PAPER Special Section on Parallel and Distributed Computing and Networking

Dynamic Job Scheduling Method Based on Expected Probability of Completion of Voting in Volunteer Computing

Yuto MIYAKOSHI[†], **Shinya YASUDA**[†a)], ***Nonmembers***, **Kan WATANABE**[†], **Masaru FUKUSHI**[††b)], ***and*** **Yasuyuki NOGAMI**[†c)], ***Members***

SUMMARY This paper addresses the problem of job scheduling in volunteer computing (VC) systems where each computation job is replicated and allocated to multiple participants (workers) to remove incorrect results by a voting mechanism. In the job scheduling of VC, the number of workers to complete a job is an important factor for the system performance; however, it cannot be fixed because some of the workers may secede in real VC. This is the problem that existing methods have not considered in the job scheduling. We propose a dynamic job scheduling method which considers the expected probability of completion (EPC) for each job based on the probability of worker's secession. The key idea of the proposed method is to allocate jobs so that EPC is always greater than a specified value (SPC). By setting SPC as a reasonable value, the proposed method enables to complete jobs without excess allocation, which leads to the higher performance of VC systems. We assume in this paper that worker's secession probability follows Weibull-distribution which is known to reflect more practical situation. We derive parameters for the distribution using actual trace data and compare the performance of the proposed and the previous method under the Weibull-distribution model, as well as the previous constant probability model. Simulation results show that the performance of the proposed method is up to 5 times higher than that of the existing method especially when the time for completing jobs is restricted, while keeping the error rate lower than a required value.

key words: *parallel computing, desktop grids, probabilistic method, sabotage-tolerance*

1. Introduction

Volunteer computing (VC) is a type of Internet-based parallel computing paradigm, which allows any participants on the Internet to contribute their idle computing resources (workers) towards solving large parallel problems. VC can promptly construct a large-scale and high-performance parallel computing platform at a low cost. Folding@home [1] is a famous example for achieving tremendous computing performance of 43 PFlops. Other successful examples of VC include SETI@home [2] and distributed.net [3], among several others [4]–[10]. It is reported in [11] that VC systems have a superiority of operating costs in comparison to Amazon EC2 cloud.

Manuscript received January 9, 2015.
Manuscript revised May 19, 2015.
Manuscript publicized September 15, 2015.

[†]The authors are with the Graduate School of Natural Science and Technology, Okayama University, Okayama-shi, 700–8530 Japan.
[††]The author is with the Graduate School of Science and Engineering, Yamaguchi University, Ube-shi, 755–8611 Japan.
a) E-mail: shinya.yasuda@s.okayama-u.ac.jp
b) E-mail: mfukushi@yamaguchi-u.ac.jp
c) E-mail: yasuyuki.nogami@okayama-u.ac.jp

DOI: 10.1587/transinf.2015PAP0027

On the other hand, VC systems have two critical problems due to the nature of voluntary contribution of workers, (1) workers may return incorrect results for allocated computation tasks (jobs) [12], and (2) workers may not return results [14].

For the first problem (1), some sabotage-tolerance technique must be employed [15]. General VC systems employ some sort of voting, where each job is distributed to multiple workers for a majority decision [16], [17]. A job is completed when the final result of the job is determined through a vote. m-first voting method, an often-used method in BOINC-based VC systems like SETI@home [2], requires more than m candidate results for each job. The use of this method degrades the performance of VC systems to less than $1/m$.

To enhance the inefficiency of the m-first voting, credibility-based voting [9] is proposed. This method performs a weighted voting based on each worker's credibility, the probability of returning correct results. The credibility of a worker is calculated based on the number of times being majority in past voting. As a feature of the credibility-based voting, the required number of candidate results to complete a job becomes smaller with time because the credibility of reliable workers increases naturally with time [18].

Considering the above feature of the credibility-based voting, credibility-based job scheduling method called ECJMAX [21] is proposed. In ECJMAX, a management node (master) calculates the requisite minimum number of candidate results to complete each job based on the current credibility of workers. This is to avoid excess allocation of jobs (excess generation of candidate results). Accordingly, ECJMAX allows to minimize the number of generated results for whole computation of the VC system.

However, ECJMAX does not consider the second problem (2), i.e., workers may not return results. In real VC, workers can join and leave the system freely because they are controlled individually by volunteer participants. For example, if a worker is shut down for participant's own reason, the allocated job may not be completed in an expected time due to the lack of candidate results. Such job needs further allocation in the next round of the job scheduling. Therefore, the efficiency of the credibility-based voting and the whole performance of the VC system are degraded.

For the second problem (2) in real VCs, in this paper, we propose a dynamic job scheduling method by extending ECJMAX. The proposed method introduces two new metrics for job to enhance the credibility-based voting, i.e., the expected probability of completion (EPC) and the specified probability of completion (SPC). EPC represents the probability of completing the job by collecting the number of candidate results. SPC is a given value that represents the threshold of EPC. The key idea of the proposed method is to dynamically allocate a job to a group of workers so that the job's EPC is always greater than SPC. By setting SPC as a reasonable value, the proposed method enables to complete jobs with higher probability which leads to the higher credibility of reliable workers and the higher performance of VC systems.

According to Javadi et al. [14], it is known that the workers on the actual VC systems repeat their joining and leaving on the Weibull–distribution. Thus, in this paper, we model worker's secession following the Weibull-distribution. We derive parameters for the distribution using actual trace data [20] and compare the performance of the proposed and the previous method under the Weibull-distribution model, as well as the previous constant probability model [24].

The rest of this paper is organized as follows. Section 2 shows the computation model of VC systems. Section 3 shows credibility-based voting method and the existing job scheduling method, ECJMAX. In Sect. 4, we propose a dynamic job scheduling method based on EPC and estimation method of the secession probability. Section 5 shows the performance evaluations of the existing and the proposed job scheduling methods. Finally, Sect. 6 shows conclusions and future works.

2. VC Model

Target VC model in this paper is the well-known master–worker model with time restricted computation tasks. In this model, a batch of tasks is generated periodically in the master side. Details of this model are described as follows (see also Fig. 1).

