Science in China Series F: Information Sciences

© 2007 Science in China Press
Springer-Verlag

Trusted dynamic level scheduling based on Bayes trust model

WANG Wei[1†] & ZENG GuoSun[2]

[1] Department of Computer Science and Engineering, Tongji University, Shanghai 201804, China;
[2] Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 201804, China

A kind of trust mechanism-based task scheduling model was presented. Referring to the trust relationship models of social persons, trust relationship is built among Grid nodes, and the trustworthiness of nodes is evaluated by utilizing the Bayes method. Integrating the trustworthiness of nodes into a Dynamic Level Scheduling (DLS) algorithm, the Trust-Dynamic Level Scheduling (Trust-DLS) algorithm is proposed. Theoretical analysis and simulations prove that the Trust-DLS algorithm can efficiently meet the requirement of Grid tasks in trust, sacrificing fewer time costs, and assuring the execution of tasks in a security way in Grid environment.

Grid computing, trustworthy scheduling, Bayes method, trustworthiness evaluation, Trust-DLS

Grid is a unified computing platform which tries to connect and share all resources in the Internet, including computation resource, storage resource, information resource, knowledge resource and equipments for scientific research, and then solves the problems of large-scale scientific engineering computing[1,2]. However, with the characteristics of dynamic, heterogeneity, distribution, openness, voluntariness, uncertainty and deception, how to obtain trustworthy Grid resource becomes a key issue in Grid research. Once Grid becomes the next generation of computing networks, tasks with requirement of high QoS in trust are going to join the Grid and utilize Grid resource, such as national intelligence analysis and banking and financial data analysis. The traditional method to solve the security problem of these application tasks is to encrypt the data of execution and analysis, or isolate them from the Internet, and then schedule them to local resources to compute and analyze. In Grid environment with uncountable numeric nodes, resource is inevitably unreliable, which has a great effect on task execution and scheduling. As Grid becomes the next generation of computation and information platform, novel algorithms are needed to schedule the jobs on the trusty nodes to execute, assure the high speed of communication, reduce the jobs execution time, lower the ratio of failure execution, and improve the security of

Received July 15, 2006; accepted February 26, 2007
doi: 10.1007/s11432-007-0021-x

†Corresponding author (email: willtongji@gmail.com, gszeng@mail.tongji.edu.cn (ZENG GuoSun))

Supported partially by the National Natural Science Foundation of China (Grant No. 60673157), the Ministry of Education Key Project (Grant No. 105071), and SEC E-Institute: Shanghai High Institutions Grid (Grant No. 200301)

execution environment of important data.

In heterogeneous computing, Grid computing, distributed computing and cluster computing environments, many static, dynamic, and even hybrid algorithms have been proposed. At the same time, some issues related to distributed scheduling, center scheduling, autonomy scheduling, intelligent scheduling and Agent negotiation scheduling are also in exploration. In static algorithms, BNP-based ISH[3], MCP[4] and ETF[5] algorithms are suited for high speed and low delay small distributed networks, which is contrary to the Grid environment; APN-based MH[6] and DSL[7] algorithms run very well in large-scale distributed system and communication delay and time cost are considered, but cannot meet the requirement of Grid nodes in trust. In dynamic algorithms, dynamic job scheduling is considered, and jobs load balancing and sharing can be guaranteed by autonomy and intelligent scheduling[8,9]. In hybrid algorithms, jobs uniform distribution and communication overhead reducing are emphasized and load balancing is achieved by considering the computation that a node performs[10−13]. However, none of these algorithms takes the characteristics of nodes into account, such as uncertainty, unreliability and deception, and the scheduling length and trustworthiness of nodes cannot be considered synchronously.

Referring to the trust relationship models of sociology, a kind of trust mechanism based task scheduling model was presented by utilizing the Bayes method. By integrating the trustworthiness of nodes into Dynamic Level Scheduling (DLS) algorithm, the Trust-Dynamic Level Scheduling (Trust-DLS) algorithm was proposed. The proposed Bayes trust model is based on others' research[14,15] and our previous work[16,17]. Larger expansion is carried out in this paper with the main contributions listed below: 1) The problems of evaluating direct and recommendation trust degree based on Bayes method is completely exposited. 2) Factors, such as time and belief degree of recommendation information, are comprehensively considered. 3) The confidence level of trust degree is evaluated by utilizing interval estimation. 4) Evaluation of the trust degree of nodes by classifying the relationship between them is completely unified.

Simulation experiments proved that the proposed Trust-DLS algorithm can efficiently reduce the ratio of task failure exertion with a little more time cost.

The rest of this paper is organized as follows. Bayes method based Trust evaluation model is introduced in section 1. Section 2 describes the details of the proposed Trust-DSL. The results and analysis of simulation experiments are in section 3. Section 4 concludes the paper with future works.

1 Bayes trust model

1.1 Basic concept

Trust is the core of relationships in social networks. Trust is the evaluation of certain entities' reliable behaviors. The trust degree of a certain entity is always decided by others' recommendations. Grid system and social networks have great similarities[18]: a node displays a message, reflecting the characteristics of its behavior when it cooperates with other nodes; a node has sufficient choices; and the node is duty bound to offer recommendations to other nodes.

Thus, the node can evaluate the copartner through its behavior (e.g., the node's ratio of successful execution). Nodes can also exchange and transmit evaluation messages in order to obtain the trust of target node and guide its cooperation decision. In this paper, we define the "trust" in Grid environment as the evaluation of the target node's ability of providing service (resource) through the reliability shown by its behavior in certain context, including the observation of its former behaviors and the recommendations from other nodes.

