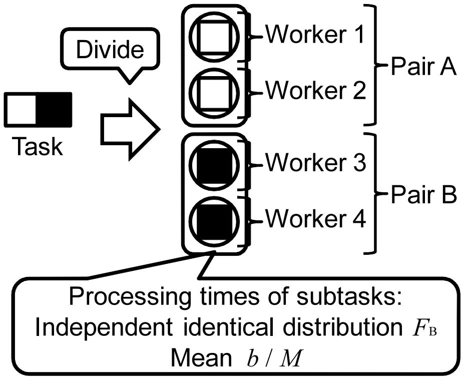
Divide
Task
Worker 1
Worker 2
Pair A
Worker 3
Worker 4
Pair B
Processing times of subtasks:
Independent identical distribution F_B
Mean b/M

FIGURE 2. Backup-task processing model (Model B).

task (consisting of $2M$ subtasks) is defined as the maximum of the processing times of the $2M$ subtasks generated from the task. As a result, the processing times of tasks are independently and identically distributed (i.i.d.) with a distribution function $G_{\rm N}$, which is given by

$$G_{\rm N}(t) = \{F_{\rm N}(t)\}^{2M}, \qquad t \ge 0.$$

(ii) Backup-task processing model (Model B)
The server consists of M pairs of workers. A task is divided into M subtasks, from each of which a backup subtask is duplicated. An original subtask and its backup subtask are assigned to a pair of workers separately. The processing times of the $2M$ subtasks (including M backup subtasks) generated from a task are i.i.d. with a distribution function $F_{\rm B}$ with mean b/M (Fig. 2). The processing of each pair of an original subtask and its backup subtask is finished when either of them is completed. The processing time of a task is defined in the same way as Model N. Thus, the processing times of tasks are i.i.d. with a distribution function $G_{\rm B}$, which is given by

$$G_{\rm B}(t) = \left[1 - \{1 - F_{\rm B}(t)\}^2\right]^M, \qquad t \ge 0.$$

In what follows, we call $F_{\rm N}$ and $F_{\rm B}$ as *worker-processing-time distributions.*

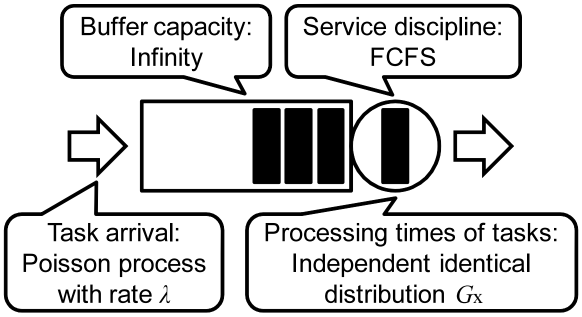
Buffer capacity: Infinity
Service discipline: FCFS
Task arrival: Poisson process with rate λ
Processing times of tasks: Independent identical distribution Gx

Figure 3. Analytical model for a large-scale parallel-distributed processing.

3. Analysis.

3.1. **Performance measures.** We consider three performance measures: the maximum throughput, mean response time, and total processing time. The maximum throughput is defined as the reciprocal of the mean processing time of a task; the mean response time as the mean sojourn time of a task in the system (from its arrival to its departure); and the total processing time as the mean of the total running time of the $2M$ workers during the processing time of a task.

Let subscript "x" denote the index symbol for the two processing models described in the previous subsection, i.e., $\mathrm{x} = \mathrm{N}$ or B. Let T_{x}, W_{x} and P_{x} $(\mathrm{x} = \mathrm{N}, \mathrm{B})$ denote the maximum throughput, mean response time and total processing time, respectively, in Model x. Note here that Model x is considered as an FCFS M/G/1 queue (Fig. 3), where the service time distribution is given by G_{x}. We then have (see, e.g., [7])

$$T_{\mathrm{x}} = \frac{1}{g_{\mathrm{x}}^{(1)}}, \qquad W_{\mathrm{x}} = \frac{\lambda g_{\mathrm{x}}^{(2)}}{2(1-\lambda g_{\mathrm{x}}^{(1)})} + g_{\mathrm{x}}^{(1)}, \qquad \mathrm{x} = \mathrm{N}, \mathrm{B}, \tag{1}$$

where $g_{\mathrm{x}}^{(1)}$ and $g_{\mathrm{x}}^{(2)}$ denote the first and second moments of distribution function G_{x}, i.e.,

$$g_{\mathrm{x}}^{(1)} = \int_0^\infty t \mathrm{d}G_{\mathrm{x}}(t), \qquad g_{\mathrm{x}}^{(2)} = \int_0^\infty t^2 \mathrm{d}G_{\mathrm{x}}(t),$$

respectively. From the definition, we also have

$$P_{\mathrm{x}} = \begin{cases} \mathsf{E}\left[\sum_{i=1}^{2M} U_{\mathrm{x},i}\right] = 2M\mathsf{E}[U_{\mathrm{x},1}], & \mathrm{x} = \mathrm{N}, \\ \mathsf{E}\left[\sum_{i=1}^{M} 2\min\{U_{\mathrm{x},i}, U_{\mathrm{x},2i}\}\right] = 2M\mathsf{E}[\min\{U_{\mathrm{x},1}, U_{\mathrm{x},2}\}], & \mathrm{x} = \mathrm{B}, \end{cases}$$

where $U_{\mathrm{x},i}$ $(i = 1, 2, \dots, 2M)$ denotes the subtask-processing-time of ith worker. Note that the $U_{\mathrm{x},i}$'s are i.i.d. random variables with distribution function F_{x}.

