QGrid: an Adaptive Trust Aware Resource Management Framework

Li Lin, Jinpeng Huai, *Member, IEEE*

***Abstract*—Recent years have witnessed the rapid development of Grid computing over the Internet, which promises to empower highly desirable resource sharing and cooperation among different organizations. However, there remains a challenging issue facing Grid environments. Malicious or selfish nodes consume precious resources without contribution or even try to destroy the system intentionally. This can severely degrade the system performance and limit the healthy development of Grid systems. To encourage resource sharing and fight against malicious behaviors, we propose an adaptive resource management framework QGrid which integrates trust factor into economic-driven allocation process. Each provider allocates resources according to the bidding price and the trust value of a requester by controlling the corresponding threshold of price and trust value. The incomplete information is a key issue for a provider in determining the two thresholds. We employ a Q-learning technique to resolve the issue, which can adapt to the dynamics of Grid environments. Furthermore, we introduce a simple isolation scheme to secure the Grid system by frustrating malicious participants from joining the system. A QGrid prototype has been successfully implemented in a real Grid test-bed, CROWN. Theoretical analysis and comprehensive experiments have been conducted, which demonstrate the efficacy of QGrid.**

***Index Terms*—resource management; adaptive; economic method and trust; CROWN**

I. INTRODUCTION

THe main objective of Grid computing is to encourage resource sharing and cooperation among different parties. It is very difficult, however, in view of several challenges that arise in real Grid environments. First, malicious nodes, which may damage resources or act against a protocol and try to attack the Grid system, have degraded the performance of system significantly [1]. Second, selfish nodes or free riders, which may consume but do not contribute resources, have been a serious issue [2]. These issues seriously discourage Grid nodes from sharing resources, and in the extreme case they lead to the "tragedy of the commons" phenomenon [3]. Consequently, it is of great importance for a Grid system to be underpinned by certain resource management scheme, which encourages maximum resource sharing among different parties while defending against malicious behaviors.

There have been many attempts [4-6] that apply pricing or micro-currency approaches with a view to dealing with resource incentive. A node earns virtual currency by selling resources, with which it can bid for other resources. However, these approaches, which only consider offered bidding price, are not able to fight against several malicious behaviors. For example, a malicious node consumes resources but does not pay for the resource occupation; or it may intentionally destroy the computing and information resources on a resource provider; or it may often boast of having more resource, which makes good nodes suffer economic losses. These behaviors can seriously discourage resource sharing among different nodes.

In contrast, there have been a lot of research efforts [7-10], which employ the notion of reputation management to defend against adversary behaviors. By contributing more valuable resources and performing more benign behaviors, a node can get higher reputations and hence have higher priority to access other resources. Reputation systems are an effective way for nodes to identify and avoid malicious nodes. Such reputation systems, however, may have several issues. First, it is not trivial to set the initial value of reputation for newcomers. The high initial values may make newcomers laze while the low ones will discourage them. Second, there is no formal specification and analysis of trust evaluation methods provided by such systems. Third, the system needs to be supported with a secure and reliable mechanism to maintain the reputation information of nodes in a distributed way, which is difficult to devise.

To address the issues mentioned above, this paper proposes a adaptive resource management framework based on Q-learning technique, named QGrid, which encourages resource sharing and fights against malicious behaviors. Integrating trust factor into economic-driven allocation process, QGrid deploys different strategies based on different roles of consumer and provider. Consumers try to maximizing their own benefits under constraints of budget and deadline. Providers allocate resources according to the bidding price and the trust value of a requester by controlling the corresponding threshold of price and trust value. QGrid formulates the decision problems for resource consumers and providers respectively, and then gives an appropriate solution for the corresponding optimal decision

Manuscript received August 31, 2007. This work is partially supported by grants from China 973 Fundamental R&D Program (No. 2005CB321803) and National Natural Science Funds for Distinguished Young Scholar (Project No. 60525209).

Li Lin is with the Institute of Advanced Computing Technology, Beihang University, Beijing, China (e-mail: linli@act.buaa.edu.cn).

Jinpeng Huai is with the Institute of Advanced Computing Technology, Beihang University, Beijing, China (corresponding author to provide phone: 8610-8231755; fax: 8610-82316796; e-mail: huaijp@buaa.edu.cn).

policy, which makes nodes have enough incentive to stay and play in the Grid. QGrid is built on the framework TIM proposed in our previous paper [11]. QGrid fertilizes the TIM framework by implementing an adaptive allocation mechanism for resource providers.

The incomplete information is a key issue for a provider in determining the threshold of price and trust value. Exploiting Q-learning technique, we introduce the learning capability to providers, by which they are able to infer the dynamics of the grid environment, and to adjust their thresholds efficiently. We have successfully applied the Q-learning technique to adjust the risk factor in [12], which is used to balance the relative importance of security and economic advantage. In this paper, we further extend the Q-learning technique to adjust the thresholds of price and trust value separately. In comparison, convergence analysis and test under different settings are not presented in [12].

We have successfully implemented QGrid in our CROWN system [11]. Extensive theoretical analysis and experiments demonstrate the efficacy of QGrid in large-scale real Grid systems. Especially, we conduct comprehensive experiments to further study convergence of both price and trust threshold under different settings. The results have showed that each provider can learn to an acceptable policy of adjusting thresholds, even if he locates a complete dynamic environment. Our implementation experiences and experimental results are valuable to study node behaviors and capability.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 gives the system model and problem statement. Section 4 introduces QGrid in detail, which includes price determination of consumers, allocating mechanism of providers, and the scheme for dealing with node joins and departures. Section 5 presents our experimental results. Section 6 concludes the paper and shows some possible future work.

II. Related work

Extensive researches have been conducted on Grid resource management. We give an overview of related work in this section. Emphasis is put on economics-driven approaches, incentive provision and those exploiting game theoretical approaches in resource allocation.

Buyya et al. [13, 14] present some of the economic models which have been used in the human society, such as auction models, commodity market models, contract-net models, bargaining or negotiation models, and bartering models. They discussed possible directions how the economic system in the human society can be applied to Grid computing. The discussion, however, is at a conceptual level and no implementation has been presented. It is still a major challenge to implement these models.

Ongoing efforts are being made to exploit economic models in Grid environments [15-18]. Most of them focus on commodity market models and auction models.

In commodity market based approaches, a market is the mediator between consumers and providers, which mediates all information from both consumers and providers. Although these attempts try to imitate real markets in the human society, the centralized market server introduces many limitations, such as the single point of failure and limited scalability. Furthermore, the central server requires additional organizations for regular maintenance.

In the auction model, resource suppliers and demanders interact with each other through a trusted third party (auctioneer), which is responsible for making auction rules, collecting resources or bids information, and making matches of end users to resource owners. Moreover, this model also has been extended to agent-based electronic commerce [19-22], where artificial software agents participate in auction-based business and make independent decisions based on their own interests. Nevertheless, security concerns of different parties are not (or seldom) addressed in these works.

The automatic negotiation [23] has received a great deal of attention from the multi-agents community. Two or more parties seek a mutual agreement through explicit or implicit exchange of views. Not relying on a trusted third party, the bargaining (negotiation) model has been applied for Grid resource management [24-27]. Weiming Shen et al. [25] present some of the recent work on adaptive negotiation strategies for agent-based load balancing and Grid computing. By reviewing existing bargaining mechanisms and comparing different negotiation strategies and protocols in these mechanisms, Sim [27] presents a comprehensive survey of bargaining model for Grid resource allocation. He attempts to highlight the need of negotiation activities in a Grid environment and discusses the major challenges in Grid resources negotiation. Furthermore, he develops a Grid simulation testbed [26], which can adapt to heterogeneous e-market where participating agents have different types of negotiation strategies. Based on the proposed testbed, he compares the performance of heterogeneous negotiation agents and achieves some favorable empirical results. These works focus on the bargaining of some QoS metrics, such as price, and deadline. However, the focus of these papers is on economic models for Grid resource management and not on security.

Motivated by the need to support transient Grid collaboration, Kaizar Amin et al. [28-30] enhance the existing commodity Grid architecture and propose a new architecture, called Ad-Hoc Grid. The objective is to offer structure-, technology-, and control-independent Grid solutions. The functional principles of Ad-Hoc Grid, such as technology abstract, QoS management, reputation management, and security etc., have significant overlapped with our CROWN Grid (China R&D Environment Over Wide-area Network) [11], but there are some notable differences between them. For example, in QGrid, employing the Q-learning method in trust-aware resource allocation and nodes management, we resolve the issue of incomplete information, which is better able to adapt to the dynamic change of Grid environments. In contrast, our work and the results in the context of Ad-Hoc Grid can be complementary to each other.