In general, trust relationship is variable. Node A trusts node B's ability of providing service with the increase of their successful cooperations. A will gradually change and adjust its trustworthiness to B as time goes on. There is a direct trust relationship between A and B, which can be described by the success of their former cooperation. Besides, there is a recommendation relationship in trust[19]. If the node has not communicated with another, it can only receive recommendation from other nodes and judge the recommendation by its strategy. Trust is not a simple thing, and the trust is a continuous process in which one node's trust to another is to a certain extent, e.g., from low to medium, and then to high. The above analysis of attributes of trust relationship is the necessary basis of trust evaluation and the design of the trust computing model.

For the sake of simplicity, we only considered a Grid system within the same context during a period of time. For two nodes x and y, the successful cooperation probability between them is denoted by θ. They may have direct interactions between them, and they may also have other intermediate peers and each of them has direct experiences with x and y. On one hand, if there are direct interactions between x and y, we can obtain direct probability of successful cooperation, which is called *direct trust degree*, denoted by θ_{dt} . On the other hand, if there is an intermediate node z between x and y, and there are interactions between x and z, and z and y, then we can also obtain an indirect probability of successful cooperation between x and y, which is called *recommendation trust degree*, denoted by θ_{rt}. Thus, there are two kinds of probabilities of successful cooperation, which can be aggregated into global trust degree as follows:

$$\theta = f(\lambda_0 \cdot \theta_{dt}+(1-\lambda_0)\cdot \theta_{rt}),\ \lambda_0 \in (0, 1), \tag{1}$$

where $f(\cdot)$ is trust degree combination function, satisfying the property of convex function, that is, let $S \subset R^n$ be a nonempty convex set, and f a function defined on S. f is a convex function on S if for every θ_{dt}, $\theta_{rt} \in S$, $\lambda \in (0, 1)$, we have

$$f(\lambda \cdot \theta_{dt}+(1-\lambda)\cdot \theta_{rt}) \leqslant \lambda f(\theta_{dt})+(1-\lambda) f(\theta_{rt}). \tag{2}$$

$f(\cdot)$ is decided by the subject factors of x, such as personality and emotion. For example, a common trust degree combination function is $\hat{\theta} = \lambda\theta_{dt}+ (1-\lambda)\theta_{rt}$, $\lambda \in (0, 1)$, and a node will choose $\lambda>0.5$ if it trusts more his direct experiences rather than others' recommendations. In light of this, we analyze how to obtain these two kinds of trust degree by Bayes method.

Bayes method is based on subject probability, which is an opinion and degree of rational belief. It is the probability that someone believes something will happen[20]. This degree as a kind of belief is subjective, but also decided by reasoning according to the experience and knowledge of the object facts, judging, analyzing and synthesizing according to the related information.

1.2 Direct trust degree

For the interaction probability here, we use Bayes approach to compute its estimator.

Proposition 1. Let x and y be two nodes in the Grid, and their interaction results are described by binomial events (successful/failure). When there are n times interactions between them, u times successful cooperation, v times failure cooperation, and define $\hat{\theta}_{dt}$ as the probability of successful cooperation at n+1 times. Then, the posterior distribution of successful cooperation between x and y is a *Beta* distribution with the density function:

$$Beta(\theta \mid u,v) = \frac{\Gamma\left(u+v+2\right)}{\Gamma\left(u+1\right)\Gamma\left(v+1\right)}\theta^{u}\left(1-\theta\right)^{v}, \tag{3}$$

and

$$\hat{\theta}_{dt} = E(Beta(\theta \mid u+1, v+1)) = \frac{u+1}{u+v+2}, \tag{4}$$

where $0<\theta<1$, and u, $v>0$.

Proof. Let p=P(S) denote the probability of successful cooperation in one interaction. The prior probability of p can be a random variable in (0. 1). Give no more information about p and according to the Bayes theorem, p can be assumed to a uniform distribution U (0, 1) with the prior distribution $\pi(p)$. When there are n times interaction, which is new information, and let event A = "u times successful results in n interaction", then the result of the interaction is binomial events, which is $P(A \mid \boldsymbol{p} = p) = p^{u}(1-p)^{n-u}$. According to the continuous form of the Bayes theorem,

$$P(A) = \int_0^1 P(A \mid \boldsymbol{p} = p)\pi(p)\,\mathrm{d}\,p \int_0^t p^{u}(1-p)^{n-u}\,dp = \frac{(n-u)!u!}{(n+1)!}. \tag{5}$$

The posterior distribution function $f\,(p|A)$ reflects the information updating of event A. According to the Bayes theorem,

$$f_{f|A}(p \mid A) = \frac{(n+1)!}{(n-u)!u!}p^{u}(1-p)^{n-u} = \frac{\Gamma\left(u+v+2\right)}{\Gamma\left(u+1\right)\Gamma\left(v+1\right)}p^{u}\left(1-p\right)^{v}. \tag{6}$$

The posterior distribution function is not the uniform but *Beta* distribution *Beta* (u+1, v+1) in formula (6).

We can use this distribution function to predict the probability in the future. Let event B = "know u times successful cooperation in n times interaction, and the n+1 time is also successful", then

$$P(B) = \int_0^1 P(B \mid \boldsymbol{p} = p)f(p \mid A)\,\mathrm{d}\,p = \int_0^1 p\frac{(n+1)!}{(n-u)!u!}p^{u}(1-p)^{n-u}\mathrm{d}p = \frac{u+1}{n+2}. \tag{7}$$

According to the properties of *Beta* distribution, the expect value of this distribution $E(Beta(\theta \mid u+1, v+1))$ is formula (7). QED

According to Proposition 1, direct trust degree is related to the probability of successful service provider of the target node and the number of total interactions. It reflects the ability of reliable service a target node provides in the network.