- A VC system consists of a master (a management node) and W different workers (participant nodes).
- The master allocates a job to a worker w_i, where $1 \leq i \leq W$, based on a request from the worker.
- Each worker executes the allocated job in each time step and returns the result to the master.
- In each time step, N tasks (jobs) are generated with time restriction E_i, where $1 \leq i \leq N$.
- The computation proceeds until the time step reaches TD, where TD is a predefined duration of the computation which is set in the master.

This model is better suited for various VC applications which have severe deadline for the computation or data dependency between jobs generated in the adjacent time steps. For example, in VC applications such as weather forecast, measured data must be processed by a deadline for on-time forecast. Also, in VC applications which involve some iterative processing, computations in the t-th iteration must be

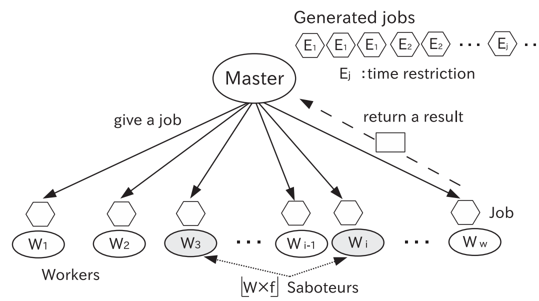


Fig. 1 Master–worker model with time restricted jobs

done as fast as possible to start the next $(t+1)$-th iteration.

Due to the nature of voluntary contribution of workers, some workers may return incorrect results in real VC. Those workers are called saboteurs and modeled as follows [22].

- There are $\lfloor f \times W \rfloor$ saboteurs, where f is a faulty fraction.
- Saboteurs return incorrect results with a constant probability s, which is known as the sabotage rate.
- The values of f and s are unknown to the master. Instead, the master assumes the maximum faulty fraction f_{max} $(\geq f)$.

The performance of a VC system is measured by two factors; the number of completed jobs, N_{comp}, and the error rate of completed jobs, ϵ. N_{comp} is the number of jobs that the final results are determined by a voting method before their time restriction. A job completed with an incorrect result is called incorrect job. ϵ is the ratio of the number of incorrect jobs to all completed N_{comp} jobs. ϵ represents the reliability of whole computation in the VC system and becomes an important factor of the sabotage-tolerance methods such as voting.

3. Credibility-Based Voting and Job Scheduling

3.1 Credibility-Based Voting

Voting is a basic sabotage-tolerance method for VC to eliminate incorrect results returned from workers. A simple m-first voting is used in BOINC [13], major VC middleware. In the m-first voting, m matching results are collected to complete a job. When m is too small, error rate ϵ may become unacceptably large. Thus, the practical VC systems such as SETI@home [2] set m as an enough large value (e.g. $m = 3$). Because the amount of calculation is more than tripled when $m = 3$, the performance of the VC system decreases significantly.

To enhance the inefficiency of the m-first voting, credibility-based voting [9] is proposed. This method defines a credibility for each system element such as worker, result, and job, and performs a weighted voting. Credibility represents the probability of the element being correct. The use of the credibility has the following two advantages; (1) required number of candidate results to complete a job can be smaller than that in the m-first voting, thus resulting in higher performance VC, and (2) it enables to guarantee the correctness of the computation; error rate ϵ is smaller than an acceptable error rate.

Each credibility is defined as follows [19].

The credibility of a worker w_i, denoted by $C_W(w_i)$, is given by Eqs. (1) - (2).

$$C_W(w_i) = \begin{cases} 1 - f_{max}, & \text{if } k = 0, \\ 1 - \frac{f_{max}}{1-f_{max}} \times min(1, max(\frac{1}{ke\theta}, (1-\theta)^k)), & \text{otherwise,} \end{cases} \quad (1)$$

$$\theta = 1 - \epsilon_{acc}. \quad (2)$$

Here, k represents how many times w_i becomes majority in past votes and ϵ_{acc} represents the acceptable error rate which is set at the master side, thus corresponding to a reliability requirement imposed for the VC.

The credibility of a result r returned from worker w_i, denoted by $C_R(r)$, is equal to $C_W(w_i)$.

$$C_R(r) = C_W(w_i) \quad (3)$$

Suppose that several results returned for a job j are grouped into result groups $G_1, \dots, G_g$, each of which includes all results having the same value. The credibility of a result group G_a, denoted by $C_G(G_a)$, is given as the conditional probability that results in G_a are correct and all other results are incorrect.

$$C_G(G_a) = \frac{P_T(G_a) \prod_{i \neq a} P_F(G_i)}{\prod_{i=1}^{g} P_F(G_i) + \sum_{n=1}^{g} P_T(G_n) \prod_{i \neq n} P_F(G_i)}, \quad (4)$$

$$P_T(G_a) = \prod_{r \in G_a} C_R(r), \quad (5)$$

$$P_F(G_a) = \prod_{r \in G_a} 1 - C_R(r). \quad (6)$$

The credibility of job j, denoted by $C_J(j)$, is equal to $C_G(G_x)$, where G_x is a result group which has a maximum credibility among all result groups for job j.

$$C_J(j) = C_G(G_x) = \max_{1 \leq a \leq g} C_G(G_a). \quad (7)$$

When $C_J(j)$ reaches θ (= $1 - \epsilon_{acc}$), the result of the group G_x is accepted as the final result of job j, and then job j is completed.

3.2 Credibility-Based Job Scheduling

In the credibility-based voting, job's completion is largely dependent on the credibility of workers executing the job. Thus, the credibility of each worker should be a factor to develop an efficient job scheduling method. ECJMAX [21] is a job scheduling method which considers the credibility and aims at increasing worker's credibility as fast as possible to minimize the total number of candidate results produced for overall computation.

