Computing and Informatics, Vol. 33, 2014, 1139–1167

A UTILITY-BASED REPUTATION MODEL FOR GRID RESOURCE MANAGEMENT SYSTEM

Olga Kussul

National Technical University of Ukraine "Kyiv Polytechnic Institute"
Institute of Physics and Technology
37 Prospect Peremogy, Kyiv 03056, Ukraine
e-mail: olgakussul@gmail.com

Nataliia Kussul, Sergii Skakun

National Technical University of Ukraine "Kyiv Polytechnic Institute"
Institute of Physics and Technology
37 Prospect Peremogy, Kyiv 03056, Ukraine
&
Space Research Institute NASU-SSAU
40 Glushkov Prospekt, building 4/1, Kyiv 03680, Ukraine
e-mail: {inform, serhiy.skakun}@ikd.kiev.ua

Abstract. In this paper we propose extensions to the existing utility-based reputation model for virtual organizations (VOs) in grids, and present a novel approach for integrating reputation into grid resource management system. The proposed extensions include: incorporation of statistical model of user behaviour (SMUB) to assess user reputation; a new approach for assigning initial reputation to a new entity in a VO; capturing alliance between consumer and resource; time decay and score functions. The addition of the SMUB model provides robustness and dynamics to the user reputation model comparing to the policy-based user reputation model in terms of adapting to user actions. We consider a problem of integrating reputation into grid scheduler as a multi-criteria optimization problem. A non-linear trade-off scheme is applied to form a composition of partial criteria to provide a single objective function. The advantage of using such a scheme is that it provides a Pareto-optimal solution partially satisfying criteria with corresponding weights. Experiments were run to evaluate performance of the model in terms of resource management using data collected within the EGEE Grid-Observatory project. Results of simulations showed that on average a 45 % gain in performance can be achieved when using a reputation-based resource scheduling algorithm.

Keywords: Computational grids, reputation model, scheduling algorithm, resource allocation, utility computing, neural network

Mathematics Subject Classification 2010: 91-B16, 68-M20, 68-Q85, 62-M45

1 INTRODUCTION

At present, grid represents a distributed environment that integrates a variety of heterogeneous resources (computing power and storage capacity) within different controlled domains in a way that is impossible for a single institution to do [1]. Grid could be also considered not only for providing high-performance computations, but, in fact, can facilitate interactions between different actors by providing a standard infrastructure and a collaborative framework to share data, algorithms, storage resources, and processing capabilities [2, 3, 4, 5, 6]. Many applications in Earth science, physics, and medicine have been put onto and successfully being solved in grid environment. A set of individuals and/or institutions in grid defined by coordinated resource sharing rules for reaching common goals form a virtual organization (VO) [1]. VOs are formed dynamically, exist for some time and then resolve.

Security has always been an area of intensive research in grid computing. In recent years, a special attention has been brought to trust management in grids [7, 8, 9, 10, 11]. In [12], it is stated that trust is enabling technology and its implementation can provide the possibility to secure electronic transactions. Meanwhile, trust is described as an important and sophisticated object dealing with honesty, truthfulness and reliability of trusted person or service. Nevertheless, there is still no common definition of trust [13]. Two main definitions may be given:

> "When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that s/he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him/her. Correspondingly, when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain from doing so." [14].

> "The extent to which one party is willing to depend on something or somebody in a given situation with a feeling of a relative security, even though negative consequences are possible." [15].

Two types of trust management systems (TMSs) can be discriminated [9]: policy-based and reputation-based. In policy-based systems, entities in a VO establish trust relationships based on certain predefined policies. In reputation-based systems, certain mechanisms exist in order to evaluate the trust which is the function of reputation. Reputation is an assumption about the expected quality or reliability of a resource based on existing information or observations about his/her behaviour in the past [16].

In this paper we extend the existing utility-based reputation model for VOs in grids [17, 18], and present a novel approach for integrating reputation into grid resource management system. The proposed extensions were driven by addressing security threat scenarios [19] and include:

incorporation of statistical model of user behaviour (SMUB) that was previously developed for computer networks and distributed systems [20, 21, 22, 23]. This model was originally used in intrusion detection systems to detect anomalous patterns of user actions;

assigning initial reputation to a new entity in VOs: when organization provides a new resource to be integrated in a VO there are no records from the monitoring system to infer reputation value for this specific resource. One possible way of assigning initial reputation to a new resource is to use a methodology of an active experiment. In particular, there can be several benchmark tasks in the system to estimate the utility function and to provide initial reputation to the resource;

alliance between consumer and resource: since reputation of resource is based on measure of satisfaction of a consumer in relation to this resource we should avoid cheating via collusions among a group of entities [24]. For this purpose, it is reasonable to include into the model a factor that will reflect alliance between the consumer and resource;

time decay function: reputation of resource is based on measuring average value of utility function over certain period of time [10, 24]; but if a VO exists for a considerable period of time (e.g. for years) reputation of resource may vary considerably. That is, it is unlikely to use, for example, two years data to estimate current resource reputation if more recent records are available. So, we propose to incorporate a time lag function into the model that will provide weights depending on the time of the transaction record between consumer and resource;

score function: for different types of services offered by resource providers different reputation values will be used [25]. Namely, we will categorize services into categories, and a resource provider will get reputation value according to such a category. In grid systems, tasks can be categorised by the computational complexity. Successful execution of tasks with a complex workflow and parallel programs (for example, environmental models such as numerical weather prediction [5] or satellite data processing [26, 27, 28, 29, 30]) will provide to a resource provider a higher reputation value.

The proposed model is evaluated in terms of grid resource management. In particular, we will show how the inclusion of reputation model into the resource broker allows the better management of grid system resources. In particular, we consider a problem of integrating reputation into on-line grid scheduler as a multi-criteria optimization problem. A non-linear trade-off scheme presented in [31] is applied to form a composition of partial criteria to provide a single objective function. The advantage of using such a scheme is that it provides a Pareto-optimal solution partially satisfying criteria with corresponding weights. The proposed scheme is compared to a multiplication scheme for incorporating reputation within simulations that were run using data collected within the EGEE Grid-Observatory project.

2 RELATED WORKS

Several concepts of building and using reputation models in grid systems, and incorporating reputation into scheduling algorithms have been proposed.

In [24], a trust model for grid systems is presented which is further used to incorporate the security implications into scheduling algorithms. Three scheduling heuristics are modified to incorporate the trust notion using a multiplication scheme. It is formally shown that the makespan obtained by a trust-aware scheduler is always less than or equal to the makespan obtained by the trust-unaware scheduler that uses the same assignment heuristic.

The GridEigenTrust model [32] is the extension of the EigenTrust model [33] previously developed for P2P systems. In GridEigenTrust, a technique is introduced to derive global trust values of organizations through hierarchies, not an overall pool of individual entities. For this purpose, an eigenvalue-based trust calculation algorithm is used. The advantage of this algorithm is that it converges rapidly and introduces less overhead than computing global trust values for individual entities within every context. If the organisation will report inaccurate trust information on its entities, such organisation will be penalized by lowering the global trust of the organization. Reputation is further integrated into a QoS management system providing a way to re-evaluate resource selection and service level agreement (SLA) mechanisms.

In [8], a fuzzy-logic trust model for distributed trust aggregation through fuzzification and integration of security attributes is proposed. The following metrics are introduced to estimate trust index of a grid site: site reputation (aggregating prior job, execution success rate, cumulative site utilization, job turnaround time, and job slowdown ratio) and self-defence capability (attributed to the risk conditions and hardware and software defences deployed at a grid site). A Secure Grid Outsourcing (SeGO) system is developed for securely scheduling a large number of autonomous and indivisible jobs to grid sites. A min-min heuristic for on-line job scheduling was used in the study. Running scalable NAS and PSA workloads over simulated grids, significant performance gains after trust aggregation into the scheduler were obtained.

PathTrust [34] is a reputation system suggested for choosing members of the VO while its formation. The organization has to register with the enterprise network by providing some certificates to enter the VO. Beside of user management the enterprise network provides centralized reputation service. When the VO is resolved each member gives feedback values for reputation server and other members s/he had interacted with. The proposed model feels the lack of dynamics, because the feedback value is collected only when the VO is resolving.

The work described in this paper builds on the utility-based reputation model that can be used for users and resource providers [17, 18]. This model is based on the calculation of the utility function that expresses the satisfaction of the entity with its interaction with other entities with respect to the key features specific for the assessable entity. Reputation for users is estimated according to their resource usage and correspondence to a VO policy, while reputation for resource providers is assessed according to the quality of service (QoS) provided. Simulations were run to demonstrate the efficiency of the model for off-line jobs scheduling. In particular, for a batch of jobs submitted by the users, each resource obtained a number of service requests proportionally with its reputation. So, only one criterion (i.e. resource reputation) is used to map jobs onto grid resources. It was demonstrated that using a reputation-based scheduling, the total completion time is better with around 25 % comparing to the round-robin scheduler.

In [7], a special attention has been brought to economical issues in grids along with information asymmetry. Information asymmetry refers to the situations when resource providers and users share different portions of information on the quality of service being provided. These issues are taken into consideration while proposing a reputation-based framework for enabling grid markets. Information asymmetry in grids is discussed in details in [35]. A reputation-based mechanism allowing grid service broker to deal effectively with hidden information is presented.

