

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ (05.13.00)

УДК 004.75

DOI: 10.24160/1993-6982-2018-6-75-86

Модели, методы и алгоритмы планирования в грид и облачных вычислениях

В.В. Топорков, Д.М. Емельянов

Некоторые сложные задачи, например, обработка результатов физических экспериментов на ЛНС (ЦЕРН), задействуют значительные распределенные вычислительные ресурсы, часть из которых используется совместно с их владельцами. Этот фактор, даже в рамках виртуальных организаций, обуславливает конкуренцию за использование ресурсов как независимых пользователей, так и глобальных (пользовательских) и локальных потоков заданий собственников вычислительных узлов и существенно усложняет проблему требуемого качества обслуживания масштабируемых вычислений. Известные на сегодняшний день алгоритмы планирования, их комбинации и эвристики не позволяют создавать эффективные планы в условиях разнородных распределенных сред с динамично изменяющимся составом вычислительных ресурсов. В этих условиях весьма эффективными являются так называемые экономические модели выделения неотчуждаемых ресурсов и планирования распределенных вычислений в таких областях как грид, облачные вычисления, мультиагентные системы.

Проанализировано современное состояние исследований в области методов и алгоритмов эффективного планирования конкурирующих потоков структурированных, параллельных заданий в распределенных гетерогенных вычислительных средах — грид инфраструктурах и облачных сервисах. Основное внимание уделено методам и инструментальным средствам планирования сложных заданий и их потоков с учетом предпочтений стейкхолдеров (пользователей, владельцев ресурсов, администраторов), а также особенностей приложений. Учет этих факторов позволяет повысить эффективность использования ресурсов в распределенных вычислениях.

Представлено комплексное решение проблемы планирования сложных заданий и их потоков. Задание пользователя требует выделения множества одновременно доступных вычислительных ресурсов. Выбор подходящих ресурсов усложняется наличием приоритетных локальных заданий, динамикой загрузки и предварительными резервированиями. Актуальность расписания загрузки вычислительных узлов поддерживается с помощью горизонта планирования. Перепланирование «на лету» требует минимизации вычислительной сложности соответствующих алгоритмов. Конкуренция между множеством различных потоков заданий снижает уровень доступности ресурсов и уменьшает число альтернативных вариантов планирования. Один из возможных подходов решения этой проблемы — упреждающее планирование, позволяющее увеличить число возможных альтернативных планов с помощью специальных механизмов отбора слотов. Процедура упреждающего планирования дает возможность оценить эффективность различных комбинаций ресурсов. На основе полученных данных формируется допустимый план выполнения потока заданий, в наибольшей степени приближенный к оптимальному решению.

Ключевые слова: планирование, распределенные вычисления, метапланировщик, эффективность.

Для цитирования: Топорков В.В., Емельянов Д.М. Модели, методы и алгоритмы планирования в грид и облачных вычислениях // Вестник МЭИ. 2018. № 6. С. 75—86. DOI: 10.24160/1993-6982-2018-6-75-86.

Scheduling Models, Methods and Algorithms in Grid and Cloud Computing

V.V. Toporkov, D.M. Yemelyanov

Certain intricate problems, e.g., processing the results of physical experiments in the large hadron collider at the CERN, utilize significant distributed computing resources, part of which is shared with their owners. This factor, even within the framework of virtual organizations, causes competition for utilization of resources between both independent users and global (users') and local job flows of computing resource owners. This circumstance adds much difficulty to the problem of providing the required quality of service in scalable computing. The presently known

scheduling algorithms, their combinations and heuristics do not offer tools for producing efficient schedules under the conditions of heterogeneous distributed environments with a dynamically changing composition of computing resources. Under such conditions, so called economic models for allocating non-dedicated resources and for scheduling distributed computing turn out to be highly efficient in fields like grid and cloud computing, and multiagent systems.

The article analyzes the current state of investigations in the field of methods and algorithms for efficiently scheduling competing flows of structured and parallel jobs in distributed heterogeneous computing environments, namely, in grid infrastructures and in cloud services. Primary attention is paid to methods and tools for scheduling intricate jobs and their flows with due regard to the preferences of stakeholders (users, resource owners, and administrators), and specific features of applications. With these factors duly taken into account, more efficient utilization of the resources in distributed computing environments may be achieved.

The article presents an approach for comprehensively solving the problem of scheduling intricate jobs and their flows. Execution of a user job involves the need to allocate a set of simultaneously accessible computing resources (slot windows). The problem of selecting suitable resources is complicated by such factors as the availability of local high-priority jobs, the dynamics of resources utilization, and advanced reservations. The computing node load schedules are kept in line with the actual demands by means of a scheduling horizon. Online rescheduling implies the need to minimize the computational complexity of the relevant algorithms. Competition among the multitude of different job flows is a factor that degrades the level of resource accessibility and decreases the number of alternative scheduling options. One possible approach for solving this problem is to use anticipation scheduling, a method that makes it possible to increase the number of possible alternative schedules by means of special slot selection mechanisms. The anticipation scheduling procedure opens the possibility to evaluate the efficiency of different resource combinations. The obtained data serve as a basis for producing a feasible job flow execution schedule close to the optimal solution.

Key words: scheduling, distributed computing, metascheduler, effectiveness.

For citation: Toporkov V.V., Yemelyanov D.M. Scheduling Models, Methods and Algorithms in Grid and Cloud Computing. MPEI Vestnik. 2018;6:75—86. (in Russian). DOI: 10.24160/1993-6982-2018-6-75-86.

Введение

Одной из важнейших проблем организации распределенных вычислений является эффективное использование ресурсов с соблюдением соглашений о качестве сервиса (SLA). В ряде случаев фактическая загрузка ресурсов в центрах обработки данных, поддерживающих облачные вычисления, не превышает 10%. Одна из причин — эффект избыточного резервирования ресурсов (*over-provisioning*), что парадоксально в условиях высокого уровня виртуализации. Проблема заключается в неэффективном планировании и управлении ресурсами.

Традиционно в распределенных вычислениях выделяют и, как правило, разделяют планирование на уровне приложений (*application-level scheduling*) и потоков независимых заданий (*job-flow scheduling*). В грид первый подход ассоциируется с брокер-ориентированной моделью, а второй — с использованием метапланировщиков в рамках виртуальных организаций (ВО). Это деление весьма условно. Многочисленные исследования связаны с планированием «портфеля задач» (*bag-of-tasks*), относительно простых и слабо связанных между собой. Брокеры ресурсов адаптируются к особенностям приложений с учетом их структуры, в то время как политика ВО регламентирует взаимодействие между ключевыми стейкхолдерами (пользователями, администраторами и владельцами ресурсов) на основе экономических моделей. Тем не менее, в рамках обоих подходов следует учитывать конкуренцию потоков независимых заданий, которая приводит к снижению уровня доступности разделяемых ресурсов. При этом даже в ВО предпочтения стейкхолдеров противоречивы.

Важнейший класс структурированных заданий, состоящих из связанных задач, представляют научные

приложения типа потока работ (*workflow*). Примеры таких проектов — Montage, Epigenomics, CyberShake, SIPHT, LIGO. Каждое из заданий задействует значительные разнородные и зачастую неотчуждаемые от владельцев ресурсы, используемые также локальными заданиями. Модель планирования задания — бесконтурный оргграф (DAG), вершины которого соответствуют задачам, а дуги — информационным и логическим связям. Известно, что проблема планирования в рамках этой модели является NP-полной даже для однородных вычислительных сред. Наличие же конкурирующих потоков независимых заданий и гетерогенность среды существенно усложняют вопросы эффективного планирования и разделения ресурсов.