The idea of ECJMAX is allocating the same job to several workers intensively until the number of the workers becomes enough large to complete the job. By doing so, jobs tend to be completed fast. Accordingly, majority workers which win the vote gain large credibility shortly. Such workers contribute to reduce the candidate results to complete a job in the subsequent voting. Thus, the total number of candidate results becomes smaller, which leads to higher performance of VC systems.

The key point of ECJMAX is how to determine the number of workers to allocate the same job. Too many workers will produce many candidate results of limited use for voting (excess job allocation). In ECJMAX, expected-credibility EC_J is defined for each job, which represents the credibility of a job provided that all workers calculating the job return the correct results. If EC_J of a job is greater than the threshold θ in Eq. (2), the job will be completed without additional allocation.

The expected-credibility of a job, denoted by EC_J, is defined as follows. There exists a worker group A (A = $w_1, \dots, w_d$) who are executing job j. Then, $EC_J(A)$ is given as follows.

$$EC_J(A) = \begin{cases} \dfrac{\prod_{w_i \in A} C_W(w_i)}{\prod_{w_i \in A} C_W(w_i) + \prod_{w_i \in A} (1 - C_W(w_i))} & \text{if } g = 0, \\ \max_{1 \leq a \leq g} C_G^{'}(G_a) & \text{otherwise,} \end{cases} \quad (8)$$

where $C_G^{'}(G_a)$ is the credibility of result group G_a provided that all results returned from d workers are grouped into G_x. As shown in this equation, ECJMAX is a method based on the assumption that all d workers return their results.

3.3 The Effect of Worker's Secession

In real VC, workers can join and leave the system freely because they are controlled individually by volunteer participants. If a worker is shut down for participant's own reason, the worker will not return the result for allocated job (worker's secession). As reported in [14], worker's secession in VC is non-negligible because the computation time of a job in a worker reaches several hours or days, while participants may shut down their PC every day. Also, traditional detection techniques such as "heart-beat" are not available to detect the worker's secession in VC because communications between the pair of master and worker are initiated on the request of the worker.

The worker's secession does not have significant impacts on the error rate in credibility-based voting [23]. However, it affects the efficiency of job scheduling and the number of completed jobs. In ECJMAX, all workers in worker group A are assumed to return their results to calculate $EC_J(A)$. If one of the workers does not return the result, the job must not be completed due to the lack of candidate results. In this case, all other workers who returned their results can not gain the credibility. This will degrade the performance of VC system.

4. Dynamic Job Scheduling Method Based on Expected Probability of Completion

4.1 Summary

To solve the problem of ECJMAX, we propose a dynamic job scheduling method considering the worker's secession. The basic idea is to allocate a job to a group of workers expecting secession. To adjust the number of workers in a group, we define two parameters for each job, namely, the expected probability of completion (EPC) and the specified probability of completion (SPC). In the proposed method, the number of workers in a group changes dynamically by attempting to hold that EPC is always greater than SPC.

4.2 Secession Model

To define EPC, we first model the worker's secession. We use a probabilistic model in [13]. Let P_{down_i} be the secession probability and P_{up_i} be the rejoin probability for worker w_i. In the model, when a worker w_i gets a job from master, the worker will secede from the VC system before the next time step with P_{down_i}. In other words, the worker w_i will return the result to the master with probability 1-P_{down_i} in the next time step. The seceded worker w_i will rejoin to the VC system with a probability P_{up_i} in each time step.

The values of P_{down_i} and P_{up_i} may change with time in real VC because the secession is caused by several reasons such as hardware/software failures and rebooting. It is reported in [14] that P_{down_i} seems to follow a well-known Weibull distribution shown in Eq. (9).

$$\lambda(t) = \frac{\mu}{\eta^{\mu}} \times t^{\mu-1}, \tag{9}$$

where η is scale parameter, μ is shape parameter, and t is time step.

Although the strict model of worker's secession has not built in [14], we estimate the two parameters η and μ of the Weibull distribution using trace data in real VC. We randomly chose 100 workers in the DATASETS files of SETI@home [20] and obtained the trace data of each worker including the time of secession. Then, we calculate the secession probability assuming the time step length of 6 hours (execution of a job generally takes several hours in VC).

Figure 2 shows the values of P_{down_i} for Eq. (9) with $\eta = 0.5$, $\mu = 0.45$ and the average secession probability for the 100 workers. This figure indicates that the Weibull distribution well represents the actual behavior of worker's secession in real VC.

4.3 Proposed Job Scheduling Method

Here, we define the expected probability of completion for

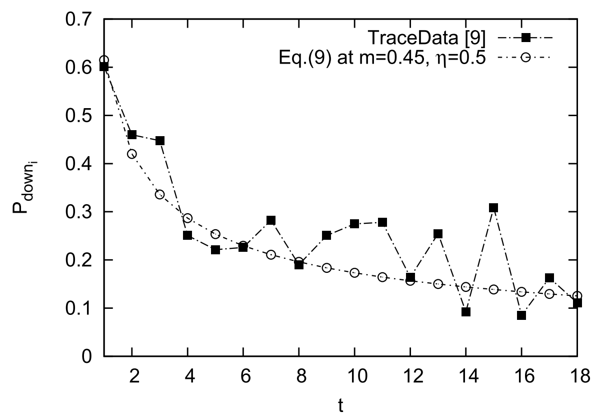


Fig. 2 The approximate performance of weibull-distribution ($\eta = 0.5, \mu = 0.45$)

job j, $EPC(j)$. $EPC(j)$ is calculated based on the probability of each worker's secession $P_{down_1}, \ldots, P_{down_d}$. Again suppose that job j is allocated to a group A consisting of d workers. Each d worker returns a result or not; therefore, there are 2^d situations for the computation of returned results. $EPC(j)$ is given by calculating the credibility of job j for the 2^d situations.

Let $A_{l,n}$ be the subset of A consisting of l workers and F_l be the set of all $A_{l,n}$. The size of F_l is given by $|F_l| =_d C_l$ for each l (i.e. $\sum_{l=0}^{d} |F_l| = 2^d$). Here, n is a running number of $A_{l,n}$ given for each l.