In [11], the grid reputation-policy trust management service (GREPTrust) for managing resource selection in computational grids is presented. This framework exploits a novel reputation-policy based trust model in which service consumers take active part in reputation evaluation process. The proposed model is further used for grid resource selection allowing setting multiple criteria definitions. In order to allocate a job onto grid resources, a round-robin scheduling algorithm and the probability of each resource to fail a computation cycle are used. Finally, resources are selected based on the predefined threshold.

A reputation-driven economic framework for grid resource management, called HOURS, is presented in [36]. The framework is targeted at automatic rescheduling, self-protection, incentives, heterogeneous resource sharing, reservation, and SLA in grid computing. A reputation-based resource scheduler is designed targeting to reduce the number of resubmissions and task/job failure rates. The simulations that were done using traces from the TeraGrid environment showed that using reputation-based resource scheduling the job failure rate can be reduced from 3.82 to 0.70 compared to sequence resource scheduling.

In [37], a genetic algorithm for job scheduling is presented capturing the heterogeneity of fault-tolerance mechanisms problem in a computational grid. The risk relationship between jobs and nodes is defined by the security demand and the trust level. The proposed algorithm has shorter makespan and the improved job failure rate comparing to the min-min and sufferage algorithms.

Analysis of relevant works shows that incorporation of reputation into grid resource management system allows us to improve efficiency of allocating jobs onto grid resources. However, in most cases linear or multiplication schemes are utilised. These schemes have the following shortcoming: they do not provide robust solutions at values close to the extreme ones (for example, 0 or 1) [31]. As to the threshold scheme, in which jobs are scheduled to the resources with reputation not less than some defined value, disadvantage comes from the need to select threshold value. This solution could not always be feasible, especially for automatic scheduling. Therefore, in this paper we propose to apply a non-linear trade-off scheme [31] to integrate reputation onto grid scheduler.

3 UTILITY-BASED REPUTATION MODEL FOR VOS IN GRIDS

The reputation model described in this paper is based on the model proposed in [17, 18]. The main modifications are associated with reputation model for VO users, as well as with addition of several new components for resource reputation model. The rationale for these modifications is to address the most important and critical security threat scenarios for trust and reputation models [19]. All proposed modifications are described in details in the following subsections.

3.1 Basic Notations

A basic concept in the reputation model is the organisation [17, 18]. The organisation provides resources, and there are users associated with this organisation. Therefore, the organisation can be described by the following attributes:

$$o = \left\{ o\_id, \bigcup_i r_i, \bigcup_j u_j \right\}, \tag{1}$$

where o_id is the organisation's identifier, $\bigcup_i r_i$ and $\bigcup_j u_j$ are resources and users associated with this organisation, respectively.

We will denote all existing organisations by O.

A virtual organization (VO) can be modelled as a set of organisations. The organisations integrate their resources on a temporary or permanent basis to achieve common goals [1]. It is to be noted that in general case an organisation may provide to a VO only a subset of its resources, and the same resource can be used in different VOs. The same stands for users of the organisation. Therefore, a VO is represented by the following set:

$$vo = \left\{ vo\_id, \bigcup_k r_k, \bigcup_l u_l, f_{vo}(), g_{vo}() \right\}, \tag{2}$$

where νo_id is a VO's identifier, $\bigcup_k r_k$ and $\bigcup_l u_l$ are resources and users from multiple organisations that participate in a VO, respectively, $f_{vo}()$ and $g_{vo}()$ are membership functions defined in the following way:

$$f_{vo} : \bigcup_k r_k \;\rightarrow\; O, \text{i.e. } f_{vo}(r_k) = o, \tag{3}$$

$$g_{vo} : \bigcup_l u_l \;\rightarrow\; O, \text{i.e. } f_{vo}(r_k) = o. \tag{4}$$

In general case, these functions can be maintained by the Virtual Organisation Management Server (VOMS). Using these functions we can retrieve any required information on membership of organisations, resources and users in VOs. For example, the set of resources provided by the organisation o in the specific νo is given by (using Equation (3)):

$$\{r \epsilon \bigcup_k r_k : f_{vo}(r) = o\} \equiv f_{vo}^{-1}(o). \tag{5}$$

In the same way we can list all users from organisation o participating in vo (Equation (4)):

$$\{u \epsilon \bigcup_l u_l : g_{vo}(u) = o\ \} \equiv g_{vo}^{-1}(o). \tag{6}$$

Suppose we want to list all organisations from O that provide resources within specific vo, or whose users participate in this VO. Such sets are given respectively by (Equations (3)–(4)):

$$\{o \epsilon O : \text{if } \exists\ r \epsilon \bigcup_k r_k \text{ that } f_{vo}(r) = o\}, \tag{7}$$

$$\{o \epsilon O : \text{if } \exists\ u \epsilon \bigcup_l u_l \text{ that } g_{vo}(u) = o\}. \tag{8}$$

Let us denote all existing VOs by VO. Suppose we want to retrieve all VOs where a resource r from specific o is used, or where the user u from o participates in. These sets are given respectively by

$$\{vo\ \ \epsilon VO : f_{vo}(r)\ \ = o\} \equiv VO|_r, \tag{9}$$

$$\{vo\ \ \epsilon VO : g_{vo}(u)\ \ = o\} \equiv VO|_u. \tag{10}$$

These basic notions are used in the following subsections to describe reputation models for resource providers and user.

3.2 Reputation Model for Resource Providers

The reputation model is based on the utility function that measures the level of satisfaction of a user in relation to service provider. In order to define utility function an auxiliary function that indicates the SLA accorded between a VO user and a resource provider for a particular resource within a VO is implemented [17]:

$$SLA : \bigcup_l u_l \times \bigcup_k r_k \times \bigcup_m vo_m \to \mathbf{R} \tag{11}$$

where **R** denotes the set of real numbers.

The SLA value represents quality of resource provider as expected by user [17]. In order to define utility function based on SLA value we describe the notion of *Event*:

$$\mathit{Event} = T \times \bigcup_l u_l \times \bigcup_k r_k \times \bigcup_m vo_m \times \{\text{QoS name}\} \times \mathbf{R} \tag{12}$$

where T is a time domain.

Therefore, the event is characterised by the following attributes

$$\{t, u, r, vo, QoS, \nu\} \tag{13}$$

where t indicates time, QoS is a name indicating QoS of interest, and ν is a real QoS value measured by grid monitoring system after user-resource interaction.

Trace corresponds to the sequence of events (Equations (12)–(13))

$$\mathit{Trace} = \bigcup_p \mathit{Event}_p = \bigcup_p \{t, u, r, vo, QoS, \nu\}_p. \tag{14}$$

Before defining a utility function and reputation we will introduce three functions: the first one will characterise possible alliance between consumer and resource in order to avoid cheating [24], the second one will account for a time when utility was estimated [10, 24], and the third one will provide different scores depending on the type of the provided service [25]. These functions provide extensions to the utility function and reputation originally proposed in [17].

Function $h(u,\ r)$ will take a value between 0 and 1 and will show the level of alliance between user u and resource r. If there is no such alliance between targets, $h(u,\ r)$ will have a higher value. For example, one possible way of defining $h(u,\ r)$ is as follows

$$h(u,r) = \begin{cases} 1, & \text{if} f_{vo}(r) \neq g_{vo}(u) \\ \theta, & \text{if} f_{vo}(r) = g_{vo}(u) \end{cases} \tag{15}$$

where θ is a parameter.

Function $z(t,\ t_c)$ will show what past records on user-resources interactions should be taken into consideration to estimate reputation of specific resource. Here t is the time, and t_c is a parameter. In the simplest form $z(t,\ t_c)$ could be a stepwise function

$$z(t,t_c) = \begin{cases} 1, & t \geq t_c \\ 0, & t < t_c. \end{cases} \tag{16}$$

Function $s(\mathit{type}(r))$ will provide different values for different types of services provided by the resource r (function $\mathit{type}(r)$ maps into category of service).

Now, we can define a utility function using Equations (11), (12), (15):

$$\mathit{utility} : \mathit{Event} \to \mathbf{R},$$

$$utility(\{t,u,r,vo,QoS,\nu\}) = \begin{cases} h(u,r)s(r), & \text{if } SLA \text{ is met,} \\ penalty(\nu, SLA)h(u,r)s(r), & \text{otherwise} \end{cases} \tag{17}$$

where SLA is the agreed SLA value between the user and resource provider, $penalty(\nu, SLA)$ is a penalty function imposed on a resource provider if the agreed SLA is not met.

