

## МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ ПЛАНИРОВАНИЯ ВЫПОЛНЕНИЯ ЗАДАЧ В СТРУКТУРНО-СЛОЖНЫХ СИСТЕМАХ НА ОСНОВЕ МОДЕЛЕЙ РЕПУТАЦИИ

В данной работе решается задача многокритериальной оптимизации планирования выполнения задач в структурно-сложных системах на основе моделей репутации. Предложен новый подход для интеграции модели репутации в подсистему планирования с использованием нелинейной схемы компромиссов. Полученные результаты экспериментов демонстрируют эффективность разработанного подхода.

In this paper we consider a problem of multi-criteria optimization of task scheduling in structural-complex systems using reputation models. We propose a new approach for integrating reputation into scheduler by applying a non-linear tradeoff scheme. Results of experiments are presented which show the effectiveness of the proposed approach.

### 1. Введение

В последние годы бурное развитие вычислительной техники и информационных технологий привело к широкому распространению сервис-ориентированных распределенных систем. Многие из них представляют собой инфраструктуру для совместного использования данных, алгоритмов, ресурсов хранения и вычислительных мощностей [1]. Примерами таких систем являются система систем GEOSS<sup>1</sup>, Grid-системы [2], системы на основе технологии Sensor Web [3], системы электронного взаимодействия (e-collaboration) [4]. Основу работы таких систем зачастую составляют виртуальные организации (ВО) – временные или постоянные объединения географически распределенных отдельных пользователей, групп, подразделений организаций или целых организаций, которые совместно используют ресурсы и информацию для достижения общих целей [5]. Виртуальные организации динамически формируются, существуют некоторое время и распадаются.

В последнее время особое внимание уделяется управлению доверием в сервисориентированных структурно-сложных системах [6, 7, 8, 9]. В работе [10] показано, что доверие – это высокоэффективная технология, и ее внедрение позволит обезопасить электронные транзакции. При этом доверие описывается как важное и сложное понятие, связанное с честностью, правдивостью и надежностью доверенного лица или сервиса. Тем не менее, единого фор-

мального определения понятия доверия не существует [11].

Системы управления доверием можно условно разделить на две группы [8]: основанные на использовании политик безопасности и основанные на вычислении репутации. В системах управления доверием основанных на использовании политик безопасности отношения доверия между субъектами основываются на конкретных, заранее оговоренных, политиках безопасности. Такие системы являются излишне жесткими и не адаптивными. В системах управления доверием, основанных на вычислении репутации, вводятся определенные формализмы для оценки доверия, которое вычисляется как функция от репутации. Репутация – это предположение о поведении агента на основе имеющейся информации или наблюдений о его поведении в прошлом [12].

Среди известных моделей репутации следует выделить известную модель, описанную в [13]. Недостатком этой модели является непродуманность инициализации модели для нового ресурса или пользователя и неустойчивость к атакам. В работах [14, 15] эта модель была модифицирована путем введения методики активного эксперимента для определения начальной репутации ресурса (поставщика услуг), учета взаимосвязей между пользователями и провайдерами ресурсов во избежание группового сговора [16], оценивания репутации на интервале времени, а не в конкретный момент [9, 16], а также *учета сложности* разных типов сервисов [17]. В частности, успешное выполнение задач со сложным потоком выполнения (workflow) или распараллеленных программ (например, в задачах экологического мониторинга, таких как

<sup>1</sup> GEOSS, <http://www.earthobservations.org>

численный прогноз погоды [18]) обеспечивает поставщику ресурсов более высокую репутацию, чем выполнение более простой задачи. Кроме того, в работах [14, 15] было усовершенствовано определение функции полезности путем включения статистической модели поведения пользователя, разработанной ранее для компьютерных сетей и распределенных систем [19, 20, 21, 22].

Данная статья посвящена применению разработанной модели для планирования выполнения задач в структурно-сложных системах. В частности, будет показано, как использование модели репутации брокером ресурсов позволяет повысить эффективность управления ресурсами сервис-ориентированных систем. Задача планирования распределения задач между ресурсами формулируется как многокритериальная задача оптимизации. Модель репутации предлагается интегрировать в подсистему планирования с использованием нелинейной схемы компромиссов [23], что обеспечивает дополнительные преимущества по сравнению с традиционной мультипликативной схемой для формирования обобщенного критерия.

## 2. Анализ существующих подходов

На сегодняшний день предложено несколько концепций построения и использования репутации в сервис-ориентированных системах и интеграции репутации в планировщик задач. В данной статье описывается подход, основанный на модели репутаций, которая вычисляется на основе функции полезности и может применяться для оценки репутации как пользователей, так и поставщиков ресурсов [13]. Эта модель основана на вычислении функции полезности, которая выражает удовлетворенность одного объекта его взаимодействием с другими объектами, учитывая ключевые качества, свойственные оцениваемому объекту. При вычислении репутации пользователей оценивается, как и какие ресурсы используются пользователем и насколько его поведения соответствует заданной политике безопасности ВО. В свою очередь, репутация поставщика ресурсов оценивается на основе параметров качества обслуживания (QoS), которое он предоставляет. Для оценки эффективности предложенной модели авторами проведено имитационное моделирование для статического планировщика задач: для заданного набора задач, отправленных пользователями в систему, каждый ресурс обрабатывал

количество запросов, прямо пропорциональное его репутацию. В работе показано, что при использовании планировщика с учетом репутации общее время выполнения задач можно уменьшить на 25%.

В работе [6] особое внимание уделяется экономическим аспектам Grid-систем с учетом информационной асимметрии. Информационная асимметрия представляет собой ситуацию, когда поставщик ресурсов и пользователь получают разную информацию о параметрах качества обслуживания. Эти аспекты учитываются при разработке модели репутации с последующим коммерческим использованием Grid-систем. Вопросы информационной асимметрии в Grid детально рассмотрены в работе [24]. Предложен механизм на основе репутаций, позволяющий эффективно учитывать скрытую (недоступную) информацию.

В работе [7] предложена модель доверия на основе методов нечеткой логики. Разработанная модель агрегирует параметры безопасности распределенных ресурсов в Grid-системе путем их фазсификации. В работе использованы следующие критерии для оценки уровня доверия Grid-узла: репутация узла (процент успешно выполненных задач, кумулятивный уровень использования узла, среднее время простоя задач, среднее время выполнения задач) и уровень защищенности узла (набор программного и аппаратного обеспечения для организации безопасности узла). Для безопасного планирования большого количества автономных и унитарных задач по Grid-узлам разработана система Secure Grid Outsourcing (SeGO). Для динамического планировщика задач используется эвристический алгоритм Min-Min. Результаты экспериментов показали существенный прирост производительности при агрегации доверия в планировщик задач.

