

Ранжирование и циклическое планирование заданий в распределенных вычислительных средах*

В.В. Топорков, Д.М. Емельянов, П.А. Потехин

Национальный исследовательский университет «МЭИ»

В работе рассматриваются подходы к формированию системы ранжированных заданий в модели циклического планирования в виртуальных организациях распределенных вычислительных сред. Предлагаются и сравниваются две различные методологии отбора заданий для планирования: первая из них базируется на решении задачи о заполнении ранца, для второй – вводится эвристический показатель «совместимости» заданий и доменов вычислительных узлов. Приводятся результаты экспериментального исследования, позволяющего оценить эффективность предложенных решений, проводится их сравнительный анализ со случайным отбором заданий в пакет.

1. Введение

Сложность планирования вычислений в таких средах, как грид, обусловлена динамичностью, географической распределенностью ресурсов, а также разнородностью заданий от пользователей виртуальных организаций (ВО) и требований к их выполнению [1]. Важнейшей задачей планирования в ВО является обеспечение эффективного использования доступных ресурсов с учетом интересов основных участников вычислений: пользователей, владельцев ресурсов и администраторов. В распределенных вычислительных средах (РВС) с большим числом участников с различными и, зачастую, противоречащими друг другу, интересами свою эффективность показали подходы к планированию на основе экономических принципов [2-6].

Решения задач планирования можно классифицировать в зависимости от используемого типа диспетчеризации заданий. При децентрализованной диспетчеризации планировщики ресурсов, как правило, работают локально на стороне клиента для реализации интересов конкретного пользователя (AppLeS [7], PAUA [8]). Централизованная (иерархическая) диспетчеризация предполагает наличие метапланировщика, обеспечивающего более эффективное использование доступных ресурсов. Метапланировщик РВС в процессе планирования работает с метазаданиями, сопровождающимися ресурсными запросами – спецификациями характеристик вычислительных ресурсов, необходимых для выполнения задания. Иерархическая модель используется, например, в таких системах, как X-Com [9], GrADS [10]. Таким образом, перед метапланировщиком стоит задача распределения потока заданий между доступными ресурсами и доменами ресурсов РВС на основе метаданных, заданных пользователем в ресурсном запросе. Возможна оценка ресурсных требований заданий на основе статистики или с использованием экспертных систем [11]. Как правило, задача планирования потока заданий решается с использованием стандартных наборов средств и алгоритмов, включающих FCFS (First Come First Served), бэжфиллинг, механизмы вычисления приоритетов пользователей и заданий, и разделения ресурсов [12-14]. Важным аспектом в этих подходах является преимущественное соблюдение очереди и пользовательских приоритетов при выполнении заданий. Еще более «справедливое» формирование очереди [6] основывается на экономических принципах и учитывает особенности отдельных заданий и их влияние на выполнение всей очереди.

Циклическое планирование потока заданий [15] позволяет реализовать политику ВО с использованием различных критериев. Планирование осуществляется на основе динамически обновляющихся данных о состоянии загрузки доступных вычислительных узлов домена РВС. Та-

* Работа выполнена при содействии Совета по грантам Президента Российской Федерации для поддержки молодых российских ученых и ведущих научных школ (шифры МК-4148.2015.9, НШ-362.2014.9), РФФИ (проекты 15-07-02259, 15-07-03401), Минобрнауки России, задание № 2014/123 на выполнение государственных работ в сфере научной деятельности в рамках базовой части государственного задания (проект 2268).

ким образом, в каждом цикле планирования потока заданий решаются следующие задачи: распределение заданий из глобальной очереди в потоки; формирование системы (пакетов) заданий; планирование системы заданий в соответствии с принятой в ВО политикой. При планировании пакета заданий, интересы ВО, как правило, выше приоритетов отдельных заданий, что позволяет оптимизировать общие параметры выполнения системы заданий. Например, в [16] при планировании пакета заданий решается задача минимизации суммарного потребления электроэнергии. При этом дисциплина очереди может быть и нарушена.

В циклической схеме планирования (ЦСП) [15] предполагается реализация двух этапов: во-первых, поиск для каждого задания нескольких вариантов (альтернатив) выполнения на интервале планирования и, во-вторых, последующий выбор комбинации альтернатив (по одной для каждого задания) по критерию эффективности, принятому в ВО [17]. Наличие нескольких альтернатив выполнения создает возможности для оптимизации плана выполнения пакета независимых заданий. Выбор комбинации осуществляется методами динамического программирования. В качестве примера политики ВО может рассматриваться задача минимизации суммарного времени выполнения пакета заданий при ограничении на общий бюджет виртуальной организации в текущем цикле планирования. Для учета интересов пользователей ВО пакет формируется из наиболее приоритетных заданий потока (например, стоящих в начале очереди). Поиск альтернатив выполнения осуществляется последовательно для каждого задания, что подчеркивает важность приоритетов заданий в пакете. При введении пользовательских критериев оптимизации отдельных заданий можно обеспечить «справедливое» планирование пакета заданий [15, 17]. Однако стоит отметить, что отбор заданий согласно простым приоритетам пользователей может негативно сказаться на эффективности планирования всего потока заданий. Другими словами, для повышения эффективности планирования потока заданий в целом, в соответствии с политикой ВО, необходимо рассмотреть альтернативные способы ранжирования системы заданий.