The form of penalty function depends on the QoS in place. For example, for time metrics which are usually to be minimised a penalty function can be represented by [10]

$$penalty(\nu, SLA) = \begin{cases} 1, & \text{if } \nu \leq SLA \\ \frac{SLA}{\nu}, & \text{if } \nu > SLA. \end{cases} \tag{18}$$

Let us denote a set of traces that are used to estimate the reputation of resource r in a vo up to the current time t with

$$Trace|_{(vo,r,t)} = \{\{t',u',r',vo\_id',QoS',\nu'\} \in Trace : r = r', vo\_id = vo', t' \leq t\}. \tag{19}$$

Let us denote a set of $utility()$ function values derived from traces $Tracej_{(vo,r,t)}$ with

$$O_{(vo,r,t)} = \{z(t,t_c) \cdot utility(\{t,u,r,vo,QoS,\nu\}) | \{t,u,r,vo,QoS,\nu\} \in Trace|_{(vo,r,t)}\}. \tag{20}$$

A reputation is expectation of $utility()$ function (in terms of probability theory)

$$rep(vo,r,t) = E[utility(O_{(vo,r,t)})] = \int utility(O_{(vo,r,t)}) p_{utility}(O_{(vo,r,t)}) dO_{(vo,r,t).} \tag{21}$$

If we do not want to discriminate values from $utility()$ function by time then we might use $z(t,t_c) = 1$.

In order to approximate expectation we can use a sample mean (Equation (20))

$$rep(vo,r,t) = \frac{1}{|O_{(vo,r,t)}|} \sum_{x \in O_{(vo,r,t)}} x, \tag{22}$$

where $|\cdot|$ denotes the cardinality of the set.

The reputation of an organisation o in VO is the aggregation of the reputation of all resources it provides to VO (using Equations (5), (22)):

$$rep(vo,t) = \frac{1}{|f_{vo}^{-1}(o)|} \sum_{r \in f_{vo}^{-1}(o)} rep(vo,r,t). \tag{23}$$

The reputation of a resource in all VOs can be estimated as follows (using Equations (9), (22)):

$$rep(r,t) = \frac{1}{|VO|_r|} \sum_{vo \in VO|_r} rep(vo,r,t), \tag{24}$$

3.3 Reputation Model for Users

In the reputation model proposed in [17] the corresponding model for user is built using a penalty function. If a user performs an action that does not conform to a VO or resource policy, the user is charged with a penalty. This penalty is used to estimate utility function for the user, and subsequently a reputation of the user; but audit of user actions and checking these actions against the VO or resource policy is a complex problem, especially from implementation perspective. There should be clearly defined criteria to assess user actions and corresponding software components should be available to do that.

In order to assess user reputation we propose to exploit methods and models traditionally used in intrusion detection systems (IDSs). Some of the research has been made on assessing and discussing the use of IDSs for grids [38, 39].

In our case, we propose to use a statistical model of user behaviour (SMUB) that was developed by us for computer networks and further extended for distributed systems, in particular grid systems [20, 21, 22, 23]. This model is based on the analysis of statistical data that is gathered after a user executes actions, in particular jobs, in a grid system. The model was built and verified on real data that were collected from GILDA infrastructure[1] of the EGEE project. In conducted experiments the following performance metrics were achieved: the model was able to discriminate behaviour of different types of users with 86 % overall rate with average false alarm rate being 7.48 % and average miss rate being 20.9 %.

In particular, the model accounts for different statistical parameters given by the following attributes:

$$\{S, ET, CPU, WT, CW, ES, CT, STD, RAM, VM, VO, RB\}, \tag{25}$$

where S is a site where a user job was executed, ET is a execution target, CPU is a job CPU time, WT is a job wall time, CW is defined as CPUWall = CPU/W, ES is a job exit status (success or with errors), CT is a job creation time in grid system, STD is a start time difference, i.e. difference between time when a job started to execute on computational resource and time when job was sent to grid system by the user, RAM is a volume of RAM used by the job, VM is a volume of virtual memory used by the job, VO is a virtual organization a user belongs to, RB is a resource broker hostname that was used to schedule the user job.

This set of parameters is used to discriminate user behavioural patterns from other users and to detect how a current user action corresponds to actions made in the past. Such patterns could include, for example, situations when a job is running for a significant amount of time, or when the CPU utilization is up to 100 % [9]. In order to detect such patterns from data that was logged during user activity we use intelligent techniques, namely neural networks [40]. For each user a neural network is trained to form an opinion to discriminate between the normal and abnormal user behaviour. When neural network is trained a target output is

[1] GILDA infrastructure: `https://gilda.ct.infn.it`

set to 1 for input data that corresponds to the normal user behaviour, and to 0 for data that corresponds to the abnormal user behaviour. In order to represent both cases (normal and abnormal) in training data sets we use records from grid monitoring system: records about past user actions represent the normal behaviour while records from other users and synthetically generated data represent the abnormal user behaviour. The synthetic data can be incorporated into the training data sets in order to represent abnormal patterns that were not present in data sets from monitoring system. Therefore, neural network acts as a classifier. If we put an independent sample (not present in the training data set) to the neural network input the output value will be between 0 and 1. This value can be treated as a posterior probability of normal/abnormal user behaviour: higher values correspond to normal actions while lower values correspond to potential anomalous patterns. In such a way, we propose to use the output of the SMUB model in order to estimate the reputation of a user in the VO. The output of the neural network could be treated as a "user reputation" in that sense how current user actions refer to the actions performed in the past. Such a user model will be specific to the VO depending on its goals and types of jobs being executed. For example, VO can be oriented on applications that require execution of large number of jobs with relatively small amount of data to be processed by a single job. In such a VO, jobs that consume almost full amount of RAM and virtual memory on resource would be considered as an anomalous pattern. In turn, other VOs can be oriented on applications where a single job consists of a number of elementary jobs each processing large amount of data. Such an example includes Earth science domain and satellite data processing [2, 4, 5, 26, 41].

The advantages of such a model are the ability to detect deviations from user behavioural patterns, to discriminate between different users, and to incorporate known patterns into the neural network training process. The disadvantages of the model in our case are rather high miss rate and the need to retrain the model over the course of time because user behaviour changes. Though the proposed user reputation model has both advantages and disadvantages, it is to be noted that this model should be used with other mechanisms to allow for more efficient assessment of user reputation.

Let us provide a formal description of the reputation model for user based on the SMUB model as it was done for the reputation model for resource provider in previous subsection.

We define Event for the user by

$$Event = \{t, u, r, vo, \mathbf{x}\} \tag{26}$$

where $\mathbf{x} = (S, ET, CPU, WT, CW, ES, CT, STD, RAM, VM, VO, RB)$.

Trace corresponds to the sequence of events

$$Trace = \bigcup_p Event_p = \bigcup_p \{t, u, r, vo, \mathbf{x}\}_p. \tag{27}$$

An analogy of *utility*() function (as it was defined for resource providers) is defined in the following way (Equation (27))

$$utility : Event \to \mathbf{R}, \tag{28}$$

$$utility(\{t, u, r, vo, \mathbf{x}\}) = SMUB_{(u,vo)}(\mathbf{x}), \tag{29}$$

where $SMUB_{(u,vo)}(\mathbf{x})$ is an output of the SMUB model.

It is worth noting that, in general case, under *utility*() function for users we can use other user behaviour models (e.g. [42, 43]) or a combination of different models to capture different aspects of user behaviour.

In our case, the SMUB transformation is performed by a neural network, and the model is specific to the user and VO. Let us denote a set of traces that are used to estimate the reputation of user u in vo up to the current time t with

$$Trace|_{(vo,u,t)} = \{\{t', u', r', vo', \mathbf{x}'\} \in Trace : u = u', vo = vo\_id', t' \leq t\}. \tag{30}$$

Let us denote a set of *utility*() function values derived from traces $Trace|_{(vo,u,t)}$ with (Equations (29)–(30))

$$O_{(vo,u,t)} = \{z(t, t_c) \cdot utility(\{t, u, r, vo, \mathbf{x}\}) | \{t, u, r, vo, \mathbf{x}\} \in Trace|_{(vo,u,t)}\}. \tag{31}$$

Reputation is expectation of *utility*() function

$$rep(vo, u, t) = E[utility(O_{(vo,u,t)})] = \int utility(O_{(vo,u,t)}) p_{utility}(O_{(vo,u,t)}) dO_{(vo,u,t).} \tag{32}$$

In order to approximate expectation we can use a sample mean (Equation (31))

$$rep(vo, u, t) = \frac{1}{|O_{(vo,u,t)}|} \sum_{x \in O_{(vo,u,t)}} x. \tag{33}$$

The reputation of an organisation in VO (from users' perspective) is the aggregation of the reputation of all users that participate in VO (Equations (6), (33)):

$$rep(\nu o, t) = \frac{1}{|g_{vo}^{-1}(o)|} \sum_{r \in g_{vo}^{-1}(o)} rep(vo, u, t). \tag{34}$$

The reputation of a user in all VOs can be estimated as follows (Equations (9), (33))

$$rep(r, t) = \frac{1}{|VO|_u|} \sum_{vo \in VO|_u} rep(vo, u, t). \tag{35}$$

4 INCORPORATING REPUTATION INTO GRID SCHEDULER

We consider a problem of on-line job scheduling in which jobs in the grid are scheduled immediately (unlike [17] where off-line job scheduling is considered). Here, we propose a novel approach on how reputation can be integrated into the scheduling algorithm using a non-linear trade-off scheme [31].