Настоящая статья представляет собой аналитический обзор состояния работ по планированию потоков заданий и структурированных приложений в распределенных гетерогенных средах.

Метапланирование потоков заданий на неотчуждаемых ресурсах

В грид с неотчуждаемыми ресурсами вычислительные узлы обычно используются локальными высокоприоритетными заданиями владельцев ресурсов. Таким образом, доступные для внешних пользователей ресурсы представляются набором временных интервалов (слотов), в течение которых отдельные узлы могут выполнять задачи, входящие в параллельные пользовательские задания. В гетерогенных средах у слотов имеются различные времена старта и завершения на ресурсах соответствующей производительности. Это усложняет проблему согласованного назначения параллельных заданий, составляющие задачи которых должны стартовать одновременно. Дополнительные трудности вызывает фрагментация ресурсов, когда большая часть узлов занимается низкоприоритетными

мелкими заданиями [1, 2]. При этом высокоприоритетные задания «зависают» в очереди, поскольку не могут набрать требуемое количество слотов (окон) соответствующей длительности. Планирование на уровне приложений базируется на использовании доступных ресурсов и, как правило, не предполагает общей политики их предоставления и потребления [3]. Планирование потоков заданий в ВО основано на неких общих правилах разделения ресурсов, построенных, в частности, на экономических моделях [2 — 7]. Такой подход позволяет повысить эффективность планирования потоков заданий и разделения ресурсов. Политика ВО может быть направлена на оптимизацию планирования с учетом предпочтений стейкхолдеров. Примеры постановок задач планирования в ВО: оптимизация пользовательских критериев или функций полезности для отобранных заданий [6, 8], балансировка загрузки ресурсов [9, 10], соблюдение строгой очередности либо приоритетов заданий [11], оптимизация производительности системы по совокупности критериев [12, 13].

На принципе регламентированного участия пользователей в ВО и доступа к ресурсам строится ряд крупнейших грид-инфраструктур, в частности, EGI (European Grid Infrastructure). В настоящее время наблюдается тенденция гибридации грид технологий, облачных сервисов и платформ [14 — 17]. Интересен российский опыт интеграции облачной платформы Everest [14] как PaaS и грид инфраструктуры EGI. Создание масштабируемых вычислительных сред и поддержка их функционирования, помимо грид технологий (Globus Toolkit, Unicore, gLite, gUSE), облачных сервисов и платформ (Amazon EC2, IBM Cloud, VMware vCloud, Microsoft Azure, Everest), базируется на основе применения метапланировщиков, брокеров ресурсов и систем управления потоками заданий. К числу известных проектов относятся:

- PanDA — система управления потоком заданий, разработка которой начата группами Брукхейвенской Национальной лаборатории (BNL) и Техасского университета в Арлингтоне (UTA); хорошая масштабируемость была продемонстрирована в ходе эксперимента ATLAS на LHC (ЦЕРН), подсистема пилотов реализует позднюю привязку заданий к ресурсам, контролирует процесс их выполнения и позволяет скрыть неоднородность вычислительной среды, PanDA обеспечивает интеграцию в единую среду различных грид инфраструктур, облачных платформ, кластеров, суперкомпьютеров;

- DIRAC — система, позволяющая интегрировать разнородные ресурсы (вычислительный грид, облачные платформы, кластеры) для решения задач в области физики высоких энергий, успешно применяется в эксперименте LHCb (ЦЕРН), весьма эффективен для диспетчеризации заданий в вычислительной среде на базе требования JDL и приоритетов пользователей;

- Ganga — инструментарий для управления вычислительными заданиями и доступа к ресурсам грид, используется в экспериментах ATLAS и LHCb; отличительной особенностью является работа со структурированными заданиями, они могут включать в себя подзадания, выполняющиеся параллельно и, таким образом, может быть реализована оптимизация выполнения на уровне отдельных приложений;

- метапланировщик GridWay — допускает эффективное разделение вычислительных ресурсов (кластеров, суперкомпьютеров, одиночных серверов) в пределах нескольких административных доменов, может использоваться в большинстве существующих грид инфраструктур и в рамках платформ облачных вычислений;

- Nimrod/G — один из первых проектов, в котором реализованы экономические принципы разделения ресурсов для управления выполнением заданий с учетом предпочтений пользователей.

Все описанные проекты реализуют отдельные аспекты комплексного сочетания диспетчеризации потоков заданий, планирования на уровне приложений и осуществления политики предоставления и потребления ресурсов с учетом предпочтений участников ВО.

Предпочтения различных стейкхолдеров ВО могут быть противоречивыми. С одной стороны, пользователи заинтересованы в наискорейшем завершении их заданий с наименьшими издержками (стоимостью выполнения). В то же время предпочтения ВО (администраторов и владельцев ресурсов) связаны с балансировкой загрузки доступных ресурсов либо максимизацией доходов собственников узлов. С другой стороны, политика ВО должна отражать интересы всех стейкхолдеров. При этом важнейшим аспектом является справедливость правил предоставления и потребления ресурсов. В ряде работ понятие справедливости интерпретируется в соответствии с теорией кооперативных игр [8], т. е. имеются справедливые распределение потоков заданий [10], квоты [18, 19], приоритезация пользовательских заданий [11], а также немонетарное распределение ресурсов [20]. Во многих исследованиях предпочтения стейкхолдеров ВО учитываются лишь отчасти. Например, владельцы ресурсов конкурируют за выполнение заданий, оптимизируя лишь пользовательские критерии [6, 21], либо основной целью является эффективное использование ресурсов без учета интересов пользователей [22]. В ряде случаев применяются мультиагентные экономические модели [3, 23, 24]. Однако они не позволяют оптимизировать выполнение потока заданий в целом.

Недостаток большинства централизованных подходов к метапланированию заключается в том, что они утрачивают свою эффективность и возможность оптимизации в распределенных средах с ограниченным уровнем ресурсов. Так, в [2] показано, что традиционный бэкфилинг [1] демонстрирует лучшие резуль-

таты планирования по сравнению с различными оптимизационными схемами на ресурсах с минимально допустимой для заданий производительностью. При этом некоторые задания могут зависать в очереди. В этих условиях приоритеты пользовательских заданий и дисциплина очереди существенно влияют на результаты планирования. В то же время брокеры приложений позволяют учесть предпочтения пользователей и оптимизировать выполнение заданий на основе рыночных механизмов [3, 4, 6, 7].

Циклическое планирование

Циклическая схема планирования (ЦСП) [2, 25] позволяет реализовать справедливое распределение ресурсов, т. е. каждый участник ВО обладает механизмами влияния на результаты планирования, исходя из собственных предпочтений [26].

Циклическое планирование потока заданий осуществляется по пакетам, причем планирование каждого из них ассоциируется с соответствующим временным интервалом (циклом). Процедура включает два основных этапа: на первом происходит поиск альтернативных сценариев выполнения каждого задания (далее — альтернатив), которые по сути являются окнами слотов в пространстве «ресурсы – время» [27]. На втором этапе отбирается оптимальная комбинация альтернатив для выполнения пакета заданий с использованием методов динамического программирования [25]. Для каждого из заданий выбирается одна альтернатива на основе критериев ВО и пользовательских предпочтений [26]. В качестве пользовательских критериев выступают время выполнения задания (makespan), стоимость использования ресурсов и т.д. Они отражают пользовательские предпочтения и содержат дополнительную информацию для поиска альтернатив. Оптимизационная задача планирования в ВО заключается в максимизации или минимизации соответствующих критериев при заданных ограничениях на другие критерии, либо поиске Парето-оптимальной стратегии в пространстве критериев [4, 25, 28].