In the aspect of incentive research, there are two schemes: soft incentive [7-10] and hard incentive [4-6].

Pricing schemes and token-exchange approaches [4-6] fall in the hard incentive category, in which resource allocation is solely based on bidding prices (or tokens) of consumers. With the higher price, a node is able to get more resources. However, only considering the bidding price can not fight against malicious behaviors.

The soft incentive category includes two representative methods, peer-approved and service-quality. Peers can only access resources of those nodes having a lower or equal use ratings. In addition, the QoS provided to these peers can also be differentiated accordingly [9]. Feldman [8] introduces the concept of generosity and proposes a decision function to help nodes to cooperate with the more generous nodes. Ma et al. [10] elaborate the concept of contribution to maximize utilization in bandwidth allocation. In essence, the work mentioned above is a reputation system, where the reputation [7, 9] (or generosity [8], contribution [10]) of a node is consistent with the quantity of resources contributed by the node. It neither encourages newcomers with low reputations nor discourages malicious nodes from acting as consumers.

Most existing work has modeled Grid nodes as strategic players using the non-cooperative game theory. Ma et al. [10] analyze a completive game among users in the P2P community, and present a resource distribution mechanism RBM-IU, whose objective is to maximize the aggregated utility of bandwidth resources. Kwok et al. [7] focus on an intra-site job execution game in a Grid system. They analyze how a participating computer formulates its mixed game strategy to maximize its own utility and show that the Nash equilibrium could be derived with the help of an intra-site scheduler. Feldman et al. [4] present a price-anticipating scheme in resource allocation and propose a best response algorithm for users to calculate their bids. All the work discussed above assumes that a player has acquired all information of the others in a game, which is not practical for the distributed Grid system.

A few other research efforts have been made in the environment with incomplete information. For example, Wang et al. [31] investigate the problem of bandwidth allocation and propose a market-driven approach. Using a decentralized algorithm, service providers can strategically decide their respective prices to maximize their economic revenues and minimize losses in the long run. The major difference of our work from this work is that we also develop an allocation scheme with trust evaluation in order to fight against malicious consumption of network resources.

Our previous work [32] tries to build a secure Grid resource market. We make the first step to incorporate the trust concept in resource allocation. Each provider determines its allocation schemes based on its own estimation to the utility of consumers, where the competition among different providers is not covered. Owing to the inherent decentralization of the market, it is not trivial to achieve good estimation, especially in consideration of the competition among different providers. In this work, we propose QGrid which is completely distributed and deals well with system dynamism. Exploiting the Q-learning technique, we introduce the learning capability to providers, by which they are able to infer the dynamism of the Grid environment, and to adjust their thresholds of price and trust value. QGrid has better scalability and adaptability when managing realistic resources in the Grid. In addition, we propose an active isolation scheme that circumvent malicious participants and prevent malicious newcomers from joining the system. Thus, QGrid can establish a secure and balanced Grid system in the long run.

III. System Model and Problem Statement

In this section, we introduce the resource organization of the Grid system and market model for resource allocation that we consider. We highlight the key decision problems of Grid nodes. Finally, we describe the trust evaluation model which is critical to the design of QGrid.

A. Resource organization and market models

The distributed resource organization in CROWN Grid adopts a two-tier architecture [12], as illustrated in Fig. 1. The super-sink layer consists of high-capacity servers that collect and exchange information for child nodes in the lower-child layer. Each node in the lower layer is attached to a reliable sink node. The set of child nodes, which is attached to the same sink node, comprise a club, such as $Club_k$ in Fig. 1. A physical organization is a real-world example of a club, which includes all kinds of resource in the organization. The architecture has received enormous attentions [13, 33-35], such as the super-peer architecture in project JXTA [35]. It is important to organize and manage resources effectively so that resource discovery and job execution can be performed efficiently. We believe that this hierarchical organization is a preferable solution, since it increases the scalability of resource management. It can adapt to the distributed network environment where nodes dynamically join and leave.

Following the market model in [12], we assume that a sink node is the resource manager of its club. It is responsible for selling resources and publishing information of available resources. Meanwhile, a child node competes for resources by sending a bid to the sink node. After using the resource, the child node pays an amount of currency for the resource occupation.

B. Decision problem of nodes

In the market, the objective of any selfish node is to maximize its own interest.

On the one hand, a child node (act as a buyer) issues the quantity of resources demanded by its task, and submits a bid to

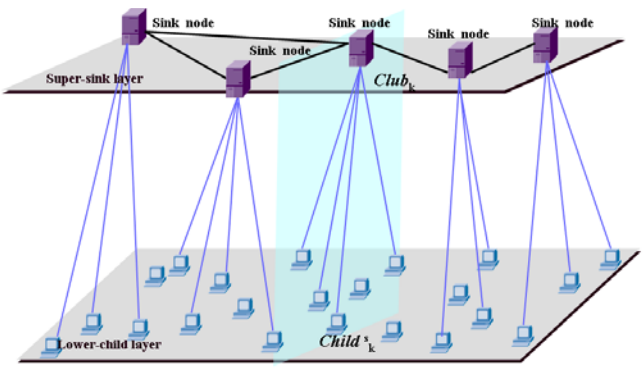

Fig. 1. Club-based two tier organizations

a selected sink node. Apparently, a consumer will be more satisfactory if a lower price gets the completion of its task. Therefore, *it is the key issue for each user to determine its bidding price, which tries to meet deadline and budget constraints*.

On the other hand, similar to traditional markets in the human society, resource price should fluctuate when resource supply and demand change. In addition, it is difficult, if not impossible, for a sink node (act as a seller) to maximize the total revenue of its club or minimize the waste of idle resources, if it neglects any potential malicious nodes. Therefore, it is inappropriate for each sink node to allocate resource merely based on the bidding prices instead of seeking for trustful bidders. This study examines the integration of the notion of "trust" into resource management such that the processes of allocation and dealing with node joins and leaves are aware of security implications. The sink nodes implement admission control and allocate resources according to the price and trust value of each buyer by setting the suitable threshold of price and trust value.

When most of Grid nodes have high trust values, a smaller trust threshold usually can increase the club's revenue. Conversely, increasing the trust threshold can reduce resource waste if there are many malicious nodes in a Grid system. Meanwhile, the balance of supply and demand can be achieved by adjusting the price threshold. Hence, *it is the key issue for each sink node to adjust the two thresholds*.

C. Trust evaluation models

Since nodes in the super layer are able to collect historical records of their child nodes, we evaluate a node's trust value with the widely used history-based method. Following the model in [12], we consider that nodes in the same club can have the direct trust relationship and nodes in different clubs can build up the indirect trust relationship. Based on the above two-tier architecture, we adopt the following method of trust evaluation, where each sink node plays a key role in deducting trust.

Direct trust For $Child^i_k$ in $Club_k$, the other nodes in $Club_k$ will report positive or negative experiences to $Club_k$'s sink node, after their resources were used to execute jobs by $Child^i_k$. Given the numbers of positive and negative experiences are u and v, respectively, then *the direct trust of* $Club_k$*-to-*$Child^i_k$ is

$$directtrust^i_k = \begin{Bmatrix} 1-\lambda^{u-v} & u > v \\ 0 & else \end{Bmatrix}$$

where λ is the probability of success with a single task.

Recommendation trust After nodes in $Club_k$ have used the resources of nodes in $Club_j$ to execute jobs, the nodes in $Club_j$ will report positive or negative experiences to its sink node. Given the numbers of positive and negative experiences are m and n, for the sink node of $Club_j$, *the recommendation trust value of* $Club_k$*-to-*$Club_j$ is

$$rectrust_{jk} = \begin{Bmatrix} 1-\lambda^{m-n} & m > n \\ 0 & else \end{Bmatrix}$$

λ as above.