Let y be a criterion associated with the scheduler, i.e. criterion which is minimized to map jobs onto resources of a grid system. These could be job earliest completion time, fair execution time [44], a number of jobs in resource queue or a failure rate. Let $rep(r_i)$ denote reputation value of resource r_i. When incorporating a reputation into a job scheduler the following multi-criteria optimization problem arises: we want to minimize $y(r_i)$ value associated with the scheduler while running a job on a resource with maximum reputation $rep(r_i)$. Therefore, it is necessary to form a composition of these partial criteria to provide a single objective function. For this purpose, we propose to utilize a non-linear trade-off scheme presented in [31].

In such a scheme, normalized partial criteria y_k are integrated using the following equation:

$$Y(x) = \sum_{k=1}^{s} \alpha_k [1 - y_k(x)]^{-1}; \alpha_k \geq 0, \sum_{k=1}^{s} \alpha_k = 1, \tag{36}$$

where α_k are parameters having two-fold meaning [31]: weighted coefficients that provide weights for partial criteria, and regression coefficients of the regression utility function that is built using the concept of non-linear trade-off scheme. The advantage of using an integrated function in Equation (36) is that it provides a Pareto-optimal solution partially satisfying criteria with corresponding weights.

Thus, reputation can be incorporated into the scheduler in the following way:

$$Y(r_i) = \frac{\alpha_1}{1 - y_n(r_i)} + \frac{\alpha_2}{rep(r_i)}, \tag{37}$$

where $y_n(r_i)$ is the normalized value of the criterion associated with the scheduler.

The job is assigned to the resource which minimizes the value given in Equation (37).

$$r^* = \operatorname*{argmin}_{r_i} Y(r_i). \tag{38}$$

In our experiments we used the earliest completion time heuristic to assign jobs onto resource of the grid system. Let $ECT(r_i)$ be the estimated completion time of running a job on resource r_i, and $ECT_n(r_i)$ is the corresponding normalized value:

$$ECT_n = \frac{ECT(r_i)}{ECT_{max}}, \tag{39}$$

where ECT_{max} is the upper bound value for the ECT value. The *ECT-reputation* scheduler assigns job to a resource that minimizes the following expression:

$$r^* = \operatorname*{argmin}_{r_i} \left[\frac{\alpha_1}{1 - y_n(r_i)} + \frac{\alpha_2}{rep(r_i)} \right]. \tag{40}$$

The proposed approach was compared to a multiplication scheme in which the job is allocated to the resource that minimizes the following expression:

$$r^* = \underset{r_i}{\operatorname{argmin}}[ECT(r_i)(1 - rep(r_i))]. \qquad (41)$$

To incorporate user reputation into the scheduler several approaches can be applied. The first one is that the resource uses a predefined threshold so the users with reputation lower than this threshold would not be able to execute jobs on this particular resource. In such a case, there should be some mechanisms established in the grid environment so such users would be able to execute jobs, for example through purchasing either reputation at VO or processor time at a particular resource. Another approach consists in giving low priority to the jobs submitted by users with low reputation, or submitting jobs of users with low reputation only to the resources with low reputation. These approaches and efficiency of using user reputation in scheduling jobs onto grid resources will be investigated in the future works.

5 RESULTS OF EXPERIMENTS

In this section, results of experiments are presented to assess the performance of the described model. The performance is evaluated in terms of improving resource management in the grid.

5.1 Data Description

In order to generate workload within experiments, i.e. jobs inter-arrival time and jobs execution time, we used data traces provided by the Grid Observatory project[2]. This project provides data on job cycle in the EGEE grid infrastructure. In particular, we used data collected by the Real Time Monitor (RTM) system that summarizes various information on jobs executed in the grid. In total, the trace registers 37 attributes categorized into Information, Timestamps and Metrics [45].

5.2 Experimental Parameters

All experiments were run for a grid infrastructure of 20 resources with resource productivity (in unitless standard units) being uniformly selected from the range $[1, 200]$. Job complexity (also, in unitless standard units) was generated from traces provided by the Grid Observatory project lying in the range $[1, 56\,000]$. Distribution of job complexity is shown in Figure 1. Job execution time on a resource was estimated as *jobComplexity/resourceProductivity*. Jobs inter-arrival time and

[2] Grid Observatory: `www.grid-observatory.org`

workload were also generated from the EGEE traces. Figures 2 and 3 show cumulative number of submitted jobs over the time and job arrival rate (in jobs/min), respectively.

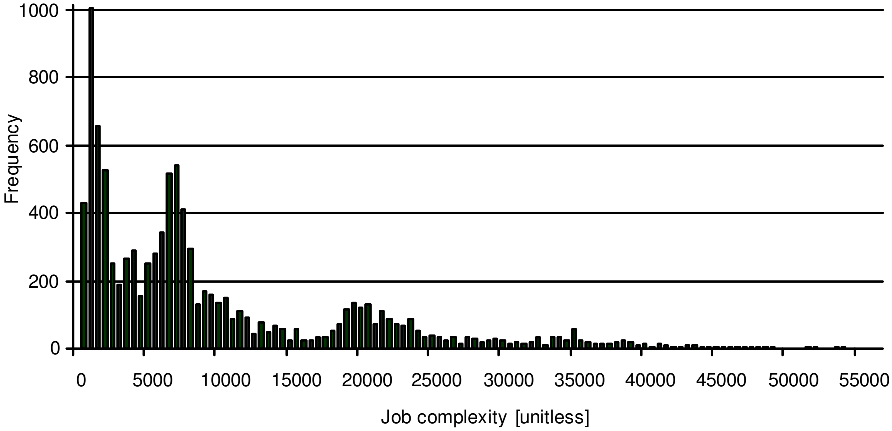


Figure 1. Distribution of job complexity within experiments (for 10 000 jobs)

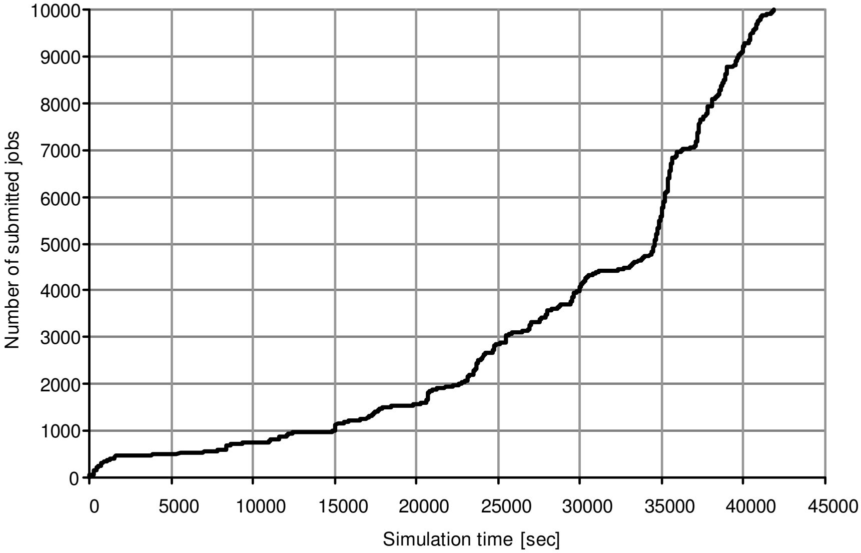


Figure 2. Cumulative number of submitted jobs within experiments

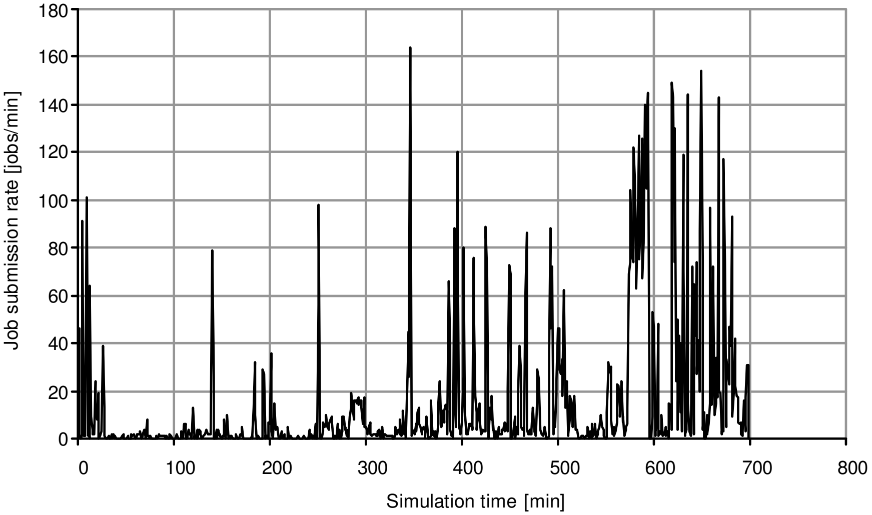


Figure 3. Job arrival rate within experiments

Within the experiments the following QoS metrics were considered: job waiting time, job execution time and job total completion time. The agreed SLA values were modelled as follows: the agreed waiting time was selected randomly from the range $[1, 30\,000]$ sec, and the agreed execution time was selected as *jobComplexityminResourceProductivity*. In order to model a scenario when a resource did not respect the agreed execution time the following approach was used: a random value from the interval $[1, 2\,500]$ sec was added to the actual execution time value. The penalty function and reputation were estimated using Equations (18) and (22), respectively. If not stated otherwise, the utility function (Equation (17)) was calculated for the job completion time QoS metric.