Для планирования пакета заданий необходимо выделение для каждого из заданий множества альтернативных, не накладывающихся окон в пространстве «узлы – время» (рис. 1). В противном случае могут возникнуть неразрешимые коллизии, обусловленные конкуренцией за ресурсы между заданиями, разделяющими одни и те же слоты. Процедуры последовательного поиска и резервирования ресурсов позволяют предотвратить подобный сценарий.

В экстремальном случае, когда ресурсы ограничены либо перегружены, для каждого задания может быть зарезервирована лишь одна альтернатива. Это вызывает фрагментацию ресурсов, поскольку их большая часть занята низкоприоритетными заданиями (см. рис. 1, а). Ситуация, показанная на рис 1, а, иллюстрирует одно из основных ограничений исходной циклической схемы.

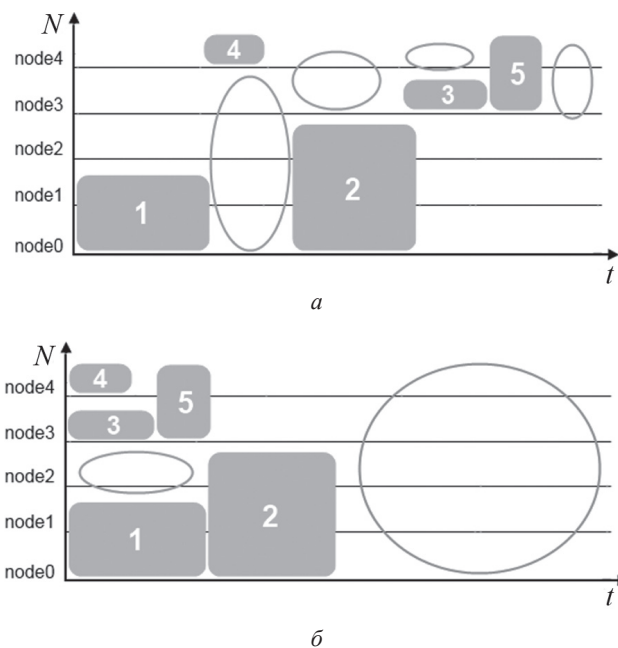


Рис. 1. Альтернативное планирование пакета заданий:

N – число узлов; t – время

Существует и альтернативный план, который мог бы быть реализован, если бы на этапе отбора слотов выбиралась не первая подходящая альтернатива (окно), а просматривались другие возможные варианты назначения (см. рис. 1, б). В этом и состоит идея упреждающего (предвосхищающего) планирования, представляющего развитие ЦСП.

Алгоритмы поиска окон слотов

Для запуска структурированного (параллельного) задания необходимо согласованное выделение необходимого числа слотов, как впрочем, и в классическом варианте бэкфиллинга [1]. Из списка доступных к началу цикла планирования слотов следует отобрать окно параллельных слотов, временная протяженность которого соответствует времени резервирования задания для соответствующего ресурса. Пользовательские требования включаются в ресурсный запрос, содержащий время резервирования ресурсов, характеристики вычислительных узлов (тактовую частоту, емкость оперативной памяти, дискового пространства, тип операционной системы), а также ограничение на максимальную стоимость окна.

На сегодняшний день известно множество алгоритмов и эвристик для выбора ресурсов и планирования параллельных заданий и связанных задач в распределенных вычислительных средах [3 — 6, 12, 13, 27, 29 — 37]. В рамках некоторых из них выбирается первый подходящий набор ресурсов в зависимости от состояния среды без оптимизации планирования [29, 31 — 33]. Другие модели не учитывают такие особенности сред с неотчуждаемыми ресурсами, как динамику загрузки ресурсов, конкуренцию независимых пользователей, глобальных и локальных заданий [3, 4, 6]. Так,

в [6] предложены эвристики для отбора слотов на основе определяемых пользователем функций полезности при заданном ограничении на стоимость формируемого окна. Однако оптимизация выполняется лишь на стадии отбора лучшей из доступных альтернатив ресурсов.

Ряд алгоритмов согласованного назначения на ресурсы (коаллокации) не предполагает предварительного резервирования и основан на синхронной очередности задач, составляющих задание [35]. Известно, что предварительное резервирование в ряде случаев повышает эффективность коаллокации, алгоритмы коаллокации с предварительным резервированием описаны в [13, 31 — 33, 36].

Алгоритмы отбора первого подходящего набора ресурсов (First Fit) [29, 31 — 33] выполняют назначение любого задания на набор слотов, удовлетворяющий требованиям ресурсного запроса без какой-либо оптимизации. Предлагаемый в [29] выбор ресурсов в ВО на основе предпочтений не поддерживает механизм планирования заданий. Задание назначается на первый же найденный набор ресурсов в соответствии с пользовательскими предпочтениями. В [5] изложен подход к согласованному выделению ресурсов среди различных ВО на основе комбинации иерархической и одноранговой (peer-to-peer) моделей взаимодействия метапланировщиков.

Алгоритмы, описанные в [12, 13, 29], реализуют исчерпывающий поиск ресурсов на доступных слотах. Подходы, предложенные в [12, 13], базируются на моделях целочисленного линейного или смешанного целочисленного программирования. Пользователи могут определять временные рамки использования ресурса: раннее и позднее время старта задания, длительность его выполнения, а также предпочтения по времени резервирования. Эти условия накладывают ограничения на отбор слотов лишь в пределах временных рамок. Модель целочисленного линейного программирования в сочетании с генетическим алгоритмом дана в [30]. Она позволяет сформировать план с минимальной стоимостью использования ресурсов для групп независимых пользователей. В [12] на основе модели смешанного целочисленного программирования формируется наилучший план для совокупности совместно функционирующих кластеров, в [37] затрагиваются вопросы перепланирования «на лету» в зависимости от фактической динамики загрузки ресурсов. Подходы к планированию, изложенные в [12, 13, 29, 30, 36, 37], эффективны при заданных целевых критериях: стоимости и времени обработки, уровне загрузки ресурсов, балансировке загрузки для связанных задач. Следует заметить, что сложность данных алгоритмов планирования резко возрастает в зависимости от степени разнородности ресурсов и при необходимости коаллокации задач для параллельных заданий в различных ресурсных доменах.

В работах [27, 34, 38 — 49] исследованы эффективные алгоритмы отбора слотов, основанные на предпо-

чтениях пользователей, администраторов ВО и владельцев вычислительных ресурсов. Некоторые из них (ALP, AMP и AEP) характеризуются линейной временной сложностью в зависимости от числа доступных слотов в текущем интервале планирования [27, 34]. Для отбора слотов пользователи имеют возможность вводить соответствующий критерий в формат ресурсного запроса, например, на языке JSDL.

AEP из множества неупорядоченных слотов строит список, в котором подходящие по ресурсным требованиям слоты отсортированы по неубыванию времени старта (рис. 2). В списке с каждым слотом связывается ожидаемое время выполнения задачи, стоимость ресурса и числовая характеристика, соответствующая критерию отбора окна. Изначально алгоритм предназначен для сканирования списка допустимых слотов и выбора одной альтернативы в соответствии с ресурсным запросом пользователя и оптимальной по заданному критерию. Он позволяет находить окно слотов с ближайшим временем старта, завершения, минимальными временем или стоимостью выполнения задания.