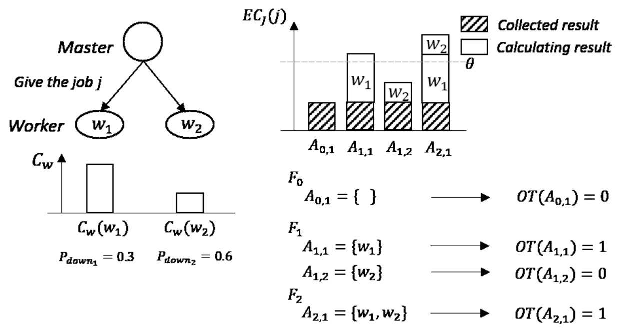
Suppose a situation that the workers in $A_{l,n}$ return results for job j and the remaining $d - l$ workers in $A^c_{l,n} = A - A_{l,n}$ do not. In this case, if the expected-credibility of j, i.e. $EC_J(A_{l,n})$, exceeds the threshold θ, it is expected that j will be completed with results from $A_{l,n}$. On the other hand, if $EC_J(A_{l,n})$ does not reach θ, it means that j must not be completed even if all $A_{l,n}$ workers return results. The formula of $EPC(j)$ is given by Eq. (10) using the variable $OT(A_{l,n})$.

$$EPC(j) = \sum_{l=0}^{d} \sum_{A_{l,n} \in F_l} OT(A_{l,n}) \prod_{w_i \in A_{l,n}} (1 - P_{down_i}) \prod_{w_i \in A^c_{l,n}} P_{down_i} \tag{10}$$

$$OT(A_{l,n}) = \begin{cases} 1 & \text{if } \theta \leq EC_J(A_{l,n}), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

$OT(A_{l,n})$ represents whether j is completed or not when all $A_{l,n}$ workers return their results.

In order to easily understand Eq. (10), Fig. 3 shows a calculation example of $EPC(j)$. In this example, two workers, w_1 and w_2, are calculating job j (i.e. $A = \{w_1, w_2\}$ and $d = 2$). Suppose that these workers have the credibility $C_w(w_i)$ and the secession probability P_{down_i} as shown in the upper-left part of Fig. 3. Since $d = 2$, possible F_l are $F_0 = \{\}$, $F_1 = \{A_{1,1}, A_{1,2}\}$, and $F_2 = \{A_{2,2}\}$. With the supposed worker's credibility, also suppose that workers in $A_{1,1}$ or $A_{2,1}$ make $EC_J(j)$ greater that the threshold θ, and therefore, $OT(A_{1,1})$ and $OT(A_{2,1})$ become 1. $EPC(j)$ is calculated using $OT(A_{l,n})$ and P_{down_i} as defined in Eq. (10).



$$
\begin{aligned}
EPC(j) = & \underbrace{OT(A_{0,1}) \times P_{down_1} \times P_{down_2}}_{A_{0,1}} + \underbrace{OT(A_{1,1}) \times (1 - P_{down_1}) \times P_{down_2}}_{A_{1,1}} \\
& + \underbrace{OT(A_{1,2}) \times P_{down_1} \times (1 - P_{down_2})}_{A_{1,2}} + \underbrace{OT(A_{2,1}) \times (1 - P_{down_1}) \times (1 - P_{down_2})}_{A_{2,1}} \\
& = 0 \times 0.3 \times 0.6 + 1 \times 0.7 \times 0.6 + 0 \times 0.3 \times 0.4 + 1 \times 0.7 \times 0.4 \\
& = 0.7
\end{aligned}
$$

Fig. 3 A calculation example of $EPC(j)$ ($A = \{w_1, w_2\}$)

Using the above $EPC(j)$, the proposed method repeats the allocation of job j for workers who request job allocations until $EPC(j)$ reaches the specified probability of completion, denoted by SPC. Every time job j is allocated to a worker, $EPC(j)$ is updated. By the proposed method, even if some workers secede the VC system, it is expected that each job can be completed with the probability more than or equal to SPC.

There are two optional ideas to implement the proposed method; (a) the upper limit of the number of workers, and (b) an efficient use of high credibility worker. About (a), if the number of workers in group A, i.e. d, becomes large, the calculation cost of EPC increases exponentially which may be a bottle neck of the performance of job scheduling in the master. Hence, the proposed method sets the upper limit of d, denoted by EW_{max}. If d for job j exceeds EW_{max}, the master begins to allocate another job to workers regardless of values of $EPC(j)$. About (b), if the credibility of a worker reaches the threshold θ, it means the worker is enough creditable to complete every job by his own result. It is a waste to use such worker together with others. Thus, if the master finds such high credibility worker, the worker is exceptionally assigned to a job having the minimal EC_J.

4.4 Estimation of the Secession Probability

The proposed method needs to know the secession probability of each worker, $P_{down_1}, \ldots, P_{down_d}$, to calculate EPC. However, in real VC, the secession probability is naturally unknown before the computation and is known to be different in every worker [14]. The proposed method requires a way to get the estimation of the secession probability for each worker to calculate EPC.

In this paper, we propose a simple estimation way of the secession probability as follows. For a worker w_i, the estimated probability of secession, denoted by exP_{down_i}, is given as the ratio of the number of jobs being returned no result n_x to the number of jobs being allocated n_r. The lower limit of exP_{down_i} is set to 0.05 because every worker will secede with a small probability as shown in [20].

$$
exP_{down_i} = \begin{cases} 0.05 & \text{if } n_r \leq 1, \\ max(0.05, \frac{n_x}{n_r - 1}) & \text{otherwise.} \end{cases} \quad (12)
$$

Against exP_{down_i}, actual probability of secession is denoted by $reaP_{down_i}$ hereafter.