5.3 Performance Analysis

In all our simulations jobs are scheduled immediately after arrival. The following schedulers were compared in simulations:

- A heuristic on-line scheduler which maps a job to a resource which provides the job earliest completion time (ECT).
- A reputation-based scheduler with a non-linear trade-off scheme (Equation (40)). For this scheduler, we used the following values for parameters: $\alpha_1 = \alpha_2 = 0.5$.
- A reputation-based scheduler with a multiplication scheme (Equation (41)).

The following performance metrics were used to evaluate scheduling algorithms with/without incorporating information on the resource reputation:

Makespan: the difference between the finish time of the last job and the release time (i.e. the arrival time in the grid system) of the first job.

Average Job Execution Time: the average time a job being executed (job execution time is the difference between finish time and start time of the job on resource).

Average Job Queue Waiting Time: the average job waiting time in the queue of the resource (job waiting time is the difference between the start time and the release time of the job).

Average Job Excess Time: the average time by which a job misses the agreed SLA (job excess time is the difference between job finish time and agreed SLA).

SLA missed: the number of cases when an agreed SLA was missed on job execution.

Average utility.

Resource utilization: the number of jobs completed.

Two sets of experiments were run varying different parameters, in particular workload (number of jobs) and resource trustworthiness.

For the first set of experiments, performance of schedulers was compared within simulations that were run for different number of jobs that corresponded to different job arrival rates (Table 1).

Number of jobs in simulation	Arrival rate [jobs/min]
1 000	4
2 000	5
3 000	7
4 000	8
5 000	9
6 000	10
7 000	11
8 000	12
9 000	13
10 000	14

Table 1. Number of jobs and jobs arrival rate used in simulation to compare schedulers

We allowed 20 % of the resources always to be untrustworthy, i.e. the agreed SLA is always violated by such resource providers. Initial reputation for these resources was set to 0.1. Figures 4–10 show performance metrics for the schedulers depending on system workload.

Figures 4–10 show that the *ECT-reputation* scheduler using a non-linear trade-off scheme outperformed the *ECT-reputation* scheduler using a multiplication scheme and a scheduler without knowledge of reputation for all metrics. Table 2 shows average improvement for the metrics used in the study. The use of a non-linear

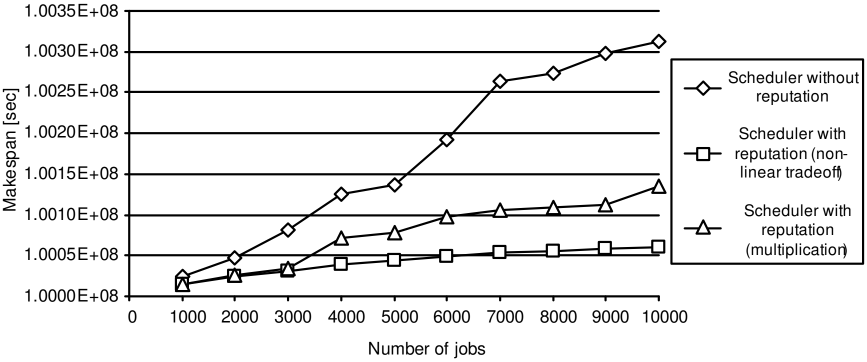


Figure 4. Makespan for *ECT* and *ECT-reputation* schedulers depending on system workload

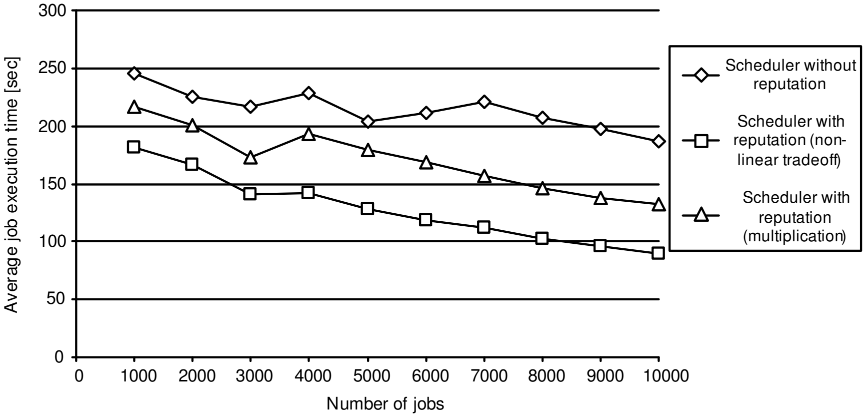


Figure 5. Average job execution time for *ECT* and *ECT-reputation* schedulers depending on system workload

trade-off scheme for incorporating reputation allowed us to improve the scheduler more than 2 times comparing to a multiplication scheme.

It should be, however, noted that when incorporating reputation into a specific scheduler, the reputation should be estimated using QoS metrics related to the scheduler. Otherwise, there could be no improvements of using reputation-based scheduler for some of the metrics.

Within the second set of experiments, we varied resource trustworthiness. We allowed 20 % of the resources to be untrustworthy but with different degree of trustworthiness (we call it *trustworthiness rate*). For example, if resource trustworthiness rate is equal to 0.6 then it meets the agreed SLA on average in 60 % of cases. The following approach was used to simulate such scenarios: when un-

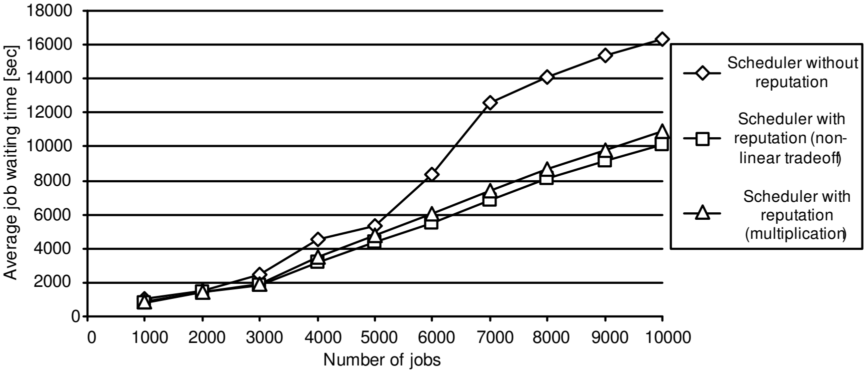


Figure 6. Average job queue waiting time for *ECT* and *ECT-reputation* schedulers depending on system workload

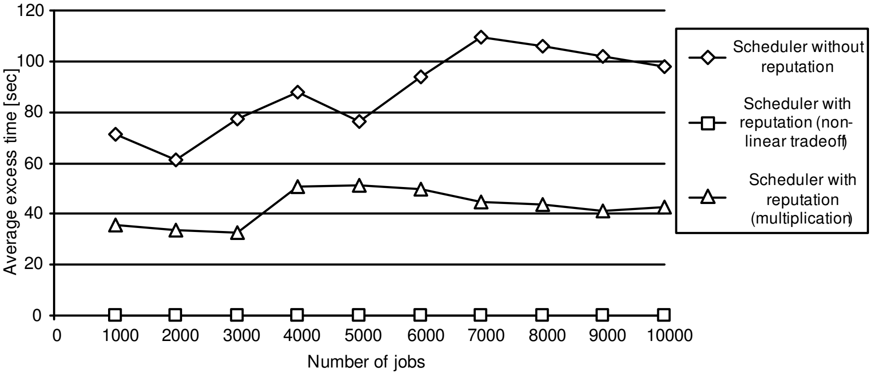


Figure 7. Average job excess time for *ECT* and *ECT-reputation* schedulers depending on system workload

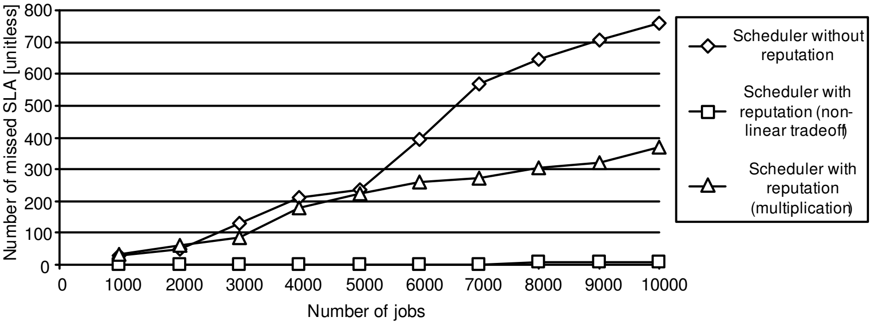


Figure 8. The number of SLA missed for *ECT* and *ECT-reputation* schedulers depending on system workload