Trust deduction According to the theorem "Suppose T_1 is the recommendation trust value from B to A, and T_2 is the direct trust value from B to C, then the trust value from A to C is $1-(1-T_2)^{T_1}$" (see [36]), *the trust value of* $Club_j$*-to-*$Child^i_k$ can be computed by:

$$trust^i_j = 1-\left(1-directtrust^i_k\right)^{rectrust_{jk}}$$

It is intuitive that a node is more willing to cooperate with a node having a higher trust value, and the whole system should be secure by isolating those nodes with lower trust levels. In QGrid, the trust evaluation method will be used in the allocation process and the active isolation to malicious participants. It should be noted that the method is not simple. The deduction might be boring and time-consuming and thus nodes intend not to report experiences. To address this issue, each sink node is allowed to take the initiative to set a deadline of deduction time, beyond which it considers the trust value of the corresponding node is close to zero.

IV. The Design of QGrid

In this section, we present the design of QGrid. It is composed of three major components. The first component is bidding price determination for resource buyers. The second component is allocation mechanism for resource providers. As pointed out in the previous section, we have two separate decision problems for resource sellers and buyers, respectively. The first component makes the pricing decision for resource buyers (i.e., child nodes), and the second component makes the resource allocating decision for resource providers (i.e., sink nodes). The third component deals with the dynamic management of the system where new nodes may join and existing node may leave.

For the first component, we essentially inherit the bidding strategy that is proposed in our previous paper [11]. In QGrid, the resource price of a club is set by its sink node. Child nodes bid for resources based on the price. In general, they offer bids that can maximize their own interests. Mathematically, we may characterize a node's interest by a utility function, which includes the empirical benefits and costs incurred in trading resources. The goal of a buyer is to maximize the surplus subject to deadline and budget constraints. The deals of the bidding price strategy are not repeated here and can be found in [11]. In comparison, in QGrid, each sink node can adaptively reset the resource price by adopting certain learning technique, instead of basing on its own estimation to available resource quantity.

Next, we delve into the two key components of QGrid. The next subsection details the allocating mechanism for sellers. And the final subsection discusses the isolation scheme for dealing with node joins and leaves.

A. Allocating mechanism for sellers

In this section, we describe resource allocating mechanism for sink nodes, which exploits the Q-learning technique.

1) Decision problem formulation

In QGrid, by controlling the corresponding threshold of price and trust value, each sink node allocates resources according to the bidding price and the trust value of a requester.

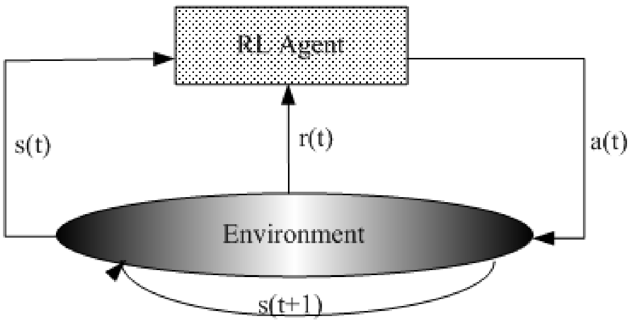

Fig. 2. Interaction illustration between an RL system and the environment

Suppose the environment is at state $s(t)$ in time slot t, after the RL agent obeys certain internal rule to perform action $a(t)$, it shifts to state $s(t+1)$. The agent then receives a immediate reward $r(t)$.

As stated in section III, it is the key issue for each sink node to adjust the two thresholds. During the process of adjusting the thresholds, a sink node should consider not only the historical behaviors of buyers but also the competition against the other sink nodes. This adjusting process can be modeled as a sequential game.

However, in the distributed and dynamic Grid systems, it is highly challenging to acquire complete information about all buyers and sink nodes. It results from the fact that it is infeasible for a sink node to observe how the other nodes' policies impact its utility, or how allocation outcomes is reached. Hence, we formulate the adjusting process of the two thresholds as *a dynamic sequential game with incomplete information*, where the sink node has insufficient knowledge to derive their adjusting strategies.

Markov decision processes (MDPs, [37]) have been popular models for dynamic sequential decision problems, which are generally specified by four components: a finite set S of environment states; a finite set A of actions; a state transition function P and rewards function R. A decision policy is defined as a mapping, i.e., $\pi:(s,a) \rightarrow \pi(s,a)$, where $\pi(s,a)$ is the probability of taking action a at state s. The goal of a MDP is to locate a decision policy that can maximize the long-term return of the decision maker.

We use MDP-based approach to predict the optimal policy of the thresholds adjustment. In the discrete-time domain, the interaction between a sink node and its environment can be modeled as follows:

Definition 1 Let Sr, Dr be the supply amount and demand amount of resources in a period, respectively. Let Ur be the amount of resources that have been allocated in the period, and Wr in Ur be that have been wasted. It is obvious that $\xi=Dr/Sr$ indicates the ratio of demand to supply, and $\zeta=Wr/Ur$ indicates the ratio of resource waste. We divide [0, 1] into $l \geq 1$ intervals, and suppose the current environment is at state s_{ij} if there is $\xi \in [(i-1)/l, i/l)$ and $\zeta \in [(j-1)/l, j/l)$. Thus, $S=\{s_{ij}\}_{i=1,2,\ldots,l;\, j=1,2,\ldots,l}$ is the environment state set of a club.

Definition 2 For the sink node of a club, we interpret its actions as the possible incremental change to be made to the two thresholds:

$$A = \left\{ \left(\Delta pr, \Delta tr\right) \mid \Delta pr \in \Delta Pr, \Delta tr \in \Delta Tr \right\}$$

where $\Delta Pr=\{-1,0,1\}$ is the adjustment set on price threshold and $\Delta Tr=\{-0.01,0,0.01\}$ is the one on trust threshold.

Definition 3 In a period, the utility of a club is defined by

$$r = \eta\left(\log\left(3-\zeta\right)\right)+\psi \qquad (2)$$

where ζ as above, ψ represents the club's total revenue, and η is a positive parameter that represents the degree of dissatisfaction about resource wastage. Different clubs may have different η values. The formulation (2) indicates that a sink node is more satisfied when its club gets higher revenue or its resources are less destroyed. Therefore, for a sink node, the utility of its club can be used to reflect the immediate reward it has obtained from the environment.

In general, when a decision maker has the exact knowledge about functions P and R, he can use dynamic programming techniques to compute the optimal decision policy π^* by starting from any feasible policy π. However, as above noticed, the thresholds adjustment is a dynamic sequential problem with incomplete information, so it is highly challenging to acquire complete information of functions P and R. Fortunately, a sink node is still capable of observing its own action and utility. Hence, we develop an appropriate solution to learn the optimal decision policy directly. An attractive Reinforcement learning technique, *Q-Learning*, has become our choice.

2) Computation of optimal price and trust thresholds

Reinforcement learning is an adaptive method for making the optimal decision in a stochastic and partially observable environment. The interaction between a RL agent and its environment is illustrated by Fig.2. Without prior knowledge about functions P and R, a RL agent incrementally improves its decision policy towards an optimal one through a large amount of trial-and-error with its environment. If an action results in a positive reward, then the probability of the system taking such action will be increased; otherwise, the probability is reduced. The most familiar example of RL is the training of a chess player: a chess player gradually learns the best moves at different positions, by repeatedly taking his moves, and receiving rewards (e.g. $r>0$) or penalties (e.g., $r<0$) from the trainer.

QL is a recent form of Reinforcement learning algorithm that can be used on line. QL works by learning a Q-value function, defined as $Q:(s,a) \rightarrow Q(s,a)$, $s \in S$, $a \in A$, where $Q(s,a)$ is the Q-value associated with the state-action pair (s,a), and represents the expected return when taking action a in state s and then following the current policy to the end. Obviously, once these values have been learned, the optimal action from

TABLE I
THE PROCESS OF LEARNING Q-VALUE FUNCTION

Step 1. For all state-action pair (s,a), it initializes $Qe(s,a)$ to 0, where $Qe(s,a)$ is the estimation of $Q(s,a)$.
Step 2. Repeat (for each episode): From the current state s, it selects an action a. This will cause a receipt of an immediate reward r, and arrival at a next state s'. Based on (s, a, s', r), it updates $Qe(s,a)$ according to the following rule. $$Qe(s,a) \leftarrow (1-\beta)Qe(s,a)+\beta[r+\gamma \max_{a'} Qe(s',a')] \quad (3)$$ where $\beta=1/(1+visit(s,a))$ indicates a learning rate, $\gamma \in [0,1]$ is a discounting factor that discriminates the impact of rewards. $visit(s,a)$ is the total number of state-action pair (s,a) that has appeared before.

any state is the one with the highest *Q*-value. That is, the optimal policy π^* can be found by simply identifying the action that maximizes *Q*-value under the state *s*. Therefore, the essential task of QL is to learn correct *Q*-value function.