Although formula (4) gives the method of computing direct trust degree, there are still two problems. First, a node may not have interacted with other nodes before, so it cannot measure the trust degree of them. Second, a node may have few interactions with the target nodes, which is not enough to perform trust evaluation. Under both situations, due to the lack of evidences (observations), it is not suitable to use $\hat{\theta}_{dt}$ as the trustworthy of nodes. We need to estimate the confidence value of $\hat{\theta}_{dt}$. In fact, the measure of "reliability" about these intermediates is required. We evaluate the confidence level of trust degree by interval estimation in this paper.

Let ($\hat{\theta}_{dt} - \varepsilon$, $\hat{\theta}_{dt} + \varepsilon$) be the confidence interval with degree γ of $\hat{\theta}_{dt}$, ε is the error level. Confidence degree of $\hat{\theta}_{dt}$ can be modeled as

$$\gamma = P(\hat{\theta}_{dt} - \varepsilon < \theta_{dt} < \hat{\theta}_{dt} + \varepsilon) = \frac{\int_{\hat{\theta}_{dt}-\varepsilon}^{\hat{\theta}_{dt}+\varepsilon} \theta^{u-1}(1-\theta)^{v-1}\mathrm{d}\theta}{\int_0^1 \theta^{u-1}(1-\theta)^{v-1}\mathrm{d}\theta} = \frac{\Gamma(u)\Gamma(v)}{\Gamma(u+v)}\int_{\hat{\theta}_{dt}+\varepsilon}^{\hat{\theta}_{dt}-\varepsilon} \theta^{u-1}(1-\theta)^{v-1}\mathrm{d}\theta. \quad (8)$$

The confidence degree and accuracy of interval estimation are two tradeoff factors. When the number of interactions is fixed, they cannot be improved together. Therefore, according to the rules of Neyman proposed in ref. [20], we consider confidence degree first, and improve accuracy as high as possible in this condition. We can select a threshold of confidence level γ_0, and then improve the accuracy by increasing the number of interactions. When the accuracy is at an acceptable level, that is, $\gamma \geqslant \gamma_0$, the trust degree can be evaluated with the samples (evidences) at this time. The method of increasing the number of samples is collecting the target node's interaction results with other nodes. This kind of trust evaluation is the recommendation trust degree mentioned above. The relationship between number of samples n_0, ε and γ_0 can be modeled as follows:

$$n_0 \geqslant -\frac{1}{2\varepsilon^2}\ln\left(\frac{1-\gamma_0}{2}\right). \quad (9)$$

1.3 Recommendation trust degree

With respect to recommendation trust, we also use the approach above to evaluate it, as the recommendation is formed by several direct interactions. The selection of recommend nodes can also be decided by the trust degree of them.

Proposition 2. Let the interactions between x and y, z and y be independent, and the number of interactions between them be n_1 and n_2 separately, in which the successful cooperation is u_1 and u_2, and failure cooperation is v_1 and v_2. Then, the trust degree of x to y by z can be modeled as follows:

$$\hat{\theta}_{rt} = E(Beta(\theta \mid u_1 + u_2 + 1,\ v_1 + v_2 + 1)) = \frac{u_1 + u_2 + 1}{n_1 + n_2 + 2}. \quad (10)$$

Proof. n_1 and n_2 are independent with the same distribution. According to Proposition 1, the prior distribution of n_1 is the *Beta* distribution. When x observed the interaction results between y and z, it could update its prior information by the Bayes theorem. According to the properties of the *Beta* function, the posterior of it is still *Beta* distribution with the expect value $E(Beta(\theta \mid u_1 + u_2 + 1,\ v_1 + v_2 + 1))$. The proof detail can be found in ref. [21]. Thus, formula (10) is the trust evaluation x to z. QED

When there are several recommendation nodes, it is easy to extend formula (10), and combined with the accuracy analysis above, we can obtain the following:

$$\hat{\theta}_{rt} = \frac{\sum_{\gamma \geqslant \gamma_0} u + 1}{\sum_{\gamma \geqslant \gamma_0} (u+v) + 2}. \quad (11)$$

In formula (11), the following assumption is given: a node can always obtain the interaction history of other nodes by searching the whole network, and increase the number of samples. When satisfying the condition of $\gamma \geqslant \gamma_0$, the searching can be stopped and the trust degree is evaluated by formula (11). Considering the confident level of trust degree, we can define the confidence of the recommendation y to x as the real number of interactions to the total required numbers between them.

$$w_{xy} = \begin{cases} \dfrac{n_{xy}}{n_0}, & \text{if } n_{xy} < n_0, \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

Considering the global trust degree is affected by positive and negative feedbacks separately, the value of $\hat{\theta}_{rt}$ can be mapped onto [−1, 1]. Therefore, formula (11) can be modified as follows:

$$\hat{\theta}_{rt} = \frac{\sum w \cdot (u - v)}{\sum w \cdot (u + v) + 2}. \tag{13}$$

In formula (13), it is not necessary to search the whole network to obtain the number of interactions n_0; for example, a peer can only query the related information by asking its neighbor nodes. This is promising in reducing the communication efficiency of the network.