3.2. **Special cases for worker-processing-time distribution.** We consider two types of the worker-processing-time distributions. For convenience, let $M_{\mathrm{N}} = 2M$ and $M_{\mathrm{B}} = M$.

(i) Weibull distribution

$$F_{\mathrm{x}}(t) = 1 - \exp\left\{-\left(t/\eta_{\mathrm{x}}\right)^{\alpha}\right\}, \qquad t \ge 0,$$

with scale parameter $\eta_{\mathrm{x}} = b/\{M_{\mathrm{x}}\mathit{\Gamma}(1+1/\alpha)\}$ $(\alpha > 0)$;

(ii) Pareto distribution

$$F_{\mathrm{x}}(t) = \begin{cases} 1 - \left(\mu_{\mathrm{x}}/t\right)^{\beta}, & t \ge \mu_{\mathrm{x}}, \\ 0, & 0 \le t < \mu_{\mathrm{x}}, \end{cases}$$

where $\mu_{\mathrm{x}} = b(\beta-1)/(M_{\mathrm{x}}\beta)$ $(\beta > 2)$.

For the two worker-processing-time distributions, we can calculate the moments $g_{\mathrm{N}}^{(n)}$ and $g_{\mathrm{B}}^{(n)}$ $(n = 1, 2)$ as follows:

(i) Weibull processing time case:

(a) Model N

$$g_{\mathrm{N}}^{(1)} = \frac{b}{2M}\sum_{k=1}^{2M}(-1)^{k-1}\binom{2M}{k}\frac{1}{k^{\frac{1}{\alpha}}},$$

$$g_{\mathrm{N}}^{(2)} = \frac{\mathit{\Gamma}\left(1+\frac{2}{\alpha}\right)}{\left\{\mathit{\Gamma}\left(1+\frac{1}{\alpha}\right)\right\}^{2}}\frac{b^{2}}{4M^{2}}\sum_{k=1}^{2M}(-1)^{k-1}\binom{2M}{k}\frac{1}{k^{\frac{2}{\alpha}}}.$$

(b) Model B

$$g_{\mathrm{B}}^{(1)} = \frac{b}{2^{\frac{1}{\alpha}}M}\sum_{k=1}^{M}(-1)^{k-1}\binom{M}{k}\frac{1}{k^{\frac{1}{\alpha}}},$$

$$g_{\mathrm{B}}^{(2)} = \frac{\mathit{\Gamma}\left(1+\frac{2}{\alpha}\right)}{\left\{\mathit{\Gamma}\left(1+\frac{1}{\alpha}\right)\right\}^{2}}\frac{b^{2}}{2^{\frac{2}{\alpha}}M^{2}}\sum_{k=1}^{M}(-1)^{k-1}\binom{M}{k}\frac{1}{k^{\frac{2}{\alpha}}}.$$

(ii) Pareto processing time case:

(a) Model N

$$g_{\mathrm{N}}^{(1)} = (\beta-1)\frac{b}{2M}\sum_{k=1}^{2M}(-1)^{k-1}\binom{2M}{k}\frac{k}{\beta k-1},$$

$$g_{\mathrm{N}}^{(2)} = \frac{(\beta-1)^{2}}{\beta}\frac{b^{2}}{4M^{2}}\sum_{k=1}^{2M}(-1)^{k-1}\binom{2M}{k}\frac{k}{\beta k-2}.$$

(b) Model B

$$g_{\mathrm{B}}^{(1)} = 2(\beta-1)\frac{b}{M}\sum_{k=1}^{M}(-1)^{k-1}\binom{M}{k}\frac{k}{2\beta k-1},$$

$$g_{\mathrm{B}}^{(2)} = \frac{2(\beta-1)^{2}}{\beta}\frac{b^{2}}{M^{2}}\sum_{k=1}^{M}(-1)^{k-1}\binom{M}{k}\frac{k}{2\beta k-2}.$$

Combining (1) with the above equations, we can obtain T_{x} and W_{x}. We can also calculate P_{x} as follows:

(i) Weibull processing time case:

$$P_{\mathrm{x}} = \begin{cases} b, & \mathrm{x} = \mathrm{N}, \\ 2^{\left(1-\frac{1}{\alpha}\right)}b, & \mathrm{x} = \mathrm{B}. \end{cases}$$

TABLE 1. Values of α and β.

Coefficient of variation	8.307	2.236	1.000	0.5227	0.2805
α	0.2500	0.5000	1.000	2.000	4.000
β	2.007	2.095	2.414	3.159	4.702

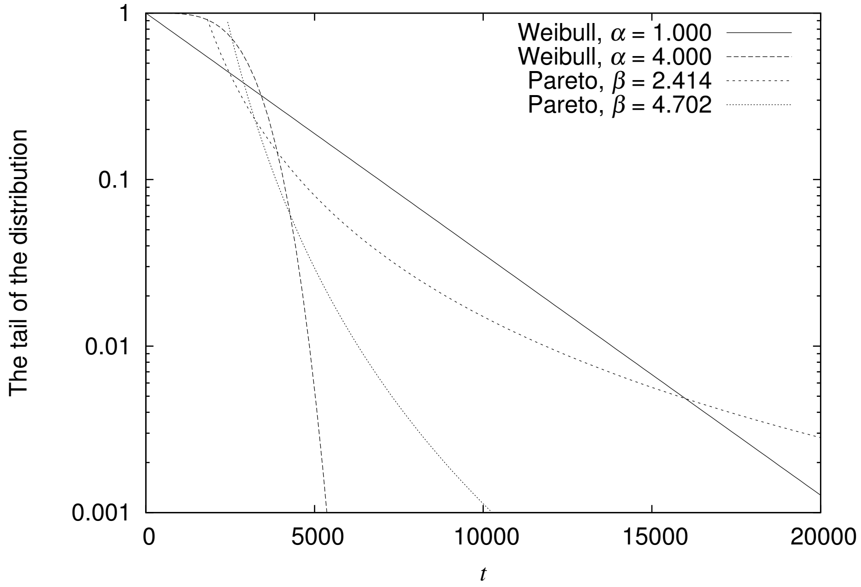

FIGURE 4. The tail of the worker-processing-time distribution.

