ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ №3(34), 2017, с. 3–29

УДК 004.75

И. А. Чернов, Е. Е. Ивашко, Н. Н. Никитина

Обзор методов планирования заданий в Desktop Grid

Аннотация. В статье представлен научный обзор современного состояния исследований в области планирования заданий в вычислительных системах типа Desktop Grid. Проведен обзор научных публикаций за период с 1999 по 2017 гг. и выполнен анализ критериев и методов оптимизации, предложенных в научных работах для повышения эффективности планирования заданий в Desktop Grid.

Ключевые слова и фразы: Desktop Grid, планирование заданий, BOINC, Enterprise Desktop Grid, высокопроизводительные вычисления, распределенные вычисления.

Введение

Desktop Grid известна как многообещающая парадигма высокопроизводительных вычислений (high-throughput computing). Desktop Grid — это грид-система, объединяющая через Интернет неспециализированные вычислители (персональные компьютеры, ноутбуки, смартфоны и пр.) и использующая их простаивающие вычислительные ресурсы (idle computing time) для выполнения вычислительноемких расчетов. Как вычислительный инструмент Desktop Grid ведет свою историю с 1999 г., когда был запущен первый масштабный проект распределенных добровольных вычислений SETI@home. На сегодня Desktop Grid является важной частью отрасли высокопроизводительных вычислений наряду с вычислительными кластерами и вычислительными Грид. Однако, в отличие от указанных систем, фундаментальные основы функционирования Desktop Grid значительно менее изучены. При этом успех существующих и появление новых научных проектов на основе Desktop Grid приводят к возникновению новых научных задач.

Работа частично поддержана грантами РФФИ 16-07-00622_а, 15-07-02354_а и 15-29-07974_офи_м.

© И. А. Чернов, Е. Е. Ивашко, Н. Н. Никитина, 2017
© Институт прикладных математических исследований КарНЦ РАН, 2017
© Программные системы: теория и приложения, 2017

DOI: 10.25209/2079-3316-2017-8-3-3-29

Планирование заданий играет важную роль в работе Desktop Grid, так как значимо влияет на производительность вычислительной системы. В сравнении с вычислительными Грид и кластерами, планирование заданий в Desktop Grid осложняется целым рядом факторов, таких как большая аппаратная и программная разнородность, недоверенность вычислительных узлов, неопределенность доступности и надежности узлов и т.д.

Различными исследовательскими группами в научных работах был предложен широкий спектр алгоритмов и эвристик для решения отдельных задач планирования в Desktop Grid. При этом использовалась различная дополнительная информация, такая как оценка периодов доступности узлов, рейтинги надежности и др. В представленной работе анализируются различные критерии и методы, используемые в научной литературе для улучшения отдельных характеристик Desktop Grid.

На сегодняшний день в области исследования Desktop Grid был выполнен ряд обзоров литературы, в том числе анализ работ, связанных с планированием заданий. Эти публикации фокусируются на классификации задач планирования [**1**], эвристиках [**2**], алгоритмах, основанных на знаниях [**3**, **4**]; указанные работы более подробно обсуждаются в разделе 2.3. В отличие от предыдущих работ, настоящий обзор

- сосредотачивает внимание на критериях и методах оптимизации;
- включает большое число научных работ, опубликованных за период с 1999 г. вплоть до 2017 г.

Статья имеет следующую структуру: в разделе 1 дано краткое описание концепции Desktop Grid и программной платформы BOINC как наиболее часто используемого промежуточного программного обеспечения Desktop Grid. В разделе 2 представлен анализ обзорных и исследовательских статей в области планирования заданий в Desktop Grid на основе BOINC: описаны объекты исследования, дан обзор задач, критериев и методов оптимизации и результатов.

1. Desktop Grid и BOINC

1.1. Вычислительные системы Desktop Grid

Desktop Grid как инструмент высокопроизводительных вычислений представляют большой интерес из-за присущих им достоинств, которые включают в себя следующее:

- простота развертывания и поддержки;
- низкая стоимость;
- высокая масштабируемость;
- большая потенциальная пиковая производительность;
- простота разработки приложений;
- возможность использования унаследованных приложений.

Однако существует также целый ряд недостатков, существенно сокращающих класс задач, которые могут эффективно решаться в Desktop Grid:

- низкая скорость каналов связи;
- низкая вычислительная мощность отдельных узлов;
- высокая аппаратная и программная разнородность;
- отсутствие информации о доступности узлов;
- низкая надежность узлов;
- недоверенность узлов.

Указанные недостатки приводят к существенному снижению эффективной вычислительной производительности по сравнению с пиковой, а также к значительному влиянию политики планирования заданий на производительность.

В литературе рассматривается несколько невзаимоисключающих видов Desktop Grid, особенности которых могут быть учтены при планировании заданий:

- **проект добровольных вычислений** (volunteer computing): вычислительные узлы являются компьютерами добровольцев и соединены с сервером сетью Интернет. Планирование заданий затрудняется высокой гетерогенностью, низкой скоростью соединения, ненадежностью узлов и пр.;
- **Enterprise Desktop Grid**: вычислительные узлы являются рабочими станциями некоторой организации и соединены с сервером по локальной сети. В такой системе доступность и надежность узлов выше в сравнении с добровольными вычислениями;
- **гибридная система распределенных вычислений** на узлах суперкомпьютеров и вычислительных кластеров [5–7]. Краткий обзор гибридных систем распределенных вычислений и особенностей управления заданиями в подобных системах приведен в работе [8];

- **иерархическая Desktop Grid**: существует иерархия серверов вычислительной сети, в которой каждый сервер более низкого уровня иерархии получает набор заданий от сервера более высокого уровня и должен распределить эти задания между своими подчиненными серверами или вычислительными узлами (на самом низком уровне иерархии). Каждый сервер решает свою частную задачу планирования заданий [9];
- **пиринговая Desktop Grid**: вычислительные узлы связаны друг с другом и могут обмениваться данными. Планировщик заданий может принимать во внимание граф связей между заданиями для ускорения вычислений, снижения нагрузки на сервер, уменьшения времени коммуникаций и др. [9, 10].

В представленной работе рассматриваются наиболее распространенные Desktop Grid, используемые для организации добровольных вычислений.

1.2. Архитектура BOINC и базовый планировщик заданий

Существует целый ряд программных систем для организации Desktop Grid. Однако свободно распространяемая (Open Source) платформа BOINC [11] является де-факто стандартом для добровольных вычислений. Всего на текущий момент существует более 60 активных проектов на основе BOINC, объединяющих более 15 млн. вычислительных узлов по всему миру [12].

Платформа BOINC основана на клиент-серверной архитектуре. Клиентская часть — это программа, задействующая простаивающие вычислительные ресурсы компьютера для выполнения вычислений в рамках одного или нескольких проектов. Клиентские программы доступны практически для любого аппаратного и системного программного обеспечения вычислительных узлов.

Серверная часть состоит из нескольких подсистем, отвечающих за генерацию заданий, их назначение вычислительным узлам, получение и обработку результатов и т.д.

В вычислительной модели BOINC каждое задание имеет несколько копий, называемых *репликами*. Каждая реплика рассчитывается на отдельном вычислительном узле, и результаты нескольких реплик сравниваются между собой. Концепция *кворума* используется для получения «эталонного» результата задания. При необходимости (например, при большом числе ошибочных или потерянных заданий),

количество реплик для одного задания может быть увеличено в процессе расчетов. Для всех заданий в BOINC устанавливается *срок выполнения (deadline)*, в течение которого вычислительный узел должен вернуть результат. Если в течение установленного срока результат сервером не получен, то считается, что задание потеряно.

Механизм репликаций является формой избыточных вычислений и преследует несколько целей: во-первых, это повышение надежности с повышением вероятности получения в срок корректного результата, даже если некоторые узлы были выключены или выведены из вычислительной сети, не завершив выполнения задания. Это, в свою очередь, повышает производительность, так как снижает среднее время получения результата от набора ненадежных узлов. Также репликация и голосование играют важную роль в борьбе с саботажем. Репутационный кворум, при котором доверенные узлы имеют при голосовании больший вес, чем обычные узлы, — это метод, использующий одновременно и механизмы репутации, и репликации.