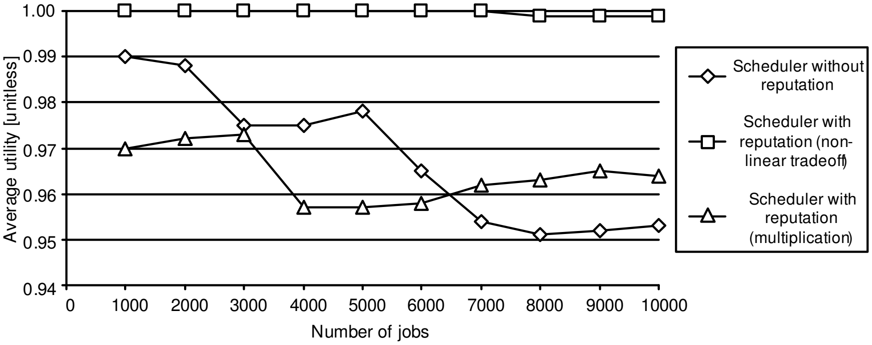
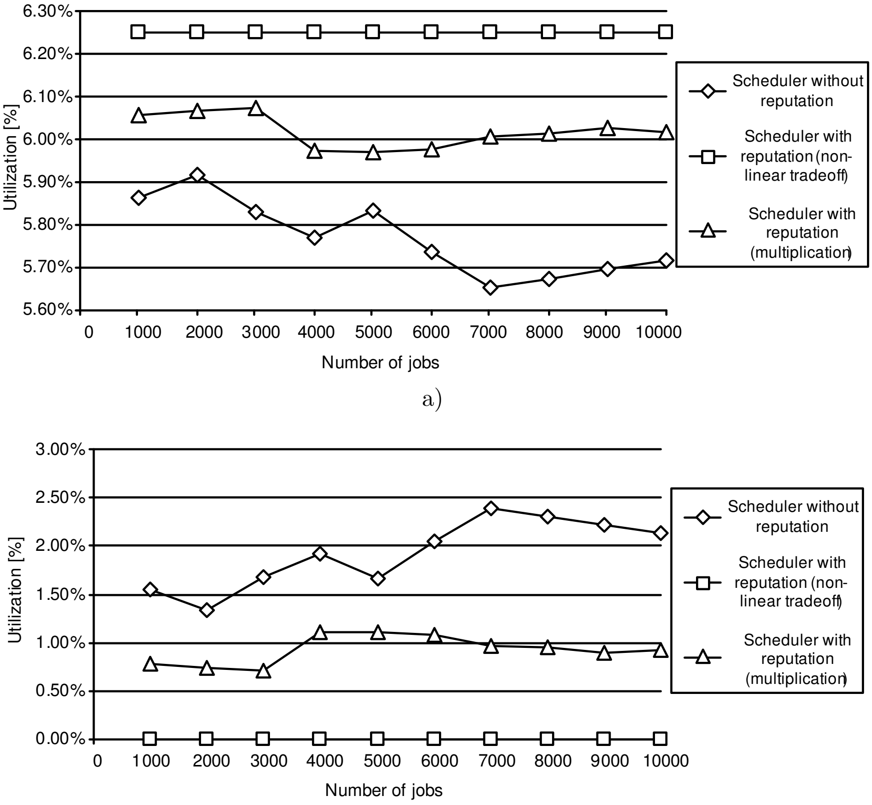


Figure 9. Average utility for *ECT* and *ECT-reputation* schedulers depending on system workload



a)

b)

Figure 10. Average resource utilization (a) trustworthy resources; b) untrustworthy resources) for *ECT* and *ECT-reputation* schedulers depending on system workload

Performance metric	Average improvement [%]	
	Non-linear trade-off scheme	Multiplication scheme
Average exec time	40.9	20.9
Average wait time	30.5	26.1
Average excess time	100.0	50.9
Missed SLA	99.6	26.9
Average utility	3.2	0.4
Makespan	0.1	0.1
Average	45.7	20.9

Table 2. Average improvement of incorporating reputation into grid scheduler

trustworthy resource was scheduled to execute a job, a random value uniformly distributed in the $[0; 1]$ range was generated. If this random value was less than resource trustworthiness rate, then the resource met the agreed SLA. Otherwise, the agreed SLA is violated by the resource provider. Figures 11–17 show performance metrics for the schedulers depending on trustworthiness rate of untrustworthy resources.

Figures 11–17 show that the *ECT-reputation* scheduler using a non-linear trade-off had a better performance comparing to other schedulers. This scheduler provided a better resource management in terms of scheduling jobs to untrustworthy resources: no jobs were scheduled until resource reputation became higher (in our cases until resource reputation was 0.5, see Figure 17 b)). Moreover, *ECT-reputation* scheduler using a non-linear trade-off can be adjusted to schedule jobs on untrustworthy resources via α_1 and α_2 parameters (Equation (40)).

6 CONCLUSIONS

In this paper we extended the existing utility-based reputation model [17] by incorporating a statistical model of user behaviour (SMUB) and several components such as assigning initial reputation to a new entity in VO, capturing alliance between consumer and resource, time decay function, and score function. The inclusion of the SMUB model provides robustness and dynamics to the user reputation model comparing to the policy-based user reputation model in terms of adapting to user actions. Though, other IDS models and techniques should be further evaluated to provide a better performance.

The proposed model was evaluated in terms of resource management in grids. A problem of on-line job scheduling was considered, and a novel approach for integrating reputation into the job scheduling algorithm using a non-linear trade-off scheme was presented. The advantage of using such a scheme is that it provides a Pareto-optimal solution partially satisfying criteria with corresponding weights. The results of experiments showed that the scheduler with knowledge of reputation using a non-linear trade-off scheme outperformed a scheduler without knowledge of reputation on average 45 % for all performance metrics used in the study. A non-

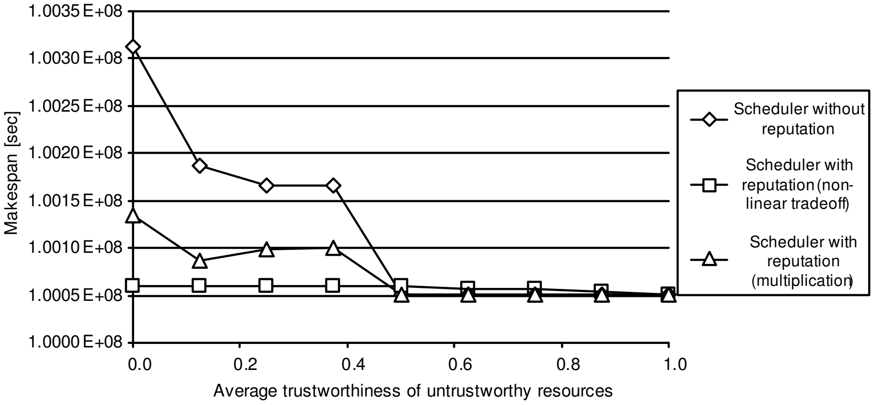


Figure 11. Makespan for *ECT* and *ECT-reputation* schedulers depending on system workload

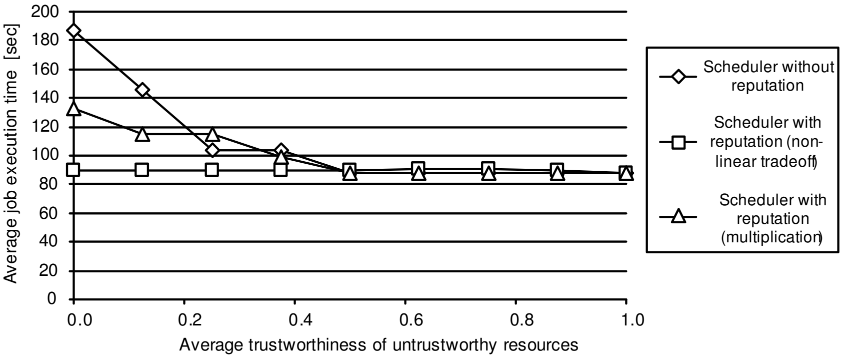


Figure 12. Average job execution time for *ECT* and *ECT-reputation* schedulers depending on resource trustworthiness rate

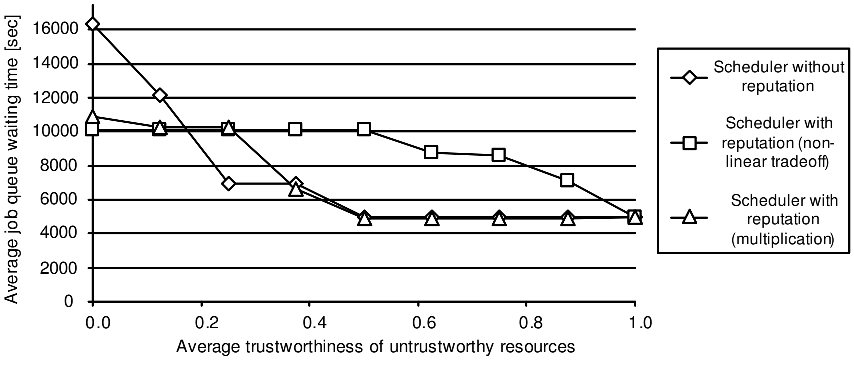


Figure 13. Average job queue waiting time for *ECT* and *ECT-reputation* schedulers depending on resource trustworthiness rate