В случае наличия ограничения или перегрузки ресурсов для каждого задания может быть зарезервирована лишь одна альтернатива. В этом случае результат альтернативного планирования ничем не будет отличаться от алгоритмов класса First Fit [2, 29] и приведет к фрагментации. Известный алгоритм обратного заполнения (бэкфиллинг) [1] решает эту проблему, однако изначально он был разработан для параллельных систем и имеет ряд ограничений для распределенных сред. Одно из них — неэффективное использование ресурсов по критериям, отличным от времени старта. Другое — принципиальная невозможность определять политику ВО, используя соответствующие критерии потребления и представления ресурсов.

Модификация AEP для упрещающего планирования заключается в том, чтобы сохранить все промежуточные результаты поиска в качестве множества допустимых сценариев выполнения задания. Это возможно с помощью ЦСП, в основу которой заложены методы динамического программирования. Промежуточные результаты представляют собой альтернативы, условно оптимальные по заданному критерию (например, условно минимальные по времени выполнения при

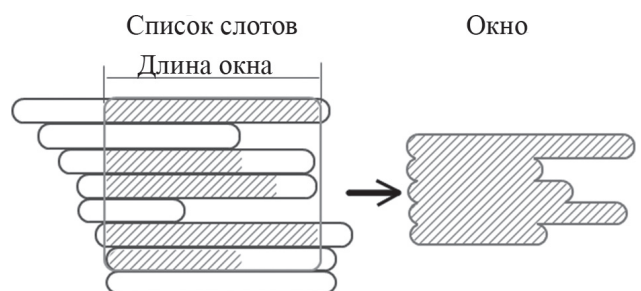


Рис. 2. Отбор окна на списке слотов алгоритмом AEP

соответствующем значении стоимости использования ресурса).

Планирование потоков работ

Эффективность выполнения потока работ в гетерогенных средах критично зависит от планирования задач (tasks) составных частей задания (job). Проблема планирования для модели задания в виде бесконтурного орграфа является *NP*-полной для однородной вычислительной среды [50]. Это не позволяет получать оптимальные решения даже для простейших постановок задачи планирования. Поэтому основное направление исследований в области планирования в модели «поток работ» для гетерогенных сред — построение субоптимальных решений на основе несложных эвристик и метаэвристик [51]. Для сравнения эффективности алгоритмов планирования используются различные среды имитационного моделирования: SimGrid [52] или WorkflowSim [53].

Алгоритмы планирования потока работ можно разделить на статические [54 — 57] и динамические [58 — 60]. К статическим относятся алгоритмы: HEFT (Heterogeneous Earliest Finish Time) [54], HCPT (Heterogeneous Critical Parent Trees) [55], LA (Lookahead) [56], PEFT (Predict Earliest Finish Time) [57]. Они строят расписание выполнения задач до их запуска и могут использоваться для перепланирования потока работ с учетом реальной динамики загрузки ресурсов. Статические алгоритмы используют априорные оценки времени выполнения задач и обмена данными. К известным динамическим алгоритмам принадлежат: OLB (Opportunistic Load Balancing) [58], MCT (Minimum Completion Time), Min-Min, Max-Min и Suffrage [59, 60]. Алгоритмы OLB и MCT назначают задачи, а Min-Min, Max-Min и Suffrage — пакеты задач. Динамические алгоритмы в течение каждого цикла планирования задают задачи, выполнение предшественников которых завершено.

Результаты сравнительного исследования алгоритмов для планирования таких приложений, как CyberShake, Epigenomics, LIGO Inspiral и Montage, приведены в [61]. В ряде работ предложены варианты развития наиболее распространенного алгоритма HEFT, ориентированные на решение определенных проблем при планировании потоков работ. Так, PDHEFT минимизирует затраты на передачу данных между узлами путем дублирования задач, PO-HEFT обеспечивает оценку вычислительных характеристик задачи исходя из ее типа и значений входных параметров, прогнозируя время выполнения задачи, пределы масштабирования и объем передаваемых данных [62].

Упреждающее планирование

Идея упреждающего планирования состоит в том, чтобы генерировать своего рода «заготовку» множества альтернативных планов относительно опорного

решения, а затем из нее выбрать допустимый план, наиболее близкий к опорному в соответствии с заданной метрикой.

Принципиальное отличие идеи упреждающего планирования от известных подходов заключается в следующем.

Шаг 1. Возможные альтернативы выполнения каждого задания отбираются без учета возможных коллизий («пересечений» слотов) и резервирования ресурсов. Результирующий набор альтернатив отражает полный диапазон возможных сценариев выполнения задания на текущем интервале планирования.

Шаг 2. Реализуется выбор комбинации альтернатив (одна альтернатива для каждого задания пакета), оптимальных в соответствии с заданным критерием. Результирующая комбинация альтернатив, вероятнее всего, соответствует недопустимому плану из-за возможного «пересечения» слотов, обуславливающих коллизии при назначении на ресурсы. Основная идея этого этапа заключается в том, что полученное, возможно недопустимое, решение берется как опорное для поиска допустимого плана.

Шаг 3. Допустимое назначение на ресурсы формируется с помощью альтернатив, отобранных на первом этапе, благодаря процедуре репликации, суть которой заключается в отыскании и резервировании допустимых альтернатив выполнения задания, в наибольшей степени соответствующих отобранным, близким к оптимальным, но недопустимым планам. В основу репликации заложена модификация оригинального алгоритма поиска наборов слотов AEP [27, 38].

Множество альтернатив, найденных на первом шаге, отражает диапазон возможных сценариев выполнения каждого задания (упреждающего планирования).

Опорное решение, полученное на втором шаге, представляет собой комбинацию альтернативных окон, в которой каждому заданию соответствует одно окно.

Допустимый план на третьем шаге формируется с использованием заданной метрики близости окон слотов для задания. Таким образом, упреждающее планирование позволяет диверсифицировать множество возможных альтернативных планов и повысить эффективность использования ресурсов. AEP с использованием метрики подобия сканирует список допустимых слотов и сохраняет каждую допустимую их комбинацию.

Главное отличие модификации AEP от исходного алгоритма заключается в том, что вместо поиска окна с экстремальным значением заданного критерия, возвращается окно с минимальным расстоянием от опорной альтернативы. В общем случае, это расстояние должно отражать предпочтения стейкхолдеров по выполнению потока заданий в терминах множества критериев: стоимости и времени выполнения, времени старта и финиша и т. д.

Заключение

Проведенный анализ показал, что известные подходы реализуют лишь отдельные аспекты планирования на уровне приложений [1, 3, 6, 51 — 62], либо на уровне потоков заданий [4, 5, 7 — 17]. Снижение уровня доступности ресурсов в условиях конкуренции потоков независимых заданий приводит к уменьшению числа возможных альтернатив выполнения заданий и, следовательно, ограничению возможностей оптимизации планов. В этом случае альтернативное планирование практически сводится к процедуре выбора «первой подходящей». Принципиально иной подход — упреждающее планирование. Оно дает результаты, сопоставимые с бэкфиллингом по времени старта и завершения задания, и в то же время позволяет учитывать предпочтения стейкхолдеров [63 — 65].