Данная статья посвящена вопросам формирования системы заданий для ЦСП. Предлагается эвристический коэффициент «совместимости» характеристик заданий и домена вычислительных узлов для принятия решений о распределении потоков заданий между доменами или для отбора заданий в пакет. Предлагаются и сравниваются два различных подхода к формированию пакета: первый базируется на решении известной задачи о заполнении ранца, второй использует показатель «совместимости» заданий с доменами ресурсов.

Остальная часть статьи организована следующим образом. В разделе 2 описываются предлагаемые решения для формирования системы заданий. Раздел 3 посвящен экспериментальным исследованиям. В заключении приводятся основные результаты и обсуждаются направления дальнейшей работы.

2. Формирование системы заданий

Ресурсный запрос задания представляет собой отражение пользовательских ожиданий качества сервиса, а также пользовательскую часть соглашения о предоставляемых услугах. Высокая степень разнородности и разнообразия систем планирования распределенных и параллельных вычислений приводит к существованию множества различных форм представления ресурсного запроса. В ЦСП выделяются следующие основные требования к вычислительным ресурсам: минимально необходимая для выполнения задания производительность p , максимальная суммарная стоимость (бюджет) S выполнения задания, требуемое число n вычислительных узлов, а также время t , на которое необходимо зарезервировать ресурсы (в расчете на ресурсы производительности p). Эта модель опирается на пользовательскую оценку времени выполнения задания. Использование относительных единиц производительности ресурса позволяет оценить время выполнения задания на ресурсах различной производительности. Время выполнения может сокращаться в зависимости от производительности узлов, на которые назначено задание. Если пользовательская оценка неточна, выполнение задания может быть либо прервано по истечении времени выделения ресурсов, либо ресурсы могут быть освобождены преждевременно.

Система независимых заданий в каждом цикле планирования представляет собой пакет определенным образом отобранных из потока заданий от пользователей. Отбор заданий в паке-

ты позволяет повысить общую эффективность планирования в виртуальной организации по сравнению с планированием каждого задания в отдельности за счет оптимизации общего критерия, формализующего политику ВО, и справедливого распределения ресурсов на основе предпочтений ключевых стейкхолдеров [2-6, 11, 15].

2.1 Ограничение размера пакета

Важным этапом, на первый взгляд, не имеющим прямого отношения к эффективности циклического планирования потока заданий в ВО, является определение размера пакета заданий в каждом цикле. За счет варьирования ограничения на размер пакета заданий, который может быть выражен в количестве заданий, их суммарной стоимости и т.д., можно повысить эффективность планирования согласно одному или нескольким различным критериям. В нашей модели можно выделить следующие критерии эффективности планирования.

- Загрузка вычислительных узлов. В максимизации данного параметра заинтересованы собственники вычислительных ресурсов.
- Критерий оптимизации, формализующий политику ВО. Примером может служить минимизация времени выполнения потока заданий при ограничении на его суммарную стоимость.
- Число альтернатив для выполнения каждого задания. Данный критерий напрямую связан с предыдущим. В ЦСП большее число альтернатив создает больше возможностей для оптимизации планирования.
- Количество циклов планирования, требуемых для выполнения некоторого множества (потока) заданий. Минимизация данного показателя обеспечивает более высокую пропускную способность распределенной вычислительной среды.

Приведенные выше критерии эффективности планирования в ЦСП являются взаимозависимыми и противоречивыми. К примеру, чем большее количество альтернатив выполнения удастся найти для заданий пакета в каком-либо цикле планирования, тем ниже будет относительная загрузка вычислительных узлов на этом интервале, потому что для выполнения задания будет выбрана только одна из найденных альтернатив. С другой стороны, чем меньше альтернатив выполнения, тем выше потенциальная загрузка доступных ресурсов, но тем меньше возможностей для оптимизации по критерию ВО и, следовательно, хуже значение принятого критерия. Таким образом, нельзя предложить решение, которое бы удовлетворяло всем приведенным типам критериев. Однако этап формирования и определения размера пакета заданий перед каждым циклом может быть использован администраторами ВО в качестве дополнительного механизма управления процессом планирования. Например, можно предположить, что больший размер пакета заданий в общем случае обеспечит более высокую загрузку вычислительных ресурсов, а также уменьшит число циклов планирования. При этом уменьшится и число найденных альтернатив выполнения, что негативно отразится на значении критерия ВО.

Непосредственное задание размера пакета, например, администраторами ВО, нецелесообразно. В условиях, когда локальные расписания доступных вычислительных узлов динамически изменяются, а характеристики входящих заданий существенно различаются и базируются на пользовательских, зачастую неточных, оценках, невозможно заранее выбрать ограничение, которое бы позволило повысить эффективность планирования согласно выбранному в ВО критерию. Более гибкий механизм ограничения размера пакета можно построить на основе соотношения требований заданий и характеристик вычислительной среды. В качестве таких характеристик в рамках экономических принципов логично выбрать стоимость использования и время резервирования (предоставления) ресурсов.