В работе [25] для управления ресурсами Grid-системы с учетом репутации предложен каркас HOURS. Система ориентирована на автоматический перезапуск задач, самозащищенность, совместное использование гетерогенных ресурсов, резервирование и установление гарантированного уровня качества обслуживания в Grid-вычислениях. Для уменьшения количества повторно отправляемых в систему задач и уровня невыполнения задач разработан планировщик ресурсов, учитывающий репутацию вычислительного узла. Эксперименты проводились на основе реальных данных, полученных в системе TeraGrid. Результаты моделирования

показали, что использование планировщика задач с репутацией может снизить уровень невыполнения задач с 3.82 до 0.70 относительно стандартного планировщика.

В статье [16] представлена модель доверия для Grid-систем, которая используется для планирования задач с учетом требуемых параметров безопасности. Для интеграции репутации в эвристические планировщики задач используется мультипликативная схема. Формально показано, что время обработки всех задач при использовании планировщика с учетом репутации всегда будет меньше времени выполнения при использовании планировщика без репутации (при условии, что базовая эвристика распределения задач в обоих случаях - одинакова).

В работе [26] разработан генетический алгоритм для планирования задач, который учитывает разнородность механизмов обеспечения отказоустойчивости в вычислительных Grid-системах. Риск запуска задачи на конкретном узле оценивается на основе выдвигаемых требований безопасности и уровнем доверия. Предложенный алгоритм обеспечивает меньшее время выполнения задач и меньшее количество невыполненных задач по сравнению с алгоритмами Min-Min и Sufferage.

Система PathTrust [27] – это система репутаций, предложенная для выбора членов ВО на этапе ее формирования. Для того, чтобы войти в ВО, организация должна зарегистрироваться с инфраструктурой сети предприятия (enterprise network) путем предоставления некоторых сертификатов. Помимо управления пользователем сеть предприятия предоставляет централизованный сервис репутаций. Когда ВО распадается, каждый член оставляет определенные значения обратной связи для сервера репутаций для других членов, с кем он вступал во взаимодействие. Эти значения обратной связи могут быть положительными или отрицательными. Данной схеме не хватает динамики, поскольку данные обратной связи агрегируются только в момент распада виртуальной организации.

Модель доверия GridEigenTrust [28] является расширением модели EigenTrust [29], разработанной ранее для систем P2P (peer-to-peer). В модели GridEigenTrust используется метод получения глобальной репутации с учетом иерархий и вычисления доверия на основе собственных чисел (eigenvalue-based trust calculation algorithm). Преимуществами данного подхода являются быстрая сходимость и экономность по сравнению с вычислением глобального зна-

чения доверия для отдельных субъектов для каждой ситуации. В случае, когда организация предоставит неправдивую информацию о своих субъектах, она будет оштрафована путем понижения глобального уровня доверия всей организации. В дальнейшем значение репутации интегрируется в систему управления обеспечением качества обслуживания (QoS management system), предоставляя возможность переопределить механизм выбора ресурсов и гарантированного уровня качества обслуживания (SLA).

### 3. Модель репутации на основе вычисления функции полезности в сервис-ориентированных системах

В качестве модели репутации будем исследовать модификацию известной модели [13], предложенную в [14, 15].

#### 3.1. Модель репутации для поставщиков ресурсов

Модель репутации поставщиков ресурсов, предложенная и детально описанная в [14, 15], основана на функции полезности, которая определяет уровень удовлетворенности пользователя предоставленным сервисом. Она содержит две составляющие: модель репутации поставщика ресурсов и модель репутации пользователей. Обе составляющие основываются на вычислении функции полезности, введенной в [14, 15], и определяющей уровень сервиса. Приведем формальное описание этих моделей.

Функция полезности имеет вид:

$$utility : Event \rightarrow \mathbf{R},$$

где пространство событий *Event* представляет собой декартово произведение времени, ресурсов, пользователей ВО и качества обслуживания:

$$Event = T \times \bigcup_l u_l \times \bigcup_k r_k \times \bigcup_m vo_m \times \{QoS\ name\} \times \mathbf{R} \quad (1)$$

где *T* – временной интервал.

Функция полезности вычисляется следующим образом:

$$utility(\{t, u, r, vo, QoS, v\}) = \begin{cases} h(u, r) s(r), & \text{если } SLA \text{ предоставлен} \\ penalty(v, SLA) h(u, r) s(r), & \text{иначе} \end{cases}, \quad (2)$$

где *t* – время, *u* – пользователь, *r* – ресурс, *vo* – ВО, *QoS* – качество обслуживания, *v* – реальное значение параметра качества обслуживания, *h* – функция принадлежности пользователя и ресурса к одной организации, *SLA* – заранее со-

гласованное значение уровня качества услуг между пользователем и поставщиком ресурсов,  $penalty(v, SLA)$  – функция штрафа, наложенная на поставщик ресурсов, если заранее согласованный уровень качества услуг не выдерживается.

Вид функции штрафа зависит от требований к предоставляемому качеству услуг QoS. Например, для метрик времени, которые в основном необходимо минимизировать, функция штрафа может принимать следующий вид:

$$penalty(v, SLA) = \begin{cases} 1, & \text{если } v \leq SLA \\ \frac{SLA}{v}, & \text{если } v > SLA \end{cases} \quad (3)$$

Вычислим значение функции полезности на основе последовательности событий  $Trace|_{(vo, r, t)}$ :

$$O_{(vo, r, t)} = \{ z(t, t_c) \cdot utility(\{t, u, r, vo, QoS, v\}) \mid \{t, u, r, vo, QoS, v\} \in Trace|_{(vo, r, t)} \}. \quad (4)$$

Репутация – это математическое ожидание функции полезности:

$$rep(vo, r, t) = E[ utility(O_{(vo, r, t)}) ] = \int utility(O_{(vo, r, t)}) P_{utility}(O_{(vo, r, t)}) dO_{(vo, r, t)}. \quad (5)$$

Для аппроксимации математического ожидания будем использовать выборочное среднее (4):

$$rep(vo, r, t) = \frac{1}{|O_{(vo, r, t)}|} \sum_{x \in O_{(vo, r, t)}} x, \quad (6)$$

где  $|\cdot|$  – мощность множества.

Репутация организации  $o$  в ВО – это агрегация репутации всех ресурсов, которые эта организация предоставляет в ВО (6):

$$rep(vo, t) = \frac{1}{|f_{vo}^{-1}(o)|} \sum_{r \in f_{vo}^{-1}(o)} rep(vo, r, t). \quad (7)$$

Репутацию ресурса во всех ВО можно оценить следующим образом (6):

$$rep(r, t) = \frac{1}{|VO|_r} \sum_{vo \in VO|_r} rep(vo, r, t). \quad (8)$$

### 3.2. Модель репутации для пользователей

В работе [13] модель репутации для пользователя строится с использованием функции штрафа. Если пользователь совершает дей-

ствия, которые не соответствуют политике безопасности ВО или ресурса, то на пользователя накладывается штраф. Этот штраф используется для оценки функции полезности для пользователя, а соответственно и его репутации. Но аудит действий пользователя, а также сравнение его действий с политикой безопасности ВО или ресурса является достаточно сложной и трудоемкой задачей, особенно с точки зрения реализации в реальной системе. Должны быть четкие и согласованные критерии, а также соответствующие программные компоненты, позволяющие реализовать такой аудит и проверку с заданными политиками безопасности.