Литература

1. **Dimitriadou S.K., Karatza H.D.** Job Scheduling in a Distributed System Using Backfilling with Inaccurate Runtime Computations // Proc. Intern. Conf. Complex, Intelligent and Software Intensive Syst. Krakow, 2010. Pp. 329—336.
2. **Toporkov V.V.** Heuristic Strategies for Preference-based Scheduling in Virtual Organizations of Utility Grids // J. Ambient Intelligence and Humanized Comp. 2015. V. 6(6). Pp. 733—740.
3. **Buyya R., Abramson D., Giddy J.** Economic Models for Resource Management and Scheduling in Grid Computing // J. Concurrency and Computation. 2002. V. 14 (5). Pp. 1507—1542.
4. **Kurowski K., Nabrzyski J., Oleksiak A., Weglarz J.** Multicriteria Aspects of Grid Resource Management // Grid Resource Management. International Series in Operations Research & Management Sci. 2003. V. 64. Pp. 271—293.
5. **Rodero I. e. a.** Enabling Interoperability among Grid Meta-schedulers // J. Grid Comp. 2013. V. 11 (2). P. 311—336.
6. **Ernemann C., Hamscher V., Yahyapour R.** Economic Scheduling in Grid Computing // DJSSPP. 2002. V. 18. Pp. 128—152.
7. **Baranov A., Telegin P., Tikhomirov A.** Comparison of Auction Methods for Job Scheduling with Absolute Priorities // PaCT. LNCS. 2017. V. 10421. Pp. 387—395.
8. **Rzadca K., Trystram D., Wierzbicki A.** Fair Game-theoretic Resource Management in Dedicated Grids // IEEE Intern. Symp. Cluster Comp. and the Grid. Rio De Janeiro, 2007. Pp. 343—350.
9. **Vasile M., Pop F., Tutueanu R., Cristea V., Kolodziej J.** Resource-aware Hybrid Scheduling Algorithm in Heterogeneous Distributed Computing // J. Future Generation Comp. Syst. 2015. V. 51. Pp. 61—71.
10. **Penmatsa S., Chronopoulos A.T.** Cost Minimization in Utility Computing Systems // J. Concurrency and Computation: Practice and Experience. 2014. V. 16 (1). Pp. 287—307.

Открытыми остаются следующие вопросы планирования конкурирующих потоков параллельных заданий:

- эффективность и вычислительная сложность алгоритмов выбора слотов в схеме упреждающего планирования с различными критериями и разными приоритетами заданий;
- учет комплекса предпочтений пользователя, выраженных в ресурсном запросе, при репликации опорного плана для поиска субоптимальных решений при разных метриках близости;
- эффективность алгоритмов упреждающего планирования с учетом противоречивых пользовательских критериев, политик ВО и администрирования облачных сервисов и платформ.

Данные вопросы определяют задачи, предполагаемые для решения в ходе предстоящих исследований.

References

1. **Dimitriadou S.K., Karatza H.D.** Job Scheduling in a Distributed System Using Backfilling with Inaccurate Runtime Computations. Proc. Intern. Conf. Complex, Intelligent and Software Intensive Syst. Krakow, 2010:329—336.
2. **Toporkov V.V.** Heuristic Strategies for Preference-based Scheduling in Virtual Organizations of Utility Grids. J. Ambient Intelligence and Humanized Comp. 2015;6(6):733—740.
3. **Buyya R., Abramson D., Giddy J.** Economic Models for Resource Management and Scheduling in Grid Computing. J. Concurrency and Computation. 2002; 14 (5):1507—1542.
4. **Kurowski K., Nabrzyski J., Oleksiak A., Weglarz J.** Multicriteria Aspects of Grid Resource Management. Grid Resource Management. International Series in Operations Research & Management Sci. 2003;64:271—293.
5. **Rodero I. e. a.** Enabling Interoperability among Grid Meta-schedulers. J. Grid Comp. 2013;11 (2):311—336.
6. **Ernemann C., Hamscher V., Yahyapour R.** Economic Scheduling in Grid Computing. DJSSPP. 2002;18:128—152.
7. **Baranov A., Telegin P., Tikhomirov A.** Comparison of Auction Methods for Job Scheduling with Absolute Priorities. PaCT. LNCS. 2017;10421:387—395.
8. **Rzadca K., Trystram D., Wierzbicki A.** Fair Game-theoretic Resource Management in Dedicated Grids. IEEE Intern. Symp. Cluster Comp. and the Grid. Rio De Janeiro, 2007:343—350.
9. **Vasile M., Pop F., Tutueanu R., Cristea V., Kolodziej J.** Resource-aware Hybrid Scheduling Algorithm in Heterogeneous Distributed Computing. J. Future Generation Comp. Syst. 2015;51:61—71.
10. **Penmatsa S., Chronopoulos A.T.** Cost Minimization in Utility Computing Systems. J. Concurrency and Computation: Practice and Experience. 2014;16 (1):287—307.

11. **Mutz A., Wolski R., Brevik J.** Eliciting Honest Value Information in a Batch-queue Environment // Proc. VIII IEEE/ACM Intern. Conf. Grid Comp. N.-Y., 2007. Pp. 291—297.
12. **Blanco H., Guirado F., Lrida, J.L., Albornoz V.M.** MIP Model Scheduling for Multi-clusters // Euro-par. Heidelberg: Springer, 2012. Pp. 196—206.
13. **Takefusa A., Nakada H., Kudoh T., Tanaka Y.** An Advance Reservation-based Co-allocation Algorithm for Distributed Computers and Network Bandwidth on QoS-guaranteed Grids // JSSPP. Lecture Notes in Comp. Sci. Heidelberg: Springer, 2010. V. 6253. Pp. 16—34.
14. **Сухорослов О.В.** Комбинированное использование высокопроизводительных ресурсов и грид инфраструктур в рамках облачной платформы Everest // Суперкомпьютерные дни в России: Труды Междунар. конф. 2015. С. 706—711.
15. **Bencivenni M. e. a.** Accessing Grid and Cloud Services Through a Scientific Web Portal // J. Grid Comp. 2015. V. 13. Pp. 159—175.
16. **Ronchieri E. e. a.** Accessing Scientific Applications through the WNoDeS Cloud Virtualization Framework // Proc. Intern. Symp. Grids and Clouds (ISGC). Taipei, 2013. Pp. 3—12.
17. **EGI Federated Clouds Task Force** [Официальный сайт] <https://https://www.egi.eu> (дата обращения 19.07.2017).
18. **Carroll T., Grosu D.** Divisible Load Scheduling: An Approach Using Coalitional Games // Proc. VI Intern. Symp. Parallel and Distributed Comp. 2007. Pp. 36—45.
19. **Kim K., Buyya R.** Fair Resource Sharing in Hierarchical Virtual Organizations for Global Grids // Proc. VIII IEEE/ACM Intern. Conf. Grid Comp. Austin, 2007. Pp. 50—57.
20. **Skowron P., Rządca K.** Non-monetary Fair Scheduling Cooperative Game Theory Approach // Proc. 25 Annual ACM Symp. Parallelism in Algorithms and Architectures. N.-Y., 2013. Pp. 288—297.
21. **Dalheimer M., Pfreundt F., Merz P.** Agent-based Grid Scheduling with Calana // Proc. Parallel Processing and Appl. Math. VI Intern. Conf. 2006. Pp. 741—750.
22. **Jackson D., Snell Q., Clement M.** Core Algorithms of the Maui Scheduler // Revised Papers from VII Intern. Workshop on Job Scheduling Strategies for Parallel Proc. 2002. Pp. 87—102.
23. **Thain T., Livny M.** Distributed Computing in Practice: the Condor Experience // J. Concurrency and Computation: Practice and Experience. 2005. V. 17. Pp. 323—356.
24. **Богданова В.Г., Бычков И.В., Корсуков А.С., Опарин Г.А., Феоктистов А.Г.** Мультиагентный подход к управлению распределенными вычислениями в кластерной GRID-системе // Известия РАН. Серия «Теория и системы управления». 2014. № 5. С. 95—105.
25. **Toporkov V. e. a.** Metascheduling and Heuristic Co-allocation Strategies in Distributed Computing // J. Comp. and Informatics. 2015. V. 34 (1). Pp. 45—76.
11. **Mutz A., Wolski R., Brevik J.** Eliciting Honest Value Information in a Batch-queue Environment. Proc. VIII IEEE/ACM Intern. Conf. Grid Comp. N.-Y., 2007:291—297.
12. **Blanco H., Guirado F., Lrida, J.L., Albornoz V.M.** MIP Model Scheduling for Multi-clusters. Euro-par. Heidelberg: Springer, 2012:196—206.
13. **Takefusa A., Nakada H., Kudoh T., Tanaka Y.** An Advance Reservation-based Co-allocation Algorithm for Distributed Computers and Network Bandwidth on QoS-guaranteed Grids. JSSPP. Lecture Notes in Comp. Sci. Heidelberg: Springer, 2010;6253:16—34.
14. **Suhoroslov O.V.** Kombinirovannoe Ispol'zovanie Vysokoproizvoditel'nykh Resursov i Grid Infrastruktur v Ramkah Oblachnoy Platformy Everest. Superkomp'yuternye Dni v Rossii: Trudy Mezhdunar. Konf. 2015:706—711. (in Russian).
15. **Bencivenni M. e. a.** Accessing Grid and Cloud Services Through a Scientific Web Portal. J. Grid Comp. 2015;13:159—175.
16. **Ronchieri E. e. a.** Accessing Scientific Applications through the WNoDeS Cloud Virtualization Framework. Proc. Intern. Symp. Grids and Clouds (ISGC). Taipei, 2013:3—12.
17. **EGI Federated Clouds Task Force** [Ofits. Sayt] <https://https://www.egi.eu> (Data Obrashcheniya 19.07.2017).
18. **Carroll T., Grosu D.** Divisible Load Scheduling: An Approach Using Coalitional Games. Proc. VI Intern. Symp. Parallel and Distributed Comp. 2007:36—45.
19. **Kim K., Buyya R.** Fair Resource Sharing in Hierarchical Virtual Organizations for Global Grids. Proc. VIII IEEE/ACM Intern. Conf. Grid Comp. Austin, 2007:50—57.
20. **Skowron P., Rządca K.** Non-monetary Fair Scheduling Cooperative Game Theory Approach. Proc. 25 Annual ACM Symp. Parallelism in Algorithms and Architectures. N.-Y., 2013:288—297.
21. **Dalheimer M., Pfreundt F., Merz P.** Agent-based Grid Scheduling with Calana. Proc. Parallel Processing and Appl. Math. VI Intern. Conf. 2006:741—750.
22. **Jackson D., Snell Q., Clement M.** Core Algorithms of the Maui Scheduler. Revised Papers from VII Intern. Workshop on Job Scheduling Strategies for Parallel Proc. 2002:87—102.
23. **Thain T., Livny M.** Distributed Computing in Practice: the Condor Experience. J. Concurrency and Computation: Practice and Experience. 2005;17:323—356.
24. **Bogdanova V.G., Bychkov I.V., Korsukov A.S., Oparin G.A., Feoktistov A.G.** Mul'tiagentnyy podhod k upravleniyu raspredelennymi vychisleniyami v klasternoy GRID-sisteme. Izvestiya RAN. Seriya «Teoriya i sistemy upravleniya». 2014;5:95—105. (in Russian).
25. **Toporkov V. e. a.** Metascheduling and Heuristic Co-allocation Strategies in Distributed Computing. J. Comp. and Informatics. 2015;34 (1):45—76.