При использовании ограничения по времени для каждого задания вычисляется суммарное время занятия слотов в пересчете на ресурс единичной производительности. Для выполнения задания в РВС необходимо выделение набора подходящих слотов, каждый из которых характеризуется временем старта, длительностью и стоимостью использования [17]. Данный набор слотов образует «окно», для которого можно рассчитать суммарное используемое процессорное время и суммарную стоимость. Стоит отметить, что для нормализации их значений дополнительно требуется произвести пересчет на ресурс единичной производительности. Таким образом, суммарное процессорное время можно вычислить как произведение $p \cdot t \cdot n$. Для РВС рассчитывается суммарная длина слотов, также в пересчете на ресурс с производительностью,

принятой за единицу. Задания в пакет следует набирать таким образом, чтобы суммарное время занятия слотов не превышало суммарной длины слотов домена РВС, взятой с некоторым коэффициентом $\text{limitCoefficient} \in (0; 1]$. Администраторы ВО могут оперировать этим коэффициентом для регулирования процесса выполнения потока заданий в РВС.

Ограничение по стоимости аналогично ограничению по времени: максимальный бюджет S выполнения задания задается пользователем в ресурсном запросе. Для РВС в текущем цикле планирования рассчитывается суммарная стоимость доступных для использования слотов.

В отличие от пакета с фиксированным количеством заданий, ограничения по времени и по стоимости позволяют подстраивать размер пакета в условиях динамически изменяющейся загрузки узлов вычислительной среды и разнородности потока заданий. Для этого возможно задавать значение ограничивающего коэффициента $\text{limitCoefficient} \in (0; 1]$. Далее, в разделе 3, проводится экспериментальное исследование этого подхода.

2.2 Показатель совместимости задания и вычислительной среды

При отборе заданий в пакет можно руководствоваться различными принципами. Для увеличения дохода собственников ресурсов можно отбирать задания с наибольшими значениями доступного бюджета. Однако при выполнении потока, задания с меньшим бюджетом также должны быть гарантированно выполнены. Схемы формирования пакета, предлагаемые в данной работе, производят выбор заданий в пакет на основании совместимости характеристик задания и вычислительной среды. Таким образом, в пакет отбираются задания, которые по ресурсным запросам наилучшим образом подходят для выполнения в текущем интервале планирования. В качестве меры совместимости отдельного задания и домена ресурсов в РВС предлагается эмпирический коэффициент Dq (Distribution quality). Коэффициент Dq характеризует шансы успешного планирования и выполнения задания в текущем состоянии вычислительной среды. Dq может принимать как положительные (высокий шанс выполнения), так и отрицательные значения (низкий шанс выполнения). Для расчета коэффициента Dq и выделения значимых характеристик заданий и доменов вычислительных ресурсов были проведены экспериментальные исследования, основанные на моделировании нескольких тысяч циклов планирования. В результате выявлены следующие характеристики среды и параметры ресурсного запроса, которые оказывают наибольшее влияние на вероятность удачного исхода планирования.

1. Соотношение «цена/качество» узлов среды (Q_0) и заданий (Q). Для отдельного вычислительного узла Q_0 рассчитывается как отношение удельной стоимости использования ресурса к его показателю производительности $\frac{c}{p}$. Таким образом, чем выше производительность и меньше стоимость использования узла, тем ниже значение данного коэффициента. Для домена ресурсов берется среднее значение Q_0 по всем узлам. Для отдельного задания показатель формируется аналогично: $Q = \frac{S}{ntp}$, где среднее значение максимальной допустимой пользователем удельной стоимости одного узла равна $c = \frac{S}{nt}$.

2. Количество доступных ресурсов (узлов) n_0 в домене РВС и, соответственно, количество вычислительных n узлов, необходимых для выполнения задания.

3. Средняя длина слота l_s в интервале планирования в пересчете на ресурс единичной производительности и требуемая длительность резервирования ресурсов в пересчете на ресурс с базовой производительностью $t \cdot p$.

4. Суммарное процессорное время V_s в расчете на ресурс единичной производительности (суммарная длина всех используемых слотов), и необходимое для выполнения задания процессорное время $t \cdot p \cdot n$.

Таким образом, коэффициент Dq состоит из четырех основных слагаемых, соответствующих приведенным характеристикам домена РВС и пользовательского задания. Для каждого слагаемого вводятся подстроечные параметры: K_q , K_n , K_l и K_v – весовые коэффициенты слагаемых; C_q , C_n , C_l и C_v – пороговые значения, определяющие величину, при которой для задания будет найдена хотя бы одна альтернатива выполнения. Значения подстроечных параметров могут формироваться на основе статистики предыдущих циклов планирования или экспертной оценки.

Коэффициент Dq представляет собой сумму следующих слагаемых.

1. $Dq_1 = \frac{K_q}{C_q} \left(\frac{Q}{Q_0} - C_q \right)$. Слагаемое нормализует соотношение коэффициентов Q задания и Q_0 среды. Чем больше отношение $\frac{Q}{Q_0}$, тем выше значение данного слагаемого и выше вероятность успешного выполнения задания. К примеру, чем больше бюджет S , выделяемый пользователем для выполнения задания, тем больше будет значение данного слагаемого и всего показателя Dq .