Для того чтоб оценить репутацию пользователя в данной работе были использованы методы и модели, которые традиционно применялись в системах выявления вторжений (IDSs). На данный момент уже предпринимались попытки использования систем выявления вторжений для Grid-систем [30, 31].

В данной работе была использована статистическая модель поведения пользователя (SMUB), изначально разработанная для компьютерных сетей, а в дальнейшем расширена для распределенных сетей [19, 20, 21, 22]. Эта модель основана на анализе статистической информации, которая собирается после выполнения пользователем действий в системе. Данная модель верифицировалась на реальных данных, собранных в инфраструктуре GILDA проекта EGEE. При проведении экспериментов были получены следующие результаты: с помощью модели можно было определить аномальное поведение пользователей в среднем в 86%, ошибка первого рода составила 7.48%, а второго – 20.9%.

Выход модели может интерпретироваться, как «репутация пользователя» в том смысле, что текущие действия пользователя соответствуют его поведению в прошлом. Такая модель пользователя будет специфической для каждой виртуальной организации в зависимости от ее целей и типов задач, которые выполняются в ней. Например, ВО может быть ориентирована на приложения, которые требуют выполнения большого количества задач со сравнительно небольшим объемом данных. В таких ВО задачи, требующие всей оперативной и виртуальной памяти ресурса, будут рассматриваться как аномальные шаблоны поведения. С другой стороны, другие ВО могут быть ориентированы на приложения, где одна задача состоит из нескольких элементарных подзадач,

каждая из которых требует обработки большого количества данных. Такие задачи характерны для области наук про Землю и обработки спутниковых данных [1, 32].

К преимуществам такой модели стоит отнести возможность выявлять отклонения от моделей поведения пользователя, отличать разных пользователей и интегрировать известные шаблоны поведения в процесс настройки параметров модели. Недостатками такой модели являются достаточно высокий уровень ложных срабатываний и необходимость переобучать модель через некоторое время. Несмотря на то, что у предложенной модели есть как достоинства, так и недостатки необходимо отметить, что ее целесообразно использовать вместе с другими механизмами безопасности, что позволит более эффективно оценить репутацию пользователя.

Приведем формальное описание модели репутации для пользователя, основанной на статистической модели поведения пользователя.

Для пользователя определим набор *Event* следующим образом:

$$Event = \{t, u, r, vo, \mathbf{x}\}. \quad (9)$$

где  $\mathbf{x}$  – вектор параметров статистической модели поведения пользователя, описанный в [14, 15].

Последовательность событий (*Events*) имеет следующий вид:

$$Trace = \bigcup_p Event_p = \bigcup_p \{t, u, r, vo, \mathbf{x}\}_p. \quad (10)$$

Функцию полезности *utility()* определим следующим образом (10):

$$utility : Event \rightarrow R, \quad (11)$$

$$utility(\{t, u, r, vo, \mathbf{x}\}) = SMUB_{(u, vo)}(\mathbf{x}), \quad (12)$$

где  $SMUB_{(u, vo)}(\mathbf{x})$  – выход статистической модели поведения пользователя, который может принимать значения из отрезка [0; 1].

Важно понимать, что в общем случае в качестве функции полезности для пользователя *utility()* могут быть использованы другие модели поведения пользователя, например [33, 34], или ансамбль разных моделей для того, чтобы учесть различные аспекты поведения пользователя.

Статистическая модель поведения пользователя основана на использовании нейронных сетей [35] и является специфической для пользователя и ВО [15]. Определим набор параметров, которые используются для оценки репутации пользователя *u* в ВО *vo* до текущего времени *t* следующим образом:

$$Trace|_{(vo, u, t)} = \left\{ \begin{array}{l} \{t', u', r', vo', \mathbf{x}'\} \in Trace : \\ u = u', vo = vo', t' \leq t \end{array} \right\} \quad (13)$$

Вычислим значение функции полезности на основе последовательности событий  $Trace|_{(vo, u, t)}$  (12-13):

$$O_{(vo, u, t)} = \{z(t, t_c) \cdot utility(\{t, u, r, vo, \mathbf{x}\}) \mid \{t, u, r, vo, \mathbf{x}\} \in Trace|_{(vo, u, t)}\}. \quad (14)$$

Значение репутации вычисляется следующим образом:

$$rep(vo, u, t) = E[utility(O_{(vo, u, t)})] = \int utility(O_{(vo, u, t)}) P_{utility}(O_{(vo, u, t)}) dO_{(vo, u, t)}. \quad (15)$$

Для аппроксимации математического ожидания будем использовать выборочное среднее (14)

$$rep(vo, u, t) = \frac{1}{|O_{(vo, u, t)}|} \sum_{x \in O_{(vo, u, t)}} x. \quad (16)$$

Репутация организации в ВО (с точки зрения пользователя) – это агрегация репутаций всех пользователей, которые принимают участие в ВО (16):

$$rep(vo, t) = \frac{1}{|g_{vo}^{-1}(o)|} \sum_{r \in g_{vo}^{-1}(o)} rep(vo, u, t). \quad (17)$$

Репутация пользователя во всех ВО можно оценить следующим образом (16):

$$rep(r, t) = \frac{1}{|VO|_u} \sum_{vo \in VO|_u} rep(vo, u, t). \quad (18)$$

#### 4. Интеграция репутации в планировщик задач для grid-систем

В данной работе рассматривается задача динамического планирования, при которой задачи в Grid-системе распределяются по ресурсам сразу после поступления (в отличие от статического планирования, которое использовалось в работе [13]). Для интеграции репутации в алгоритм планирования предлагается новый подход с использованием нелинейной схемы компромиссов [23].

Обозначим критерий, ассоциированный с планировщиком посредством *y*, т.е. этот критерий минимизируется при распределении задач на ресурсы Grid-системы. В качестве такого критерия могут выступать: минимальное время выполнения задачи, приемлемое время выполнения [36], текущее количество задач в очереди ресурса или уровень отказов. Пусть  $rep(r_i)$  – это значение репутации ресурса  $r_i$ . При интеграции репутации в планировщик ресурсов возникает следующая многокритериальная задача опти-

мизации: необходимо выбрать ресурс, который минимизирует значение  $y(r_i)$ , но при этом имеет максимальную репутацию  $rep(r_i)$ . Таким образом, необходимо синтезировать в единую скалярную функцию частные критерии, которая бы в различных ситуациях выражала бы разные принципы оптимальности. Для этого предлагается использовать нелинейную схему компромиссов [23].