Встроенный механизм планирования заданий в BOINC состоит из серверной и клиентской частей. Серверная часть служит для назначения заданий вычислительным узлам в соответствии с некоторой политикой планирования. Клиентская часть выстраивает порядок выполнения заданий от нескольких проектов, так, чтобы, например, не нарушать сроки выполнения, честно разделять процессорное время узла между различными проектами и т.д.

Ряд исследований был посвящен оптимизации локального (клиентского) планировщика заданий. Например, в статье [**13**] предложен алгоритм многокритериальной оптимизации. Алгоритм не гарантирует удовлетворение всех ограничений, установленных пользователем, однако позволяет обеспечить лучшую производительность по сравнению с исходным алгоритмом планирования BOINC, что показано на экспериментах, проведенных с помощью эмулятора EmBOINC. В работе [**14**] рассматривается задача планирования нескольких наборов заданий, т.е. сначала выбирается набор заданий, а затем конкретное задание из выбранного набора. В работе не предполагается использование дополнительной информации о вычислительных узлах или заданиях; для назначения заданий используется алгоритм планирования WQR-FT, а для выбора набора заданий — одна из пяти эвристик. Результаты тестируются с помощью имитационной модели.

BOINC также использует механизмы *планирования с привязкой к узлам (locality scheduling)* и *однородной избыточности (homogeneous*

redundancy). Планирование с привязкой к узлам призвано минимизировать объем данных, пересылаемых клиентам: задание по возможности назначается тем узлам, на которых уже имеются необходимые входные данные. Механизм однородной избыточности реализован для борьбы с численными различиями в результатах, вызванными различиями в характеристиках вычислительных узлов: узлы объединяются в группы на основе типа операционной системы и архитектуры процессора; реплики одного и того же задания назначаются узлам одной и той же группы.

Используемый в BOINC серверный планировщик заданий является «наивным» и учитывает лишь несколько достаточно простых технических ограничений. Производительность Desktop Grid можно повысить, используя более сложные алгоритмы планирования, основанные на информации о структуре вычислительной сети, характеристиках вычислительных узлов, специфике предметной области и т.п. Такого рода информация может быть собрана в процессе вычислений.

Тестирование алгоритмов планирования в реальных вычислительных проектах является сложным и времяемким, поэтому для этой цели часто используются эмуляторы или имитационные модели Desktop Grid, разработке, описанию и изучению свойств которых посвящен ряд научных работ. Так, в статье [**15**] предложен эмулятор клиента BOINC, позволяющий протестировать различные стратегии планирования заданий при разных сценариях работы, возникающих при варьировании аппаратных характеристик, шаблонов доступности вычислительного узла, наличия ошибок выполнения задания и т.д. Дано описание и проводится оценка политик планирования BOINC в соответствии с несколькими метриками производительности; показано какие стратегии являются наиболее выгодными при различных условиях. В статье [**16**] предложена система имитационного моделирования SimGrid. Глава книги [**3**] частично посвящена обзору и описанию систем SimBA, SimBOINC и EmBOINC. Эмуляция Desktop Grid с помощью EmBOINC детально описана в работах [**17**, **18**], внимание также уделено средствам имитационного моделирования. Система имитационного моделирования BOINC SimBA описана в [**19**].

В следующем разделе представлен анализ исследований, посвященных различным аспектам планирования заданий сервером Desktop Grid на базе BOINC.

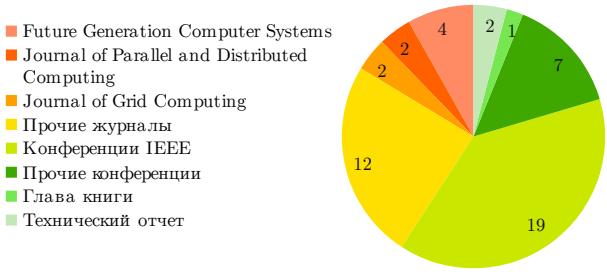

Рис. 1. Источники публикаций, вошедших в обзор (1999–2017)

2. Аналитический обзор научных работ

2.1. Исходные данные

При отборе статей по планированию заданий в Desktop Grid мы опирались на библиографическую базу научных статей Scopus. Всего нами было проанализировано 113 статей, посвященных планированию заданий и смежным вопросам, опубликованных за период с 1999 по 2017 гг.; из них мы отобрали 49 статей для более подробного анализа в рамках представленного обзора.

Отобранные для обзора статьи опубликованы в общей сложности в 20 научных журналах, в трудах 26 конференций, в одной главе книги и двух технических отчетах. Распределение статей по источникам публикации представлено на рис. 1.

2.2. Критерии и методы оптимизации

Планирование заданий — это сложная задача, даже для случаев с полной информацией [**2**, **20**–**22**]; в условиях реальных вычислительных проектов она становится еще сложнее. Во-первых, это связано с тем, что полный набор заданий редко сформирован заранее и дополняется динамически; во-вторых, набор доступных вычислительных узлов изменяется со временем: узлы становятся недоступны (временно или насовсем), возвращают неверные результаты, к проекту подключаются новые вычислительные узлы (что особенно актуально для добровольных вычислений).

При разработке алгоритмов планирования заданий используется несколько распространенных критериев оптимизации. Выбор определенного критерия существенно влияет на характеристики алгоритма планирования, т.к. некоторые критерии оптимизации могут быть

противоречивыми (например, надежность вычислений и производительность). Ниже представлен список наиболее популярных критериев оптимизации:

- *пропускная способность (throughput)*: число завершенных заданий или прошедших валидацию результатов, полученных в единицу времени;
- *время обработки (makespan)* набора вычислительных заданий;
- *время завершения (turnaround time)* вычислительного задания;
- *надежность (reliability)*: вероятность получить, отправив вычислительному узлу задание, правильный, допустимый или хоть какой-либо результат;
- *доступность (availability)*: доля результатов, полученных от вычислительных узлов, из общего числа отправленных сервером заданий;
- *уровень загрузки (load)* сервера или вычислительных узлов;
- *накладные расходы (overhead)* вследствие дублирования заданий, нарушения сроков обработки или ошибок.

Помимо перечисленных, используются также и менее распространенные критерии, такие как устойчивость к саботажу (актуальная для добровольных вычислений), оптимальная обработка зависимых между собой заданий или заданий из нескольких проектов, предсказание доступности вычислительных узлов, оптимальное использование разнородных вычислительных узлов [**10**, **23**, **24**] и др.

Как правило, исследователи предлагают и оценивают эффективность специальных методов оптимизации выбранного критерия или набора критериев. Несмотря на свои различия, большинство используемых методов оптимизации могут быть сгруппированы по следующим типам:

- *группировка или кластеризация вычислительных узлов* в соответствии с какими-либо свойствами (например, надежность, доступность, вычислительная мощность) часто используется как отдельно, так и в комбинации с другими методами. Концепция групп используется для голосования, а задания часто назначаются конкрентной группе и распределяются внутри нее доступным вычислительным узлам.

- *Репутация вычислительных узлов или их групп* оценивается по истории их работы и отражает надежность и/или доступность. Репутация может быть определена как вероятность получить ответ в установленный срок, вероятность злонамеренных действий или др. «Доверительные» методы также основаны на репутации: создание «черных списков», рекомендации и т.п.
- *Дублирование заданий (репликация)* совместно с механизмом кворума позволяет повысить устойчивость вычислений против случайных и преднамеренных ошибок, а также снизить количество незавершенных в срок заданий и время ожидания ответа в целом.
- *Эвристики*, зачастую основанные на «жадных» алгоритмах и проверяемые экспериментально с помощью имитационных моделей или эмуляторов. Необходимо отметить, что имитационные модели используются значительно чаще: только в двух статьях ([**3**, **19**]) из перечисленных в таблице 1 используются эмуляторы.
- *Аналитические модели*, основанные на использовании методов теории вероятностей, теории игр, марковских цепей и др.