1.4 Effect of time factor to trust evaluation

Besides the discussions above, we also consider the factor of time similar to ref. [15] in our model. As the trust degree is also affected by time, the impact of time varies according to the trust degree. The more recent the history information is, the more impact the factor has. We introduce a decay factor to reflect the importance of the history information, which decreases as the time passes on. When it decreases to a certain level, it should be discarded. The concept of time segment is used here, which can be a minute, an hour, a day, a month, or even a year. In the practical applications, day is a reasonable unit. It can not only reflect the change of trust degree with time, but also make the computation perform efficiently. The interaction of nodes is composed with a serial of time sequences. Given a certain sequence i, and the number of the successful and failure interactions are u_i and v_i separately, the following formula with the decay factor can be modeled:

$$u(n) = \sum_{i=1}^{n} u_i \cdot \eta^{(n-i)}, \qquad v(n) = \sum_{i=1}^{n} v_i \cdot \eta^{(n-i)}, \tag{14}$$

where $u(n)$ and $v(n)$ are the number of successful and failure interactions after nth sequence, and $0 \leqslant \eta \leqslant 1$. When $\eta = 1$, nothing is affected by history interactions, the whole record is aggregated; when $\eta=0$, the latest history record is considered. The problem of formula (14) is that the whole history interactions are needed to record, and we can solve it by proposing the following recursive algorithm:

$$u(i) = u(i-1) \cdot \eta + u_i, \quad v(i) = v(i-1) \cdot \eta + v_i, \tag{15}$$

where $u(i)$ and $v(i)$ are the number of successful and failure interactions at the ith sequence. The direct and recommendation trust degree at time i can be evaluated by formulas (4) and (13) with $u(i)$ and $v(i)$ introduced in formula (15).

1.5 Analyzing of trust relationship between nodes

The relationships between two nodes, x and y, can be classified into four categories according to whether there are direct interactions and/or recommendations between them[22]. Suppose dt=1 (or 0) represent that there are (not) interactions between x and y, and rt=1 (or 0) represent there are (not) intermediate nodes between them. Then, the four kinds of relationships can be described as $TR(dt, rt)$. We analyze the evaluation of trust degree in those relationships one by one.

1) $TR(dt, rt)$ = (0, 0). It means there is neither recommendations nor interactions between x and

y. Therefore, we should select Uniform distribution, the non-information prior distribution, to be prior distribution. Thus, the estimator of total trust value $\hat{\theta}=1/2$.

2) $TR(dt, rt)$ = (1, 0). It means there are only direct interactions between x and y. Given the threshold γ_0, if $\gamma \geqslant \gamma_0$, the estimator of successful cooperation probability can be evaluated according to formula (4), or trust value $\hat{\theta}=1/2$.

3) $TR(dt, rt)$ = (0, 1). It means there are only recommendations peers between x and y, so direct trust value still can be 1/2, and recommendation trust value can be computed by formula (13). According to formula (1), the total trust value can be computed.

4) $TR(dt, rt)$ = (1, 1). It means there are both recommendations and interactions between x and y. When $\gamma<\gamma_0$, the direct experience is not reliable. This situation is degraded to (3), and more interaction records need to be collected. If $\gamma \geqslant \gamma_0$, the total trust value can be evaluated by formula (1), in which $\hat{\theta}_{dt}$ and $\hat{\theta}_{rt}$ can be computed by formula (4) and (13) separately.

All the discussions above lead to Table 1 below.

Table 1 Evaluating the value of trust in four kinds of relationships

$TR(dt, rt)$	γ	$\hat{\theta}_{dt}$	$\hat{\theta}_{rt}$	$\hat{\theta}$
(0, 0)	–	1/2	0	1/2
(1, 0)	$\gamma \geqslant \gamma_0$	$\frac{u+1}{u+v+2}$	0	$\hat{\theta}_{dt}$
	$\gamma<\gamma_0$	1/2	0	1/2
(0, 1)	–	1/2	$\frac{\sum w\cdot(u-v)}{\sum w\cdot(u+v)+2}$	$\lambda\cdot\hat{\theta}_{dt}+(1-\lambda)\cdot\hat{\theta}_{rt}$
(1, 1)	$\gamma \geqslant \gamma_0$	$\frac{u+1}{u+v+2}$	$\frac{\sum w\cdot(u-v)}{\sum w\cdot(u+v)+2}$	$\lambda\cdot\hat{\theta}_{dt}+(1-\lambda)\cdot\hat{\theta}_{rt}$
	$\gamma<\gamma_0$	1/2	$\frac{\sum w\cdot(u-v)}{\sum w\cdot(u+v)+2}$	$\lambda\cdot\hat{\theta}_{dt}+(1-\lambda)\cdot\hat{\theta}_{rt}$

2 Dynamic level scheduling based on trust model

2.1 Trust dynamic level scheduling algorithm

According to the trust model presented in section 1, this paper extends the traditional DLS algorithm by considering trustworthiness of resource nodes. This algorithm meets the requirement of user tasks in trust, and makes tasks scheduling based on directed acyclic graph (DAG) more reasonable.

The dynamic level scheduling (DLS) algorithm is a compile time, static list scheduling heuristic which has been developed to allocate a DAG-structure application to a set of heterogeneous machines to minimize the execution time of the application[11,13]. At each scheduling step, the DLS algorithm chooses the next task to schedule and the machine on which that task is to be executed by finding the ready task and machine pair that have the highest dynamic level. The dynamic level of a task-machine, (v_i, m_j) is defined to be

$$DL(v_i,m_j)=SL(v_i)-\max\{t_{i,j}^{A},\ t_j^{M}\}+\Delta(v_i, m_j), \tag{16}$$

where $SL(v_i)$ is called the static level of the task, $\max\{t_{i,j}^{A},\ t_j^{M}\}$ is the time when task v_i can begin execution on machine m_j, $t_{i,j}^A$ denotes the time when the data will be available if task v_i is scheduled on machine m_j, and t_j^M denotes the time when machine m_j will be available for the execution of task $v_i \cdot \Delta(v_i, m_\mathrm{j}) = t_i^E t_{i,j}^E$ reflects the computing performance of the machine, t_i^E denotes the execution time of the task v_i on all the free machines, and $t_{i,j}^E$ denotes the execution time of task v_i on machine m_j.