2. $Dq_2 = \frac{K_n}{C_n} \left(C_n - \frac{n}{n_0} \right)$. Нормализует соотношение количества подходящих вычислительных узлов в домене ресурсов с количеством узлов, необходимых для выполнения задания $\frac{n}{n_0}$.

3. $Dq_3 = \frac{K_l}{C_l} \left(C_l - \frac{t \cdot p}{l_s} \right)$. Данное слагаемое нормализует отношение требуемого заданию времени резервирования ресурсов и средней длины слотов системы.

4. $Dq_4 = \frac{K_v}{C_v} \left(C_v - \frac{t \cdot p \cdot n}{V_s} \right)$. Определяет отношение необходимого для выполнения задания процессорного времени к суммарному процессорному времени в текущем цикле планирования.

Отметим, что параметры Q_0 , n_0 , l_s и V_s рассчитываются на множестве ресурсов домена, которые подходят заданию по ограничению на минимальную производительность p . Если таких ресурсов в рассматриваемом домене нет или же их количество меньше указанного в ресурсном запросе значения n , коэффициент Dq полагается равным минус бесконечности.

Выбирать задания в пакет при помощи коэффициента Dq возможно различными способами. В рамках одного из возможных подходов в пакет на каждом цикле планирования отбираются задания с максимальным значением Dq . Однако в таком случае возможна ситуация, аналогичная отбору по критерию стоимости: при успешном планировании наиболее «ценных» заданий потока в начальных циклах планирования возможно резкое снижение эффективности выполнения заданий, оставшихся в очереди, в последующих циклах планирования.

Методики формирования системы заданий, предлагаемые в данной работе, используют другую политику и основаны на отборе заданий с минимальным положительным значением Dq , т.е. самых «проблемных» заданий из тех, которые могут быть успешно выполнены в текущем интервале планирования. Данная политика позволяет сбалансировать выполнение потока заданий в течение множества циклов и обеспечить наиболее эффективное использование ресурсов. Задания же с очень высоким или отрицательным значением показателя Dq , могут быть перемещены в другие потоки и выполнены в других доменах вычислительных узлов РВС.

2.3 Методики формирования пакета заданий

В работе предлагаются две принципиально различные методики формирования пакета.

Первая из них сводит процесс формирования пакета заданий к решению задачи динамического программирования об оптимальном заполнении ранца. Данный подход представляется наиболее естественным, так как позволяет формализовать процедуру отбора заданий при заранее известных характеристиках заданий и домена РВС. Вторая методика основана на использовании коэффициента Dq и позволяет гибко подстраиваться под динамически изменяющийся состав ресурсов и заданий потока. Для сравнения обе методики используют описанный в разделе 2.1 подход к выбору ограничения размера пакета заданий, а выбор состава пакета осуществляется на основе вычисления показателя Dq для каждого задания.

Идея использования задачи о рюкзаке при организации планирования не нова, однако в известных подходах [16-18] она чаще всего используется для оптимальной аллокации заданий на неотчуждаемых ресурсах. Нами же предлагается использовать ее для заполнения пакета заданий, в качестве подготовительного этапа перед планированием.

Известно, что задача о рюкзаке формулируется следующим образом. Имеется набор предметов, каждый из которых характеризуется двумя основными параметрами: весом и ценностью. Имеется рюкзак ограниченной вместимости по весу. Задача состоит в том, чтобы собрать рюкзак с максимальной ценностью, соблюдая ограничение по весу. Нетрудно заметить, что модель задачи о рюкзаке хорошо подходит для описанной задачи формирования пакета заданий. Весовое ограничение пакета и вес каждого задания могут быть стоимостными или временными, в

зависимости от выбранного типа ограничения. Весовое ограничение выбирается на основе суммарных характеристик домена вычислительных ресурсов с некоторым коэффициентом $limitCoefficient$, как описано в разделе 2.1. Ценность каждого задания предлагается вычислять как $\frac{1}{Dq}$: то есть, чем меньше значение коэффициента Dq , тем больше ценность задания. Данное предположение основано на логике выбора коэффициента Dq , описанной в разделе 2.2, и позволяет обеспечить более равномерную загрузку ресурсов в течение нескольких циклов планирования. Можно отметить, что задания, ценность которых меньше или равна нулю, никогда не попадут в пакет, так как они не вносят положительного вклада в суммарную ценность пакета, но при этом используют часть «полезного веса».

При наборе пакета заданий на основе описанного выше показателя совместимости задания и домена ресурсов использование лишь коэффициента Dq становится явно недостаточным: при добавлении задания в пакет необходимо учитывать параметры уже отобранных заданий. Для учета вклада заданий, находящихся в пакете, необходимо внести дополнительные слагаемые. Например, в случае использования ограничения по времени, в слагаемое, нормализующее отношение процессорного времени, необходимого заданию и свободного процессорного времени в домене, добавляется суммарное процессорное время заданий, отобранных в пакет ранее. В этом случае $Dq_4 = \frac{K_v}{C_v} (C_v - \frac{t \cdot p \cdot n + \sum_{i=1}^N t_i \cdot p_i \cdot n_i}{V'_s})$, где сумма берется по всем N заданиям, уже отобранным в пакет, а V'_s – суммарное процессорное время для всех заданий. В случае, если используется ограничение по стоимости, в числителе этого отношения присутствует суммарный бюджет выполнения уже отобранных заданий пакета, а в знаменателе – суммарная стоимость всех используемых слотов.