В данной схеме нормализованные частные критерии  $y_k$  синтезируются в единую свертку, используя следующее выражение:

$$Y(x) = \sum_{k=1}^s \alpha_k [1 - y_k(x)]^{-1}; \alpha_k \geq 0, \sum_{k=1}^s \alpha_k = 1, \quad (19)$$

где  $\alpha_k$  – параметры имеющие двоякий физический смысл [23]. С одной стороны – это весовые коэффициенты, выражающие значения отдельных критериев, с другой – это коэффициенты регрессии содержательной регрессионной модели функции полезности, построенной на основе концепции нелинейной схемы компромиссов. Преимуществом использования в качестве интегрирующей функции (19) является ее способность в промежуточных случаях приводить к парето-оптимальным решениям, дающим различные меры частичного удовлетворения критериев.

Репутация может быть интегрирована в планировщик ресурсов следующим образом:

$$Y(r_i) = \frac{\alpha_1}{1 - y_n(r_i)} + \frac{\alpha_2}{rep(r_i)}, \quad (20)$$

где  $y_n(r_i)$  – нормализованное значение критерия, ассоциированного с планировщиком.

Таким образом, задача отправляется на ресурс, который минимизирует функцию (20):

$$r^* = \arg \min_{r_i} Y(r_i). \quad (21)$$

В проведенных экспериментах для распределения задачи на ресурс использовалась эвристика  $ECT$  (определение минимального времени выполнения задачи). Пусть  $ECT(r_i)$  – оценка время выполнения задачи на ресурсе  $r_i$ , а  $ECT_n(r_i)$  – соответствующее нормализованное значение:

$$ECT_n = \frac{ECT(r_i)}{ECT_{max}}, \quad (22)$$

где  $ECT_{max}$  – максимальное значения  $ECT$ .

Планировщик  $ECT-reputation$ , учитывающий репутацию, отправляет задачу на тот ресурс, который минимизирует следующее выражение:

$$r^* = \arg \min_{r_i} \left[ \frac{\alpha_1}{1 - ECT_n(r_i)} + \frac{\alpha_2}{rep(r_i)} \right]. \quad (23)$$

Для оценки эффективности предложенная схема сравнивалась мультипликативной схемой, при которой задача отправляется на ресурс, минимизирующий следующее выражение:

$$r^* = \arg \min_{r_i} [ECT(r_i)(1 - rep(r_i))]. \quad (24)$$

## 5. Результаты экспериментов

В данном разделе представлены результаты экспериментов для оценки эффективности предложенной модели. Эффективность оценивалась с точки зрения улучшения управления ресурсами в Grid-системе. Для проведения имитационного моделирования различных сценариев было разработано специализированное программное обеспечение.

### 5.1. Описание данных

Для генерации загрузки системы при проведении экспериментов использовались реальные данные, полученные проектом Grid Observatory<sup>2</sup>. В рамках этого проекта предоставляются данные «жизненного цикла» задач в Grid-инфраструктуре EGEE. В частности, были использованы данные, собранные системой Real Time Monitor (RTM), которая интегрирует различную информацию о задачах, выполняемых в Grid. Для каждой задачи имеется 37 атрибутов, которые делятся на следующие категории: информационные (Information), метки времени (Timestamps) и метрики (Metrics) [37].

### 5.2. Параметры экспериментов

Все эксперименты проводились для Grid-инфраструктуры, состоящей из 20 ресурсов. Производительность ресурсов (в условных единицах) равномерно выбиралась из промежутка [1, 200]. Сложность задач генерировалась на основе данных проекта Grid Observatory и находилась в отрезке [1, 56000]. Распределение сложности задачи представлено на рис. 1. Время выполнения задачи на ресурсе оценивалось как отношение сложности задачи к производительности ресурса  $jobComplexity/resource\ Productivity$ . Время между поступлениями задач и нагрузка на вычислительную систему также генерировались на основе статистических данных Grid Observatory. На рис. 2 и 3 представле-

<sup>2</sup> Grid Observatory: [www.grid-observatory.org](http://www.grid-observatory.org)

но изменение общего количества отправленных задач от времени и скорость поступления задач соответственно.

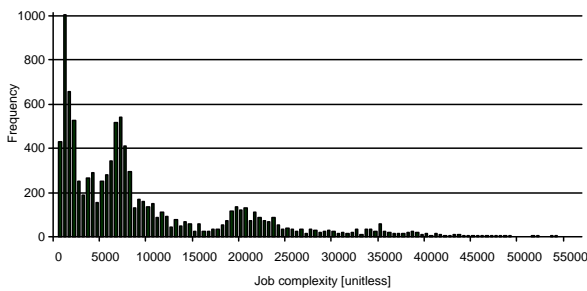


Рис. 1. Распределение сложности задач (общее число задач - 10000)

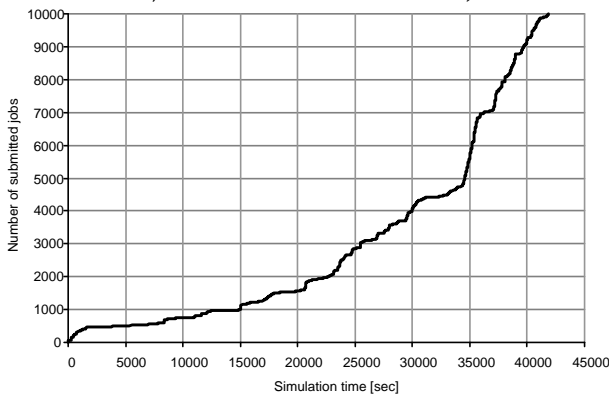


Рис. 2. Динамика изменения общего количества отправленных на выполнение задач от времени

В рамках экспериментов использовались следующие параметры качества обслуживания QoS: время ожидания задачи, время обработки задачи и общее время выполнения задачи. Значение согласованного уровня качества обслуживания SLA формировалось таким образом: время ожидания задачи выбиралось случайным образом из отрезка [1, 30000] сек, время обработки задачи выбиралось как отношение сложности задачи к минимальной производительности ресурса  $jobComplexity/minResourceProductivity$ . Для моделирования сценария, при котором ресурс не выдерживал согласованное время выполнения, применялся следующий подход: к случайному числу из отрезка [1, 2500] сек прибавлялось реальное значение времени выполнения. Функция штрафа и репутация оценивались с использованием выражений (3) и (6) соответственно. Функция полезности (2) вычислялась с использованием в качестве параметра качества обслуживания QoS общее время выполнения задачи.