When making a decision of scheduling, DLS algorithm considers the heterogeneous machines, which can adapt the heterogeneous characteristics of resources in Grid environment, but it neglects the trustworthiness of resource nodes in the Grid system. When a task is scheduled to execute on a machine, the trustworthiness of the nodes reflects the reliability of the service it supplies. To address this problem, the trust-dynamic level scheduling (Trust-DLS) algorithm in Grid environment is developed, and the trust dynamic level can be defined as follows:

$$TDL_s(v_i, n_j) = T_s(v_i, n_j)^{\alpha_i} * (SL(v_i)\max\{t_{i,j}^A, t_j^M\} + \Delta(v_i, n_j)), \qquad (17)$$

where $T_s(v_i, n_j)$ is the trustworthiness evaluation of n_j when v_i is scheduled by n_s on n_j, which is equal to $\hat{\theta}$ discussed above. α_i is the QoS factor of v_i, satisfying $0 \leqslant \alpha_i \leqslant 1$ and $\Sigma\alpha_i = 1$. To one task-machine pair (v_i, n_j), when α_i is increased, which means the requirements of task v_i in trust is increased, the scheduling priority will be lowered accordingly. Thus, the algorithm is very scalable and can meet different kinds of QoS requirements. By adjusting α_i, users' different kinds of requirements in trust are satisfied.

2.2 Basic Grid system framework based on trust scheduling

Trust-driven scheduling algorithm can be implemented as a middleware to plug into the Grid system, by which the Grid tasks can be executed on trust nodes efficiently. On one hand, the ratio of failure task execution is reduced; on the other hand, the security of data executive environment is improved. In this section, a basic Grid system framework based on Trust-DLS is presented (Figure 1). There are four tiers in this framework: the first one is the resource tier; the second one is the basic middleware; the third one is the trustworthy scheduler; and the last one is the grid client. In trustworthy scheduler, Schedule Advisor is developed based on Trust-DLS, and the Trust Model is based on the Bayes trust evaluation model.

In the trust scheduling based Grid system framework, the whole process of the task submission and execution is in the following: 1) tasks are submitted to the task queue; 2) the task scheduler fetches tasks from the queue and communicates with the schedule advisor; 3) the schedule advisor communicates with the trust model; 4) the trust model analyzes the local transactions, communicates with the grid trust middleware, obtains the detail trust resource information of task, and transfers them to task scheduler; 5) the task scheduler executes the task on the most trustworthy resource node.

3 Simulation experiments and analysis

3.1 Experiments environment and configuration

In order to evaluate the proposed algorithm, we developed a simulation platform in PlanetLab. PlanetLab is a novel Internet plan advanced by Intel, HP and a large number of famous universi-

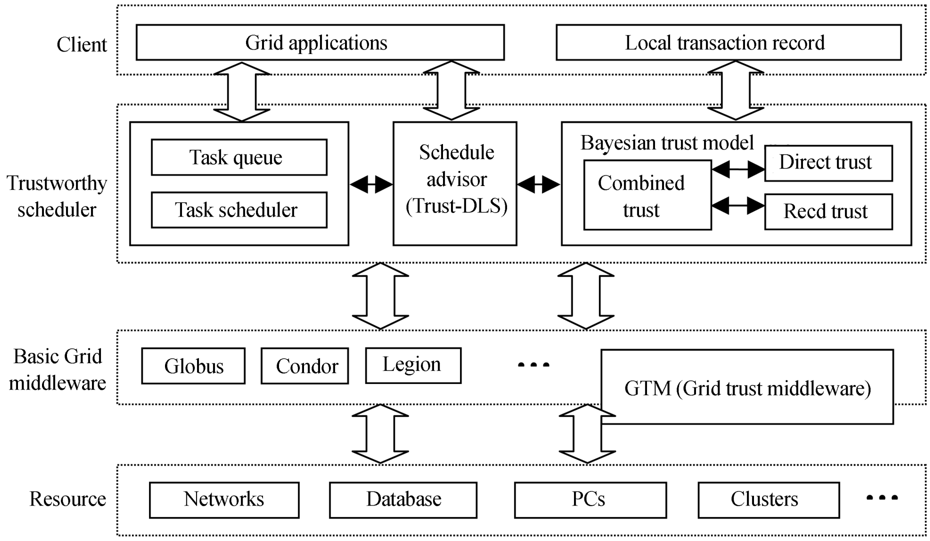

Figure 1 Basic Grid system framework based on trust scheduling.

ties all over the world. It provides network service and application research platforms based on the overlay networks. The goal of the PlanetLab is to provide a test bed through which global research groups can discover and plan the future of the next generation Internet [23,24]. By April 2006, there are 656 nodes and 320 sites over the world, and as one of the CERNET members, we joined the PlanetLab research groups in December 2004.

In the PlanetLab-based Grid environment, the number of the nodes and links is defined previously with the transmission rates of links which are assumed to be uniformly distributed between 1 and 10 Mbit/s, and the initial trust degree of the nodes is generated randomly. The execution time of each task of the task graph is assumed to be uniformly distributed between 10 and 100 s. The volume of data to be transmitted among tasks is decided by the communication to computation ratio (CCR). We assume the average communication time between a task and its successor tasks is set to the average execution time of the task times CCR. Because the performance of the algorithm is related with the application tasks and CCR, we simulate the algorithm with CCR as 0.1, 0.5, 5 or 10. The first two CCR tasks are computation intensive applications, the third one is a normal task and the last two are communication intensive applications. We emphasize two sets of simulation studies, each is carried out ten times, and the final result is the average of them. In all the experiments, the QoS factor α_i is equally in Trust-DSL algorithm, and two kinds of non-cooperative nodes are set to 20% and 30% of the total nodes separately with the failure rate of 80% and 50% when they have tasks on them. ε and γ_0 are set to 0.1 and 0.95 separately in formula (8).

We first discuss the effect of the proposed trust model, mainly focus on the effect of time decay factor η and parameters in the combination function.