Table 1 Simulation parameters

The number of workers (W)	100 workers
The number of generated jobs in each time step (N)	100 jobs
Time restrictions of jobs from the generated time (E)	4 steps
Deadline of whole computation (TD)	1 – 300 steps
Acceptable error rate (ϵ_{acc})	0.01
Faulty fraction of workers (f)	0.35
The upper limit of f (f_{max})	0.35
Sabotage rate of workers (s)	0.05
The specified probability of completion (SPC)	0.1 – 0.9
The upper limit of calculating workers (EW_{max})	16 workers

5. Performance Evaluation

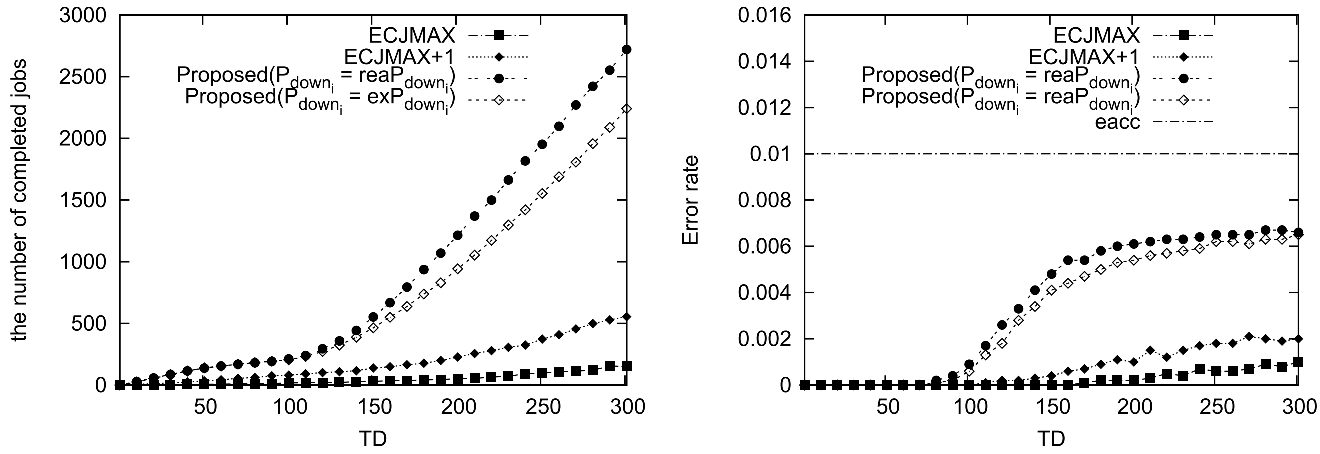
5.1 Simulation Conditions

We evaluate the effectiveness of the proposed job scheduling method through the simulation of VCs. The number of completed jobs N_{comp} and the error rate ϵ are evaluated as the average of 1000 simulation results for the following four different job scheduling methods.

- ECJMAX: existing method [21]
- ECJMAX+1: a simple extension of ECJMAX in which each job is allocated to one extra worker to collect preliminary results.
- Proposed (P_{down_i}=$reaP_{down_i}$): the proposed method using $reaP_{down_i}$ for the calculation of EPC assuming that the master knows P_{down_i} beforehand.
- Proposed (P_{down_i}=exP_{down_i}): the proposed method using exP_{down_i} based on Eq. (12) as a feasible case in real VC.

The parameters used in our simulation are determined as in [21] and shown in Table 1. Following the common assumption [19], [21], [22], workers are assumed to return their result in one time step. The length of a time step depends on the VC applications (i.e. the granularity of a job) and it is decided by master side. Each time restriction of job, i.e. E_i, is uniformly set to $E_i = E$ for $i = 1, \ldots, N$.

For the actual probability of worker's secession, we use two models. The first one is the constant $reaP_{down_i}$ model, in which $reaP_{down_i}$ is a constant value and given by $reaP_{down}$ for $i = 1 \ldots W$ as in [24] and the Weibull–distribution model, in which $reaP_{down_i}$ changes with time based on Eq. (9) for $i = 1, \ldots, W$. The parameters of Weibull–distribution, $\eta = 0.5$, $\mu = 0.45$, are used as shown in Fig. 2. For both models, P_{up_i} is assumed to 1 for $i = 1, \ldots, W$ because we assumed that one time step is 6 hours,



(a) The number of completed jobs (b) Error rate

Fig. 4 The number of completed jobs P and the error rate ϵ for $reaP_{down}$=0.6, E = 4 and SPC = 0.4 under the constant $reaP_{down}$ model

which is sufficiently large to rejoin the VC system.

5.2 Simulation Results

5.2.1 Simulation Results under the Constant $reaP_{down}$ Model

Figure 4 shows the number of completed jobs N_{comp} and error rate ϵ as a function of TD. Figure 4 (a) shows that the difference between the proposed method and ECJMAX becomes larger as TD increases. Especially, when $TD = 300$, the performance of the proposed method (P_{down_i}=$reaP_{down_i}$) is 5 times larger than that of ECJMAX.

ECJMAX+1 also improves the performance in comparison to ECJMAX; however, the impact is limited because only one extra worker is allocated a job. As the proposed method does, the number of workers for a job, d, should be changed dynamically corresponding to the change of the worker's credibility. Note that d should not be too large because it causes excess allocations of jobs which degrades performance of VC system.

The performance difference between the Proposed (P_{down_i}=$reaP_{down_i}$) and Proposed (P_{down_i}=exP_{down_i}) in this figure shows that the performance depends on the degree of accuracy of P_{down_i}. In Proposed($P_{down_i} = exP_{down_i}$), if P_{down_i} is estimated lower than the actual, a group of workers decided by the method is insufficient to complete the job j, which requires reallocation of the job. While, in the opposite case, too many workers are allocated for the job j. Both cases will result in inefficient usage of workers. This is the reason for the performance diffrerence shown in Fig. 4 (a).