Таким образом, самый высокий приоритет имеют задания с минимальным положительным значением коэффициента Dq . При увеличении количества заданий в пакете значение Dq_4 уменьшается и может принимать отрицательные значения. Формирование пакета продолжается до тех пор, пока в потоке остаются задания с положительным значением коэффициента Dq .

Следует отметить, что в данной методике формирования пакета ограничивающим коэффициентом, которым оперируют администраторы ВО, является подстроечный коэффициент C_v .

В отличие от методики формирования пакета на основе решения задачи о рюкзаке, ограничение размера пакета для данной методики не является жестким. При решении задачи о рюкзаке ограничение является строгим и не может быть нарушено. При использовании данной методики превышение ограничения приведет к тому, что Dq_4 примет отрицательное значение, однако сам коэффициент Dq может остаться положительным, и тогда задание попадет в пакет.

Подход на основе модифицированного коэффициента Dq позволяет более гибко соблюдать политику выбора задания, так как всякий раз выбирает задание с минимальным положительным значением коэффициента. Методика, решающая задачу о рюкзаке, оптимизирует сумму ценностей, при этом ценность отдельного задания может и не быть самой высокой. Основное преимущество методики на основе коэффициента Dq состоит в возможности учета характеристик заданий, уже отобранных в пакет. Несмотря на то, что в данной работе рассматриваются такие характеристики, как процессорное время (или стоимость) уже находящихся в пакете заданий, на практике можно учесть и другие показатели. Модель задачи о рюкзаке не предполагает возможности учета характеристик находящихся в пакете заданий, так как формирует пакет целиком, на основе предварительно вычисленного значения Dq для каждого задания. Следует отметить, что для обеих описанных методик формирования пакета используется ранжирование заданий в соответствии с коэффициентом Dq по убыванию, т.е. в начало пакета помещаются самые «проблемные» задания. Это позволяет улучшить показатели планирования.

3. Экспериментальные исследования

Эффективность предложенных методик формирования пакета заданий исследуется с использованием симулятора РВС [19]. Эксперимент заключается в планировании потока заданий в домене вычислительных ресурсов с использованием различных методик формирования пакета и с различными настройками. Исследуются следующие схемы формирования пакетов заданий:

1. Random – в пакет попадает фиксированное число заданий, которые выбираются из потока случайным образом;
2. KnapsackT – пакет заданий формируется на основе решения задачи о рюкзаке, при этом используется ограничение по времени;
3. KnapsackC – задания в пакет отбираются на основе решения задачи о рюкзаке с ограничением по стоимости;
4. MDqT – формирование пакета с использованием коэффициента Dq , вводится ограничение по времени;
5. MDqC – формирование пакета на основе коэффициента Dq с ограничением по стоимости.

Планирование потока в каждом эксперименте осуществляется циклично: перед началом цикла планирования из потока отбирается пакет заданий, а затем происходит его планирование и выполнение. Задания, которые не удастся выполнить в данном цикле планирования, возвращаются в поток. Циклический процесс планирования продолжается до тех пор, пока все задания потока не будут выполнены. На основе проведенных экспериментальных исследований получены средние значения показателей эффективности планирования в зависимости от используемой схемы формирования пакета заданий. Кроме того, исследована эффективность планирования при различных значениях ограничивающего коэффициента `limitCoefficient`. Для схемы Random размер пакета выбирался в соответствии со средним значением размера пакетов, набранных с помощью остальных подходов.

В таблице 1 приведены значения ограничений, введенных при моделировании.

Таблица 1. Параметры схем формирования пакета в зависимости от серии экспериментов

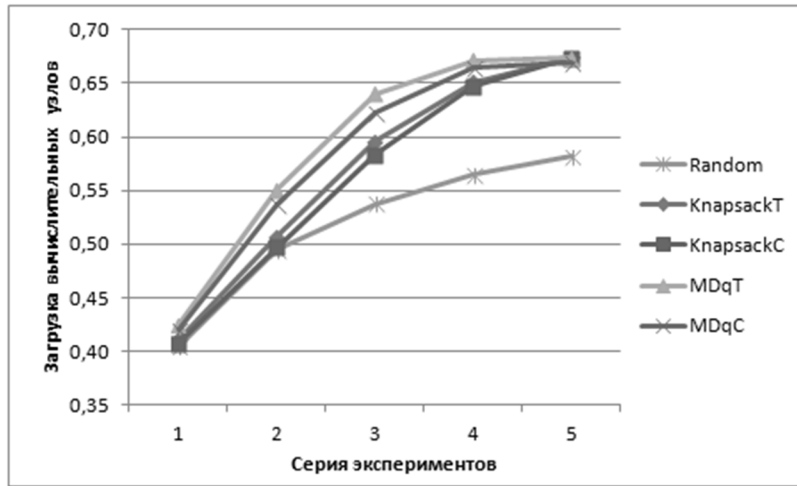
№ серии экспериментов	1	2	3	4	5
Размер пакета для схемы Random	6	20	30	40	50
Значение <code>limitCoefficient</code> для схем KnapsackT, KnapsackC, MDqT, MDqC	0,1	0,3	0,5	0,7	0,9

Оценим эффективность рассматриваемых схем формирования пакета по следующим критериям: уровень загрузки вычислительных узлов, время выполнения задания, количество альтернатив выполнения задания, число циклов планирования, требуемых для выполнения потока заданий.