26. **Toporkov V., Yemelyanov D., Bobchenkov A., Potekhin P.** Fair Resource Allocation and Metascheduling in Grid with VO Stakeholders Preferences // IEEE 45 Intern. Conf. on Parallel Processing Workshops. 2016. Pp. 375—384.
27. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Slot Selection Algorithms in Distributed Computing // Journal of Supercomputing. 2014. V. 69 (1). Pp. 53—60.
28. **Farahabady M.H., Lee Y.C., Zomaya A.Y.** Pareto-optimal Cloud Bursting // IEEE Transac. on Parallel and Distributed Sys. 2014. V. 25. Pp. 2670—2682.
29. **Cafaro M., Mirto M., Aloisi, G.** Preference-Based Matchmaking of Grid Resources with CP-Nets // J. Grid Comp. 2013. V. 11 (2). Pp. 211—237.
30. **Garg S.K., Konugurthi P., Buyya R.** A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids // J. Par., Emergent and Distr. Syst. 2011. V. 26. Pp. 493—517.
31. **Aida K., Casanova H.** Scheduling Mixed-parallel Applications with Advance Reservations // 17th IEEE Int. Symposium on HPDC. N.-Y., 2008. Pp. 65—74.
32. **Ando S., Aida K.** Evaluation of Scheduling Algorithms for Advance Reservations // Information Proc. Soc. Japan SIG Notes. 2007. Pp. 37—42.
33. **Elmroth E., Tordsson J.** A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability // J. of Concurrency and Computation. 2009. V. 21(18). Pp. 2298—2335.
34. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Slot Selection Algorithms in Distributed Computing with Non-dedicated and Heterogeneous Resources // PaCT. LNCS. Heidelberg: Springer, 2013. V. 7979. Pp. 120—134.
35. **Azzedin F., Maheswaran M., Arnason N.** A Synchronous Co-allocation Mechanism for Grid Computing Systems // Cluster Comp. 2004. V. 7. Pp. 39—49.
36. **Castillo C., Rouskas G.N., Harfoush K.** Resource Co-allocation for Large-scale Distributed Environments // 18th ACM Intern. Symp. High Performance Distributed Comp. N.-Y. 2009. Pp. 137—150.
37. **Olteanu A., Pop F., Dobre C., Cristea V.** A Dynamic Rescheduling Algorithm for Resource Management in Large Scale Dependable Distributed Systems // Computers and Mathematics with Appl. 2012. V. 63 (9). Pp. 1409—1423.
38. **Топорков В.В., Емельянов Д.М.** Экономическая модель планирования и справедливого разделения ресурсов в распределенных вычислениях // Программирование. 2014. № 1. С. 54—65.
39. **Топорков В.В., Бобченков А.В., Емельянов Д.М., Целищев А.С.** Методы и эвристики планирования в распределенных вычислениях с неотчуждаемыми ресурсами // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2014. Т. 3. № 2. С. 43—62.
26. **Toporkov V., Yemelyanov D., Bobchenkov A., Potekhin P.** Fair Resource Allocation and Metascheduling in Grid with VO Stakeholders Preferences. IEEE 45 Intern. Conf. on Parallel Processing Workshops. 2016:375—384.
27. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Slot Selection Algorithms in Distributed Computing. Journal of Supercomputing. 2014;69 (1):53—60.
28. **Farahabady M.H., Lee Y.C., Zomaya A.Y.** Pareto-optimal Cloud Bursting. IEEE Transac. on Parallel and Distributed Sys. 2014;25:2670—2682.
29. **Cafaro M., Mirto M., Aloisi, G.** Preference-Based Matchmaking of Grid Resources with CP-Nets. J. Grid Comp. 2013;11 (2):211—237.
30. **Garg S.K., Konugurthi P., Buyya R.** A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids. J. Par., Emergent and Distr. Syst. 2011;26:493—517.
31. **Aida K., Casanova H.** Scheduling Mixed-parallel Applications with Advance Reservations. 17th IEEE Int. Symposium on HPDC. N.-Y., 2008:65—74.
32. **Ando S., Aida K.** Evaluation of Scheduling Algorithms for Advance Reservations. Information Proc. Soc. Japan SIG Notes. 2007:37—42.
33. **Elmroth E., Tordsson J.** A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability. J. of Concurrency and Computation. 2009;21(18):2298—2335.
34. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Slot Selection Algorithms in Distributed Computing with Non-dedicated and Heterogeneous Resources. PaCT. LNCS. Heidelberg: Springer, 2013;7979:120—134.
35. **Azzedin F., Maheswaran M., Arnason N.** A Synchronous Co-allocation Mechanism for Grid Computing Systems. Cluster Comp. 2004;7:39—49.
36. **Castillo C., Rouskas G.N., Harfoush K.** Resource Co-allocation for Large-scale Distributed Environments. 18th ACM Intern. Symp. High Performance Distributed Comp. N.-Y. 2009:137—150.
37. **Olteanu A., Pop F., Dobre C., Cristea V.** A Dynamic Rescheduling Algorithm for Resource Management in Large Scale Dependable Distributed Systems. Computers and Mathematics with Appl. 2012;63 (9):1409—1423.
38. **Топорков В.В., Емельянов Д.М.** Экономическая Модель Планирования и Справедливого Разделения Ресурсов в Распределенных Вычислениях. Programmirovaniye. 2014;1:54—65. (in Russian).
39. **Топорков В.В., Бобченков А.В., Емельянов Д.М., Тselishchev А.С.** Методы и Эвристики Планирования в Распределенных Вычислениях с Неотчуждаемыми Ресурсами. Vestnik YUUrGU. Seriya «Vychislitel'naya Matematika i Informatika». 2014;3;2:43—62. (in Russian).

40. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Heuristic Co-allocation Strategies in Distributed Computing with Non-dedicated Resources // *Studies in Computational Intelligence*. Heidelberg: Springer, 2014. V. 511. Pp. 109—118.
41. **Toporkov V. e. a.** Preference-based Fair Resource Sharing and Scheduling Optimization in Grid VOs // *Procedia Computer Sci.* 2014. V. 29. Pp. 831—843.
42. **Toporkov V. e. a.** Heuristic Cycle-based Scheduling with Backfilling for Large-scale Distributed Environments // *Advances in Intelligent Systems and Computing (AISC)*. Heidelberg: Springer, 2014. V. 286. Pp. 455—465.
43. **Toporkov V., Tselishchev A., Yemelyanov D., Potekhin P.** Metascheduling Strategies in Distributed Computing with Non-dedicated Resources // *Dependability Problems of Complex Information Systems, Advances in Intelligent Syst. and Comp. (AISC)*. Springer International Publ., 2014. V. 307. Pp. 129—148.
44. **Toporkov V., Yemelyanov D., Tselishchev A.** Effective Slot Selection and Co-allocation Algorithms for Economic Scheduling in Distributed Computing // *Procedia Comp. Sci.* 2013. V. 18. Pp. 2424—2427.
45. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Slot Selection Algorithms for Economic Scheduling in Distributed Computing with High QoS Rates // *New Results in Computer Systems and Dependability. AISC*. Heidelberg: Springer, 2013. V. 224. Pp. 459—468.
46. **Топорков В.В.** Пакетная обработка заданий в распределенных вычислительных средах с неотчуждаемыми ресурсами // *Автоматика и телемеханика*. 2012. № 10. С. 52—70.
47. **Toporkov V., Tselishchev A., Yemelyanov D., Bobchenkov A.** Composite Scheduling Strategies in Distributed Computing with Non-dedicated Resources // *Procedia Comp. Sci.* Elsevier, 2012. V. 9. Pp. 176—185.
48. **Toporkov V., Tselishchev A., Yemelyanov D., Bobchenkov A.** Dependable Strategies for Job-flows Dispatching and Scheduling in Virtual Organizations of Distributed Computing Environments // *Complex Systems and Dependability*. Berlin, Heidelberg: Springer-Verlag, 2012;170:240—255.
49. **Toporkov V., Toporkova A., Bobchenkov A., Yemelyanov D.** Resource Selection Algorithms for Economic Scheduling in Distributed Systems // *Proc. Comp. Sci.* Elsevier, 2011. V. 4. Pp. 2267—2276.
50. **Graham R.L., Lawler E.L., Lenstra J.K., Kan A.R.** Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey // *Annals Discrete Mathematics*. 1979;5:287—326.
51. **Yu J., Buyya R., Ramamohanarao K.** Workflow Scheduling Algorithms for Grid Computing // *Metaheuristics for Scheduling in Distributed Computing Environments. Studies in Computational Intelligence*. Berlin: Springer, 2008. V. 146. Pp. 173—214.
52. **Casanova H., Giersch A., Legrand A., Quinson M., Suter F.** Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms // *J. Parallel and Distributed Comp.* 2014. V. 74 (10). Pp. 2899—2917.
40. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Heuristic Co-allocation Strategies in Distributed Computing with Non-dedicated Resources. *Studies in Computational Intelligence*. Heidelberg: Springer, 2014;511:109—118.
41. **Toporkov V. e. a.** Preference-based Fair Resource Sharing and Scheduling Optimization in Grid VOs. *Procedia Computer Sci.* 2014;29:831—843.
42. **Toporkov V. e. a.** Heuristic Cycle-based Scheduling with Backfilling for Large-scale Distributed Environments. *Advances in Intelligent Systems and Computing (AISC)*. Heidelberg: Springer, 2014;286:455—465.
43. **Toporkov V., Tselishchev A., Yemelyanov D., Potekhin P.** Metascheduling Strategies in Distributed Computing with Non-dedicated Resources. *Dependability Problems of Complex Information Systems, Advances in Intelligent Syst. and Comp. (AISC)*. Springer International Publ., 2014;307:129—148.
44. **Toporkov V., Yemelyanov D., Tselishchev A.** Effective Slot Selection and Co-allocation Algorithms for Economic Scheduling in Distributed Computing. *Procedia Comp. Sci.* 2013;18:2424—2427.
45. **Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D.** Slot Selection Algorithms for Economic Scheduling in Distributed Computing with High QoS Rates. *New Results in Computer Systems and Dependability. AISC*. Heidelberg: Springer, 2013;224:459—468.
46. **Toporkov V.V.** Paketnaya Obrabotka Zada-niy V Raspredeleennyh Vychislitel'nyh Sredah S Neotchuzhdaemyimi Resursami. *Avtomatika i Telemekhanika*. 2012;10: 52—70. (in Russian).
47. **Toporkov V., Tselishchev A., Yemelyanov D., Bobchenkov A.** Composite Scheduling Strategies in Distributed Computing with Non-dedicated Resources. *Procedia Comp. Sci.* Elsevier, 2012;9:176—185.
48. **Toporkov V., Tselishchev A., Yemelyanov D., Bobchenkov A.** Dependable Strategies for Job-flows Dispatching and Scheduling in Virtual Organizations of Distributed Computing Environments . *Complex Systems and Dependability*. Berlin, Heidelberg: Springer-Verlag, 2012;170:240—255.
49. **Toporkov V., Toporkova A., Bobchenkov A., Yemelyanov D.** Resource Selection Algorithms for Economic Scheduling in Distributed Systems. *Proc.Comp. Sci.* Elsevier, 2011;4:2267—2276.
50. **Graham R.L., Lawler E.L., Lenstra J.K., Kan A.R.** Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals Discrete Mathematics*. 1979;5:287—326.
51. **Yu J., Buyya R., Ramamohanarao K.** Workflow Scheduling Algorithms for Grid Computing. *Metaheuristics for Scheduling in Distributed Computing Environments. Studies in Computational Intelligence*. Berlin: Springer, 2008;146:173—214.