Figure 4 (b) shows that error rate ϵ of each method is always less than the required value ϵ_{acc} =0.01. It is confirmed that the reliability condition $\epsilon \leq \epsilon_{acc}$ is guaranteed when the proposed method is used. The error rate of ECJMAX is smaller than that of the proposed method because the credibility of each worker in ECJMAX is smaller. In ECJMAX,

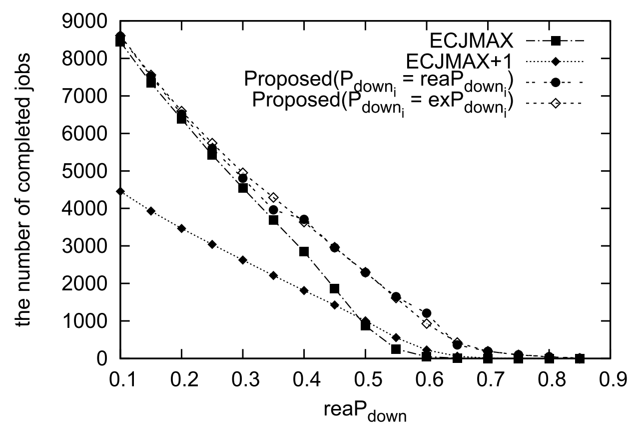


Fig. 5 The number of completed jobs vs. $reaP_{down}$ at E = 4, SPC = 0.4 and TD = 200 under the constant $reaP_{down}$ model

the chance of gaining credibility (the number of completed jobs) is smaller as shown in Fig. 4 (a). Then, ECJMAX requires more results to complete each job, which leads to smaller error rate by decreasing the probability of accepting incorrect results. This indicates there is a trade-off between the number of completed jobs and the error rate.

The main motivation of the proposed method is to maximize the number of completed jobs while keeping the reliability condition $\epsilon \leq \epsilon_{acc}$. If a master requires smaller ϵ in the proposed method, the master can achieve it easily by setting smaller ϵ_{acc}. Further experiments confirmed that the reliability condition was also guaranteed for ϵ_{acc} = 0.005 and ϵ_{acc} = 0.001, though the results are not shown here.

Figure 5 shows the number of completed jobs as a function of $reaP_{down}$. This figure shows that the number of completed jobs in the proposed method is always greater than those of ECJMAX and ECJMAX+1 for any case of $reaP_{down}$. Because the actual value of $reaP_{down}$ is unknown beforehand, this is an important feature of the proposed method. When $reaP_{down} \geq 0.7$, the performance of the pro-

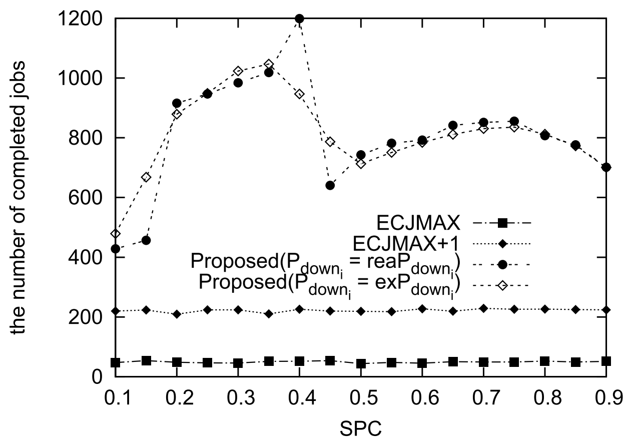


Fig. 6 The number of completed jobs vs. SPC at $reaP_{down}$ = 0.6, E = 4 and TD = 200 under the constant $reaP_{down}$ model

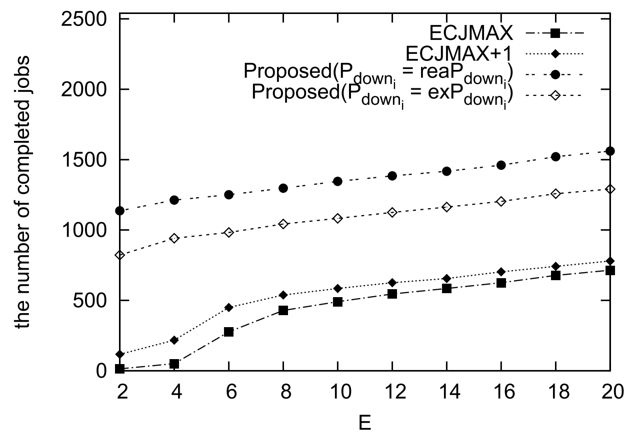


Fig. 7 The number of completed jobs vs. E at $reaP_{down}$ = 0.6, SPC = 0.4 and TD = 200 under the constant $reaP_{down}$ model

posed method dramatically decreases because d is limited by EW_{max} = 16. For such a drastic case, EW_{max} should be adjusted.

Figure 5 also shows that the performance of ECJMAX+1 is less than that of ECJMAX for $reaP_{down} \leq 0.5$. This performance degradation comes from the reason why even one extra worker in ECJMAX+1 is excess when $reaP_{down}$ is small. The simple idea of adding extra workers does not always work well as shown in this case. In contrast with ECJMAX+1, the proposed method works well even if $reaP_{down}$ is small because the master determines the number of workers d with the estimated P_{down_i} and EPC.

Figure 6 shows the number of completed jobs as a function of SPC. This figure shows that the performance of the proposed method depends largely on the value of SPC. There is an optimal value of SPC to maximize the performance of the proposed method. It is expected the optimal value in the constant $reaP_{down_i}$ model is equal to $1-reaP_{down}$ through our simulation results. The reason seems that, if worker w_i has enough credibility (larger than 1 - ϵ_{acc}) and a job j has no result group, $EPC(A = w_i)$ of j is 1 - P_{down_i}. To obtain the proof of this relationship and derive the optimal value of SPC is one of our future works.

Figure 7 shows the number of completed jobs as a function of E. As E increases, the number of completed jobs also increases because some jobs which lack candidate results can be completed by collecting additional results in extended time restriction. This figure shows that the number of completed jobs in the proposed method is always greater than those of ECJMAX and ECJMAX+1 for any E. Because the value of E depends on the type of application (e.g. E may be 24 hours for predicting the weather tomorrow), this result indicates that the proposed method can be used for various applications.