Watkins and Dayan [38] have proved that strong convergence of QL to correct *Q*-value function under certain assumption. In QGrid, the thresholds of price and trust value are adjusted by QL. The process that a sink node learns the *Q*-value function is described in Table Ⅰ.

Proposition 1(Convergence Analyses) *If a sink node updates the estimation of Q-value as the rule (3), then he can learn the correct Q-value with all states and actions having been visited infinitely often.*

Proof. The quantity of resources in each club is finite, then there exists a positive integer *H* such that $r \leqslant H$, where *r* denotes any possible utility of the club and *H* represents the maximal utility of the club without malicious wastage.

Let $Qe_n(s,a)$ be the *n*th estimation of $Q(s,a)$. For any (s,a), with $visit(s, a) \to \infty$,

$$\sum_{visit(s,a)=1}^{\infty} \frac{1}{1+visit(s,a)} = \frac{1}{2}+\left(\frac{1}{3}+\frac{1}{4}\right)+\left(\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}\right)+\cdots$$
$$> \frac{1}{2}+\left(\frac{1}{4}+\frac{1}{4}\right)+\left(\frac{1}{8}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}\right)+\cdots$$
$$= \frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\cdots = \lim_{n\to\infty}\frac{n}{2} = \infty,$$

$$\sum_{visit(s,a)=1}^{\infty} \left(\frac{1}{1+visit(s,a)}\right)^2 < \sum_{visit(s,a)=1}^{\infty} \frac{1}{visit(s,a)^2} \text{ and}$$

$$\sum_{visit(s,a)=1}^{\infty} \frac{1}{visit(s,a)^2} < \infty .$$

That is, $\sum_{visit(s,a)=1}^{\infty} \beta = \infty$ and $\sum_{visit(s,a)=1}^{\infty} \beta^2 < \infty$ are established.

By the theorem of Watkins [38], for any $\varepsilon>0$, there is $\lim_{n\to\infty} P(|Qe_n(s,a) - Q(s,a)| < \varepsilon) = 1$. □

Different exploration methods have different learning effects during the learning process. If a decision maker always selects the action with the highest estimation $Qe(s,a)$ at current state *s*, he may miss the one which is less than the highest estimation. In QGrid, we adopt an exploration method which follows the *Bolrzmann distribution*. Given all the *Q*-value estimates of state-action pairs associated with the current state *s*, the probability of taking action *a* is given by:

$$Prab(a \mid s) = \frac{e^{Qe(s,a)N}}{\sum_{a'} e^{Qe(s,a')N}} \qquad (4)$$

where *N* is a positive constant that controls the "sharpness" of differentiating actions corresponding to different $Qe(s,a)$.

Proposition 2 *Suppose that the action set of state s is {a,a'} and* $Qe(s,a)>Qe(s,a')$, *then* $Prab(a' \mid s) \to 0$ *with* $N \to \infty$.

Proof. By the formulation (4), there is

$$Prab(a' \mid s) = \frac{e^{Qe(s,a')N}}{e^{Qe(s,a')N} + e^{Qe(s,a)N}} = \frac{e^{Qe(s,a')N}}{1+e^{Qe(s,a)N - Qe(s,a')N}}$$
$$= \frac{e^{Qe(s,a')N}}{1+e^{(Qe(s,a)-Qe(s,a'))N}} .$$

Since $Qe(s,a) - Qe(s,a')>0$, $Prab(a' \mid s) \to 0$ with $N \to \infty$.□

This proposition suggests that *N* control the probability of executing actions other than the one with the highest $Qe(s,a)$. If *N* is low, or if $Qe(s,a)$ are all the same, it will pick a random action. If *N* is high and $Qe(s,a)$ are different, it will tend to pick the action with the highest $Qe(s,a)$. Hence, at the start, a sink node should select a low *N* (high exploration), and each action has a roughly equal chance of being executed. *N* increases as time goes on, and it becomes more and more likely to pick among the actions with the higher $Qe(s,a)$. Until finally, *Qe* will converge to *Q*, *N* approaches infinite (pure exploitation).

3) Allocation algorithm

A sink node allocates the resources of its club using the following algorithm, as described with the pseudo-code in Table Ⅱ, after it has collected the bids from requesters and recognized their trust values. The results from Proposition 1 and 2 highlight the most important point of the allocation algorithm: exploiting the QL method, sink nodes have the good learning capability, by which they are able to infer the dynamics of the grid environment, and then to adjust their threshold setting policies efficiently.

TABLE Ⅱ
ALLOCATION ALGORITHM IN QGRID

```
At any allocation period t
Input: Request queue and Q-value table
Output: Resource allocation and update Q-value table
Begin
1. Get s(t) by ξ(t) and ζ(t), ξ(t) is the ratio of demand to supply, ζ(t) is the
current ratio of resource waste.
2. Compute P(Δα|s(t)) by Eq (4), based on the current Q-value table,
where Δα=(Δpr, Δtr)
3. Get Δα with the probability P(Δα|s(t)), then pr(t+1) ← pr(t)+Δpr,
tr(t+1) ← tr(t)+Δtr
4. Sort request queue in descending order by C[bp_i/Σ(bp_i)]+(1 − C)
[tr_i/Σ(tr_i)], where bp_i is the bidding price of user i, tr_i is the trust value of
user i, C is the relative weight
5. while(resources > 0 and queue not empty)
      Get request in queue
      if (bp>pr(t+1) and tr>tr(t+1))
         Resource -= request. Resource
     End if
  End while
6. Get the new ratio of resource waste ζ(t+1) and the revenue ψ(t+1)
7. Compute rewards by Eq (2)
8. Update Q-value table by rule (3)
End
```

C is the relative weight, which is used to balance the relative importance of security and economic advantage. We have studied its provision in [12].

B. Dealing with node joins and departures

Some sink nodes may hope to circumvent malicious nodes of their clubs by, for example, blacklisting them. Meanwhile, due to requirement for benefits or malicious intention, nodes may join and leave the system dynamically (switch from one club to another club). Since there is no centralized trust server in QGird, we employ an active isolation scheme to decide whether to accept a node requesting to join.

According to the trust model described in section Ⅲ, the less malicious behaviors a node behave, the higher trust value it has. Apparently, every club is more willing to accept a node with higher trust value and more likely to reject a node with lower trust value. In our approach, the sink node of $Club_j$ actively propagates the constraining desire or the joining request to the other sink nodes.

Different clubs may have different trust records. This may have different opinions about the objective node. After receiving the request, all sink nodes make their own decisions based on their own trust thresholds, which are adjusted by the above QL method. If the trust value of the objective node exceeds its trust threshold, the sink node will give a positive feedback; otherwise, it will deliver a negative feedback. Note that all trust thresholds in different clubs are adjusted adaptively, so each sink node can do reasonable feedback for those new nodes with no corresponding records.

After all feedbacks have been returned to the sponsor, the sink node of $Club_j$, it balances different opinions based on the recommendation of other sink nodes and then makes a final decision. In QGrid, the sponsor can use the following constraint equation to make the final decision

$$\sum_{k=1}^{Number\ \text{of clubs}} rectrust_{jk}\, feedback_k \geq 0$$

where $rectrust_{jk}$ is the recommendation trust value of $Club_k$-to-$Club_j$, $feedback_k$ represents the feedback from the sink node of $Club_k$ and the value is defined by:

$$feedback_k = \begin{cases} 1 & \text{positive feedback} \\ -1 & \text{otherwise} \end{cases}$$

The constraint equation embodies the widely used majority principle. The principle has been explored in our previous work [11]. In this paper, we focus on realizing adaptive trust-aware scheme for dealing with node joins and departures by exploiting the QL technique. We use the simple method of trust management and assume that all sink nodes can be trusted.

Using this scheme, a sink node can also isolate local malicious nodes. In a long run, its child nodes will more rationally request resources, and this increases system resource utilization and also reduces waste. As a result, its recommendation trust will be enhanced incrementally.

V. Implementation

A. Implementation environment

The QGrid approach has been implemented in CROWN system with Java. The cooperation facility among nodes is supported by the CROWN, a fully decentralized Grid middleware infrastructure. The system architecture is illustrated in Fig.3. In our system, all kinds of software, hardware and equipment, and other resources have been packaged into services, which are invoked by users through Web.