3.2 Experiment one: the validity of trust model

We choose the trustworthy combination function $f(\cdot)$ as a simple linear function: $\theta = \lambda\theta_{dt} + (1-\lambda)\theta_{rt}$, $\lambda \in (0, 1)$, and discuss the effect of the trustworthy evaluation by λ. We set the initial direct trustworthy of node x 0.5, and then we re-evaluate the trustworthiness of it by using recommendation information of other nodes. λ is set to 0.0, 0.3, 0.8 and 1.0 separately. Figure 2 shows the results. When $\lambda = 0.0$, the average ratio of successful task execution approached 1 quickly, which fully reflected the effect of recommendation information, and when $\lambda = 1.0$, the recommendation had no effect on the trustworthy evaluation of node x, so the average ratio of successful execution is always equal to 0.5.

Then, we considered the effect of decay factor η to task execution. We set λ to 1.0 and divided time into 20 sequences. In the first ten sequences, we recommended target nodes positive evaluation every time, that is, $(u, v) = (1, 0)$, and in the last ten sequences, we recommended target nodes negative evaluation, that is, $(u, v) = (0, 1)$. η was set to 0.0, 0.3, 0.8 and 1.0 separately. The result is shown in Figure 3, from which we learn that when $n \leqslant 10$, the average ratio of successful execution declined with different degrees. The smaller the decay factor is, the more quickly the trust degree reached a stable level. This indicates that the proposed trust model is adaptable, can objectively reflect the behavior of the nodes, and assure the successful execution of jobs.

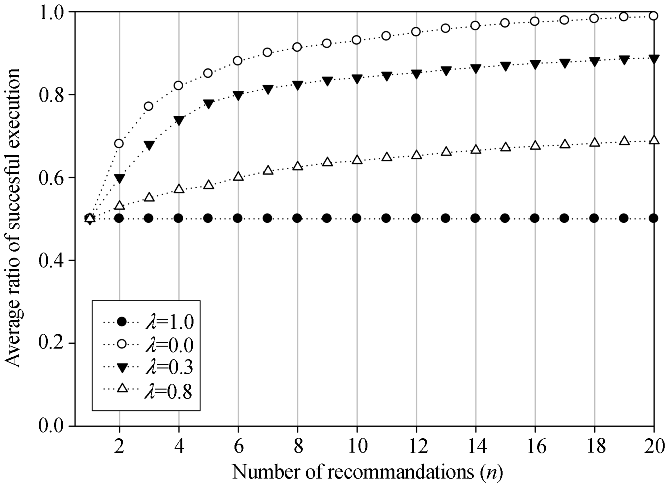

Figure 2 The effect of varying λ to trustworthy.

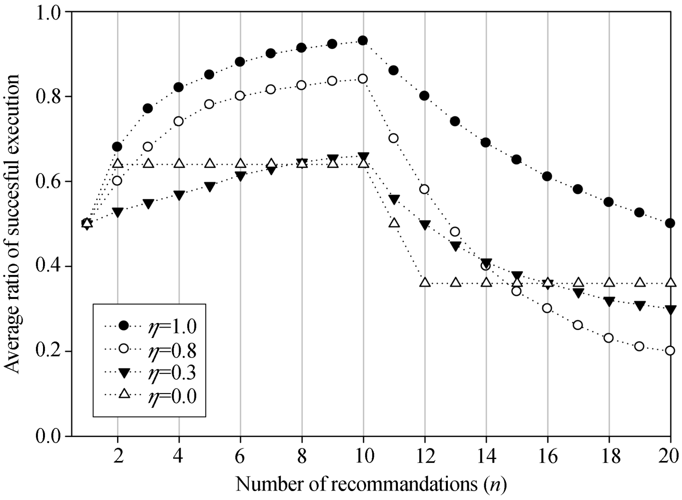

Figure 3 The effect of varying η to trustworthy.

Next, according to the experiments above, we compare the proposed Trust-DSL with DSL under different kinds of configuration. Here, λ and η are both set to 0.8.

3.3 Experiment two: varying number of tasks

In this simulation experiment, CCR is set to 1, task graph with 10 to 100 subtasks is generated randomly, and the number of Grid nodes and links both are set to 200. We compared DLS with Trust-DLS in the scheduling length and the ratio of successful execution. Figure 4 and Figure 5 show the results.

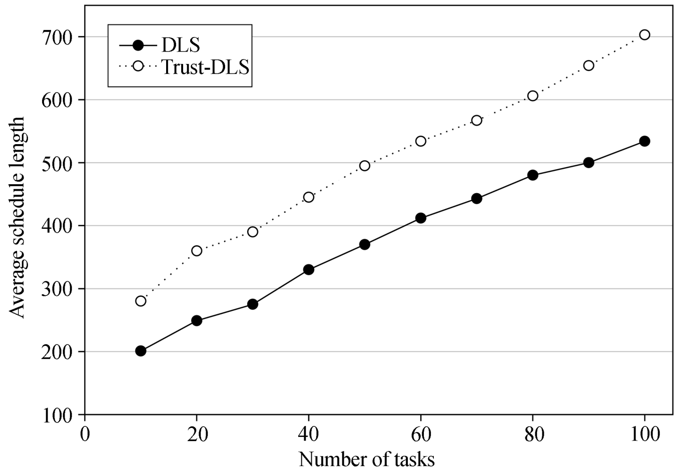

Figure 4 Comparison of scheduling length of DLS with Trust-DLS under varying number of tasks.