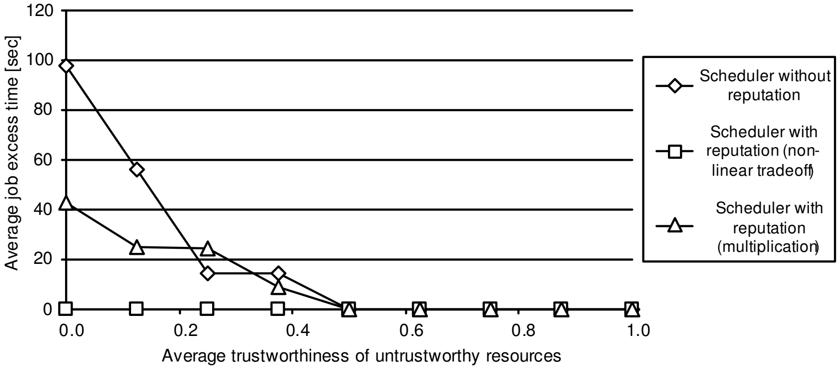


Figure 14. Average job excess time for *ECT* and *ECT-reputation* schedulers depending on resource trustworthiness rate

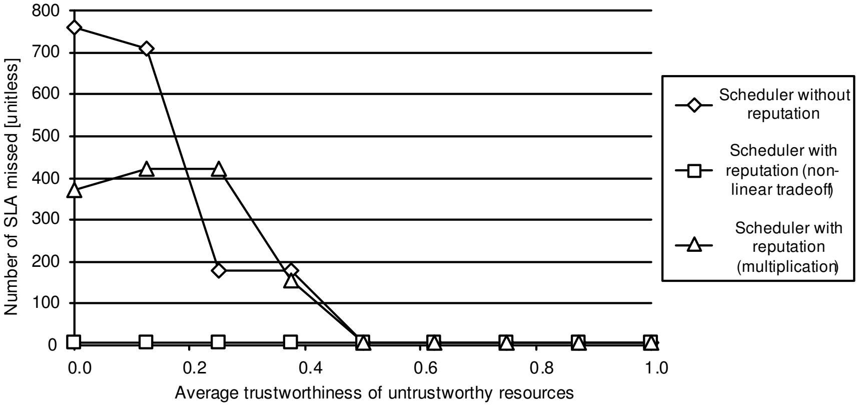


Figure 15. Number of SLA missed for *ECT* and *ECT-reputation* schedulers depending on resource trustworthiness rate

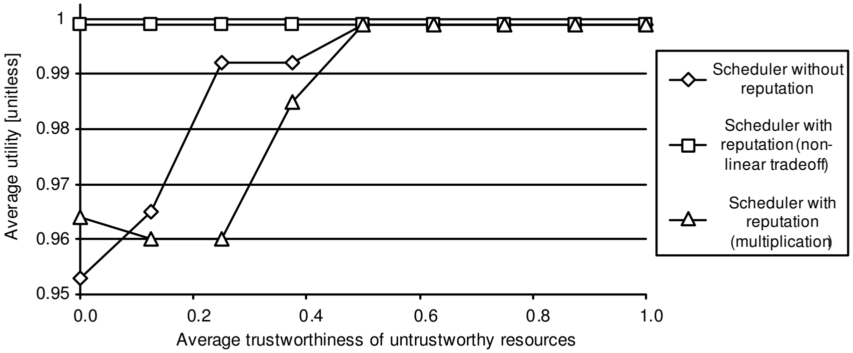
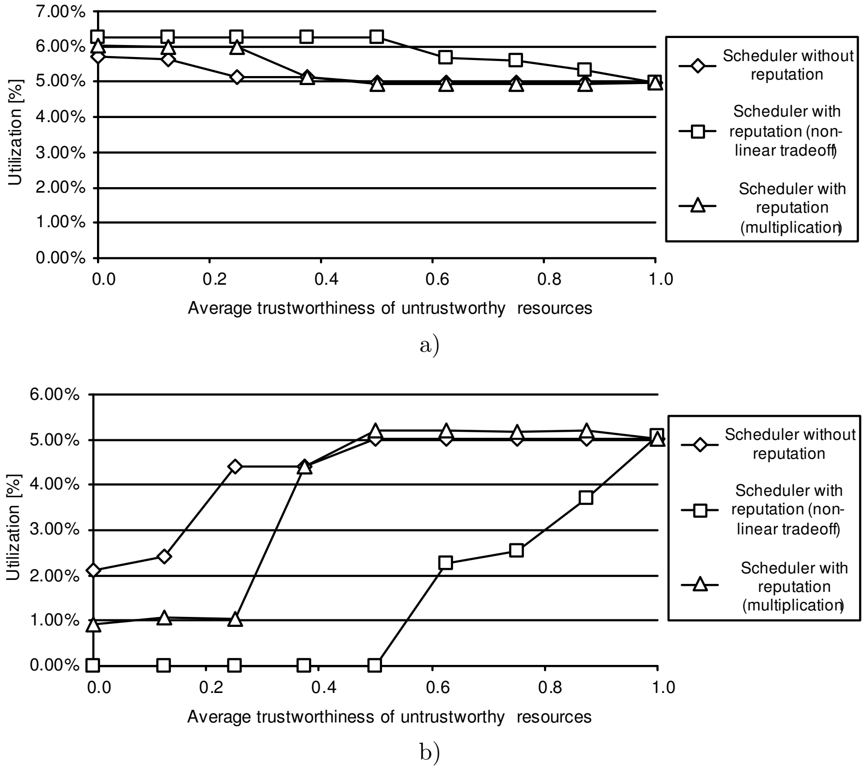


Figure 16. Average utility for *ECT* and *ECT-reputation* schedulers depending on resource trustworthiness rate



a)

b)

Figure 17. Average resource utilization (a) trustworthy resources; b) untrustworthy resources) for *ECT* and *ECT-reputation* schedulers depending on resource trustworthiness rate

linear linear trade-off scheme for incorporating reputation into the scheduler was also compared to a multiplication scheme, and showed better performance with an average factor of 2. Also, this scheduler provided a better resource management in terms of scheduling jobs to untrustworthy resources: it allows no jobs to be scheduled to untrustworthy resources until their reputation will be higher than a certain threshold value.

There are several directions for future work: incorporating a computing cost criterion into the scheduler; modelling a failure rate using Grid-Observatory data to provide a generative model for simulations; exploring applications of the model for other large-scale service-oriented systems such as the Global Earth Observation System of Systems (GEOSS).

REFERENCES

[1] Foster, I.—Kesselman, C.—Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, Vol. 15, 2001, No. 3, pp. 200–222.

[2] Kussul, N.—Mandl, D.—Moe, K.—Mund, J. P.—Post, J.—Shelestov, A.—Skakun, S.—Szarzynski, J.—Van Langenhove, G.—Handy M.: Interoperable Infrastructure for Flood Monitoring: SensorWeb, Grid and Cloud. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 5, 2012, No. 6, pp. 1740–1745.

[3] Kussul, N.—Shelestov, A.—Skakun, S.—Li, G.—Kussul O.: The Wide Area Grid Testbed for Flood Monitoring Using Earth Observation Data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 5, 2012, No. 6, pp. 1746–1751.

[4] Lecca, G.—Petitdidier, M.—Hluchý, L.—Ivanovic, M.—Kussul, N.—Ray, N.—Thieron, V.: Grid Computing Technology for Hydrological Applications. Journal of Hydrology, Vol. 403, 2011, No. 1–2, pp. 186–199.

[5] Kussul, N.—Shelestov, A.—Skakun, S.: Grid and Sensor web Technologies for Environmental Monitoring. Earth Science Informatics, Vol. 2, 2009, No. 1–2, pp. 37–51.

[6] Bouras, C.—Giannaka, E.—Tsiatos, E.: E-Collaboration Concepts, Systems and Applications. In: Kock, N. (Ed.): Information Science Reference, E-Collaboration: Concepts, Methodologies, Tools, and Applications, Vol. 1, 2009, Section I, Chapter 1.2. IGI Global.

[7] Eymann, T.—König, S.—Matros, R.: A Framework for Trust and Reputation in Grid Environments. Journal of Grid Computing, Vol. 6, 2008, No. 3, pp. 225–237.

[8] Song, S.—Hwang, K.—Kwok, Y.-K.: Trusted Grid Computing with Security Binding and Trust Integration. Journal of Grid Computing, Vol. 3, 2005, No. 1–2, pp. 53–73.

[9] Chakrabarti, A.: Grid Computing Security. Springer-Verlag, 2007.

[10] Silaghi, G.—Arenas, A.—Silva, L.: A Utility-Based Reputation Model for Service-Oriented Computing. In: Priol, T. and Vanneschi, M. (Eds.): Toward Next Generation Grids, Springer, 2007, pp. 63–72.

[11] Zetuny, Y.—Terstyanszky, G.—Winter, S.—Kacsuk, P.: Reputation-Policy Trust Model for Grid Resource Selection. In: P. Kacsuk, R. Lovas and Z. Nemeth (Eds): Distributed and Parallel Systems 2008, pp. 195–206.

[12] Grandison, T.—Sloman, M.: A Survey of Trust in Internet Applications. IEEE Communication Surveys and Tutorials, Vol. 4, 2000, No. 4, pp. 2–16.

[13] McKnight, D. H.—Chervany, N. L.: The Meaning of Trust. Technical Report, MISRC Working Paper Series 96-04, University of Minnesota Management Information Systems Research Center, 1996.