There are two kinds of nodes: sink node and child node. As a resource manager, a sink node is responsible for collecting and publishing resources information, collecting bids, inferring trust values of nodes, allocating local resources, dealing with node joins and departures. By monitoring the states of nodes in its club, a sink node can adapt to create its admission control policies and set the prices for local resources. Both provider

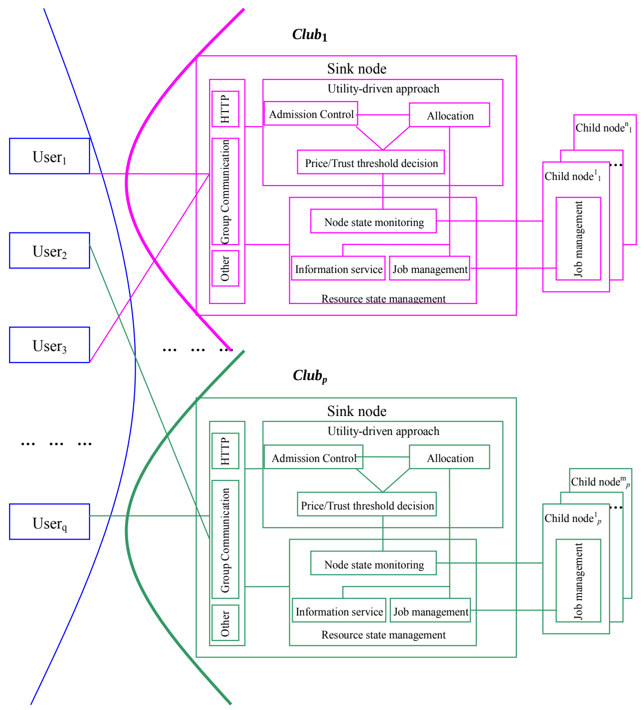

Fig. 3. The architecture of resource management system

and consumer are the roles played by child nodes, which are responsible for various actions, such as generating and submitting bids, scheduling accepted jobs, sending the joining requests. The services from sink nodes and child nodes are deployed on the node servers in CROWN. The timing of deploying a service is illustrated in [12, 32]. Each node employs the group communication software to realize asynchronous or multicast-based information flooding among nodes.

The experiment evaluation is monitored by Load Runner [39], which is a tool for resource utilization and performance analysis. The metrics include CPU/Memory utilization, total job numbers and average response time of jobs so on. In the runtime phase, a multiple servlet filter is deployed in CROWN services container to implement resource allocation control, job handling and request logging. Based on the resource utilization context and the job logging information stored in the environment state repository, an agent decision module is designed to adjust the price and trust thresholds by using QL method.

In the setup phase, we deploy 40 physical nodes equipped with Linux operation system, Intel Xeon Nocona 2.8GHz and 2G RAM. We create several clubs, and each club has a sink node and 50 virtual child nodes.

B. Impact of trust factor

The behaviors of malicious nodes will reduce resource utilization. At first, we study the impact of trust factor. The following equation gives the formal explanation of a club's utility

$$r = \eta(\log(3-\zeta)) + \psi \qquad (2)$$

Where ζ is the ratio of resource waste and ψ is the revenue. We divide consumers into four categories as follows.

(i.) Regular nodes with high bidding prices and high trust values, that is, bidding price ∈ [6,10], trust value ∈(0.5,1];

(ii.) Regular nodes with low bidding prices and high trust values, that is, bidding price ∈ [1,5], trust value ∈(0.5,1];

(iii.) New nodes with high bidding prices and low trust values, that is, bidding price ∈[6,10], trust value∈[0, 0.5].

(iv.) Malicious nodes with high bidding prices and low trust values, that is, bidding price∈[6,10], trust value ∈[0,0.5]

In this experiment, there is a club, where 50 nodes are resource providers. 100 nodes from the other two clubs act as consumers and generate bidding requests. We let the providing club receive the 100 bids each time by varying the value of trust threshold from 0.1 to 0.6 with a step of 0.1. For each scenario, a set of the percentage of No.(iii.) consumers are set to 0.3, 0.4 and 0.5. As shown in Fig.4, with an increase in the number of malicious nodes, a club with a greater trust threshold can get a higher utility. Meanwhile, with the increasing of trust threshold, the club's utility grows until it reaches a maximum value and then reduces.

C. *Evaluation of allocation mechanism*

If the percentage of malicious nodes can be acquired accurately, the above experience may give a bit of hint to provision the corresponding threshold. In practice, it is difficult to obtain the percentage of malicious nodes. Hence, in the allocation mechanism of QGrid, a sink node adjusts its trust threshold and price threshold by observing the ratio of resource waste and revenue at every time slot.

In this section, we first assess the method for adjusting trust thresholds in QGrid. Second, we study the characteristics of price threshold setting strategy. Third, we compare the allocation strategy in QGrid with other fully decentralized strategies. Finally, we examine the effect of our allocation strategy when different clubs have different needs.

In the first experiment, we create two clubs. Each club carries out 100 bidding rounds and each round receives 100 bidding requests. We fix the percentage of malicious nodes at 0.5 and assume that the value of price threshold is fixed at 40. We adopt the two methods in Table III to adjust trust threshold respectively.

TABLE III
DIFFERENT ADJUSTING METHODS OF TRUST THRESHOLD

Method	Principle
Step Increase	To initialize a trust threshold and then increase it in a same step with interaction
QGrid	To initialize a trust threshold and then adjust it by QL

As illustrated in Fig.5, a club's utility tends toward stability after multiple rounds when its trust threshold is adjusted by QL. However, in the Step Increase method, a club's utility decreases finally.

In the second experiment, we compare our price threshold strategy with Round-Robin strategy in the light of job completion time. We create three clubs and each club has 200 unit resources. Resource requests and bids are generated at an interval of 400 time units. We change the ratio of demand to supply from 0.1 to 0.9 with a step of 0.1. The trust threshold is fixed at 0.5. The initial value of price threshold is 40, and each sink node resets the price threshold with an interval of 400 time units.

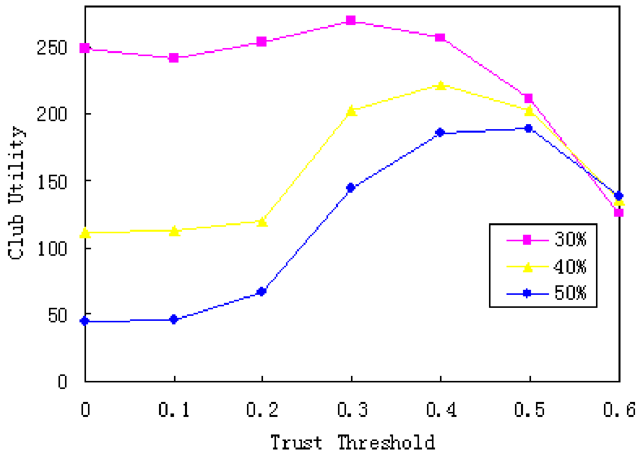

Fig. 4. Different trust threshold Vs. club utility under different percentage of malicious consumers

As seen from Fig.6, the Round-Robin strategy will rapidly increase the complete time of tasks as long as the ratio of demand to supply exceeds 0.4. Our price strategy spends less time to complete tasks compared with the Round-Robin strategy, especially at higher demand. In short, the price threshold strategy in QGrid can alleviate the system pressure by leading consumers to bid available resources rationally.

In the third experiment, we create three clubs, and let these clubs receive the same 100 bids but adopt three different allocation strategies in Table Ⅳ.

TABLE Ⅳ
DIFFERENT ALLOCATION STRATEGIES

Strategy	Principle
Greedy	To allocate resources by bidding prices of requesters in descending order.
Trust-based	To allocate resources by trust values of requesters in descending order.
QGrid	As stated in Table Ⅱ

As shown in Fig. 7, the result of the Greedy strategy is that a club can earn more revenue, when the percentage of malicious nodes is low. And with the increasing of malicious nodes, the higher revenue can be obtained using the Trust-based strategy. However, for the same club, it can always maintain higher revenue using QGrid strategy. We also compare the ratio of resource waste under the above three strategies, where the percentage of malicious nodes changes from 0.2 to 0.5. As illustrated in Fig. 8, with an increase in the number of malicious nodes, the result of the Greedy strategy is that more resources are wasted while the effect of the QGrid strategy on fighting against malicious consumption is close to the Trust-based strategy.