### 5.3. Анализ эффективности предложенного подхода

В приведенных экспериментах задачи распределялись по ресурсам сразу после поступле-

ния в систему. В рамках экспериментов сравнивались следующие планировщики с учетом и без учета репутации:

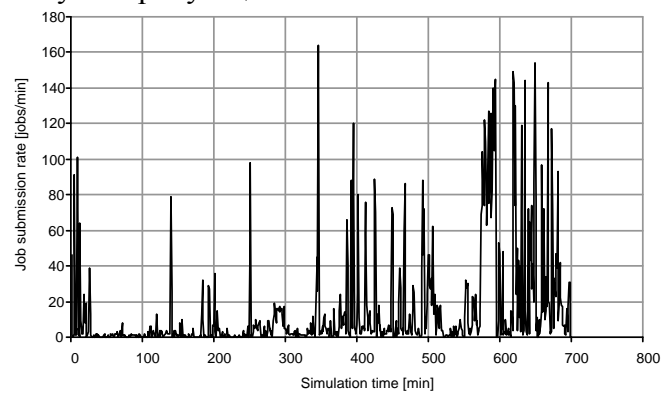


Рис. 3. Интенсивность поступления задач в систему

– Эвристический планировщик, который отправляет задачу на ресурс, минимизирующий ожидаемое время выполнения задачи (ECT).

– Планировщик, интегрирующий репутацию на основе нелинейной схемы компромиссов (23). В проведенных экспериментах были установлены следующие значения параметров:  $\alpha_1=\alpha_2=0.5$ .

– Планировщик, интегрирующий репутацию с использованием мультипликативной схемы (24).

Эффективность планировщиков сравнивалась по следующим критериям:

– время выполнения всех задач (Makespan): разница между временем завершения последней задачи и временем поступления первой задачи;

– среднее время выполнения задачи (Average Job Execution Time): среднее время, которое требуется для выполнения задачи;

– среднее время ожидания задачи в очереди (Average Job Queue Waiting Time);

– среднее время превышения выполнения задачи (Average Job Excess Time): среднее время превышения согласованного времени обслуживания (выполнения задачи).

– количество задач, для которых предоставлен ненадежный сервис (SLA missed), т.е. при обработке задачи не было предоставлено согласованное качество обслуживания;

– среднее значение функции полезности (Average utility).

– использование ресурса (Resource utilization): количество задач, выполненных данным ресурсом.

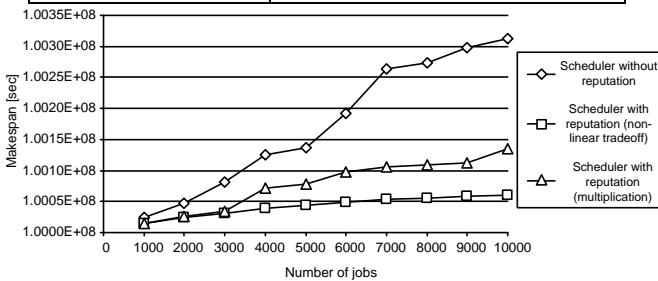
В рамках экспериментов варьировались следующие параметры: нагрузка (общее количе-

ство отправленных на выполнение задач) и надежность ресурса.

В первом случае, разное количество моделируемых задач соответствовало разной интенсивности (Табл. 1)

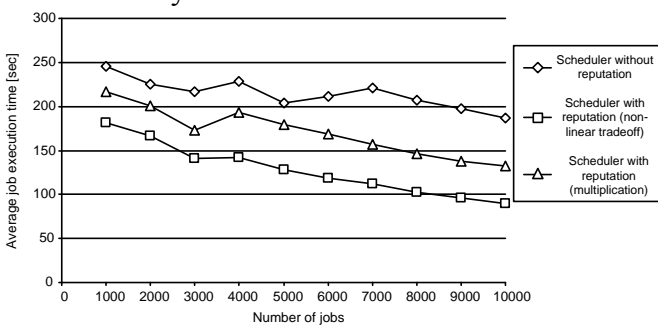
**Табл. 1. Количество и интенсивность поступления задач в рамках проведенных экспериментов**

Количество задач	Интенсивность поступления задач [задач/мин]
1000	4
2000	5
3000	7
4000	8
5000	9
6000	10
7000	11
8000	12
9000	13
10000	14

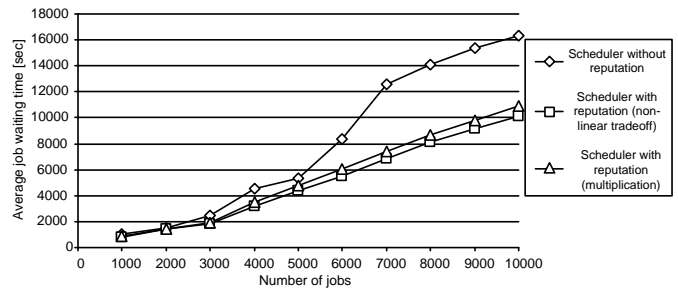


**Рис. 4. Время выполнения задач для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему**

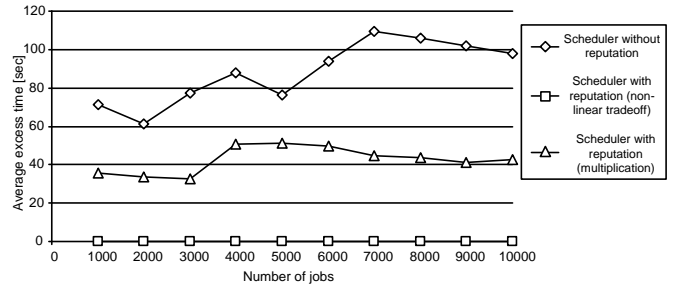
Из 20 вычислительных ресурсов 4 ресурса (20%) были установлены ненадежными, т.е. согласованный уровень качества услуг SLA всегда нарушался этими ресурсами. Начальная репутация этих ресурсов была равна 0.1. На рис. 4-10 показано сравнение планировщиков по разным критериям в зависимости от нагрузки на систему.