Выбор критерия и методов оптимизации определяет свойства конкретного алгоритма планирования заданий. В следующих разделах представлен обзор работ, использующих критерии оптимизации, направленные на повышение вычислительной производительности, и методы, связанные с реализацией специальных схем планирования заданий Desktop Grid.

2.3. Обзорные статьи

Ряд исследований за рассматриваемый период посвящен обзору различных аспектов планирования заданий, разработкам классификаций и единой таксономии Desktop Grid.

В статье [**25**] рассматриваются и сравниваются наиболее популярные современные системы для организации Desktop Grid: BOINC, XtremWeb, OurGrid и HTCondor. Приводится обзор работ, направленных на исследование факторов, влияющих на производительность Desktop Grid (как с точки зрения сервера, так и с точки зрения клиента).

В статье [**1**] предложены различные классификации Desktop Grid, в том числе с точки зрения планирования заданий. При этом планировщик рассматривается с разных ракурсов: с точки зрения приложения (зависимость заданий, делимость, вычислительная сложность и др.), ресурса (выделенность, доверительность, гетерогенность) и

планирования заданий. Последнее зависит от структуры Desktop Grid (централизованное, иерархическое или распределенное планирование заданий), режима получения заданий (PUSH или PULL), политики планирования и группировки, статического, динамического или адаптивного типа планирования заданий, обработки ошибок, регулирования нагрузки и критериев оптимизации. Работа опубликована в 2006 г. и не учитывает многие из критериев оптимизации, используемых в более современных исследованиях в этой области.

Обзор [2] посвящен вычислительным Грид: выделены источники неопределенности при планировании заданий (гетерогенность ресурсов и заданий, динамическая структура и др.), выделены различные типы планировщиков, процедура планирования заданий разделена на несколько этапов, выделены несколько типов планирования заданий в Грид. Рассматриваемые методы планирования включают в себя работу с зависимыми заданиями, различные способы организации планирования: централизованное, иерархическое и распределенное; статическое и динамическое; отдельными заданиями или группами; адаптивное; ориентированное на данные. Модели планирования заданий описаны с точки зрения доступной информации, критериев оптимизации и т.п. Основным результатом статьи являются предложенные вычислительные модели планировщика Грид и анализ нескольких эвристик и мета-эвристик планирования заданий. Авторы также отмечают, что задача планирования в общем случае является NP-полной. Несмотря на то, что статья посвящена вычислительным Грид, которые имеют значимые отличия от Desktop Grid, она вносит значительный вклад в классификацию и разработку методов планирования заданий.

В главе книги [3] представлен обзор алгоритмов планирования в Desktop Grid и задач, стоящих перед ними. Алгоритмы разделены на три группы: «наивные», не учитывающие историю работы; основанные на информации о вычислительных узлах (включая сбор статистики работы); и адаптивные — изменяющие параметры работы «на лету». Несколько алгоритмов каждой группы обсуждаются детально, включая предложенный авторами генетический адаптивный алгоритм, схожий с представленным в работе [26] (более подробное описание этого алгоритма представлено в следующем разделе данной статьи). Также обзор дополнен детальным описанием систем имитационного моделирования BOINC SimBA, SimBOINC и EmBOINC.

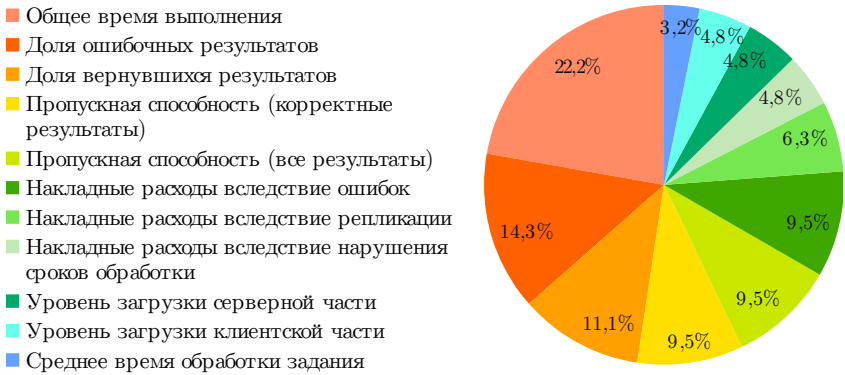

Рис. 2. Наиболее часто используемые критерии оптимизации при планировании заданий в Desktop Grid

Статья [4] посвящена обзору проблем, возникающих в проектах добровольных вычислений: измерению доступности ресурсов, выявлению шаблонов доступности, улучшению методов планирования заданий на основе шаблонов доступности. В работе также представлен обзор нескольких «наивных» и основанных на информации алгоритмов планирования заданий, методов группировки вычислительных узлов, угроз безопасности в добровольных вычислениях и мер обеспечения безопасности.

2.4. Исследовательские статьи

В данном разделе представлен обзор исследовательских работ, посвященных разработке новых алгоритмов планирования заданий в Desktop Grid. Все рассмотренные статьи акцентируются на повышении производительности Desktop Grid путем модификации базового планировщика заданий BOINC. На рис. 2 и в таблице 1 представлена сводка наиболее часто используемых критериев и методов оптимизации планирования заданий.

Для повышения производительности Desktop Grid часто используется информация о временной доступности вычислительных узлов или возможность ее предсказания. В работе [27] предложен «жадный» алгоритм планирования и обоснована его оптимальность относительно времени обработки заданий проекта. В основу алгоритма положена информация об интервалах доступности узлов; в работе также проводится тестирование алгоритма на данных реального BOINC-проекта.

Таблица 1. Критерии оптимизации, используемые в литературе при планировании заданий в Desktop Grid

Критерий оптимизации	Статьи
Пропускная способность:	
— все результаты	[13, 29–33]
— корректные результаты	[19, 26, 34–37]
Общее время выполнения	[10, 21, 22, 27, 28, 38–50]
Среднее время обработки задания	[14, 23, 51, 52]
Доля ошибочных результатов	[22, 23, 30, 32, 39, 49, 50, 53, 54]
Доля вернувшихся результатов	[10, 13, 28, 31, 42, 53, 55]
Накладные расходы:	
— вследствие репликации	[26, 54, 56, 57]
— вследствие нарушения сроков обработки	[15, 24, 58]
— вследствие ошибок	[23, 41, 42, 46, 47, 52]
Уровень загрузки:	
— клиентской части	[15, 24]
— серверной части	[15, 29, 59]

В статье [28] предложен следущий планировщик заданий, основанный на марковской цепи: сутки разделены на 48 получасовых интервалов, в течение которых каждый вычислительный узел может работать, простаивать или быть выключен. Указанные получасовые интервалы образуют состояния марковской цепи. В работе рассматриваются три схемы: «оптимистичная» без установления конечного срока завершения заданий, «пессимистичная» со строгими конечными сроками и «реалистичная» как комбинация двух предыдущих. Производительность Desktop Grid при реализации такого метода планирования проверяется экспериментально, а также обосновывается аналитически при предположении гиперэкспоненциального распределения интервалов доступности вычислительных узлов.

Авторы работы [47] решают задачу планирования при известных интервалах доступности одинаковых вычислительных узлов. При этом, неопределенность заключается в возможном сдвиге интервала доступности. Предложенный подход оценивается аналитически (включая оценку худшего случая), а также тестируется посредством имитационного моделирования.

В целом, надежность вычислительных узлов определяется не только временными интервалами их доступности. Поэтому в ряде статей исследованы более сложные модели доступности и надежности

узлов. Так, пороги надежности и доступности используются в [36] для повышения пропускной способности; этот подход проверяется с помощью эмулятора EmBOINC.

Статья [35] основана на данных проекта добровольных вычислений Docking@Home. С помощью системы имитационного моделирования SimBA авторы демонстрируют, что адаптивный выбор вычислительных узлов при назначении заданий может улучшить пропускную способность Desktop Grid. При этом, сравниваются три политики обслуживания: «первый пришел — первый ушел» (FIFO), фиксированные пороги и динамические пороги.