(ii) Pareto processing time case:

$$P_{\mathrm{x}} = \begin{cases} b, & \mathrm{x} = \mathrm{N}, \\ \dfrac{4(\beta-1)}{2\beta-1}b, & \mathrm{x} = \mathrm{B}. \end{cases}$$

4. **Numerical examples.** In this section, we show some numerical examples. First, we discuss the model validity by comparing the analytical results and Monte Carlo simulation. Then, we consider the effectiveness of backup tasks for improving the issue of stragglers by comparing Models N and B (i.e., the normal processing model and the backup-task processing model).

In the following numerical examples, we set $b = 3.000 \times 10^7$ (sec) (i.e. about a year) and $\lambda = 3.000 \times 10^{-8}$ (task/sec). Furthermore, M is varied from 1 to 10000, and the values of α and β are determined such that the coefficient of variation of the worker-processing-time distribution takes the values as shown in Table 1. Note that the coefficient of variation for Weibull (resp. Pareto) distribution becomes large with the decrease in α (resp. β), and the tail of Pareto distribution is heavier than that of Weibull distribution, although the coefficients of variation are the same (See Fig. 4).

4.1. **Model validation.** In this subsection, we discuss the model validation. In our model of backup tasks (Model B), we assume that alternative subtasks are simultaneously executed when a task process starts. In real environment, on the other hand, backup executions are activated when subtask processing times are greater than a pre-specified threshold. In order to validate our analytical model, we conducted Monte Carlo simulation experiments. In our simulation setting, a backup execution of a subtask starts when its processing time is greater than $\xi b/M$. Here, ξ is set to 1.0, 1.5 and 2.0. We calculated the 95% confidence interval of the performance measures of the maximum throughput, mean response time, and total processing time.

Figure 5 represents the mean response time for Pareto-processing time case ($\beta = 2.007$) against the number of workers in log-log plot. In Fig. 5, the mean response time for Model B is smaller than that for simulation, and the difference between them decreases with the increase in the number of workers. This implies that our model gives a lower bound of the backup-task scheduling on the mean response time, and the model assumption is valid when the number of workers is large. This trend can be seen for other parameters of Weibull and Pareto distributions. We also confirm the same trend on the maximum throughput.

Figure 6 (resp. Figure 7) illustrates the total processing time in Model B and simulation for Pareto-processing time case ($\beta = 2.007$) (resp. Weibull-processing time case ($\alpha = 0.2500$)). The horizontal axis represents the number of workers in log scale. In both figures, we observe that the total processing time remains almost constant when the number of workers increases. This is because the size of a task is a constant b and independent of the number of subtasks. Noting that the overhead of parallel processing is not taken into consideration in both analysis and simulation, the resulting total processing time is almost insensitive to the number of workers. In Fig. 6, the total processing time for Model B is larger than that for simulation, and the difference grows when the backup-task execution threshold increases. This result suggests that our model gives the worst case on resource consumption. This tendency can be seen for other parameters of Pareto distribution and Weibull distribution with $\alpha \geq 1$.

In Fig. 7, on the other hand, the total processing time for Model B is smaller than that for simulation, and the difference between them decreases with the decrease in the backup-task execution threshold. This implies that resource consumption is the smallest when backup tasks are activated from the beginning of the task processing. This trend is the same for Weibull distribution with $\alpha < 1$.

These results indicate that the maximum throughput and mean response time can be predicted quantitatively with our model when the number of workers is large. On the other hand, our model is not suitable for quantitative evaluation of the total processing time. However, the qualitative trend of the total processing time can be described well by this model.

4.2. **Impact of stragglers in Model N.**

In this subsection, we investigate the issue of stragglers by the normal processing model (Model N).

Figures 8 and 9 represent the maximum throughput for Model N against the number of workers in log-log plot. Here, the worker processing time distribution is set to Weibull (resp. Pareto) distribution in Fig. 8 (resp. Fig. 9). It is observed in Fig. 8 (resp. Fig. 9) that when α (resp. β) is small, the maximum throughput is less likely to grow with the increase in the number of workers. This is because slow workers are more likely to exist with the increase in the coefficient of variation, resulting in that the response time of a task is not significantly improved.