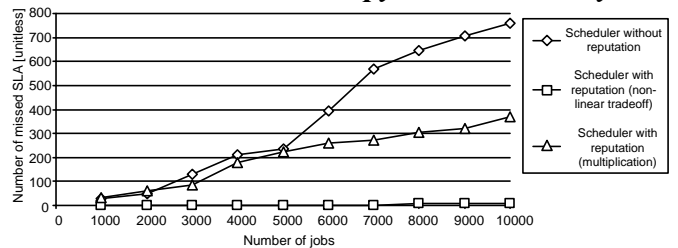
**Рис. 5. Среднее время выполнения задачи для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему**



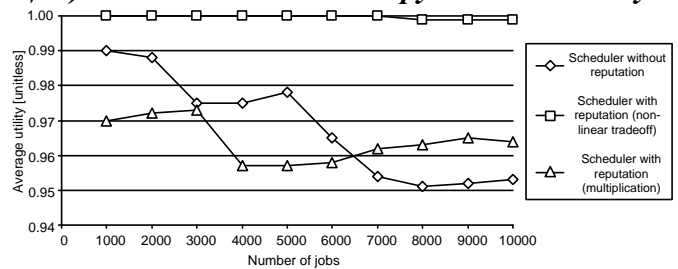
**Рис. 6. Среднее время ожидания задачи в очереди для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему**



**Рис. 7. Среднее время превышения выполнения задачи для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему**

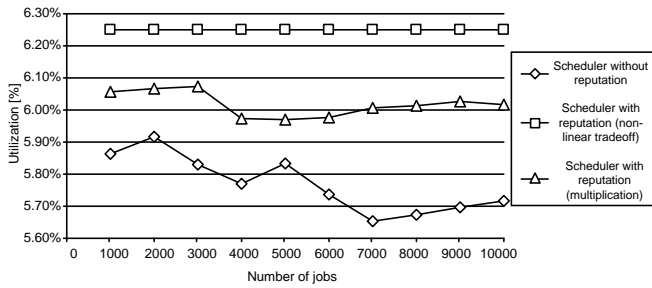


**Рис. 8. Количество задач, для которых был предоставлен ненадежный сервис, для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему**

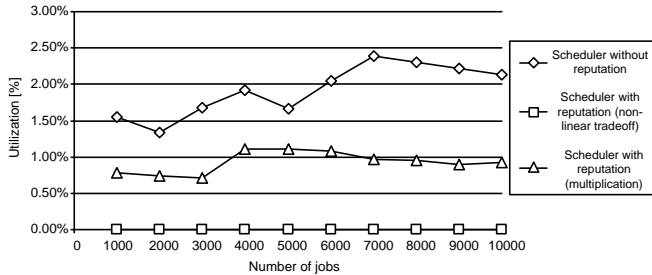


**Рис. 9. Среднее значение функции полезности для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему**





(a)



(b)

**Рис. 10. Среднее количество выполненных задач для разных планировщиков (с и без учета репутации) в зависимости от нагрузки на систему (a – надежные ресурсы; b – ненадежные ресурсы)**

Полученные результаты показали, что планировщик, интегрирующий репутацию на основе нелинейной схемы компромиссов, имеет большую эффективность по сравнению с другими планировщиками по всем критериям. Относительное улучшение значений критериев эффективности использования репутации в планировщике (с использованием обеих схем) по сравнению со стандартным планировщиком представлено в табл. 2. По сравнению с мультипликативной схемой использование нелинейной схемы компромиссов для интеграции репутации в планировщик задач позволило повысить эффективность его работы на 25%.

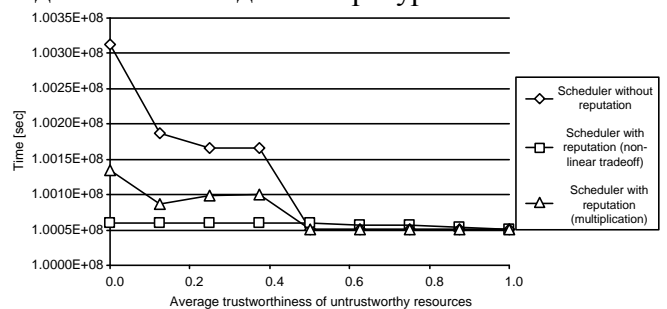
Стоит отметить, что для повышения эффективности при планировании задач репутация должна оцениваться на основе соответствующих параметров качества обслуживания. Иначе, повышение производительности по некоторым из приведенных выше критериев может не наблюдаться.

**Табл. 2. Относительное повышение значений критериев эффективности при интеграции репутации в планировщик**

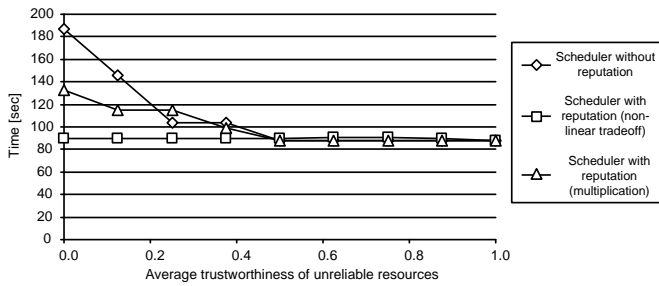
Критерий эффективности	Среднее улучшение [%]	
	Нелинейная схема компромиссов	Мультипликативная схема
Среднее время выполнения	40.9	20.9
Среднее время ожидания	30.5	26.1
Среднее время	100.0	50.9

превышения выполнения задачи		
Количество задач, для которых предоставлен ненадежный сервис	99.6	26.9
Средняя значение функции полезности	3.2	0.4
Время выполнения всех задач	0.1	0.1
Усредненное значение по всем критериям	45.7	20.9

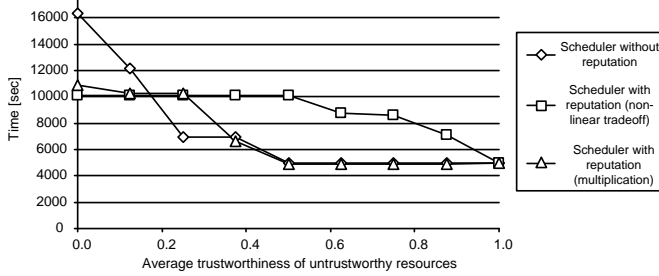
В рамках второй серии экспериментов варьировалась надежность ресурсов. Как и в первом случае, 20% ресурсов были установлены ненадежными, однако с разным уровнем надежности (*trustworthiness rate*). Например, если уровень надежности ресурса равен 0.6, это означает, что в среднем в 60% случаях заранее согласованный уровень обслуживания будет предоставлен. Для моделирования такой ситуации использовался следующий подход: в тех случаях, когда ненадежный ресурс выполнял задачу, генерировалось равномерно распределенное случайное число из отрезка [0; 1]. Если это значение меньше, чем уровень надежности ресурса, тогда ресурс предоставлял согласованный уровень качества обслуживания SLA. В противном случае, предполагалось, что поставщик ресурсов нарушил согласованный уровень качества обслуживания SLA. На рис. 11-17 показано сравнение планировщиков по разным критериям в зависимости от среднего уровня надежности ненадежных ресурсов.