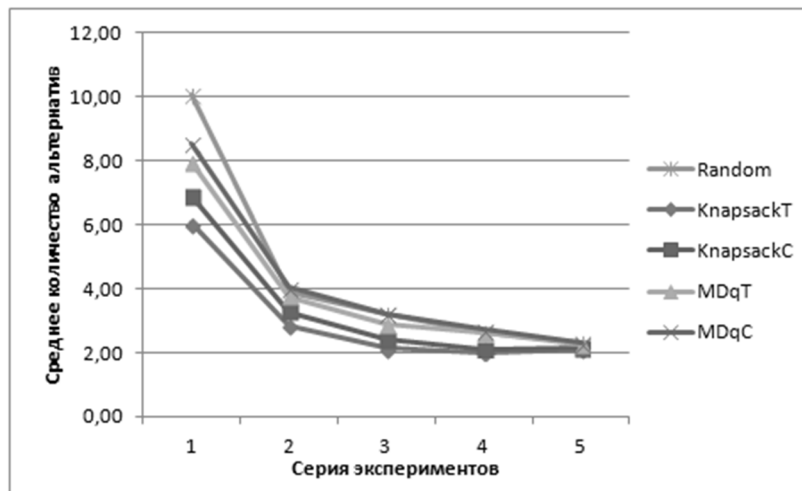
На рис. 1 (а, б) представлены значения средней загрузки домена ресурсов и среднего числа альтернатив выполнения заданий для проведенных серий экспериментов. Из графиков видно, что при увеличении размера пакета увеличивается загрузка вычислительной среды и уменьшается количество возможных альтернатив выполнения заданий.

Как было отмечено выше, рассматриваемые показатели эффективности планирования взаимозависимы. Улучшение одного показателя, как правило, ведет к ухудшению значения другого. Однако в случае, если из двух подходов к формированию пакета один дает лучшее значение одного из критериев при равнозначных значениях другого, то можно утверждать, что первый подход эффективнее. Таким образом, сравнение эффективности рассматриваемых подходов имеет смысл проводить относительно значения одного из них. Рассмотрим в качестве базового параметра значение среднего уровня загрузки вычислительных узлов.

На рис. 2 (а, б, в) представлены соответственно время выполнения, количество альтернатив и среднее число циклов планирования в зависимости от загрузки узлов. В разделе 2 было отмечено, что подход на основе решения задачи о рюкзаке выполняет ограничение размера пакета, определяемое коэффициентом `limitCoefficient`. Это наглядно видно из рис. 1 (а). С другой стороны, за счет наличия нескольких слагаемых в показателе Dq даже при превышении ограничения в слагаемом Dq_4 (по стоимости или по времени) из-за заполненности пакета, коэффициент может сохранять положительное значение. В результате этого в схемах MDqT и MDqC в пакет попадает большее число заданий, которые могут быть успешно спланированы. Это объясняет лучшие значения критериев и более высокий уровень загрузки вычислительных узлов, полученные при использовании данного подхода. Из рис. 2 (а) видно, что схемы MDqT и MDqC обеспечивают преимущество над KnapsackT и KnapsackC по времени выполнения заданий (критерий BO) в среднем на 15% при примерно равном количестве циклов планирования.



(а)



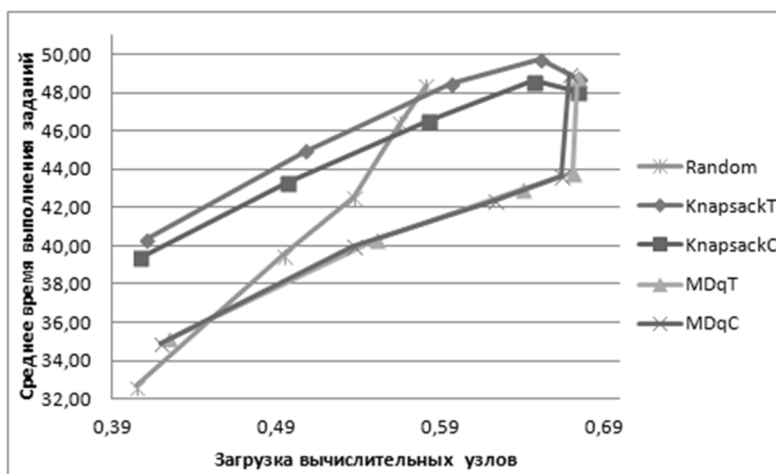
(б)

Рис. 1. Средняя загрузка узлов (а) и количество альтернатив (б) при планировании потока в разных сериях экспериментов