The configuration of the fourth experiment is similar to the first one. We suppose that the clubs have different degrees of dissatisfaction about resource wastage (namely η in Eq (2)) and all price thresholds are fixed at 40.

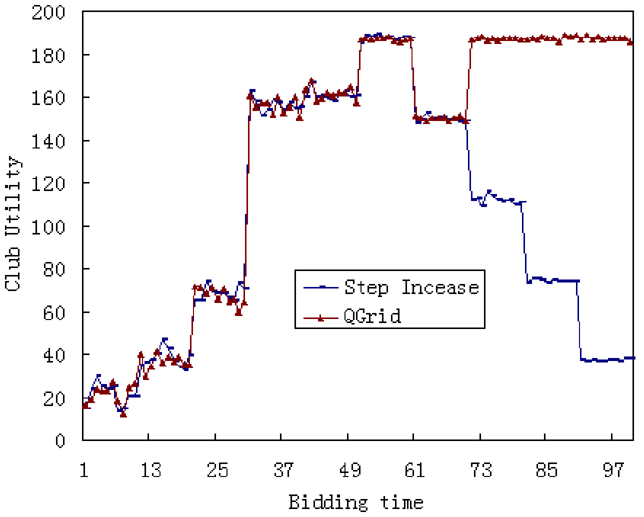

Fig. 5. Different bidding times Vs. club utility under different adjusting methods

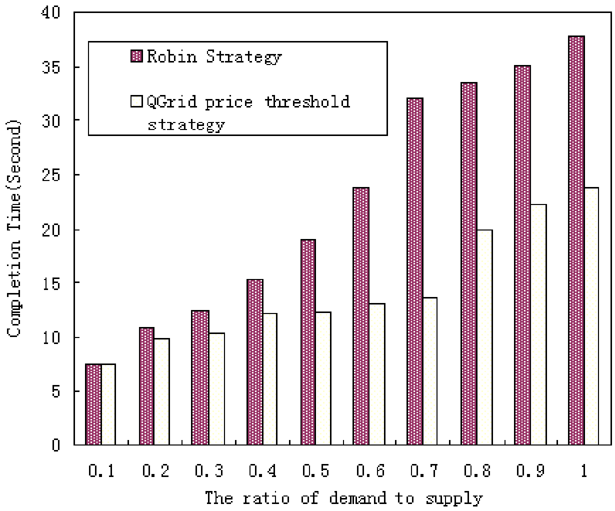

Fig. 6. Completion time Vs. the ratio of demand to supply

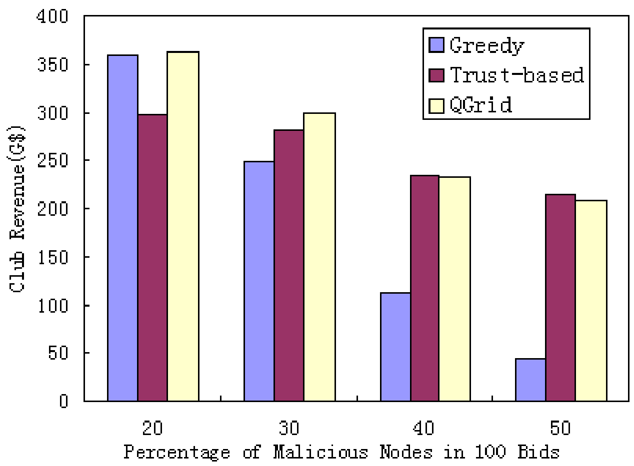

Fig. 7. Percentage of malicious consumers Vs. club revenue under different allocation strategies

As seen from Fig.9, when all clubs adjust their trust threshold using QL method, a club with a smaller η can get a higher utility. Fig. 10 illustrates the utilities of the same clubs by the Step Increase. Comparing Fig.9 with Fig.10, we conclude that a club with higher η is able to get a better utility by reducing its trust threshold.

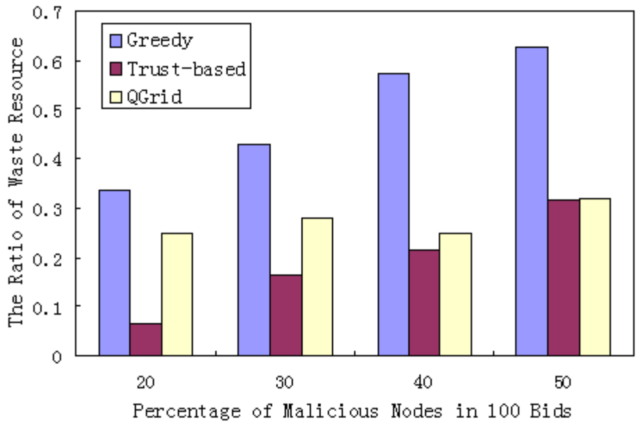

Fig. 8. Percentage of malicious consumers Vs. the ratio of waste under different allocation strategies

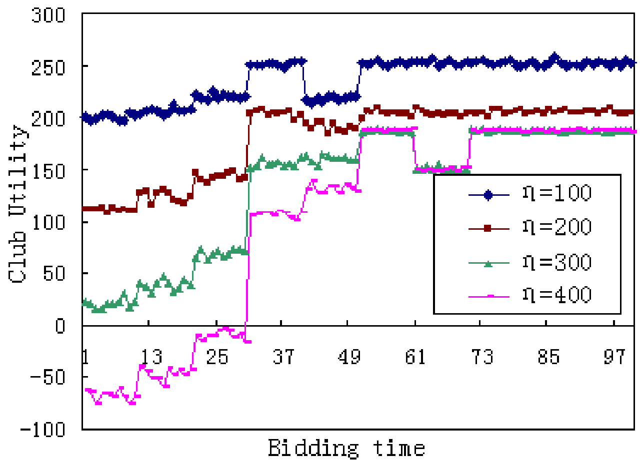

Fig. 9. Different bidding times Vs. utility of different club in QGrid

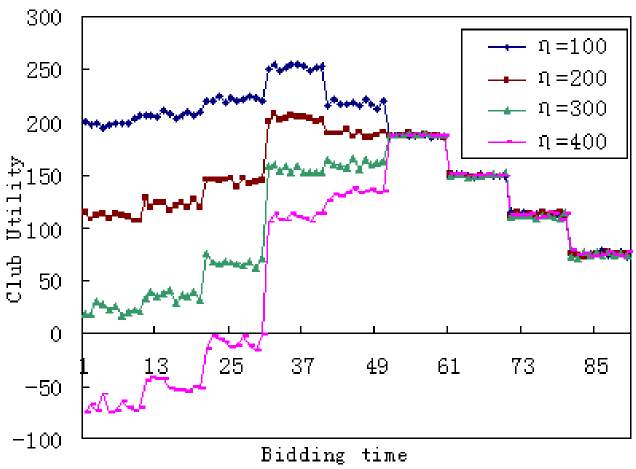

Fig. 10. Different bidding times Vs. utility of different club in Step Increase

D. *Effectiveness of QGrid isolation scheme*

Malicious nodes may newly join the system and existing nodes may switch from one club to another. The result is that the good nodes lose their bids. As a result, both the recommended trust and the utility of the requested club decrease. In this section, we evaluate the isolation scheme of QGrid.

In the case, the same 50 nodes from $Club_3$ want to join $Club_1$ and $Club_2$, where $Club_1$ adopts the isolation scheme of QGrid (as described in section Ⅳ.*B*) while $Club_2$ does not. Let both $Club_1$ and $Club_2$ carry out 80 bidding rounds and each round receives 100 bidding requests. Meanwhile, we set the percentage of malicious joining nodes from 0 to 0.9 with a step of 0.1. If the percentage of malicious joining nodes is 0.4, the sink node of $Club_3$ will give negative feedback to 20 joining nodes after the sink node of $Club_1$ sponsors the feedback requests.