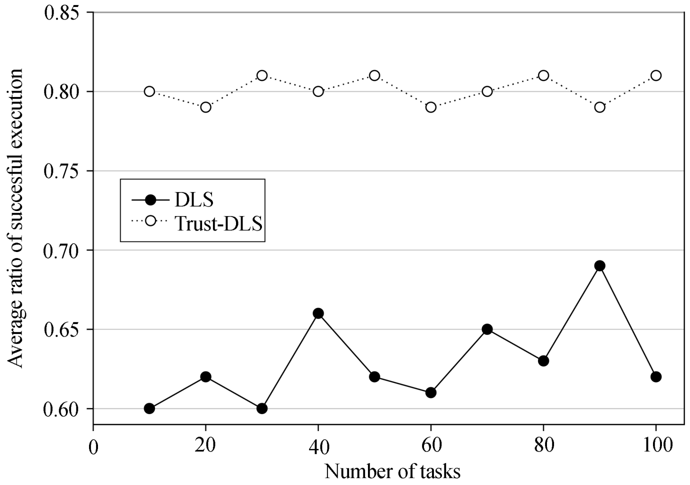

Figure 5 Cmparison ratio of successful execution of DLS with Trust-DLS under varying number of tasks.

In Figure 4, with the number of tasks increased, the scheduling length of the two algorithms both increased as well. The scheduling length of Trust-DLS is a little longer than DLS's. However, the ratio of successful execution of Trust-DLS is much higher than DLS's, as shown in Figure 5. This indicates that the trust mechanism based scheduling algorithm can assure the successful execution of tasks, but increase some scheduling length.

With the same configuration, we experiment the tasks scheduling with CCR equal to 0.1, 0.5, 5, and 10 separately, and analyze the results. Table 2 shows the results of comparing to DLS in scheduling length and ratio of successful execution with Trust-DLS.

Table 2 Under five different CCR, comparison of DLS in scheduling length and ratio of successful execution with Trust-DLS (varying number of tasks)

CCR	Scheduling length	Ratio of successful execution
0.1	31.18344%	20.61970%
0.5	27.44417%	25.66450%
1	21.64457%	29.92020%
5	17.72785%	43.60054%
10	13.55676%	69.23453%

3.4 Experiment there: varying number of nodes

In this simulation experiment, CCR is set to 1, the number of Grid nodes from 100 to 1000 is generated randomly, the number of links is set to 200, and the number of tasks is set to 300. We also compared DLS with Trust-DLS in the scheduling length and ratio of successful execution. The results are shown in Figure 6 and Figure 7. With the number of Grid nodes increase, we can draw the same conclusion: by increasing some scheduling length, trust mechanism based scheduling algorithm can assure the successful execution of tasks.

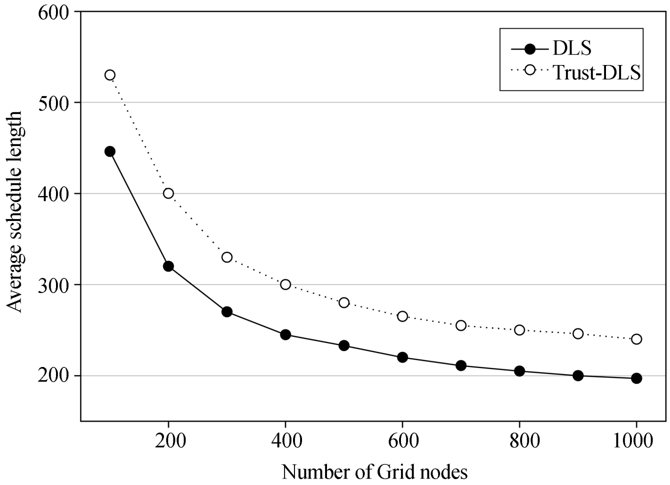

Figure 6 Comparison of scheduling length of DLS with Trust-DLS under varying number of nodes.

We also experiment with the tasks scheduling with CCR equal to 0.1, 0.5, 5, and 10 separately. Table 2 shows the result of comparing to DLS in scheduling length and ratio of successful execution with Trust-DLS.

Table 3 Under five different CCR, comparison of DLS in scheduling length and ratio of successful execution with Trust-DLS (varying number of nodes)

CCR	Scheduling length	Ratio of successful execution
0.1	47.18344%	23.61970%
0.5	41.44417%	31.66450%
1	35.64457%	36.92020%
5	17.72785%	46.60054%
10	13.55676%	71.23453%

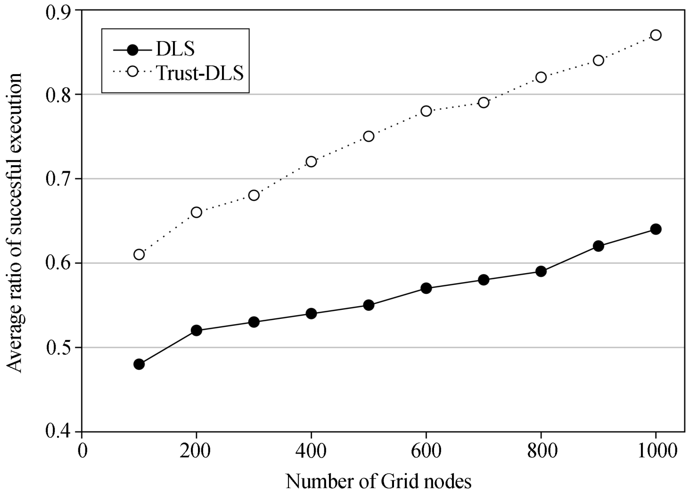

Figure 7 Comparison ratio of successful execution of DLS with Trust-DLS under varying number of nodes.

In general, the results of simulation studies can be summarized as follows:

1) The performance of the Trust-DLS algorithm heavily depends on the CCR. For small values of the CCR, Trust-DLS performs similar to DLS, whereas, for large values of the CCR, Trust-DLS algorithm is preferable due to the fact that it considerably reduces the failure probability at the expense of a relatively small increase in the execution time of applications.