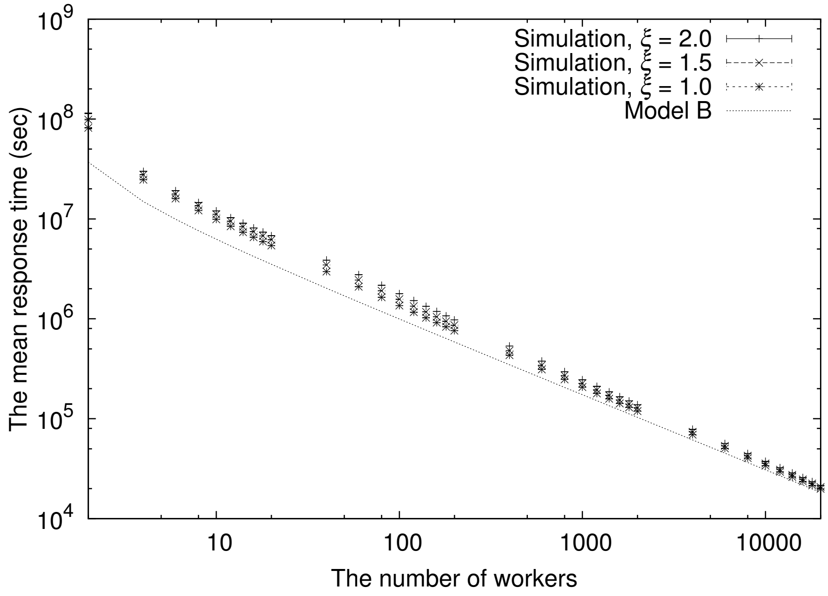

FIGURE 5. The mean response time for Model B and simulation in Pareto distribution case ($\beta = 2.007$).

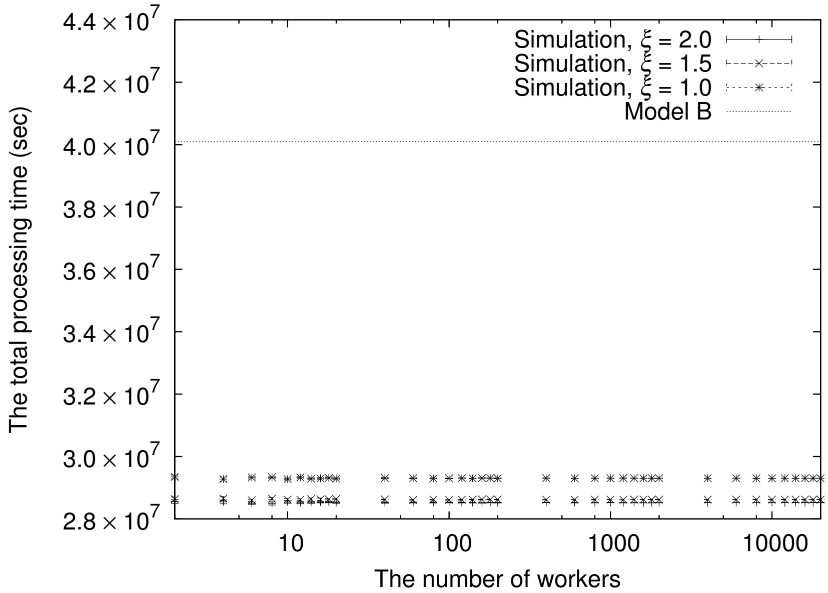

FIGURE 6. The total processing time for Model B and simulation in Pareto distribution case ($\beta = 2.007$).

Figures 10 and 11 illustrate the mean response time for Model N against the number of workers in log-log plot. In both figures, the mean response time is calculated for five values of distribution parameters. Figure 10 is Weibull-distribution

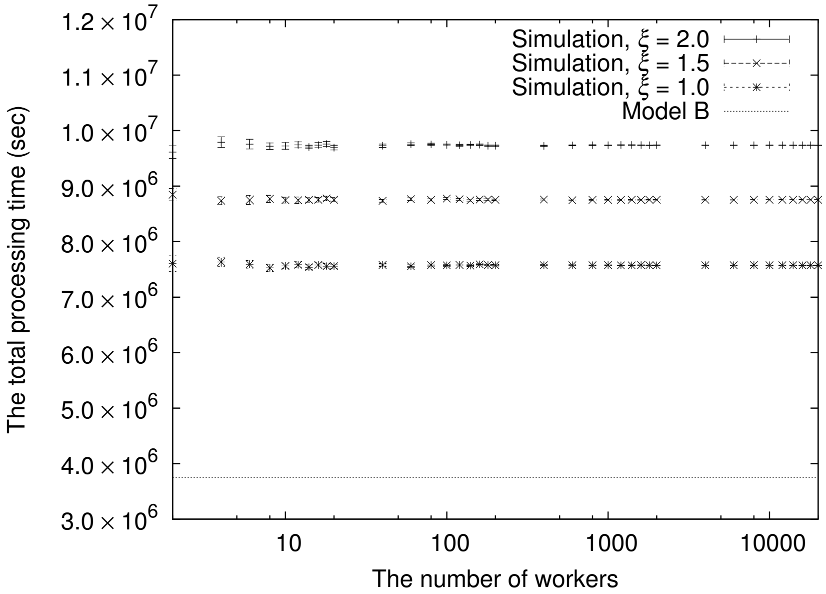

FIGURE 7. The total processing time for Model B and simulation in Weibull distribution case ($\alpha = 0.2500$).

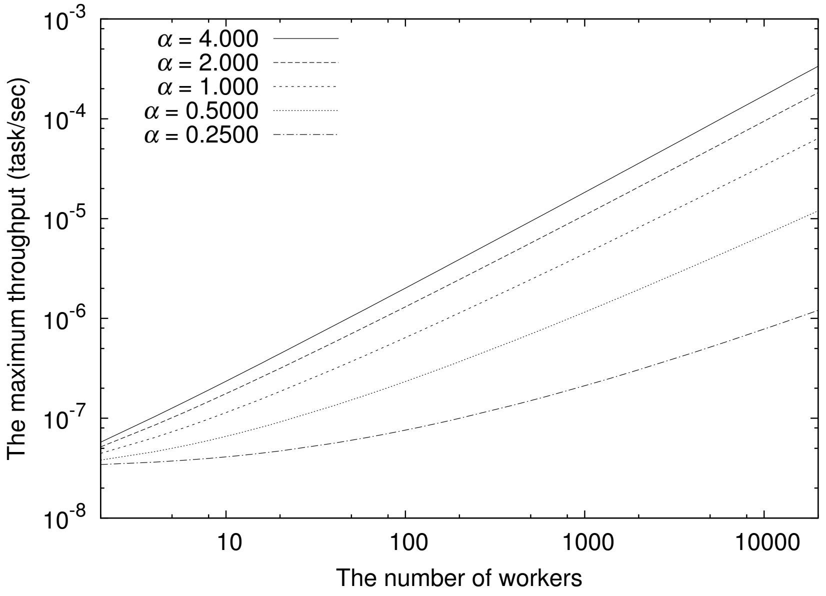

FIGURE 8. The maximum throughput for Model N in Weibull distribution case.