Fig.11 shows that as more malicious nodes enter $Club_2$, the recommendation trust value of the sink node in $Club_2$ without QGrid decreases rapidly. As shown in Fig.12 when the

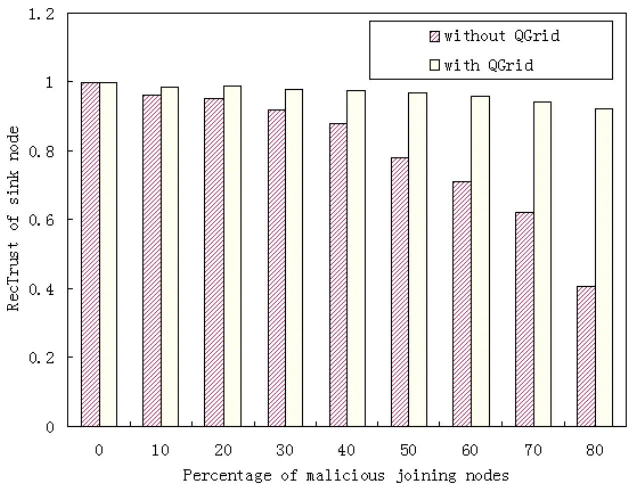

Fig. 11. Recommendation trust varying of sink node

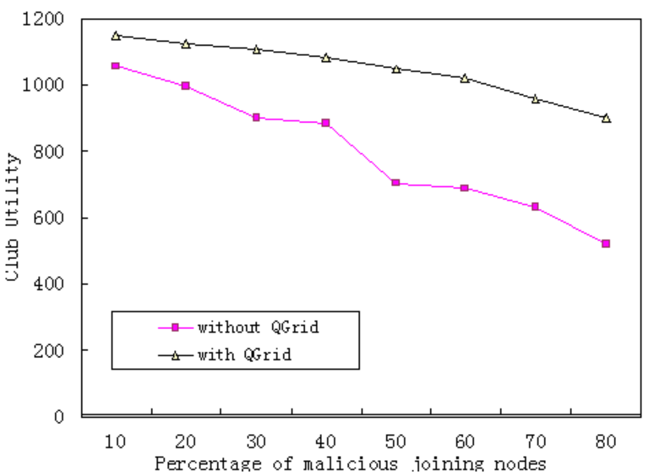

Fig. 12. Isolation of malicious nodes

percentage of malicious joining nodes increases, the utility of both clubs decreases. But the utility of $Club_2$ without QGrid is suffering much more than $Club_1$ with QGrid.

E. *Convergence under different settings*

We are further concerned with the convergence of Q-value and the resulting price and trust thresholds in our QGrid. In our simulations, we have carefully chosen the number of discrete

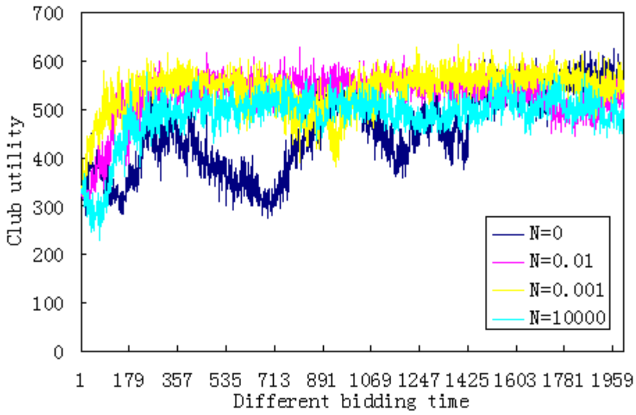

Fig. 13. Convergence of Q-value under different exploration settings

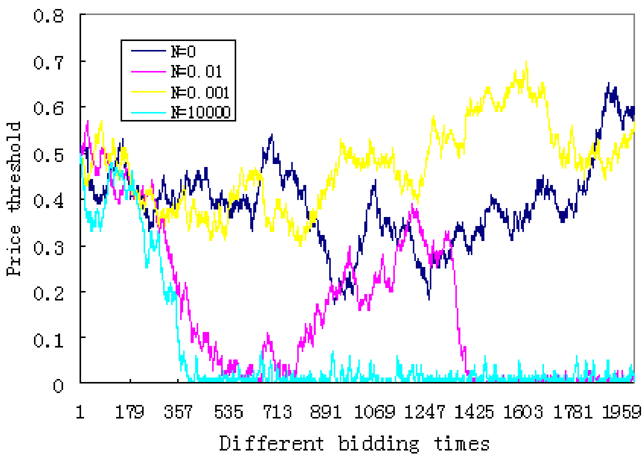

Fig. 14. Variation of price threshold under different exploration settings

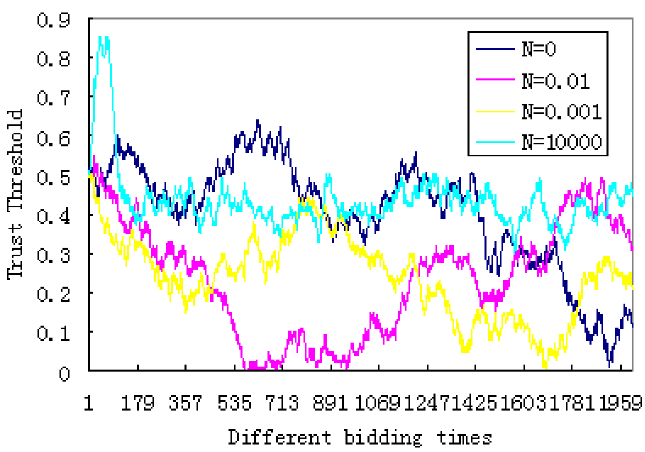

Fig. 15. Variation of trust threshold under different exploration settings

states (l in Definition 1), the discounting rate (γ in the rule (3)), and N in the formulation (4), in order to achieve fast convergence and reasonable thresholds. We use the following figures to show the effects of parameter settings, with l=100, γ=0.9.

The configuration of this experiment is the similar to the experiment in *B*, where consumers are divided into four categories. The percentages of the four categories are set to 0.1, 0.3, 0.3 and 0.3, respectively. The club has 30 resource providers and the total supply amount is 600. The club carries out 2000 rounds allocation (bidding times) and each round receives 100 bidding requests which are generated randomly. We initialize both the price threshold and the trust threshold to 0.5. It means that only those whose bidding price and trust value both exceed the half of total bidding price and trust value can be allocated recourses.

Fig.13 has shown that Q-values can converge quickly after 550 times from the starting point of the simulation as long as N is not 0. When N is 0, the sink node will pick a random adjustment for its price and trust threshold. Therefore, Q-values are unstable.

As shown in Fig.14, the price threshold deviates mostly from the initial value (0.5) when N is 0.01. It is frequently around 0.5 when N is 0.001. This is because a very small N offers little discrimination among different state-action pairs, so that the price threshold may probabilistically stay at 0.5, while a relatively larger N may bring the threshold to a reasonable value at the equilibrium.

We have also tested even larger N, e.g. N=10000, the resulting threshold may reduce swiftly and sharply (as the corresponding price threshold in Fig.14) or stay infinitely at initiate value 0.5, (as the corresponding trust threshold in Fig.15), while Q-values are lower than those derived by smaller N. As proven by Proposition 2, the main reason is that a large N essentially prevents reasonable exploration in the state-action space.

Moreover, we have recorded the results from a complete dynamic environment, where the percentages of the four categories are configured randomly. In comparison to the results in Fig.13, the main difference is that more rounds are required to converge.

VI. Conclusions

In this paper, we have addressed the problem of resource management in face of selfish and malicious nodes in Grids. Integrating the trust factor into the market-driven allocation process, we propose QGrid, which is constructed based on the roles of consumers and providers. For different roles, we discuss the key decision problems and then propose the corresponding solutions. In addition, an active isolation scheme is proposed to secure the system.

The highlights of this paper are as follows. First, we examine the integration of the trust factor into resource management. Second, we introduce the learning capability into providers, which are therefore able to deal with system dynamism, and to adjust their behaviors efficiently. Finally, performance results show that providers can effectively increase the resource utilization as well as fight against malicious behaviors, the isolation scheme help to construct a secure Grid system.

However, some problems are not covered in this paper. First, more efforts should be made to support a more complex trust model, which tackles such situations as a malicious backbone node that attempts to organize a club. Second, our ongoing research will study tradeoffs among different resources, some of which may be complementary to QGrid. Finally, a challenge is to improve the convergence speed of Q-values. This is desirable to adopt or develop some heuristic approaches.