В [32, 53] и [30] предлагается метод планирования заданий в Desktop Grid на базе BOINC, устойчивый к саботажу. Метод основан на голосовании, адаптивный, с учетом надежности узлов (называемой «credibility»). Для снижения числа заданий, не завершенных в установленные сроки, используется группировка узлов согласно их ожидаемой надежности. Предполагается, что ошибки снижают производительность системы, в основном, из-за накладных расходов репликации. Похожая идея развивается в работе [58]: учитывается статистика надежности и доверия к узлу для снижения потерь времени из-за нарушения сроков возврата заданий.

Авторы [41] разрабатывают политику планирования, направленную на минимизацию разницы между реальным временем завершения проекта и временем, которое бы понадобилось при постоянной доступности всех узлов. Для назначения заданий используется «репутация», отражающая вычислительную мощность узла.

Механизм репликации заданий является отдельной значимой областью исследований и служит для достижения множества целей. В первую очередь, репликация может применяться для обеспечения желаемого уровня корректных результатов. Так, в работе [54] рассматривается проблема саботажа, когда вычислительные узлы могут преднамеренно возвращать неправильные результаты. Задача решается в предположении, что пространство ответов достаточно велико, чтобы исключить непреднамеренное совпадение неправильных результатов. Целью работы является нахождение уровня репликации, обеспечивающего, с одной стороны, низкие накладные расходы, а с другой — низкую вероятность принятия сервером неправильного результата.

Авторы [34, 37] определяют надежность вычислительного узла, как его вероятность вернуть результат, не нарушив срока обработки. Уровень надежности вычисляется на основе числа полученных правильных ответов. При этом, правильность ответа определяется

голосованием группы узлов, решающих одинаковое задание. Размер группы зависит от надежности составляющих ее членов. В статьях предложены и протестированы эвристики формирования групп.

В работе [60] доказано, что по критерию минимизации времени выполнения, репликация целесообразна только в «хвосте» вычислений, то есть на том этапе, когда число заданий становится меньше числа вычислительных узлов (последний этап вычислений в рамках проекта). Оптимизация «хвоста» анализируется в работе [38]. Предложены три стратегии задействования свободных ресурсов с целью минимизации общего времени обработки, учитывающие производительность узлов; стратегии репликации проверены экспериментально. Автор [50] определяет оптимальный уровень репликации и оптимальные размеры вычислительных заданий для минимизации времени выполнения «хвоста» и сокращения числа ошибок.

Модель PULL назначения заданий описана в [55] и нацелена на минимизацию числа нарушений сроков возврата результатов. Для предложенного алгоритма доказан нулевой уровень числа таких нарушений при полной надежности вычислительных узлов. Для случая ненадежных узлов и случайных запросов от сервера используется репликация для повышения общей надежности системы.

В работе [56] предложена группировка узлов в соответствии с их производительностью и историей работы. При этом, задания, назначаемые ненадежным узлам, дублируются.

Для планирования заданий в гетерогенной вычислительной системе в [51] предлагается группировка заданий по сложности одновременно с группировкой узлов по вычислительной мощности; затем решается задача линейной оптимизации для нахождения максимальной пропускной способности без накопления заданий и с предоставлением достаточной вычислительной мощности каждой группе заданий. Вводится дополнительное ограничение, которое устраняет нулевые компоненты в решении, делает метод более стабильным и практически полезным. Похожий подход используется в работе [44]: вычислительные узлы и задания группируются по динамически вычисляемым характеристикам производительности и сложности соответственно; показано, что подход позволяет сократить общее время расчетов и последний этап, когда число заданий меньше числа узлов.

Группировка вычислительных узлов в соответствии с их атрибутами используется в [4]: классификация ресурсов проводится на основе взвешенной суммы набора характеристик. Авторы [42] проводят кластеризацию узлов по их доступности и надежности, где под последней понимается вероятность завершения задания в срок.

Очередное задание назначается лучшему свободному узлу в группе с лучшей доступностью.

В работе [43] используется репликация заданий, группировка и штрафы за ошибки для снижения времени обработки вычислительного проекта в Desktop Grid. В статье [52] вычисляется оптимальный уровень репликации, минимизирующий ожидаемые расходы (состоящие из среднего времени выполнения задания и штрафов за неправильные ответы). Приложение модели [52] для управления заданиями в Enterprise Desktop Grid описывается в работе [23].

В статье [46] используются контрольные точки для наилучшего планирования заданий (недосчитанное задание можно продолжить с последней контрольной точки на любом свободном вычислительном узле). Предложен алгоритм, являющийся оптимальным (с точки зрения минимальных потерь времени) для любого наперед заданного вероятностного распределения периодов недоступности узлов.

В работе [33] предлагается модель на основе математической игры заполнения и алгоритм управления заданиями на ее основе. Игроками являются вычислительные узлы, а стратегиями выбора — подмножества вычислительных заданий для расчетов. Целью алгоритма планирования заданий, в приложении к виртуальному скринингу лекарств, является обеспечение максимально разнообразного множества результатов за ограниченное время и обеспечение эффективной скорости роста множества полезных результатов.

В ряде статей предлагаются модификации генетических алгоритмов планирования заданий в Desktop Grid. Генетические алгоритмы решают оптимизационную задачу, имитируя естественный отбор и эволюцию наборов оптимизируемых параметров — «индивидуумов». Лучшие наборы значений (относительно некоторой функции полезности) производят больше «потомков». «Мутации», т.е. изменения параметров, обеспечивают поиск в окрестности удачных наборов параметров. Генетические алгоритмы различаются механизмами мутаций, способом производства «потомков» и механизмами «конкуренции» между индивидуумами и «естественного отбора».

В работе [26] предложен генетический алгоритм для нахождения оптимальной политики планирования заданий. Роль индивидуумов играют наборы правил назначения заданий. Правила описываются на языке формальной грамматики и могут содержать неравенства, логические и арифметические операции над числами, а также предопределенные переменные, такие как доступность узлов, производительность и т.д. Результат работы алгоритма — политика планирования, т.е. набор правил для назначения задания вычислительному узлу. Похожий

подход используется в [3], где отбор и скрещивание применяется для отбора лучших параметров: лучшие значения переходят в новое поколение без мутации, а пара отдельных параметров производят двух «потомков».

В статьях [39] и [22] предложен и исследуется прогнозный генетический алгоритм для отображения заданий на узлы с целью улучшения надежности и времени обработки.

Авторы [45] решают задачу построения наилучшего плана назначения заданий с помощью собственного генетического алгоритма. При этом предполагается зависимость между приложениями и большой размер «поколения»: любая ошибка приводит к замедлению развития поколения, что влечет задержку в генерации нового поколения. Наилучшие «индивидуумы» назначаются наилучшим узлам. В работе [57] исследуется похожая проблема. В частности, в следующее поколение переходят только перспективные индивидуумы — те, которые обещают улучшение поколения.

3. Заключение

Планирование заданий имеет большое значение для Desktop Grid, поскольку значительно влияет на производительность вычислительной системы. Поэтому эта область исследований привлекает пристальное внимание ученых, которые предлагают новые подходы, методы и критерии оптимизации производительности Desktop Grid с помощью специальных политик, алгоритмов и математических моделей планирования заданий.

В данной работе представлен обзор современного состояния исследований в области планирования заданий в Desktop Grid. Нами было изучено более ста научных работ по указанной тематике за период с 1999 по 2017 годы. Из этого списка было выделено 49 статей, на основе более глубокого анализа которых составлен список используемых критериев оптимизации и методов повышения эффективности планирования заданий.

Представленный обзор может стать основой для разработки новых методов планирования заданий в Desktop Grid.