53. **Chen W., Deelman E.** Workflowsim: A toolkit for Simulating Scientific Workflows in Distributed Environments // IEEE E-science (e-science) VIII Intern. Conf. 2012. Pp. 1—8.
54. **Topcuoglu H., Hariri S., Wu M.Y.** Performance-effective and Low-complexity Task Scheduling For Heterogeneous Computing // IEEE Trans. Parallel and Distributed Syst. 2002. V. 13 (3). Pp. 260—274.
55. **Hagras T., Janecek J.** A Simple Scheduling Heuristic for Heterogeneous Computing Environment // Proc. II Intern. Symp. Parallel and Distributed Comp. 2003. Pp. 104—110.
56. **Bittencourt L.F., Sakellariou R., Madeira E.R.M.** Dag Scheduling Using a Lookahead Variant of the Heterogeneous Earliest Finish Time Algorithm // Proc. XVIII Euromicro Conf. Parallel, Distributed and Network-based. 2010. Pp. 27—34.
57. **Arabnejad H., Barbosa J.G.** List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table // IEEE Trans. Parallel and Distributed Syst. 2014. V. 25 (3). Pp. 682—694.
58. **Armstrong R., Hensgen D. Kidd T.** The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-Time Predictions // IEEE Heterogeneous Computing Workshop. 1998. Pp. 79—87.
59. **Maheswaran M. e. a.** Dynamic Matching and Scheduling of a Class of Independent Tasks Onto Heterogeneous Computing Systems // IEEE Heterogeneous Computing Workshop. IEEE. 1999. Pp. 30—44.
60. **Freund R.F. e. a.** Scheduling Resources In Multi-user, Heterogeneous, Computing Environments with Smart-net // IEEE Heterogeneous Computing Workshop. 1998. Pp. 184—199.
61. **Nazarenko A., Sukhoroslov O.** An Experimental Study of Workow Scheduling Algorithms for Heterogeneous Systems // PaCT. LNCS. Springer Intern. Publ., 2017. V. 10421. Pp. 327—341.
62. **Радченко Г.И., Лыжин И.А., Неповинных Е.А.** Имплементация и сравнительное тестирование алгоритма проблемно-ориентированного планирования потоковых приложений в облачных средах PO-HEFT // Суперкомпьютерные дни в России: Труды Междунар. конф. 2016. С. 165—179.
63. **Toporkov V., Yemelyanov D., Bobchenkov A.** Job-flow Anticipation Scheduling in Grid // Proc. Comp. Sci. 2017. V. 108. Pp. 1394—1403.
64. **Toporkov V., Yemelyanov D., Toporkova A.** Anticipation Preference-based Heuristic Scheduling in Grid Virtual Organizations // Proc. Intern. Conf. Parallel Processing Workshops. 2017. Pp. 271—280.
65. **Toporkov V., Toporkova A., Yemelyanov D.** Heuristic of Anticipation for Fair Scheduling and Resource Allocation in Grid VOs // Studies in Computational Intelligence. Springer Verlag, 2018. V. 737. Pp. 27—37.
52. **Casanova H., Giersch A., Legrand A., Quinson M., Suter F.** Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms. J. Parallel and Distributed Comp. 2014;74 (10):2899—2917.
53. **Chen W., Deelman E.** Workflowsim: A toolkit for Simulating Scientific Workflows in Distributed Environments. IEEE E-science (e-science) VIII Intern. Conf. 2012:1—8.
54. **Topcuoglu H., Hariri S., Wu M.Y.** Performance-effective and Low-complexity Task Scheduling For Heterogeneous Computing. IEEE Trans. Parallel and Distributed Syst. 2002;13 (3):260—274.
55. **Hagras T., Janecek J.** A Simple Scheduling Heuristic for Heterogeneous Computing Environment. Proc. II Intern. Symp. Parallel and Distributed Comp. 2003:104—110.
56. **Bittencourt L.F., Sakellariou R., Madeira E.R.M.** Dag Scheduling Using a Lookahead Variant of the Heterogeneous Earliest Finish Time Algorithm. Proc. XVIII Euromicro Conf. Parallel, Distributed and Network-based. 2010:27—34.
57. **Arabnejad H., Barbosa J.G.** List Scheduling Algorithm for Heterogeneous Systems by an Optimistic Cost Table. IEEE Trans. Parallel and Distributed Syst. 2014;25 (3):682—694.
58. **Armstrong R., Hensgen D. Kidd T.** The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-Time Predictions. IEEE Heterogeneous Computing Workshop. 1998:79—87.
59. **Maheswaran M. e. a.** Dynamic Matching and Scheduling of a Class of Independent Tasks Onto Heterogeneous Computing Systems. IEEE Heterogeneous Computing Workshop. IEEE. 1999:30—44.
60. **Freund R.F. e. a.** Scheduling Resources In Multi-user, Heterogeneous, Computing Environments with Smart-net. IEEE Heterogeneous Computing Workshop. 1998:184—199.
61. **Nazarenko A., Sukhoroslov O.** An Experimental Study of Workow Scheduling Algorithms for Heterogeneous Systems. PaCT. LNCS. Springer Intern. Publ., 2017; 10421:327—341.
62. **Radchenko G.I., Lyzhin I.A., Nepovinnih E.A.** Implementatsiya i Sravnitel'noe Testirovanie Algoritma Problemno-orientirovannogo Planirovaniya Potokovyh Pri-lozheniy v Oblachnyh Sredah PO-HEFT. Superkomp'yuternye Dni v Rossii: Trudy Mezhdunar. Konf. 2016: 165—179. (in Russian).
63. **Toporkov V., Yemelyanov D., Bobchenkov A.** Job-flow Anticipation Scheduling in Grid. Proc. Comp. Sci. 2017;108:1394—1403.
64. **Toporkov V., Yemelyanov D., Toporkova A.** Anticipation Preference-based Heuristic Scheduling in Grid Virtual Organizations. Proc. Intern. Conf. Parallel Processing Workshops. 2017:271—280.
65. **Toporkov V., Toporkova A., Yemelyanov D.** Heuristic of Anticipation for Fair Scheduling and Resource Allocation in Grid VOs. Studies in Computational Intelligence. Springer Verlag, 2018;737:27—37.

Сведения об авторах:

Топорков Виктор Васильевич — доктор технических наук, заведующий кафедрой вычислительной техники НИУ «МЭИ», e-mail: ToporkovVV@mpei.ru

Емельянов Дмитрий Михайлович — кандидат технических наук, доцент кафедры вычислительной техники НИУ «МЭИ», e-mail: Yemelyanov.dmitry@gmail.com

Information about authors:

Toporkov Viktor V. — Dr.Sci. (Techn.), Head of Computer Engineering Dept., NRU MPEI, e-mail: ToporkovVV@mpei.ru

Yemelyanov Dmitriy M. — Ph.D. (Techn.), Assistant Professor of Computer Engineering Dept., NRU MPEI, e-mail: Yemelyanov.dmitry@gmail.com

Работа выполнена при поддержке: РФФИ (проекты № 18-07-00456, 18-07-00534); Совета по грантам Президента Российской Федерации для поддержки молодых российских ученых (проект № МК-2297.2017.9); Министерства образования и науки Российской Федерации (проект № 2.9606.2017/8.9)

The work is executed at support: RFBR (grants No.18-07-00456, 18-07-00534); Grant Council of the President of the Russian Federation for State Support of Young Russian Scientists (grants No. МК-2297.2017.9); The Ministry of Education and Science of the Russian Federation (grants No. 2.9606.2017/8.9)

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов

Conflict of interests: the authors declare no conflict of interest

Статья поступила в редакцию: 21.03.2018

The article received to the editor: 21.03.2018