case, while Fig. 11 is Pareto-distribution case. In both figures, we observe that the mean response time in case of a large coefficient variation is less likely to decrease

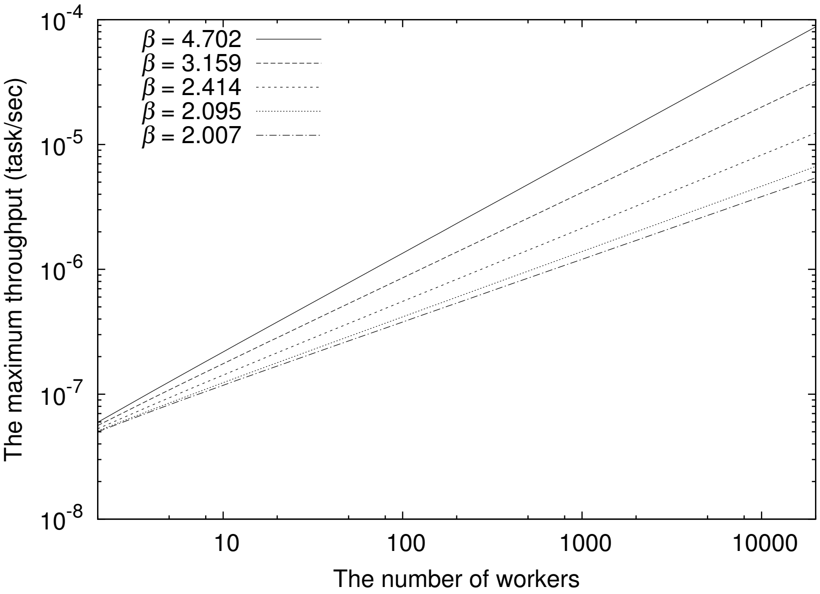

Figure 9. The maximum throughput for Model N in Pareto distribution case.

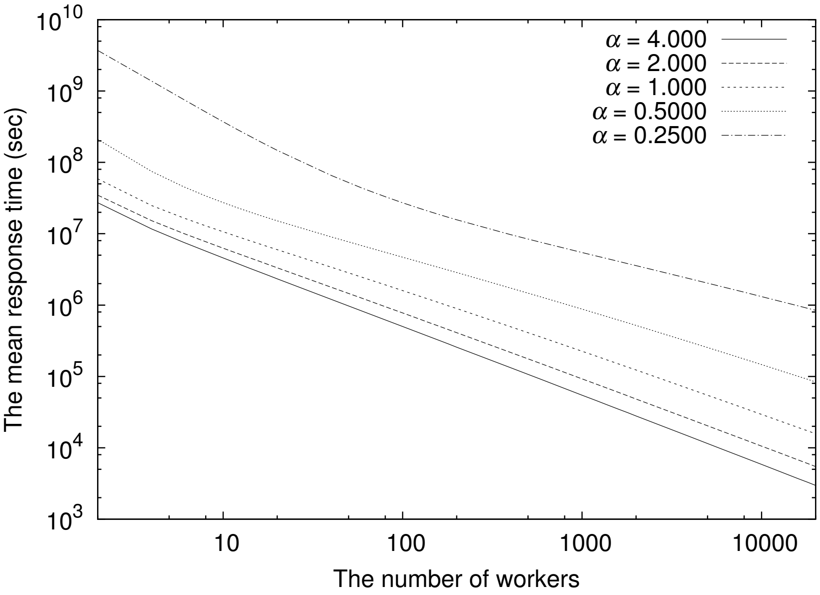

Figure 10. The mean response time for Model N in Weibull distribution case.

with the increase in the number of workers. This reason is the same as that of the maximum throughput.

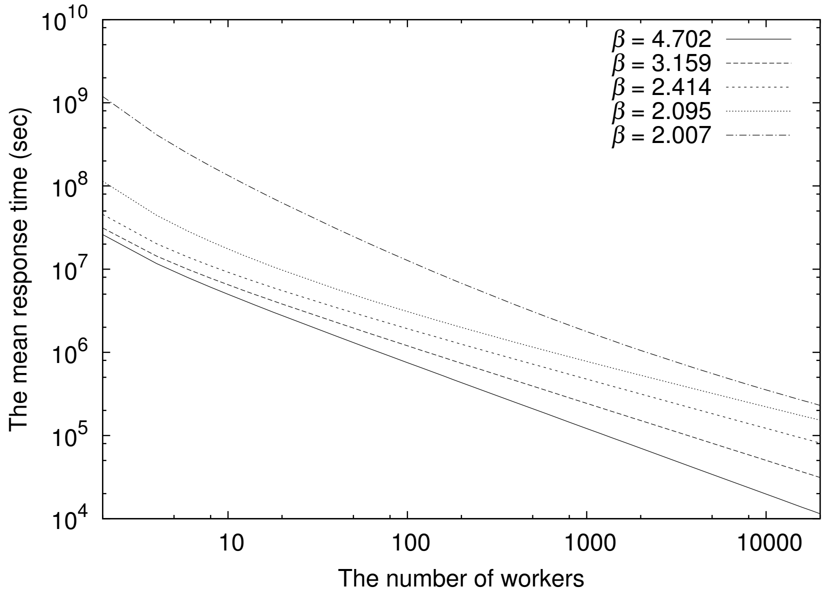

FIGURE 11. The mean response time for Model N in Pareto distribution case.