Acknowledgment

We would like to thank Yanmin Zhu, Qin Li for their kind help in the writing of this paper. We would also like to thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] L. A. DaSilva and V. Srivastava, "Node Participation in Ad Hoc and Peer-to-Peer Networks: A Game-Theoretic Formulation," *In Proceedings of the First Workshop on Games and Emergent Behaviors in Distributed Computing Environments with PPSN*, 2004.

[2] E. Adar and B. Huberman, "Free riding on Gnutella," *FirstMonday*, 2000.

[3] G. Hardin, "The tragedy of the commons," *Science*, vol. 162, pp. 1243-1248, 1968.

[4] M. Feldman, K. Lai, and L. Zhang, "A Price-Anticipating Resource Allocation Mechanism for Distributed Shared Clusters," *In Proceedings of ACM E-Commerce Conference*, Vancouver, BC, Canada, 2005.

[5] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for Sharing in Peer-to-Peer Network," *In Proceedings of ACM E-Commerce Conference*, 2001.

[6] Beverly Yang, Tyson Condie, Sepandar D. Kamvar, Hector Garcia-Molina, "Non-Cooperation in Competitive P2P Networks," *In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems(ICDCS), 2005*

[7] Y. K. Kwok, S. S. Song, and K. Hwang, "Selfish Grid Computing: Game-Theoretic Modeling and NAS Performance Results," *In Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid (CCGrid)*, 2005.

[8] M. Feldman, I. S. K. Lai, and J. Chuang, "Robust Incentive Techniques for Peer-to-Peer Networks," *In Proceedings of ACM E-Commerce Conference*, 2004.

[9] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster, "To Share or not to Share: an Analysis of Incentives to Contribute in Collaborative File Sharing Environments," *In Proceedings of Workshop on Economics of Peer to Peer Systems*, Berkeley, CA, 2003.

[10] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, "A game theoretic approach to provide incentive and service differentiation in P2P networks," *In Proceedings of ACM SIGMETRICS/PERFORMANCE*, 2004.

[11] Yu Zhang, Li Lin, Jinpeng Huai, "Balancing Trust and Incentive in Peer-to-Peer Collaborative System" *International Journal of Network Security*, ISSN 1816-353X.

[12] Li Lin, Yu Zhang and Jinpeng Huai, "Sustaining Incentive in Grid Resource Allocation: A Reinforcement Learning Approach". *In Proc. of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*,2007

[13] R. Ranjan, A. Harwood, and R. Buyya, "Grid Federation: An Economy Based, Scalable Distributed Resource Management System for Large-Scale Resource Coupling," in Grid Computing and Distributed Systems Laboratory. Australia: University of Melbourne, 2004.

[14] R. Buyya, D. Abramson, and S. Venugopal, "The Grid Economy" Special Issue on Grid Computing, *Proceedings of the IEEE*, vol. 93(3), pp. 698-714, 2005.

[15] R. Buyya and S. Vazhkudai, "Compute Power Market: Towards a Market-Oriented Grid," *In proceedings of 1st International Symposium on Cluster Computing and the Grid(CCGrid)*, 2001.

[16] R. Wolski, J. Brevik, J. Plank, and T. Bryan, "Grid Resource Allocation and Control Using Computational Economies." In F. Berman, A. Hey, and G. Fox (eds.): Grid Computing – Making the Global Infrastructure a Reality, 2003, J. Wiley, NY.

[17] R. Wolski, J. Plank, J. Brevik, and T. Bryan, "Analyzing Market-based Resource Allocation Strategies for the Computational Grid", *Int. Journal of High Performance Computing Applications*, (15) 3, Sage Publications, 2001.

[18] R.Wolski, J. Plank, and J. Brevik, "G-Commerce -- Building Computational Marketplaces for the Computational Grid", (UT Tech. Rep. #CS-00-439), 2001, http://www.cs.ucsb.edu/~rich/publications/

[19] C. Preist, A. Byde and C. Bartolini, "Economic dynamics of agents in multiple auctions", *In Proceedings of the fifth international conference on Autonomous agents(AGENT)*, 2001

[20] G. Tesuro, J. L. Bredin, "Strategic sequential bidding in auctions using dynamic programming", *In Proceedings of the fifth international conference on Autonomous agents(AGENT)*, 2002

[21] Takayuki ito, Naoki fukuta, Toramatsu shintani, Katia sycara, "BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions", *In proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS)*, 2000

[22] Huiye Ma, Ho-fung Leung, "An adaptive attitude bidding strategy for agents in continuous double auctions", *In proceedings of the 2005 IEEE international Conference on e-Technology, e-Commerce and e-Service(EEE),* 2005.

[23] Jeffrey S. Rosenschein and Gilad Zlotkin, "Rules of Encounter: Design Conventions for Automated Negotiation Among Computers", Cambridge MA: MIT Press, 1994

[24] K. Chao, R. Anane, J.-H. Chen, and R. Gatward, "Negotiating agents in a market-oriented Grid," *In proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid(CCGrid*), 2002.

[25] W. Shen, H. H. Ghenniwa, and C. Wang, "Adaptive negotiation for agent-based Grid computing", *In Proceedings of AAMAS2002 workshop on agentcities: Challenges in Open Agent Environment*s, 2002.

[26] Kwang Mong Sim, "Grid Commerce, Market-Driven G-Negotiation, and Grid Resource Management", *IEEE Transaction on systems, man, and cybernetics-part B: Cybernetics*, Vol.36, No.6, December 2006

[27] K. M. Sim, "A survey of bargaining models for Grid resource allocation", *ACM SIGECOM: E-Commerce Exchang*e, vol. 5, no. 5, pp. 22–32,Jan. 2006.

[28] K. Amin, G. von Laszewski, and A. R. Mikler, "Toward an Architecture for Ad Hoc Grids", In Proceedings of the IEEE 12th International Conference on Advanced Computing and Communications (ADCOM), 2004.

[29] K. Amin, G. von Laszewski, "Quality Assured Ad Hoc Grids", In Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services(ICAS/ICNS),2005.

[30] B. Alunkal, I.Valjkovic, G.von Laszewski, and K.Amin. "Reputation-based Grid Resource Selection", *In the Proceedings of the Workshop on Adaptive Grid Middleware (AGridM)*, 2003.

[31] W. Wang and B. Li, "Market-driven Bandwidth Allocation in Selfish Overlay Networks," *In proceedings of INFOCOM*, 2005.

[32] Yu Zhang, Li Lin, Jin-Peng Huai, XianXian Li and Liang Zhong,"A Resource Allocation Mechanism Providing Trust and Incentive in Grid", *Journal of Software*, Vol.17, No.11, November 2006, pp.2245−2254.

[33] Z. Zhuang, Y. Liu, and L. Xiao, "Dynamic Layer Management in Super-peer Architectures", *In Proceedings of the 33th International Conference on Parallel Processing*, 2004.

[34] Chawathe Y, Ratnasamy S, Breslau L, "Making gnutella-like P2P systems scalable", *In Proceedings of the Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2003.

[35] "Project JXTA", Web Page, http://www.jxta.org/

[36] T. Beth, B. Malte, and K. Birgit, "Valuation of Trust in Open Networks," *In Proceedings of the Conference on Computer Security*, 1994.

[37] Derman C. "Finite state Markovian decision process". New York: Academic Press, 1970

[38] C. J. C. H. Watkins and Q.-l. P. Dayan, "Q-Learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.

[39] "LoadRunner User's Guide ", Web Page, http://javachip.org/Tutorial/LR%20Book3.pdf

Li Lin received her B.S. degree and the M.S. degree in Mathematics from Guangxi Normal University, China, in 2001 and 2004. She is currently pursuing the Ph.D. degree in computer science at Beihang University. Her current research interest span several areas including grid computing, information security, game theory and algebra application.

Jinpeng Huai received his M.S. degree in Electrical Engineering and Computer Sciences from Harbin Institute of Technology and his Ph.D. degree in computer science from Beihang University in 1993. He is a Professor and Vice-President of Beihang University. He serves on the Steering Committee for Advanced Computing Technology Subjects, the National High-Tech Program (863) as Chief Scientist. He is a member of the Consulting Committee of the Central Government's Information Office, and Chairman of the Expert Committee in both the National e-Government Engineering Taskforce and the National e-Government Standard office. Prof. Huai and his colleagues are leading the key projects in e-Science of the National Science Foundation of China (NSFC) and Sino-UK. He has authored over 100 papers. His research interests include Service-oriented computing and Middleware, distributed computing, trustworthiness and security. He is a member of the IEEE.