2) The performance of the proposed Trust-DLS also depends on the number of Grid nodes and tasks. With the increasing number of Grid nodes and tasks, the increased performance is several times that of the cost of time, which is very practical in large-scale Grid environment.

3) There is a trade-off between the execution time and failure probability of applications, both of which cannot reach the highest point in the same time.

4 Conclusions

By evaluating the trustworthiness of machines in Grid environment, a kind of trust mechanism-based trusted dynamic level scheduling algorithm was proposed to decrease the failure probability of the task assignments, and assurance of the execution of tasks in a security environment. The main contribution of this study to scheduling systems is that it extends the traditional formulation of the scheduling problem so that both execution time and reliability of applications are simultaneously accounted for. The current trend in designing scheduling algorithms is to respect users' demands, that is, to provide Quality of Service (QoS)-based scheduling. Considering other aspects of security in Grid environment is our future work, such as the probability failure of Grid links and security software deployed in the Grid nodes. How to combine these factors so as to meet the users' requirements of QoS in different kinds of aspects is a challenge in a large heterogeneous computing system.

We are grateful to Yuan Lulai, Chen Bo and Sun Mingjun for their comments. The first author would like to thank Wang Li for her advice that shaped this paper.

1 Foster I, Kesselman C, Tuecke S. The anatomy of the Grid: Enabling scalable virtual organizations. Internat J Supercomp Appl, 2001, 15(3): 200—222

2 Foster I, Kesselman C. Globus: A metacomputing infrastructure toolkit. Internat J Supercomp Appl, 1997, 11(2): 115－128

3 I-Rewinin H E, Lewis T G, Ali H H. Task Scheduling in Parallel and Distributed System. Englewood Cliffs, New Jersey: Prentice Hall, 1994. 401－403

4 Wu M, Gajski D. Hypertool. A programming aid for message passing system. IEEE Trans Parallel Distrib Syst, 1990, (1): 330－343

5 Hwang J J, Chow Y C, Anger F D, et al. Scheduling precedence graphics in systems with inter-processor communication times. SIAM J Comput, 1989, 18(2): 244－257

6 I-Rewinin H E, Lewis T G. Scheduling parallel programs onto arbitrary target machines. J Parallel Distrib Comput, 1990, 9(2):138－153

7 Sih G C, Lee E A. A compile-time scheduling heuristic for interconnection-constraint heterogeneous processor architectures. IEEE Trans Parallel Distrib Syst, 1993, 4(2): 175－187

8 Iverson M, Ozguner F. Dynamic competitive scheduling of multiple DAGs in a distributed heterogeneous environment. In: Proceedings of the Seventh Heterogeneous Computing Workshop, Orland: IEEE Computer Society Press, 1998. 70－78

9 Iverson M, Ozguner F. Hierarchical, competitive scheduling of multiple DAGs in a dynamic heterogeneous environment. Distrib Syst Engin, 1999, 6(3): 112－120

10 Boeres C, Lima A, et al. Hybrid task scheduling: Integrating static and dynamic heuristics. In: Proceedings of the 15th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'03), Brazil: IEEE Computer Society, 2003. 199－206

11 Dogan A, Ozguner F. Reliable matching and scheduling of precedence-constrained tasks in heterogeneous distributed computing. In: Proc. of the 29th International Conference on Parallel Processing, Toronto, Canada: IEEE Computer Society, 2000. 307－314

12 Shatz S M, Wang J P, Goto M. Task allocation for maximizing reliability of distributed computer systems. IEEE Trans Comput, 1992, 41(9): 1156－1168

13 Dogan A, Ozguner F. Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing. IEEE Trans Parallel Distrib Syst, 2002, 13(3): 308－323

14 Mui L. Computational models of trust and reputation: Agents, evolutionary games, and social networks. PhD thesis, Cambridge: Massachusetts Institute of Technology, 2003

15 Jøsang A, Ismail R. The Beta reputation system. In: Proceedings of the 15th Bled Conference on Electronic Commerce. Bled, Slovenia: IEEE Computer Society, 2002

16 Wang W, Zeng G S, Yuan L L. A reputation multi-agent system in semantic web. In: Proceedings of the Ninth (9th) Pacific Rim International Workshop on Multi-Agents (PRIMA 2006), Guilin, LNAI, 4088, Berlin: Springer-Verlag, 2006. 211—219

17 Yuan L L, Zeng G S, Jiang L L, et al. Dynamic level scheduling based on trust model in Grid computing, Chin J Comput, 2006, 29(7): 1217－1224

18 Germano. Walking the web of trust. In: Proc the 9th Workshop on Enabling Technologies (WET ICE'2000). Los Alomitos, CA: IEEE Computer Society Press, 2000

19 Rahman A A, Hailes S. Supporting trust in virtual communities. In: Proceedings of the 33rd Hawaii International Conference on System Sciences. Hawaii: IEEE Computer Society Press, January 2000

20 Thomas L, John H. Bayesian Methods: An Analysis for Statisticians and Interdisciplinary. Cambridge: Cambridge University Press, 1999

21 Hecherman. A tutorial on learning with Bayes networks. Technical Report MSR-TR-95-06, Microsoft Research Advanced Technology Division, Microsoft Corporation, 1995

22 Qi J J, Li Z Z, Wei L. A Trust Model Based on Bayesian Approach, AWIC 2005, Lodz, Poland, LNAI 3528, Berlin: Springer-Verlag, 2005. 374－379

23 Peterson L, Anderson T, Culler D, el al. A blueprint for introducing disruptive technology into the Internet. In Proc. HotNets -I, Princeton, ACM Press, Oct 2002

24 Peterson L, Bavier A, Fiuczynski M, et al. Towards a Comprehensive PlanetLab Architecture. Technical Report PDN-05-030, PlanetLab Consortium, June 2005