These results suggest that the coefficient of variation of the worker processing time distribution significantly affects the response time performance of the large-scale parallel-distributed processing.

4.3. **Efficiency of backup-task scheduling.** In this subsection, we investigate the effect of backup-task scheduling on the performance of the large-scale parallel-distributed processing.

Figure 12 (resp. Figure 13) represents the ratio of the maximum throughput in Model B to that in Model N for Weibull-processing (resp. Pareto-processing) time case. The horizontal axis represents the number of workers in log scale, and the ratio in Fig. 12 (resp. Fig. 13) is calculated in five cases of α (resp. β). In Fig. 12, the ratio gradually decreases with the increase in the number of workers. This implies that under backup-task scheduling, increasing the number of workers does not improve the throughput performance effectively when the worker processing time follows Weibull distribution. We also observe in this figure that the ratio for a small α is significantly large, as expected.

In Fig. 13, on the other hand, the ratio grows when the number of workers is large. In addition, the ratio for a small β is larger than that for a large β. Note that the event where the worker processing time is extremely large is likely to occur for Pareto distribution. Therefore, these results suggest that backup-task scheduling is significantly effective for improving the throughput performance when the event of an extremely-large worker processing time is likely to occur.

Figure 14 (resp. Figure 15) illustrates the ratio of the mean response time in Model B to that in Model N for Weibull-processing (resp. Pareto-processing) time case. The horizontal axis represents the number of workers in log scale, and the ratio in Fig. 14 (resp. Fig. 15) is calculated in five cases of α (resp. β).

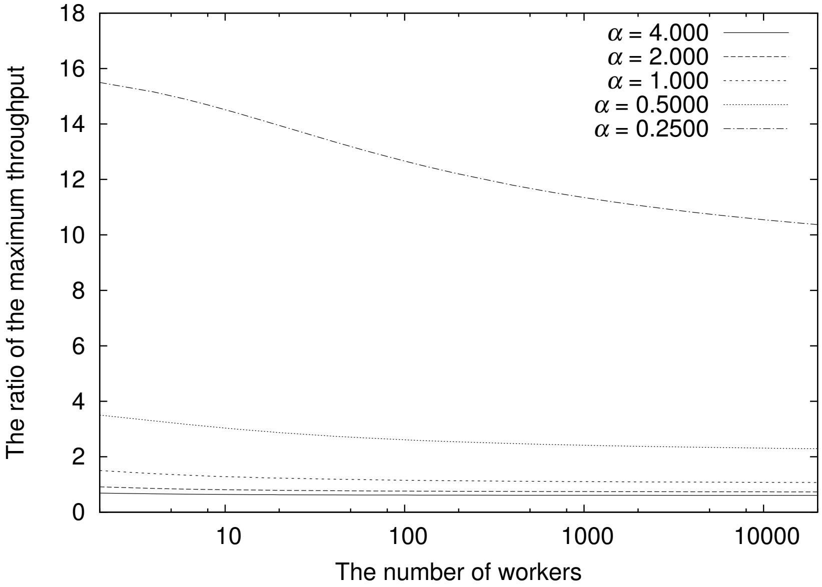

FIGURE 12. The ratio of the maximum throughput in Model B to that in Model N in Weibull distribution case.

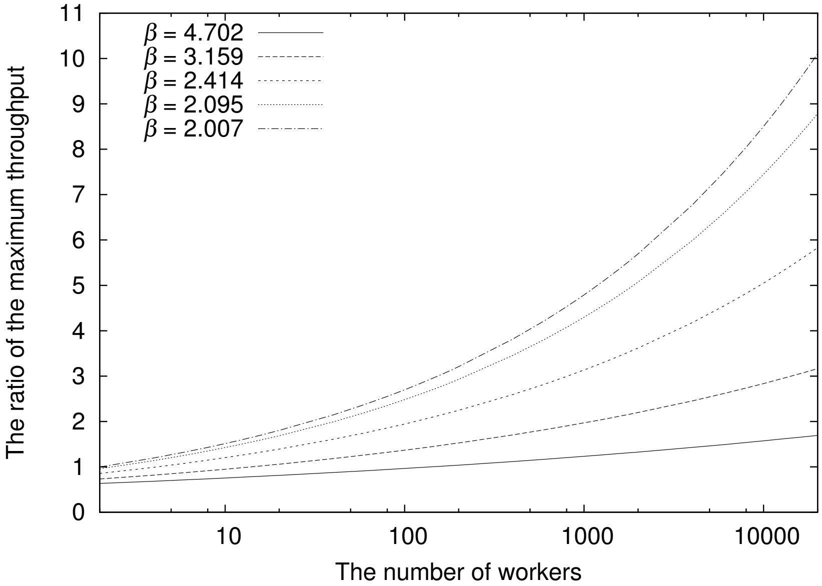

FIGURE 13. The ratio of the maximum throughput in Model B to that in Model N in Pareto distribution case.