[14] Gambetta, D.: Can We Trust Trust? In: Gambetta, D. (Ed.): Trust: Making and Breaking Cooperative Relations. Department of Sociology, University of Oxford, 1988.

[15] Josang, A.—Ismail, R.—Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems, Vol. 43, 2007, No. 2, pp. 618–644.

[16] Abdul-Rahman, A.—Hailes, S.: Supporting Trust in Virtual Communities. In: Proceedings of the IEEE 33$^{\mathrm{rd}}$ Hawaii International Conference on System Sciences (HICSS '00), Vol. 6, 2000, pp. 6007.

[17] Arenas, A.—Aziz, B.—Silaghi, G. C.: Reputation Management in Grid-Based Virtual Organisations. In: Fernandez Medina, E., Malek, M. and Hernando, J. (Eds.): Proceedings of International Conference on Security and Cryptography (SECRYPT 2008), 2008, pp. 538–545.

[18] Arenas, A. E.—Aziz, B.—Silaghi, G. C.: Reputation Management in Collaborative Computing Systems. Security and Communication Networks, Vol. 3, 2010, No. 6, pp. 546–564.

[19] Kussul, O.—Kussul, N.—Skakun, S.: Assessing Security Threat Scenarios for Utility-Based Reputation Model in Grids. Computers and Security, Vol. 34, 2013, pp. 1–15.

[20] Skakun, S.—Kussul, N.: An Agent Approach for Providing Security in Distributed Systems. In: Proceedings of International Conference Modern Problems of Radio Engineering Telecommunications and Computer Science (TCSET 2006), Slavsko, Lviv, Ukraine, 2006, pp. 212–215.

[21] Kussul, N.—Skakun, S.: Intelligent System for Users' Activity Monitoring in Computer Networks. In: IEEE Intelligent Data Acquisition and Advanced Computing Systems (IDAACS 2005): Technology and Applications, 2005, pp. 306–309.

[22] Skakun, S. V.—Kussul, N. N.—Lobunets, A. G.: Implementation of the Neural Network Model of Users of Computer Systems on the Basis of Agent Technology. Journal of Automation and Information Sciences, Vol. 37, 2005, No. 4, pp. 11–18.

[23] Kussul, N.—Skakun, S.: Neural Network Approach for User Activity Monitoring in Computer Networks. In: Proceedings of the International Joint Conference on Neural Networks, Budapest (Hungary), Vol. 2, 2004, pp. 1557–1562.

[24] Azzedin, F.—Maheswaran, M.: Integrating Trust into Grid Resource Management Systems. In: 2002 International Conference on Parallel Processing (ICPP 2002), IEEE Computer Society, Washington DC, USA, 2002, pp. 47–54.

[25] Gomez Marmol, F.—Martinez Perez, G.: Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. Computers and Security, Vol. 28, 2009, pp. 545–556.

[26] Kussul, N.—Shelestov, A.—Skakun, S.—Kravchenko, O.—Gripich, Y.—Hluchý, L.—Kopp, P.—Lupian E.: The Data Fusion Grid Infrastructure: Project Objectives and Achievements. Computing and Informatics, Vol. 29, 2010, No. 2, pp. 319–334.

[27] Skakun, S.: A Neural Network Approach to Flood Mapping Using Satellite Imagery. Computing and Informatics. Vol. 29, 2010, No. 6, pp. 1013–1024.

[28] Kussul, N.—Sokolov, B.,—Zyelyk, Y.—Zelentsov, V.—Skakun, S.—Shelestov, A.: Disaster Risk Assessment Based on Heterogeneous Geospatial Information. Journal of Automation and Information Sciences, Vol. 42, 2010, No. 12, pp. 32–45.

[29] Gallego, J.—Kravchenko, A.—Kussul, N.—Skakun, S.—Shelestov, A.—Grypych, Y.: Efficiency Assessment of Different Approaches to Crop Classification Based on Satellite and Ground Observations. Journal of Automation and Information Sciences, Vol. 44, 2012, No. 5, pp. 67–80.

[30] Shelestov, A. Yu.—Kravchenko, A. N.—Skakun, S. V.—Voloshin, S. V.—Kussul, N. N.: Geospatial Information System for Agricultural Monitoring. Cybernetics and Systems Analysis, Vol. 49, 2013, No. 1, pp. 124–132.

[31] Voronin, A. N.: A Multicriteria Problem of Distribution of Bounded Resources. Cybernetics and System Analysis, Vol. 47, 2011, No. 3, pp. 490–493.

[32] Von Laszewski, G.—Alunkal, B.—Veljkovic, I.: Towards Reputable Grids. Scalable Computing: Practice and Experience, Vol. 6, 2005, No. 3, pp. 95–106.

[33] Kamvar, S.—Schlosser, M.—Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. In: Proceedings of the 12th International Conference on World Wide Web, ACM Press, New York, NY, USA 2003, pp. 640–651.

[34] Kerschbaum, F.—Haller, J.—Karabulut, Y.—Robinson, P.: PathTrust: A Trust-Based Reputation Service for Virtual Organization Formation. Lecture Notes in Computer Science, Vol. 3986, 2006, pp. 193–205.

[35] Papaioannou, T. G.—Stamoulis, G. D.: Reputation-Based Estimation of Individual Performance in Grids. In: Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID), IEEE Computer Society Washington, DC, USA, 2008, pp. 500–509.

[36] Liang, Z.—Shi, W.: A Reputation-Driven Scheduler for Autonomic and Sustainable Resource Sharing in Grid Computing. Journal of Parallel and Distributed Computing, Vol. 70, 2010, pp. 111–125.

[37] Wu, C.-C.—Sun, R.-Y.: An Integrated Security-Aware Job Scheduling Strategy for Large-Scale Computational Grids. Future Generation Computer Systems, Vol. 26, 2010, pp. 198–206.

[38] Vieira, K.—Schulter, A.—Westphall, C.—Westphall, C.: Intrusion Detection for Grid and Cloud Computing. IT Professional, Vol. 12, 2007, No. 4, pp. 38–43.

[39] Schulter, A.—Vieira, K.—Westphall, C.—Abderrahim, S.: Intrusion Detection for Computational Grids. In: Proceedings of the 2nd International Conference New Technologies, Mobility, and Security, IEEE Press 2008, pp. 1–5.

[40] Haykin, S.: Neural Networks: A Comprehensive Foundation. Upper Saddle River, New Jersey, Prentice Hall 1999.

[41] Shelestov, A.—Kussul, N.—Skakun, S.: Grid Technologies in Monitoring Systems Based on Satellite Data. Journal of Automation and Information Sciences, Vol. 38, 2006, No. 3, pp. 69–80.

[42] Oh, S. H.—Lee, W. S.: An Anomaly Intrusion Detection Method by Clustering Normal User Behavior. Computers and Security, Vol. 22, 2003, No. 7, pp. 596–612.

[43] Sun, H.-W.—Lam, K.-Y.—Chung, S.-L.—Gu, M.—Sun, J.-G.: Anomaly Detection in Grid Computing Based on Vector Quantization. Lecture Notes in Computer Science, Vol. 3251, 2004, pp. 883–886.

[44] Kretsis, A.—Kokkinos, P.—Varvarigos, E.: Developing Scheduling Policies in gLite Middleware. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society Washington, DC, USA, 2009, pp. 20–27.

[45] Germain-Renaud, C.—Cady, A.—Nauroy, J.: Grid Observatory Technical Documentation, version 1.3. Available on: `http://grid-observatory.org/fileadmin/documents/GOTechnicalDocV1.3.pdf`, 2011.

Olga Kussul is Assistant Professor at the Physics and Technology Institute of the National Technical University of Ukraine "Kiev Polytechnic Institute". She received Ph. D. degree in information technologies from the National Technical University of Ukraine "Kiev Polytechnic Institute" in 2013, M. Sc. in Information Security from the National Technical University of Ukraine "Kiev Polytechnic Institute" in 2009. Her research interests include trust management in distributed and heterogeneous systems.

Nataliia Kussul is a Deputy Director and Head of Department of Space Information Technologies and Systems at the Space Research Institute of NASU-SSAU, and Professor at the National Technical University of Ukraine "Kiev Polytechnic Institute". She received her Doctor of Sciences (second scientific degree) in applied mathematics from Space Research Institute NASU-SSAU in 2001, Ph. D. degree in applied mathematics from the Institute of Cybernetics of NASU in 1991, M. Sc. degree with honours in mathematics from Kiev State University. Her current research interests include grid computing, sensor web, intelligent computations, pattern recognition, remote sensing.

Sergii Skakun is the Head of Laboratory for Satellite Monitoring at the Space Research Institute of NASU-SSAU, and Associate Professor at the National Technical University of Ukraine "Kiev Polytechnic Institute". He his received Ph. D. degree in system analysis and theory of optimal solutions from the Space Research Institute of NASU-SSAU in 2005, M. Sc. degree with honours in applied mathematics from the Physics and Technology Institute of the National Technical University of Ukraine "Kiev Polytechnic Institute" in 2004. His research interests include grid, sensor web, Earth observation, data processing, disaster risk management.