Список литературы

[1] S. J. Choi, H. S. Kim, E. J. Byun, C. S. Hwan. *A taxonomy of desktop grid systems focusing on scheduling*, Technical report KU-CSE-2006-1120-02, Department of Computer Science and Engeering, Korea University, 2006. ↑ 4,11

[2] F. Xhafa, A. Abraham. “Computational models and heuristic methods for grid scheduling problems”, *Future Generation Computer Systems*, **26**:4 (2010), pp. 608–621. ↑ [4,9,12]

[3] T. Estrada, M. Taufer. “Challenges in designing scheduling policies in volunteer computing”, *Desktop Grid Computing*, eds. C. Cérin, G. Fedak, CRC Press, 2012, pp. 167–190. ↑ [4,8,11,12,18]

[4] N. M. Durrani, J. A. Shamsi. “Volunteer computing: requirements, challenges, and solutions”, *Journal of Network and Computer Applications*, **39** (2014), pp. 369–380. ↑ [4,13,16]

[5] А. П. Афанасьев, И. В. Бычков, О. С. Заикин, М. О. Манзюк, М. А. Посыпкин, А. А. Семенов. «Концепция многозадачной грид-системы с гибким распределением свободных вычислительных ресурсов суперкомпьютеров», *Известия Российской академии наук. Теория и системы управления*, **4** (2017) (в печати). ↑ [5]

[6] A. Afanasiev, Y. Evtushenko, M. Posypkin. “The layered software infrastructure for solving large-scale optimization problems on the grid”, *International Journal of Computer Research*, **18**:3/4 (2011), pp. 307. ↑ [5]

[7] D. Thain, T. Tannenbaum, M. Livny. “Distributed computing in practice: the Condor experience”, *Concurrency-Practice and Experience*, **17**:2–4 (2005), pp. 323–356. ↑ [5]

[8] М. А. Посыпкин, В. А. Сухомлин, Н. П. Храпов. «Комбинированные распределенные инфраструктуры в науке и образовании», *Современные информационные технологии и ИТ-образование*, **1**:11 (2015), с. 31–36. ↑ [5]

[9] J. Rius, F. Cores, F. Solsona. “Cooperative scheduling mechanism for large-scale peer-to-peer computing systems”, *Journal of Network and Computer Applications*, **36**:6 (2013), pp. 1620–1631. ↑ [6]

[10] Y. Ch. Lee, A. Y. Zomaya, H. J. Siegel. “Robust task scheduling for volunteer computing systems”, *Journal of Supercomputing*, **53**:1 (2010), pp. 163–181. ↑ [6,10,14]

[11] D. P Anderson. “BOINC: A system for public-resource computing and storage”, *Grid Computing*, Proceedings Fifth IEEE/ACM International Workshop on IEEE, 2004, pp. 4–10. ↑ [6]

[12] BOINCstats (accessed Mar 2017), URL: `https://boincstats.com` ↑ [6]

[13] J. Atlas, T. Estrada, K. Decker, M. Taufer. “Balancing scientist needs and volunteer preferences in volunteer computing using constraint optimization”, *Computational Science*, ICCS 2009, Springer, 2009, pp. 143–152. ↑ [7,14]

[14] C. Anglano, M. Canonico. “Scheduling algorithms for multiple bag-of-task applications on desktop grids: A knowledge-free approach”, *IEEE International Symposium on Parallel and Distributed Processing*, IPDPS 2008, IEEE, 2008, pp. 1–8. ↑ [7,14]

[15] D. P. Anderson. “Emulating volunteer computing scheduling policies”, *IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, IPDPSW 2011, IEEE, 2011, pp. 1839–1846. ↑ [8,14]

[16] O. Beaumont, L. Bobelin, H. Casanova, P.-N. Clauss, B. Donassolo, L. Eyraud-Dubois, S. Genaud, S. Hunold, A. Legrand, M. Quinson. *Towards scalable, accurate, and usable simulations of distributed applications and systems*, Research Report RR-7761, Institut National de Recherche en Informatique et en Automatique, 2011. ↑ 8

[17] T. Estrada, M. Taufer, D. P. Anderson. "Performance prediction and analysis of BOINC projects: An empirical study with EmBOINC", *Journal of Grid Computing*, **7**:4 (2009), pp. 537–554. ↑ 8

[18] T. Estrada, M. Taufer, K. Reed, D. P. Anderson. "EmBOINC: An emulator for performance analysis of BOINC projects", *IEEE International Symposium on Parallel & Distributed Processing*, IPDPS 2009, IEEE, 2009, pp. 1–8. ↑ 8

[19] M. Taufer, A. Kerstens, T. Estrada, D. A. Flores, P. J. Teller. "SimBA: A discrete event simulator for performance prediction of volunteer computing projects", *21st International Workshop on Principles of Advanced and Distributed Simulation*. V. 7, 2007, pp. 189–197. ↑ 8,11,14

[20] H. Casanova, F. Dufossé, Y. Robert, F. Vivien. "Scheduling parallel iterative applications on volatile resources", *2011 IEEE International Parallel & Distributed Processing Symposium*, IEEE, 2011, pp. 1012–1023. ↑ 9

[21] M. Maheswaran, Sh. Ali, H. J. Siegel, D. Hensgen, R. F. Freund. "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems", *Journal of Parallel and Distributed Computing*, **59**:2 (1999), pp. 107–131. ↑ 9,14

[22] X. Wang, Ch. Sh. Yeo, R. Buyya, J. Su. "Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm", *Future Generation Computer Systems*, **27**:8 (2011), pp. 1124–1134. ↑ 9,14,18

[23] I. A. Chernov, N. N. Nikitina. "Virtual screening in a desktop grid: Replication and the optimal quorum", *International Conference on Parallel Computing Technologies*, Lecture Notes in Computer Science, vol. **9251**, ed. V. Malyshkin, Springer, 2015, pp. 258–267. ↑ 10,14,17

[24] D. Kondo, D. P. Anderson, J. McLeod VII. "Performance evaluation of scheduling policies for volunteer computing", *Third IEEE International Conference on E-Science and Grid Computing*, IEEE, 2007, pp. 221–227. ↑ 10,14

[25] M. Kh. Khan, T. Mahmood, S. I. Hyder. "Scheduling in desktop grid systems: Theoretical evaluation of policies and frameworks", *International Journal of Advanced Computer Science and Applications*, **8**:1 (2017), pp. 119–127. ↑ 11

[26] T. Estrada, O. Fuentes, M. Taufer. "A distributed evolutionary method to design scheduling policies for volunteer computing", *ACM SIGMETRICS Performance Evaluation Review*, **36**:3 (2008), pp. 40–49. ↑ 12,14,17

[27] D. Kondo, H. Casanova. *Computing the optimal makespan for jobs with identical and independent tasks scheduled on volatile hosts*, Technical Report CS2004-0796, Dept. of Computer Science and Engineering, University of California, 2004. ↑ 13,14

[28] E. J. Byun, S. J. Choi, M. S. Baik, J. M. Gil, C. Y. Park, C. S. Hwang. "MJSA: Markov job scheduler based on availability in desktop grid computing environment", *Future Generation Computer Systems*, **23** (2007), pp. 616–622. ↑ 14

[29] V. V. Mazalov, N. N. Nikitina, E. E. Ivashko. "Task scheduling in a desktop grid to minimize the server load", *International Conference on Parallel Computing Technologies*, Lecture Notes in Computer Science, vol. **9251**, ed. V. Malyshkin, Springer, 2015, pp. 273–278. ↑ 14

[30] Sh. Yasuda, Ya. Nogami, M. Fukushi. "A dynamic job scheduling method for reliable and high-performance volunteer computing", *2nd International Conference on Information Science and Security*, ICISS 2015, IEEE, 2015, pp. 1–4. ↑ 14,15

[31] N. M. Durrani, J. A. Shamsi, S. Fatima. "Towards efficient resource grouping in heterogeneity-aware volunteer computing", *Sindh University Research Journal*, 2013, pp. 1–11. ↑ 14

[32] K. Watanabe, M. Fukushi, M. Kameyama. "Adaptive group-based job scheduling for high performance and reliable volunteer computing", *Journal of Information Processing*, **19** (2011), pp. 39–51. ↑ 14,15

[33] Н. Н. Никитина, Е. Е. Ивашко. «Цена анархии в игре заполнения для поиска лекарств», *Вероятностные методы в дискретной математике*, расширенные тезисы, IX Международная Петрозаводская конференция (30 мая–3 июня 2016 г., Петрозаводск, Россия), 2016, с. 67–69. ↑ 14,17

[34] J. Sonnek, A. Chandra, J. Weissman. "Adaptive reputation-based scheduling on unreliable distributed infrastructures", *IEEE Transactions on Parallel and Distributed Systems*, **18**:11 (2007), pp. 1551–1564. ↑ 14,15