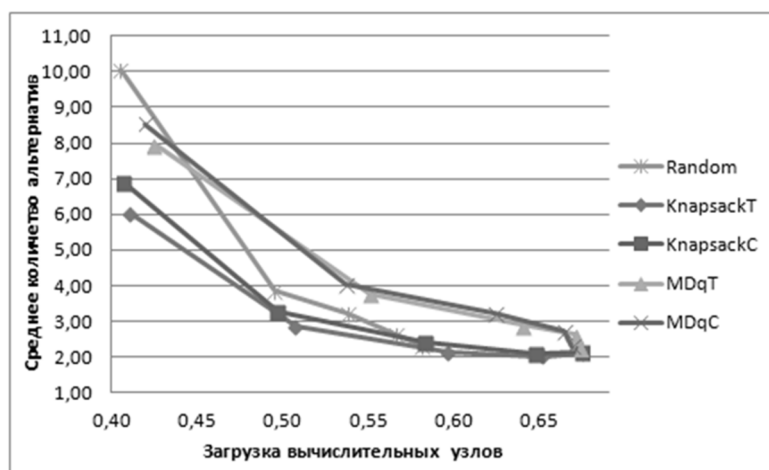
Как можно видеть из рис. 2 (а), почти на всем рассматриваемом интервале по оси загрузки ресурсов лучший результат обеспечивают схемы MDqT и MDqC. Графики на рис. 2 (б) подтверждают результаты, представленные на рис. 2 (а): подходы, обеспечившие наилучшие значения критерия ВО, позволяют получить и большее количество альтернатив для выполнения заданий пакета. Средние значения количества циклов планирования в зависимости от загрузки вычислительных узлов (рис. 2 (в)) в этих схемах примерно одинаковы.

На рис. 3 приведены средние значения числа альтернатив выполнения задания, количества заданий в пакете и числа возвратов в поток для рассматриваемых подходов по циклам планирования в серии экспериментов № 3 (табл. 1). Как было отмечено выше, схемы с использованием коэффициента Dq отбирают задания с минимальным положительным значением этого коэффициента. Это означает, что в пакет прежде всего попадают задания, требующие больше ресурсов или имеющие меньшее значение коэффициента «цена/качество» Q . Поэтому в начальных циклах планирования для этих схем размер пакета меньше, а число возвратов больше, чем для схем KnapsackT и KnapsackC (далее KnapsackT(C)). На последующих циклах планирования для схем MDqT и MDqC (далее MDqT(C)) в потоке остаются менее «проблемные» для планирования задания, что следует из значительного уменьшения числа возвратов. Кроме того, схемы на основе Dq обеспечивают большее количество альтернатив выполнения на каждом цикле планирования по сравнению с KnapsackT(C) несмотря на то, что средний размер пакета для схем 4 и 5 оказался больше, чем для схем 2 и 3 (38,4 и 35,7 против 30,0 и 28,7 заданий), а для выполнения потока заданий понадобилось на один цикл планирования меньше. В целом, схемы

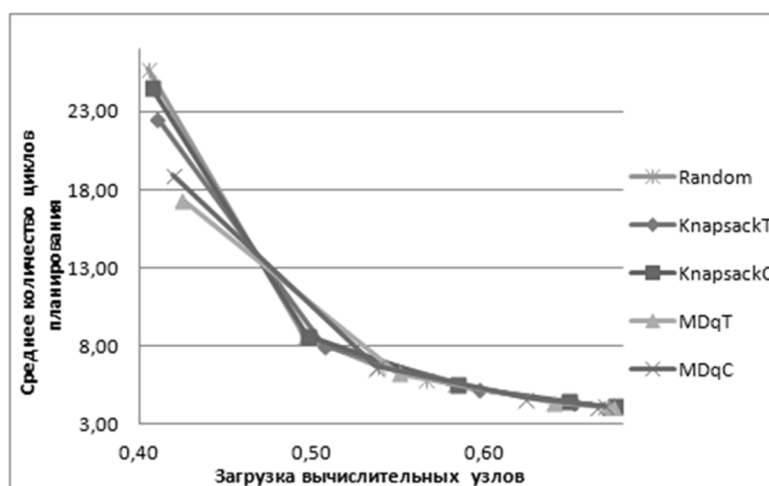
MDqT(C) действуют более агрессивно, отбирая менее «перспективные» задания в начальных циклах, чем и обеспечивается большая эффективность.



(а)

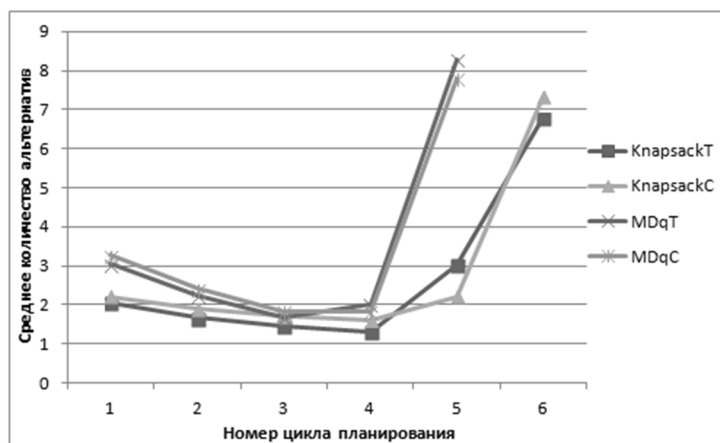


(б)