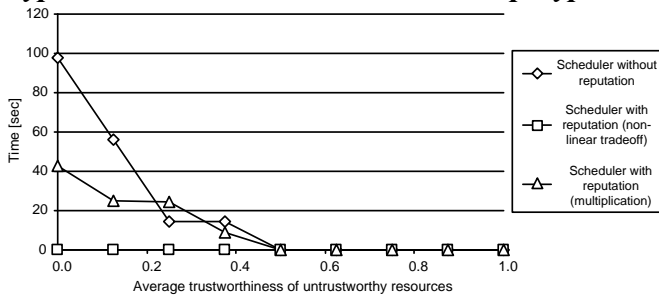
**Рис. 11. Время выполнения задач для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов**



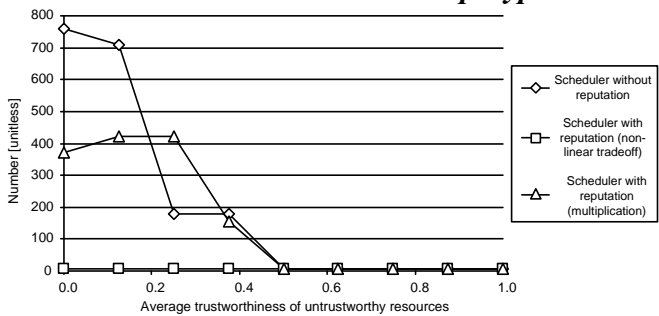
**Рис. 12.** Среднее время выполнения задачи для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов



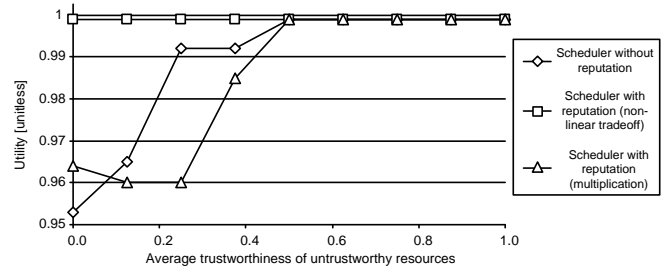
**Рис. 13.** Среднее время ожидания задачи в очереди для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов



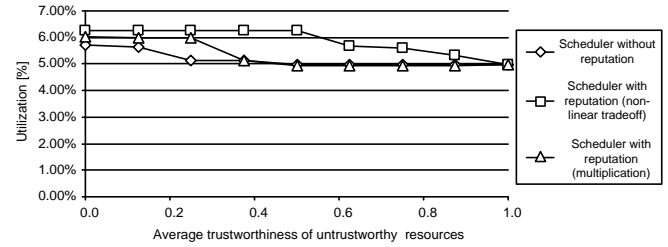
**Рис. 14.** Среднее время превышения выполнения задачи для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов



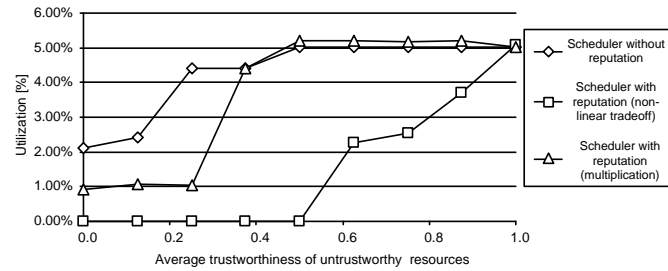
**Рис. 15.** Количество задач, для которых был предоставлен ненадежный сервис, для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов



**Рис. 16.** Среднее значение функции полезности для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов



(a)



(b)

**Рис. 17.** Среднее количество выполненных задач для разных планировщиков (с и без учета репутации) в зависимости от среднего уровня надежности ненадежных ресурсов (a – надежные ресурсы; b – ненадежные ресурсы)

Полученные результаты показали, что планировщик, интегрирующий репутацию на основе нелинейной схемы компромиссов, обеспечил более эффективное распределение задач (рис. 11-17). Например, при его использовании задачи не отправлялись на ненадежные ресурсы до тех пор, пока средний уровень надежности ненадежных ресурсов не достиг значения 0.5 (рис. 17b). Более того, такой планировщик можно адаптировать, настраивая коэффициенты  $\alpha_1$  и  $\alpha_2$  в выражении (23).

## 6. Выводы

В данной работе решается задача многокритериальной оптимизации планирования выполнения задач в структурно-сложных системах на основе моделей репутации. Предложен новый подход для интеграции модели репутации в

подсистему планирования выполнения задач с использованием нелинейной схемы компромиссов. Преимуществом данной схемы является получение парето-оптимальных решений, обеспечивающих различные меры частичного удовлетворения критериев. Полученные экспериментальные результаты свидетельствуют о том, что планировщик, построенный на основе многокритериальной оптимизации с учетом модели репутации обеспечивает более высокую эффективность планирования по сравнению с другими планировщиками без учета репутации на 20%. В свою очередь использование нелинейной схемы компромиссов в задаче многокритериальной оптимизации позволило повысить эффективность планирования задач по

сравнению с мультипликативной схемой еще на 25%. Таким образом, общее повышение эффективности планирования при использовании нелинейной схемы компромиссов с учетом модели репутации составляет примерно 45% по сравнению с традиционными системами планирования. Дополнительным преимуществом предложенного планировщика с учетом модели репутации на основе нелинейной схемы компромиссов является повышение уровня информационной безопасности, поскольку при его использовании задачи не отправляются на ненадежные ресурсы до тех пор, пока средний уровень надежности ресурсов не достигает определенного значения.