[35] M. Taufer, A. Kerstens, T. P. Estrada, D. A. Flores, R. Zamudio, P. J. Teller, R. Armen, C. L. Brooks III. "Moving volunteer computing towards knowledge-constructed, dynamically-adaptive modeling and scheduling", *IEEE International Symposium on Parallel and Distributed Processing*, IPDPS 2007, IEEE, 2007, pp. 1–8. ↑ 14,15

[36] T. Estrada, D. A. Flores, M. Taufer, P. J. Teller, A. Kerstens, D. P. Anderson. "The effectiveness of threshold-based scheduling policies in BOINC projects", *Second IEEE International Conference on E-Science and Grid Computing*, IEEE, 2006, pp. 88–88. ↑ 14,15

[37] J. Sonnek, M. Nathan, A. Chandra, J. Weissman. "Reputation-based scheduling on unreliable distributed infrastructures", *26th IEEE International Conference on Distributed Computing Systems*, ICDCS 2006, IEEE, 2006, pp. 30–30. ↑ 14,15

[38] D. Kondo, A. A. Chien, H. Casanova. “Scheduling task parallel applications for rapid turnaround on enterprise desktop grids”, *Journal of Grid Computing*, **5**:4 (2007), pp. 379–405. ↑ 14,16

[39] B. Qu, Y. Lei, Y. Zhao. “A new genetic algorithm based scheduling for volunteer computing”, *International Conference on Computer and Communication Technologies in Agriculture Engineering*. V. 3, CCTAE 2010, IEEE, 2010, pp. 228–231. ↑ 14,18

[40] M. O. Alieksieiev, O. V. Maiboroda, A. O. Onysko, O. M. Alekseev. “Repeating tasks scheduling in desktop grid systems”, *24th International Crimean Conference on Microwave & Telecommunication Technology*, CriMiCo 2014, IEEE, 2014, pp. 342–343. ↑ 14

[41] A. Essafi, D. Trystram, Z. Zaidi. “An efficient algorithm for scheduling jobs in volunteer computing platforms”, *28th International Parallel & Distributed Processing Symposium Workshops*, IEEE, 2014, pp. 68–76. ↑ 14,15

[42] J.-M. Gil, S. Kim, J. Lee. “Task scheduling scheme based on resource clustering in desktop grids”, *International Journal of Communication Systems*, **27**:6 (2014), pp. 918–930. ↑ 14,16

[43] А. С. Румянцев. «Задача оптимизации времени выполнения проекта в вычислительной сети из персональных компьютеров», *Программные системы: теория и приложения*, **5**:1 (2014), с. 175–182, URL: http://psta.psiras.ru/read/psta2014_1_175-182.pdf ↑ 14,17

[44] M. Ujhelyi, P. Lacko, A. Paulovic. “Task scheduling in distributed volunteer computing systems”, *IEEE 12th International Symposium on Intelligent Systems and Informatics*, SISY 2014, IEEE, 2014, pp. 111–114. ↑ 14,16

[45] M. Smaoui, M. Garbey. “Improving volunteer computing scheduling for evolutionary algorithms”, *Future Generation Computer Systems*, **29**:1 (2013), pp. 1–14. ↑ 14,18

[46] M. S. Bouguerra, D. Kondo, D. Trystram. “On the scheduling of checkpoints in desktop grids”, *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGrid 2011, 2011, pp. 305–313. ↑ 14,17

[47] L.-C. Canon, A. Essafi, G. Mounié, D. Trystram. “A bi-objective scheduling algorithm for desktop grids with uncertain resource availabilities”, *Parallel Processing*, Euro-Par 2011, Springer, 2011, pp. 238–249. ↑ 14

[48] J. Celaya, L. Marchal. “A fair decentralized scheduler for bag-of-tasks applications on desktop grids”, *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE, 2010, pp. 538–541. ↑ 14

[49] L. Gao, G. Malewicz. “Toward maximizing the quality of results of dependent tasks computed unreliably”, *Theory of Computing Systems*, **41**:4 (2007), pp. 731–752. ↑ 14

[50] I. Kurochkin. “Determination of replication parameters in the project of the voluntary distributed computing NetMaxhome”, International Scientific Conference ”High technologies. Business. Society” (14–17.03.2016, Borovets, Bulgaria), pp. 10–12. ↑ 14,16

[51] I. Al-Azzoni, D. G. Down. “Dynamic scheduling for heterogeneous desktop grids”, *Journal of Parallel and Distributed Computing*, **70**:12 (2010), pp. 1231–1240. ↑ 14,16

[52] И. А. Чернов. «Оптимальное дублирование заданий в вычислительной системе», *Труды Карельского научного центра Российской академии наук*, 2014, №4, с. 130–136. ↑ 14,17

[53] Yu. Miyakoshi, K. Watanabe, M. Fukushi, Ya. Nogami. “A job scheduling method based on expected probability of completion of voting in volunteer computing”, *2014 Second International Symposium on Computing and Networking*, IEEE, 2014, pp. 399–405. ↑ 14,15

[54] L.-C. Canon, E. Jeannot, J. Weissman. “A scheduling and certification algorithm for defeating collusion in desktop grids”, *2011 31st International Conference on Distributed Computing Systems*, IEEE, 2011, pp. 343–352. ↑ 14,15

[55] E. M. Heien, D. P. Anderson, K. Hagihara. “Computing low latency batches with unreliable workers in volunteer computing environments”, *Journal of Grid Computing*, **7**:4 (2009), pp. 501–518. ↑ 14,16

[56] M. Kh. Khan, S. I. Hyder, Gh. U. Ahmed, S. Begum, M. Aamir. “A group based replication mechanism to reduce the wastage of processing cycles in volunteer computing”, *Wireless Personal Communications*, **76**:3 (2014), pp. 591–601. ↑ 14,16

[57] T. Desell, M. Magdon-Ismail, B. Szymanski, C. Varela, H. Newberg, N. Cole. “Robust asynchronous optimization for volunteer computing grids”, *Fifth IEEE International Conference on E-Science*, IEEE, 2009, pp. 263–270. ↑ 14,18

[58] L. Klejnowski, Y. Bernard, C. Muller-Schloer, J. Hahner. “Using trust to reduce wasteful computation in open desktop grid systems”, *Tenth Annual International Conference on Privacy, Security and Trust*, PST 2012, IEEE, 2012, pp. 250–255. ↑ 14,15

[59] V. V. Mazalov, N. N. Nikitina, E. E. Ivashko. “Hierarchical two-level game model for tasks scheduling in a desktop grid”, *6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, ICUMT 2014, IEEE, 2014, pp. 541–545. ↑ 14

[60] G. D. Ghare, S. T. Leutenegger. “Improving speedup and response times by replicating parallel programs on a SNOW”, *Workshop on Job Scheduling Strategies for Parallel Processing*, Springer, 2004, pp. 264–287. ↑ 16

Пример ссылки на эту публикацию:

И. А. Чернов, Е. Е. Ивашко, Н. Н. Никитина. «Обзор методов планирования заданий в Desktop Grid», *Программные системы: теория и приложения*, 2017, **8**:3(34), с. 3–29.

URL: http://psta.psiras.ru/read/psta2017_3_3-29.pdf

Об авторах:

Илья Александрович Чернов

Кандидат физико-математических наук, старший научный сотрудник ИПМИ КарНЦ РАН. Доцент Кафедры математического анализа ПетрГУ. Эксперт в области математического моделирования физических процессов и краевых задач математической физики. Область научных интересов: вычислительная математика, моделирование разложения гидридов металлов, численное моделирование крупномасштабной динамики моря.

e-mail: chernov@krc.karelia.ru

Евгений Евгеньевич Ивашко

Кандидат физико-математических наук, старший научный сотрудник ИПМИ КарНЦ РАН. Руководитель ЦКП КарНЦ РАН «Центр высокопроизводительной обработки данных». Эксперт Фонда Сколково. Область научных интересов: математическая теория игр, высокопроизводительные и распределенные вычисления, Desktop Grid.

e-mail: ivashko@krc.karelia.ru

Наталия Николаевна Никитина

Кандидат технических наук, младший научный сотрудник ИПМИ КарНЦ РАН. Область научных интересов: высокопроизводительные и распределенные вычисления, планирование заданий, Desktop Grid, приложения математической теории игр.

e-mail: nikitina@krc.karelia.ru

Ilya Chernov, Evgeny Ivashko, Natalia Nikitina. *A survey of task scheduling in Desktop Grid.*

Abstract. The paper surveys the state of the art of research related to task scheduling in Desktop Grid computing systems. We overview scientific papers dated from 1999 to 2017 to analyze the optimization criteria and methods proposed by researchers for improving Desktop Grid task scheduling. (*In Russian*).