In Fig. 14, the ratio remains almost constant with the increase in the number of workers. In addition, the mean response time for Model N is smaller than that

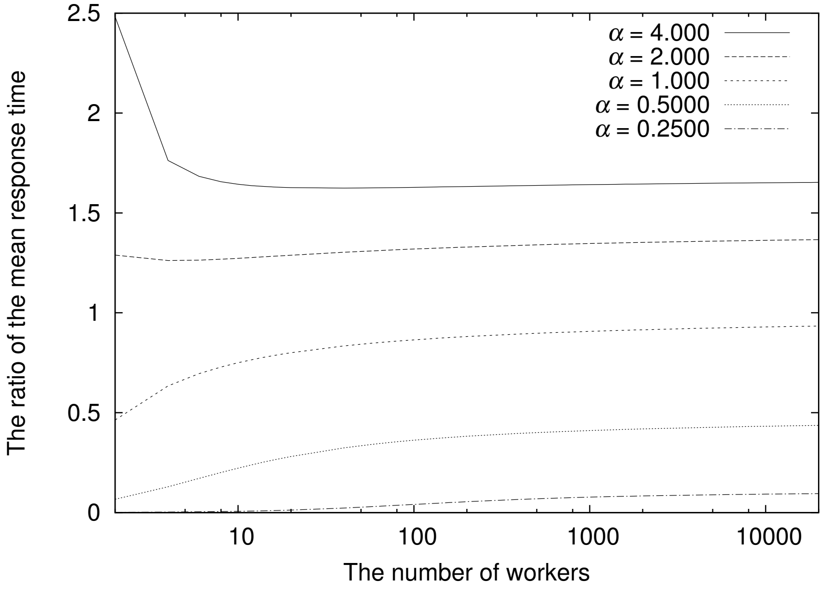

FIGURE 14. The ratio of the mean response time in Model B to that in Model N in Weibull distribution case.

for Model B when $\alpha = 2.000$ and 4.000. These results suggest that when the system is managed by backup-task scheduling and the worker processing time follows Weibull distribution, increasing the number of workers is not significantly effective in improving the response time. Note that when $\alpha = 0.2500$, 0.5000 and 1.000, the ratio is smaller than one, implying that the mean response time for Model B is smaller than that for Model N. Therefore, even for the Weibull-processing time case, backup-task scheduling can improve the performance when its coefficient of variation is large. In Fig. 15, on the other hand, the ratio decreases with the increase in the number of workers. This implies that backup-task scheduling works significantly well for Pareto-processing time case. Note that in both the figures, the ratio for the large coefficient of variation case is significantly small for any number of workers.

Figure 16 (resp. Figure 17) represents the ratio of the total processing time in Model B to that in Model N for Weibull-processing (resp. Pareto-processing) time case. The horizontal axis represents the number of workers in log scale, and the ratio in Fig. 16 (resp. Fig. 17) is calculated in five cases of α (resp. β). In both figures, the ratio is constant with the increase in the number of workers, and the ratio for a large α (resp. β) is greater than that for a small α (resp. β). That is, backup-task scheduling increases the resource consumption when the variance of the worker processing time becomes small. Remarkably, in Fig. 16, the ratio is less than one for $\alpha < 1$ and the total processing time for Model B is less than that for Model N. On the other hand, in Fig. 17, the ratio is always greater than one, and backup-task scheduling increases the total processing time compared with that of normal scheduling. This implies that backup-task scheduling can reduce the resource consumption when the worker processing time follows Weibull distribution with $\alpha < 1$.

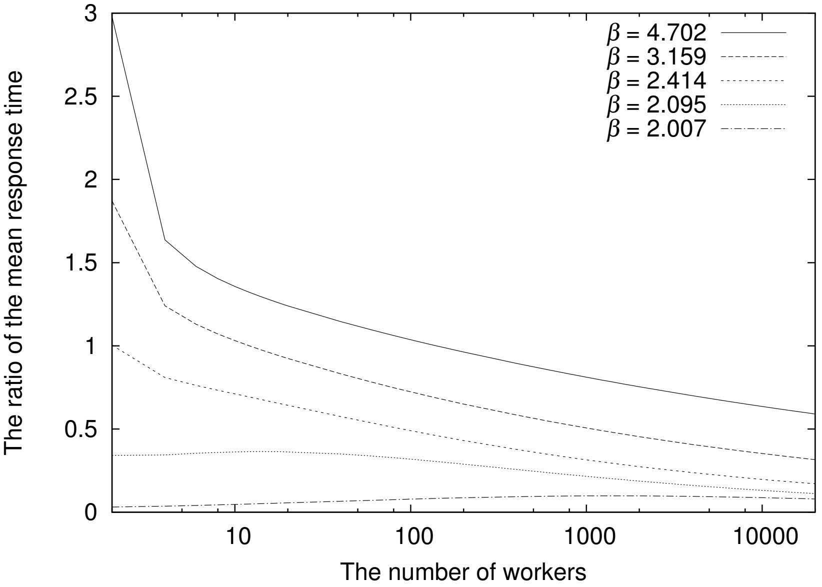

FIGURE 15. The ratio of the mean response time in Model B to that in Model N in Pareto distribution case.