### Список литературы

1. Kussul N. Grid Technologies for Satellite Data Processing and Management within International Disaster Monitoring Projects / N. Kussul, A. Shelestov, S. Skakun [S. Fiore, G. Aloisio (eds.)] Grid and Cloud Database Management. – Springer-Verlag, Berlin Heidelberg, 2011, – P. 279-306.
2. Lecca G. Grid computing technology for hydrological applications / G. Lecca, M. Petitdidier, L. Hluchy [et al.] // J. of Hydrol. – 2011. – № 403(1-2). – P.186-199.
3. Kussul N. Grid and sensor web technologies for environmental monitoring / N. Kussul, A. Shelestov, S. Skakun // Earth Sci. Inf. – 2009. – №2(1-2). – P. 37-51.
4. Bouras C. e-Collaboration concepts, systems and applications / C. Bouras, E. Giannaka, E. Tsiatos // Information Science Reference E-Collaboration: Concepts, Methodologies, Tools, and Applications / Kock N. – IGI Global, 2009. – vol. 1, Section I, Chapter 1.2.
5. Foster I. The Anatomy of the Grid: Enabling Scalable Virtual Organizations / I. Foster, C. Kesselman, S. Tuecke // Int. J. of Supercomput. Appl. – 2001. – №15(3). – P. 200-222.
6. Eymann T. A Framework for Trust and Reputation in Grid Environments / T. Eymann, S. König, R. Matros // J. of Grid Comput. – 2008. – №6(3). – P. 225-237.
7. Song S. Trusted Grid Computing with Security Binding and Trust Integration / S. Song, K. Hwang, Y.-K. Kwok // J. of Grid Comput. – 2005. – №3(1-2). – P. 53-73.
8. Chakrabarti A. Grid Computing Security / Chakrabarti Anirban. - Berlin Heidelberg, Springer-Verlag. – 2007. – 331 p.
9. Silaghi G. A Utility-Based Reputation Model for Service-Oriented Computing / G. Silaghi, A. Arenas, L. Silva // Toward Next Generation Grids / Priol, T., Vanneschi, M. – Springer, 2007. – P. 63-72
10. Grandison T. A Survey of Trust in Internet Applications / T. Grandison, M. Sloman // IEEE Commun. Surv. and Tutor. – 2000. – №4(4). – P. 2-16.
11. The Meaning of Trust / D.H. McKnight, N.L. Chervany// Technical Report MISRC Working Paper Series 96-04. – University of Minnesota. Management Information Systems Research Center. – 1996.
12. Abdul-Rahman A, Supporting trust in virtual communities / A. Abdul-Rahman, S. Hailes // Proc. of the IEEE 33rd Hawaii Int. Conf. on Syst. Sci. (HICSS '00). – 2000. – vol. 6. – P. 6007.
13. Arenas A. Reputation management in grid-based virtual organisations / A. Arenas, B. Aziz, G.C. Silaghi // Proc. Int. Conf. on Secur. and Cryptogr. (SECRYPT 2008). – 2008. – P. 538-545.
14. Куссуль О.М. Исследование эффективности применения моделей доверия на основе репутации в Grid-системах / О.М. Куссуль, А.Н. Новиков, С.С. Швец // Наукові праці ДонНТУ Серія "Інформатика, кібернетика та обчислювальна техніка". – 2010. – Випуск 12(165). – С. 126-134.
15. Kussul O. Utility-based reputation model for VO in Grids / O. Kussul, O. Novikov // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – 2009. – №50. – С. 137-145.
16. Azzedin F. Integrating Trust into Grid Resource Management Systems / Azzedin F., Maheswaran M. // Int. Conf. on Parallel Process. (ICPP 2002). – IEEE Computer Society Washington, DC, USA, 2002. – P. 47-54.
17. Gomez Marmol, F. Security threats scenarios in trust and reputation models for distributed systems / F. Gomez Marmol, G. Martinez Perez // Comput. & Secur. – 2009. – №28. – P. 545-556.

18. Kussul N. Grid system for flood extent extraction from satellite images / Kussul N., Shelestov A., Skakun S. // *Earth Sci. Inf.* – 2008. - №1(3-4). – P. 105-117.
19. Shelestov A. Intelligent Model of User Behaviour in Distributed Systems / A. Shelestov, S. Skakun, O. Kussul // *Int. J. on Inf. Theory and Appl.* – 2008. – №15(1). – P. 70-76.
20. Kussul N. Intelligent System for Users' Activity Monitoring in Computer Networks / N. Kussul, S. Skakun // *IEEE Intelligent Data Acquisition and Advanced Computing Systems (IDAACS 2005): Technology and Applications.* – 2005. – P. 306-309.
21. Skakun S. Implementation of the Neural Network Model of Users of Computer Systems on the Basis of Agent Technology / S. Skakun, N. Kussul, A. Lobunets // *J. of Autom. and Inf. Sci.* – 2005. – №37(4). – P. 11-18
22. Kussul N. Neural Network Approach for User Activity Monitoring in Computer Networks / N. Kussul, S. Skakun // *Proc. of the Int. Joint Conf. on Neural Networks (Budapest, Hungary).* – 2004. – vol. 2. – P. 1557-1562.
23. Voronin A.N. A multicriteria problem of distribution of bounded resources / A.N. Voronin // *Cybern. and Syst. Anal.* – 2011. – №47(3). – P. 490-493.
24. Papaioannou T.G. Reputation-Based Estimation of Individual Performance in Grids / T.G. Papaioannou, G.D. Stamoulis // *Eighth IEEE Int. Symp. on Cluster Comput. and the Grid (CCGRID), .* – IEEE Computer Society Washington, DC, USA, 2008. – №47(3). – P. 500-509.
25. Liang Z. A reputation-driven scheduler for autonomic and sustainable resource sharing in Grid computing / Z. Liang, W. Shi. // *J. of Parallel and Distrib. Comput.* – 2010. – №70. – P. 111-125.
26. Wu C.-C., An integrated security-aware job scheduling strategy for large-scale computational grids / Wu C.-C., Sun R.-Y. // *Future Generation Comput. Syst.* – 2010. – №26. – P. 198-206.
27. PathTrust: A Trust-Based Reputation Service for Virtual Organization Formation / F. Kerschbaum, J. Haller, Y. Karabulut, P. Robinson // *Lect. Notes in Comput. Sci.* – 2006. – №3986/2006. – P. 193-205.
28. von Laszewski G. Towards Reputable Grids. Scalable Comput / G. von Laszewski, B. Alunkal, I Veljkovic. // *Pract. and Exp.* – 2005. – №6(3). – P. 95-106.
29. Kamvar S. The EigenTrust algorithm for reputation management in P2P networks / S. Kamvar, M. Schlosser, H. Garcia-Molina // *Proc. of the 12th Int. Conf. on World Wide Web.* – ACM Press, New York, NY, USA, 2003. – №6(3). – P. 640-651.
30. Intrusion Detection for Grid and Cloud Computing / K. Vieira, A. Schulter, C. Westphall // *IT Prof.* – 2007. – №12(4). – P. 38-43.
31. Intrusion Detection for Computational Grids / A. Schulter [et al.] // *Proc. 2nd Int. Conf. New Technol., Mobil., and Secur.* – IEEE Press, 2008. – №22(7). – P. 1-5.
32. Shelestov A. Grid Technologies in Monitoring Systems Based on Satellite Data / A. Shelestov, N. Kussul, S. Skakun // *J. of Autom. and Inf. Sci.* – 2006. – №38(3). – P. 69-80.
33. Oh S.H. An anomaly intrusion detection method by clustering normal user behaviour / S.H. Oh, W.S. Lee // *Comput. & Secur.* – 2003. – №22(7). – P. 596-612.
34. Anomaly Detection in Grid Computing Based on Vector Quantization / H.-W. Sun, K.-Y. Lam, S.-L. Chung [et al.] // *Lect. Notes in Comput. Sci.* – 2004. – №3251/2004. – P. 883-886.
35. Haykin S. *Neural Networks: A Comprehensive Foundation* / Haykin S. – Saddle River, New Jersey, Prentice Hall, 1999. – 842 p.
36. Kretsis A. Developing Scheduling Policies in gLite Middleware / A. Kretsis, P. Kokkinos, E. Varvarigos // *Proc. of the 9th IEEE/ACM Int. Symp. on Cluster Comput. and the Grid.* – IEEE Computer Society Washington, DC, USA, 2009. –P. 20-27.
37. The Grid Observatory / C. Germain-Renaud, A. Cady, P. Gauron [et al.] // *11th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Comput., CCGrid.* – 2011. – P. 114-123.