Key words and phrases: Desktop Grid, task scheduling, BOINC, Enterprise Desktop Grid, high-throughput computing, distributed computing.

References

[1] S. J. Choi, H. S. Kim, E. J. Byun, C. S. Hwan. *A taxonomy of desktop grid systems focusing on scheduling*, Technical report KU-CSE-2006-1120-02, Department of Computer Science and Engeering, Korea University, 2006.

[2] F. Xhafa, A. Abraham. "Computational models and heuristic methods for grid scheduling problems", *Future Generation Computer Systems*, **26**:4 (2010), pp. 608–621.

[3] T. Estrada, M. Taufer. "Challenges in designing scheduling policies in volunteer computing", *Desktop Grid Computing*, eds. C. Cérin, G. Fedak, CRC Press, 2012, pp. 167–190.

[4] N. M. Durrani, J. A. Shamsi. "Volunteer computing: requirements, challenges, and solutions", *Journal of Network and Computer Applications*, **39** (2014), pp. 369–380.

[5] A. P. Afanas'yev, I. V. Bychkov, O. S. Zaikin, M. O. Manzyuk, M. A. Posypkin, A. A. Semenov. "A concept of multi-tasking Grid system with flexible distribution of idle computational resources of supercomputers", *Journal of Computer and Systems Sciences International*, 2017 (to appear).

[6] A. Afanasiev, Y. Evtushenko, M. Posypkin. "The layered software infrastructure for solving large-scale optimization problems on the grid", *International Journal of Computer Research*, **18**:3/4 (2011), pp. 307.

[7] D. Thain, T. Tannenbaum, M. Livny. "Distributed computing in practice: the Condor experience", *Concurrency-Practice and Experience*, **17**:2–4 (2005), pp. 323–356.

[8] M. A. Posypkin, V. A. Sukhomlin, N. P. Khrapov. "Combined Distributed Infrastructures in Science and Education", *Sovremennyye informatsionnyye tekhnologii i IT-obrazovaniye*, **1**:11 (2015), pp. 31–36 (in Russian).

[9] J. Rius, F. Cores, F. Solsona. "Cooperative scheduling mechanism for large-scale peer-to-peer computing systems", *Journal of Network and Computer Applications*, **36**:6 (2013), pp. 1620–1631.

[10] Y. Ch. Lee, A. Y. Zomaya, H. J. Siegel. "Robust task scheduling for volunteer computing systems", *Journal of Supercomputing*, **53**:1 (2010), pp. 163–181.

© I. A. Chernov, E. E. Ivashko, N. N. Nikitina, 2017
© Institute of Applied Mathematical Research KarRC of RAS, 2017
© Program systems: Theory and Applications, 2017

DOI: 10.25209/2079-3316-2017-8-3-3-29

[11] D. P Anderson. “BOINC: A system for public-resource computing and storage”, *Grid Computing*, Proceedings Fifth IEEE/ACM International Workshop on IEEE, 2004, pp. 4–10.

[12] BOINCstats (accessed Mar 2017), URL: https://boincstats.com

[13] J. Atlas, T. Estrada, K. Decker, M. Taufer. “Balancing scientist needs and volunteer preferences in volunteer computing using constraint optimization”, *Computational Science*, ICCS 2009, Springer, 2009, pp. 143–152.

[14] C. Anglano, M. Canonico. “Scheduling algorithms for multiple bag-of-task applications on desktop grids: A knowledge-free approach”, *IEEE International Symposium on Parallel and Distributed Processing*, IPDPS 2008, IEEE, 2008, pp. 1–8.

[15] D. P. Anderson. “Emulating volunteer computing scheduling policies”, *IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, IPDPSW 2011, IEEE, 2011, pp. 1839–1846.

[16] O. Beaumont, L. Bobelin, H. Casanova, P.-N. Clauss, B. Donassolo, L. Eyraud-Dubois, S. Genaud, S. Hunold, A. Legrand, M. Quinson. *Towards scalable, accurate, and usable simulations of distributed applications and systems*, Research Report RR-7761, Institut National de Recherche en Informatique et en Automatique, 2011.

[17] T. Estrada, M. Taufer, D. P. Anderson. “Performance prediction and analysis of BOINC projects: An empirical study with EmBOINC”, *Journal of Grid Computing*, **7**:4 (2009), pp. 537–554.

[18] T. Estrada, M. Taufer, K. Reed, D. P. Anderson. “EmBOINC: An emulator for performance analysis of BOINC projects”, *IEEE International Symposium on Parallel & Distributed Processing*, IPDPS 2009, IEEE, 2009, pp. 1–8.

[19] M. Taufer, A. Kerstens, T. Estrada, D. A. Flores, P. J. Teller. “SimBA: A discrete event simulator for performance prediction of volunteer computing projects”, *21st International Workshop on Principles of Advanced and Distributed Simulation*. V. 7, 2007, pp. 189–197.

[20] H. Casanova, F. Dufossé, Y. Robert, F. Vivien. “Scheduling parallel iterative applications on volatile resources”, *2011 IEEE International Parallel & Distributed Processing Symposium*, IEEE, 2011, pp. 1012–1023.

[21] M. Maheswaran, Sh. Ali, H. J. Siegel, D. Hensgen, R. F. Freund. “Dynamic mapping of a class of independent tasks onto heterogeneous computing systems”, *Journal of Parallel and Distributed Computing*, **59**:2 (1999), pp. 107–131.

[22] X. Wang, Ch. Sh. Yeo, R. Buyya, J. Su. “Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm”, *Future Generation Computer Systems*, **27**:8 (2011), pp. 1124–1134.

[23] I. A. Chernov, N. N. Nikitina. “Virtual screening in a desktop grid: Replication and the optimal quorum”, *International Conference on Parallel Computing Technologies*, Lecture Notes in Computer Science, vol. **9251**, ed. V. Malyshkin, Springer, 2015, pp. 258–267.

[24] D. Kondo, D. P. Anderson, J. McLeod VII. “Performance evaluation of scheduling policies for volunteer computing”, *Third IEEE International Conference on E-Science and Grid Computing*, IEEE, 2007, pp. 221–227.

[25] M. Kh. Khan, T. Mahmood, S. I. Hyder. “Scheduling in desktop grid systems: Theoretical evaluation of policies and frameworks”, *International Journal of Advanced Computer Science and Applications*, **8**:1 (2017), pp. 119–127.

[26] T. Estrada, O. Fuentes, M. Taufer. “A distributed evolutionary method to design scheduling policies for volunteer computing”, *ACM SIGMETRICS Performance Evaluation Review*, **36**:3 (2008), pp. 40–49.

[27] D. Kondo, H. Casanova. *Computing the optimal makespan for jobs with identical and independent tasks scheduled on volatile hosts*, Technical Report CS2004-0796, Dept. of Computer Science and Engineering, University of California, 2004.

[28] E. J. Byun, S. J. Choi, M. S. Baik, J. M. Gil, C. Y. Park, C. S. Hwang. “MJSA: Markov job scheduler based on availability in desktop grid computing environment”, *Future Generation Computer Systems*, **23** (2007), pp. 616–622.

[29] V. V. Mazalov, N. N. Nikitina, E. E. Ivashko. “Task scheduling in a desktop grid to minimize the server load”, *International Conference on Parallel Computing Technologies*, Lecture Notes in Computer Science, vol. **9251**, ed. V. Malyshkin, Springer, 2015, pp. 273–278.