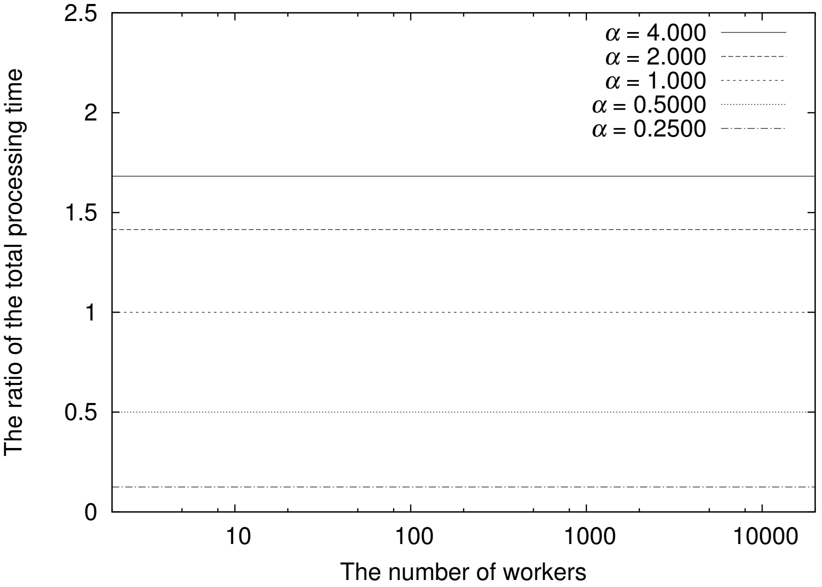

FIGURE 16. The ratio of the total processing time in Model B to that in Model N in Weibull distribution case.

In order to conclude this section, note first that backup-task scheduling is not effective for a small M because the mean worker-processing time for Model B is b/M, which is greater than that for Model N $b/2M$. Note also that the issue of

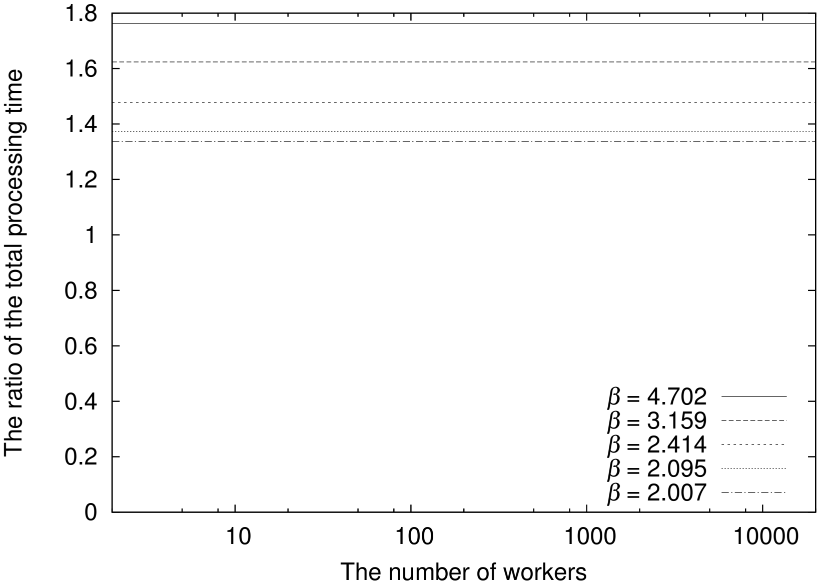

Figure 17. The ratio of the total processing time in Model B to that in Model N in Pareto distribution case.

stragglers rarely occurs for small M. When M is extremely large, the mean worker-processing time for Model B is still greater than that for Model N. However, the difference between them is small, and the issue of stragglers is likely to occur when the variation of the worker-processing time is large.

From the numerical results, we can claim that backup-task scheduling is significantly efficient for improving the performance when the variation of the worker processing time is large. Moreover, the effect of backup tasks depends on the worker processing time distribution, although the coefficient of variation is the same. Especially, for many workers case, the effect of backup-task scheduling on the maximum throughput and mean response time for Weibull distribution is very different from that for Pareto distribution. Therefore, we should pay attention to the distribution as well as the first and second order statistics of worker processing time when we consider the efficiency of backup-task scheduling.

5. **Conclusion.** In this paper, we considered the efficiency of backup-task scheduling in a large-scale parallel-distributed processing. We modeled the task-scheduling server as a single-server queue with many workers, deriving the maximum throughput, mean response time, and total processing time. From the numerical results, we can claim that backup-task scheduling is significantly efficient for improving performance in case of large variance of the worker processing time. Note that the effect of backup-task scheduling depends on the distribution of the worker processing time even when the means and variances of the distributions are the same.

REFERENCES

[1] 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.

[2] L. A. Barroso and U. Hölzle, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines," Morgan & Claypool, California, 2009.

[3] 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.

[4] J. Dean and S. Ghemawat, *MapReduce: simplified data processing on large clusters*, Communications of the ACM, **51** (2008), 107–113.

[5] J. Dejun, G. Pierre and C.-H. Chi, *EC2 performance analysis for resource provisioning of service-oriented applications*, Proc. Service-Oriented Computing: ICSOC/ServiceWave 2009 Workshops, Lecture Notes in Computer Science, **6275/2010** (2010), 197–207.

[6] 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.

[7] D. Gross, J. F. Shortle, J. M. Thompson and C. M. Harris, "Fundamentals of Queueing Theory," 4^{th} edition, John Wiley & Sons, New Jersey, 2008.

[8] K. Xiong and H. Perros, *Service performance and analysis in cloud computing*, Proc. 2009 IEEE Congress on Services (Service 2009), (2009), 693–700.

[9] N. Yigitbasi, A. Iosup, D. Epema and S. Ostermann, *C-Meter: a framework for performance analysis of computing clouds*, Proc. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, (2009), 472–477.

Received September 2012; 1st revision December 2012; 2nd revision June 2013.

E-mail address: tsugu@sys.i.kyoto-u.ac.jp
E-mail address: masuyama@sys.i.kyoto-u.ac.jp
E-mail address: kasahara@is.naist.jp
E-mail address: takahashi@i.kyoto-u.ac.jp