5.2.2 Simulation Results under Weibull–Distribution Model

Figure 8 shows the number of completed jobs N_{comp} as a

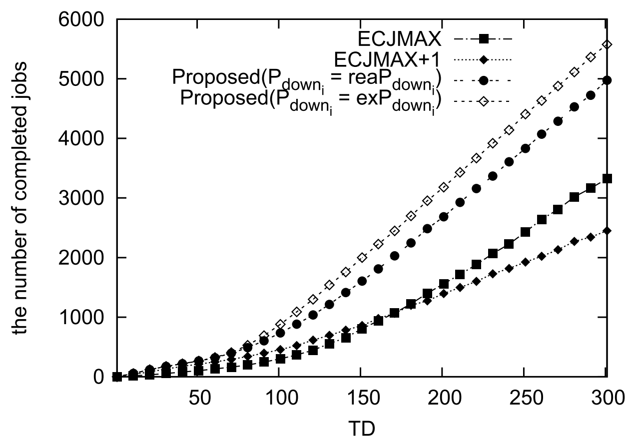


Fig. 8 The number of completed jobs vs. TD at E = 4 under Weibull–distribution model (η = 0.5, μ = 0.45)

function of TD. This figure shows that the number of completed jobs in the proposed method is always greater than those of ECJMAX and ECJMAX+1 even in the Weibull–distribution model. Compared to the case of the constant $reaP_{down}$ model shown in Fig. 4 (a), the performance difference between ECJMAX and the proposed methods becomes smaller. This is because P_{down_i} in the Weibull–distribution model (i.e., P_{down_i} changes from 0.61 to 0.12 as shown in Fig. 2) is smaller than that in the constant $reaP_{down}$ model (P_{down_i} is always 0.6 as in Fig. 4 (a)). When P_{down_i} is small, the performance difference tends to be small as shown in Fig. 5.

Figure 9 shows the number of completed jobs N_{comp} as a function of SPC. Similarly in Fig. 6, this figure also shows that the performance of the proposed method depends on the value of SPC and there is an optimal SPC to maximize the performance. As discussed in the above paragraph, P_{down_i} in the Weibull–distribution model is smaller than that in the constant $reaP_{down}$ model (P_{down_i} = 0.6), while the optimal SPC is around 0.4 in both cases. This results indicates the difficulty of estimation for the optimal SPC.

Compared to the constant $reaP_{down}$ model and the

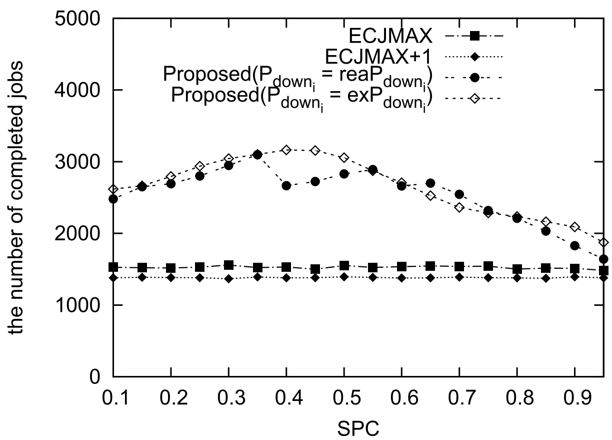


Fig. 9 The number of completed jobs vs. SPC at $E = 4$ under Weibull–distribution model ($\eta = 0.5, \mu = 0.45$)

Weibull–distribution model, the main difference is the superiority of the Proposed ($P_{down_i} = reaP_{down_i}$). Contrary to Fig. 4 (a), Fig. 8 shows that the number of completed jobs in Proposed ($P_{down_i} = reaP_{down_i}$) is smaller than that in Proposed ($P_{down_i} = exP_{down_i}$). The reason seems that $reaP_{down_i}$ given by Eq. (9) is a decreasing function at $\mu = 0.45 \leq 1$. In this case, the estimation formula of exP_{down_i} shown in Eq. (12) gives larger value than $reaP_{down_i}$. If exP_{down_i} becomes smaller, each job requires more result to satisfy $EPC \geq SPC$. Thus, more worker can gain credibility when each job is completed.

This result indicates that, when P_{down_i} changes with time (like in the Weibull–distribution model), the optimal value of SPC may be changed with time. For example, if SPC is large at the beginning of computation, more worker can gain credibility when a job is completed. After gaining enough credibilioty, each worker is allocated to various jobs by setting smaller SPC to increase the performance of whole VC system. To derive such a function for SPC and obtain the proof of its optimality are our future works.

6. Conclusion

We proposed a dynamic job scheduling method based on the expected probability of completion of voting (EPC) in VC where worker's secession happens with unknown probability P_{down_i}. In the proposed method, a master performs job scheduling based on EPC so that EPC is higher than a specified value SPC to complete a voting with probability SPC or higher. We also proposed a simple estimation function of P_{down_i} based on each worker's history for a feasible use of the proposed method. Simulation results show that the proposed method improved the number of completed jobs compared to ECJMAX and guaranteed the error rate of VC lower than the required value. In the future works, we will improve the estimation function of P_{down_i} and derive the optimal function of SPC. Another interesting research is the development of a theoretical framework to customize the length of time step and examine the impact on the performance of VCs under more general conditions that workers behaviors (i.e. both returning results and seceding from VC) follow probability distributions.