[30] Sh. Yasuda, Ya. Nogami, M. Fukushi. “A dynamic job scheduling method for reliable and high-performance volunteer computing”, *2nd International Conference on Information Science and Security*, ICISS 2015, IEEE, 2015, pp. 1–4.

[31] N. M. Durrani, J. A. Shamsi, S. Fatima. “Towards efficient resource grouping in heterogeneity-aware volunteer computing”, *Sindh University Research Journal*, 2013, pp. 1–11.

[32] K. Watanabe, M. Fukushi, M. Kameyama. “Adaptive group-based job scheduling for high performance and reliable volunteer computing”, *Journal of Information Processing*, **19** (2011), pp. 39–51.

[33] N. N. Nikitina, Ye. Ye. Ivashko. “The Price of Anarchy in a Congestion Game for Drug Discovery”, *Veroyatnostnyye metody v diskretnoy matematike*, rasshirennyye tezisy, IX Mezhdunarodnaya Petrozavodskaya konferentsiya (30 maya–3 iyunya 2016 g., Petrozavodsk, Rossiya), 2016, pp. 67–69 (in Russian).

[34] J. Sonnek, A. Chandra, J. Weissman. “Adaptive reputation-based scheduling on unreliable distributed infrastructures”, *IEEE Transactions on Parallel and Distributed Systems*, **18**:11 (2007), pp. 1551–1564.

[35] M. Taufer, A. Kerstens, T. P. Estrada, D. A. Flores, R. Zamudio, P. J. Teller, R. Armen, C. L. Brooks III. “Moving volunteer computing towards knowledge-constructed, dynamically-adaptive modeling and scheduling”, *IEEE International Symposium on Parallel and Distributed Processing*, IPDPS 2007, IEEE, 2007, pp. 1–8.

[36] T. Estrada, D. A. Flores, M. Taufer, P. J. Teller, A. Kerstens, D. P. Anderson. “The effectiveness of threshold-based scheduling policies in BOINC projects”, *Second IEEE International Conference on E-Science and Grid Computing*, IEEE, 2006, pp. 88–88.

[37] J. Sonnek, M. Nathan, A. Chandra, J. Weissman. “Reputation-based scheduling on unreliable distributed infrastructures”, *26th IEEE International Conference on Distributed Computing Systems*, ICDCS 2006, IEEE, 2006, pp. 30–30.

[38] D. Kondo, A. A. Chien, H. Casanova. “Scheduling task parallel applications for

rapid turnaround on enterprise desktop grids", *Journal of Grid Computing*, **5**:4 (2007), pp. 379–405.

[39] B. Qu, Y. Lei, Y. Zhao. "A new genetic algorithm based scheduling for volunteer computing", *International Conference on Computer and Communication Technologies in Agriculture Engineering*. V. 3, CCTAE 2010, IEEE, 2010, pp. 228–231.

[40] M. O. Alieksieiev, O. V. Maiboroda, A. O. Onysko, O. M. Alekseev. "Repeating tasks scheduling in desktop grid systems", *24th International Crimean Conference on Microwave & Telecommunication Technology*, CriMiCo 2014, IEEE, 2014, pp. 342–343.

[41] A. Essafi, D. Trystram, Z. Zaidi. "An efficient algorithm for scheduling jobs in volunteer computing platforms", *28th International Parallel & Distributed Processing Symposium Workshops*, IEEE, 2014, pp. 68–76.

[42] J.-M. Gil, S. Kim, J. Lee. "Task scheduling scheme based on resource clustering in desktop grids", *International Journal of Communication Systems*, **27**:6 (2014), pp. 918–930.

[43] A. S. Rumyantsev. "Optimizing the Execution Time of a Desktop Grid Project", *Programmnyye sistemy: teoriya i prilozheniya*, **5**:1 (2014), pp. 175–182 (in Russian), URL: http://psta.psiras.ru/read/psta2014_1_175-182.pdf

[44] M. Ujhelyi, P. Lacko, A. Paulovic. "Task scheduling in distributed volunteer computing systems", *IEEE 12th International Symposium on Intelligent Systems and Informatics*, SISY 2014, IEEE, 2014, pp. 111–114.

[45] M. Smaoui, M. Garbey. "Improving volunteer computing scheduling for evolutionary algorithms", *Future Generation Computer Systems*, **29**:1 (2013), pp. 1–14.

[46] M. S. Bouguerra, D. Kondo, D. Trystram. "On the scheduling of checkpoints in desktop grids", *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGrid 2011, 2011, pp. 305–313.

[47] L.-C. Canon, A. Essafi, G. Mounié, D. Trystram. "A bi-objective scheduling algorithm for desktop grids with uncertain resource availabilities", *Parallel Processing*, Euro-Par 2011, Springer, 2011, pp. 238–249.

[48] J. Celaya, L. Marchal. "A fair decentralized scheduler for bag-of-tasks applications on desktop grids", *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE, 2010, pp. 538–541.

[49] L. Gao, G. Malewicz. "Toward maximizing the quality of results of dependent tasks computed unreliably", *Theory of Computing Systems*, **41**:4 (2007), pp. 731–752.

[50] I. Kurochkin. "Determination of replication parameters in the project of the voluntary distributed computing NetMaxhome", International Scientific Conference "High technologies. Business. Society" (14–17.03.2016, Borovets, Bulgaria), pp. 10–12.

[51] I. Al-Azzoni, D. G. Down. "Dynamic scheduling for heterogeneous desktop grids", *Journal of Parallel and Distributed Computing*, **70**:12 (2010), pp. 1231–1240.

[52] I. A. Chernov. "Optimal Replication of Tasks in a Computing Grid", *Trudy Karel'skogo nauchnogo tsentra Rossiyskoy akademii nauk*, 2014, no.4, pp. 130–136 (in Russian).

[53] Yu. Miyakoshi, K. Watanabe, M. Fukushi, Ya. Nogami. “A job scheduling method based on expected probability of completion of voting in volunteer computing”, *2014 Second International Symposium on Computing and Networking*, IEEE, 2014, pp. 399–405.

[54] L.-C. Canon, E. Jeannot, J. Weissman. “A scheduling and certification algorithm for defeating collusion in desktop grids”, *2011 31st International Conference on Distributed Computing Systems*, IEEE, 2011, pp. 343–352.

[55] E. M. Heien, D. P. Anderson, K. Hagihara. “Computing low latency batches with unreliable workers in volunteer computing environments”, *Journal of Grid Computing*, **7**:4 (2009), pp. 501–518.

[56] M. Kh. Khan, S. I. Hyder, Gh. U. Ahmed, S. Begum, M. Aamir. “A group based replication mechanism to reduce the wastage of processing cycles in volunteer computing”, *Wireless Personal Communications*, **76**:3 (2014), pp. 591–601.

[57] T. Desell, M. Magdon-Ismail, B. Szymanski, C. Varela, H. Newberg, N. Cole. “Robust asynchronous optimization for volunteer computing grids”, *Fifth IEEE International Conference on E-Science*, IEEE, 2009, pp. 263–270.

[58] L. Klejnowski, Y. Bernard, C. Muller-Schloer, J. Hahner. “Using trust to reduce wasteful computation in open desktop grid systems”, *Tenth Annual International Conference on Privacy, Security and Trust*, PST 2012, IEEE, 2012, pp. 250–255.

[59] V. V. Mazalov, N. N. Nikitina, E. E. Ivashko. “Hierarchical two-level game model for tasks scheduling in a desktop grid”, *6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, ICUMT 2014, IEEE, 2014, pp. 541–545.

[60] G. D. Ghare, S. T. Leutenegger. “Improving speedup and response times by replicating parallel programs on a SNOW”, *Workshop on Job Scheduling Strategies for Parallel Processing*, Springer, 2004, pp. 264–287.

Sample citation of this publication:

Ilya Chernov, Evgeny Ivashko, Natalia Nikitina. “A survey of task scheduling in Desktop Grid”, *Program systems: Theory and applications*, 2017, **8**:3(34), pp. 3–29. (*In Russian*).

URL: http://psta.psiras.ru/read/psta2017_3_3-29.pdf