References

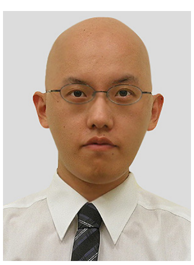
[1] http://folding.stanford.edu/ July 1, 2014.
[2] http://setiathome.ssl.berkeley.edu/ July 1, 2014.
[3] http://www.distributed.net/ July 1, 2014.
[4] http://einstein.phys.uwm.edu/ May 17, 2015.
[5] http://fightaidsathome.scripps.edu/ May 17, 2015.
[6] http://www.mersenne.org/ May 17, 2015.
[7] http://milkyway.cs.rpi.edu/ May 17, 2015.
[8] http://www.worldcommunitygrid.org/ May 17, 2015.
[9] L.F.G. Sarmenta and S. Hirano, "Bayanihan: building and studying web-based volunteer computing systems using Java," Future Generation Computer Systems, vol.15, no.5–6, pp.675–686. 1999.
[10] B. Knispel, B. Allen, J.M. Cordes, J.S. Deneva, D. Anderson, C. Aulbert, N.D.R. Bhat, O. Bock, S. Bogdanov, A. Brazier, F. Camilo, D.J. Champion, S. Chatterjee, F. Crawford, P.B. Demorest, H. Fehrmann, P.C.C. Freire, M.E. Gonzalez, D. Hammer, J.W.T. Hessels, F.A. Jenet, L. Kasian, V.M. Kaspi, M. Kramer, P. Lazarus, J. van Leeuwen, D.R. Lorimer, A.G. Lyne, B. Machenschalk, M.A. McLaughlin, C. Messenger, D.J. Nice, M.A. Papa, H.J. Pletsch, R. Prix, S.M. Ransom, X. Siemens, I.H. Stairs, B.W. Stappers, K. Stovall, and A. Venkataraman, "Pulsar Discovery by Global Volunteer Computing," Science, vol.329, no.5994, p.1305, 2010.
[11] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D.P. Anderson, "Cost-Benefit Analysis of Cloud Computing versus Desktop Grids," Proc. 2009 IEEE International Symposium on Parallel and Distributed Processing, pp.1–12, 2009.
[12] D. Kondo, F. Araujo, P. Malecot, P. Domingues, L.M. Silva, G. Fedak, and F. Cappello, "Characterizing Error Rates in Internet Desktop Grids," 13th European Conference on Parallel and Distributed Computing, pp.361–371, 2007.
[13] D.P. Anderson, "BOINC: A System for Public-Resource Computing and Storage," 5th IEEE/ACM International Workshop on Grid Computing, pp.4–10, Nov. 2004.
[14] B. Javadi, D. Kondo, A. Iosup, and D. Epema "The Failure Trace Archive: Enabling the Comparison of Failure Measurements and Models of Distributed Systems," Journal of Parallel and Distributed Computing, vol.73, no.8, pp.1208–1223, 2013.
[15] P. Domingues, B. Sousa, and L.M. Silva, "Sabotage-tolerance and trust management in desktop grid computing," Future Generation Computer Systems, vol.23, no.7, pp.904–912, 2007.
[16] Y.A. Zuev, "On the Estimation of Efficiency of Voting Procedures," Theory of Probability and its Applications, vol.42, no.1, pp.73–81, 1998.
[17] H. Casanova, "Benefits and Drawbacks of Redundant Bathc Requests," Journal of Grid Computing, vol.5, no.2, pp.235–250, 2007.
[18] K. Watanabe, M. Fukushi, and S. Horiguchi, "Expected-credibility-based job Scheduling for Reliable Volunteer Computing," IEICE Trans. Inf. & Syst., vol.E93-D, no.2, pp.306–314, Feb. 2010.
[19] K. Watanabe and M. Fukushi, "Generalized Spot-checking for Reliable Volunteer Computing," IEICE Transactions on Information and Systems, vol.E93-D, no.12, pp.3164–3172, Dec. 2010.
[20] http://fta.scem.uws.edu.au/ July 1, 2014.
[21] K. Watanabe, M. Fukushi, and M. Kameyama "Adaptive Group-Based Job Scheduling for High Performance and Reliable Volunteer Computing," Journal of Information Processing, vol.19, pp.39–51, Feb. 2011.
[22] L.F.G. Sarmenta, "Sabotage-Tolerance Mechanisms for Volunteer Computing Systems," Future Generation Computer Systems, vol.18, no.4, pp.561–572, 2002.
[23] K. Watanabe and M. Fukushi, "Sabotage-tolerant Job Scheduling for Reliable Volunteer Computing," Volume 2, Computer Science

Research and Technology, pp.117–160, ISBN: 978-1-61122-073-5 (online: 978-1-61122-724-6), Nova Science Publishers, 2011.

[24] Y. Miyakoshi, K. Watanabe, M. Fukushi, and Y. Nogami, "A Job Scheduling Method Based on Expected Probability of Completion of Voting in Volunteer Computing," Proc. 2014 Second International Symposium on Computing and Networking, CANDAR 2014, pp.399–405, 2014.

Yuto Miyakoshi graduated from Okayama University in 2013 and then received the master grade in 2015. He has researched applications of volunteer computing.

Shinya Yasuda graduated from Okayama University in 2014. He is a master grade student in Okayama univerisity. He is now interested in volunteer computing.



Kan Watanabe graduated from Tohoku University in 2006 and received the Ph.D. degree in 2011 from Tohoku University. He was an assistant professor of Okayama University until March 2015.

Masaru Fukushi received his B.Sc. and M.Sc. degrees from Hirosaki University in 1995 and 1997, respectively, and Ph.D. degree in information science from the Graduate School of Information Science at Japan Advanced Institute of Science and Technology (JAIST) in 2002. He is currently an associate professor in the Graduate School of Science and Engineering at Yamaguchi University. Prior to joining in the Yamaguchi University, he was an assistant professor in the School of Information Science at JAIST from 2002 to 2004, and in the Graduate School of Information Sciences at Tohoku University from 2004 to 2012. His research interests include dependable parallel VLSI architectures, dependable and high-performance distributed systems, and parallel and distributed computing. Dr. Fukushi is a member of IEEE, IPS of Japan, IEICE of Japan.

Yasuyuki Nogami graduated from Shinshu University in 1994 and received the Ph.D. degree in 1999 from Shinshu University. He is now an associate professor of Okayama University. His main fields of research are finite field theory and its applications such as recent public key cryptographies. He is now studying about elliptic curve cryptography, pairing-based cryptography, Lattice-based cryptography, pseudo random number generator, Advanced Encryption Standard, and homomorphic encryptions. Recently, he is a member of security research group in Okayama university and particularly focusing on IoT security from the viewpoints of software and hardware implementations. He is a member of IEICE and IEEE.