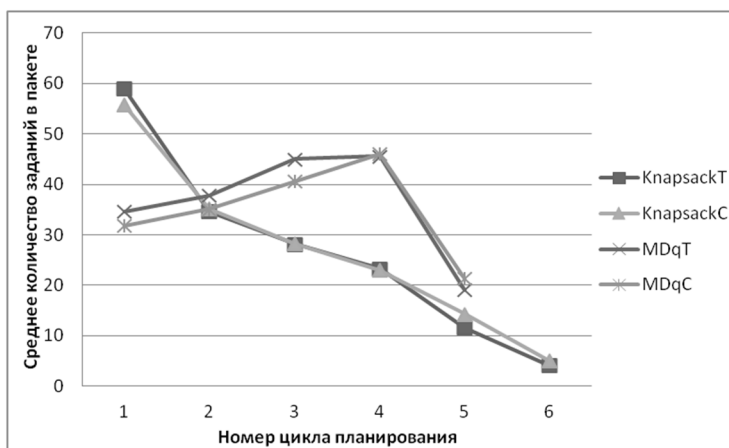


(в)

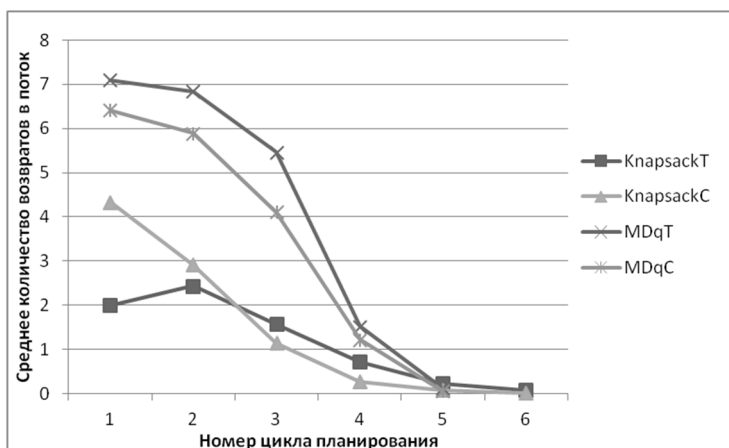
Рис. 2. Средние значения времени выполнения (а), количества альтернатив (б), числа циклов планирования (в) в зависимости от средней загрузки узлов



(а)



(б)



(в)

Рис. 3. Средние значения количества альтернатив на задание (а), заданий в пакете (б) и возвратов в поток (в) в зависимости от цикла планирования

Полученные результаты позволяют сделать вывод о том, что для схем на основе решения задачи о рюкзаке лучшие значения рассматриваемых критериев получены с использованием ограничения по стоимости.

Для подходов MDqT(C) принципиальных различий в зависимости от типа ограничения не отмечается.

Схема случайного формирования пакета Random в некоторых экспериментах обеспечивает наилучшее значение таких критериев, как время выполнения заданий или количество альтернатив выполнения. Однако, объясняется это тем, что пакет заданий набирается без учета «совместимости» заданий и состояния домена ресурсов, а единственным ограничением является предварительно заданный размер пакета.

При фиксированном размере пакета задания группируются неоднородно: их суммарные показатели могут существенно быть больше или меньше ограничения, используемого, например, в KnapsackT(C). Как следствие, получается относительно низкий уровень загрузки вычислительных узлов и большее число циклов планирования, необходимых для выполнения всего потока. Из этого можно сделать вывод о неэффективном использовании ресурсов данной схемой.

4. Заключение

В работе исследована проблема формирования системы заданий при планировании ресурсов в виртуальной организации распределенной вычислительной среды. Для повышения эффективности планирования в рамках циклической схемы предложен общий вид коэффициента «совместимости» Dq задания и домена РВС. Предложены и исследованы две методики формирования пакета заданий, основанные на принципиально различных подходах к отбору заданий на основе Dq .

Одна из предложенных методик KnapsackT(C) формирует пакет на основе решения задачи о рюкзаке для предварительно вычисленного значения Dq для каждого задания и имеет жесткое ограничение на суммарное время или стоимость выполнения заданий пакета.

Другой подход MDqT(C) использует для отбора заданий коэффициент Dq , который динамически изменяется в зависимости от характеристик заданий, уже находящихся в пакете, и использует более мягкое ограничение на размер пакета заданий.

Результаты экспериментов демонстрируют значительное преимущество эвристического подхода MDqT(C) над подходом на основе решения задачи о рюкзаке KnapsackT(C). Объясняется это, прежде всего, тем, что в условиях динамически изменяющегося состава и локального расписания вычислительных узлов, а также разнородности заданий потока, важно учитывать как можно больше динамических параметров состояния среды. Таким образом, мягкое ограничение на размер пакета и динамично изменяемое значение коэффициента Dq позволяет подходу MDqT(C) обеспечить наилучшие значения рассматриваемых критериев эффективности выполнения потока заданий в РВС.

Следует отметить, что обе предложенные методики обеспечили лучшие результаты планирования и большую эффективность использования доступных ресурсов по сравнению с традиционной схемой формирования пакета заданий случайным образом.

Дальнейшие исследования будут посвящены вопросам формирования системы заданий с учетом предпочтений всех участников вычислений ВО.

Литература

1. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International J. Supercomputer Applications. 2001. Vol. 15, No. 3. P. 200-220.
2. Garg S.K., Buyya R., Siegel H.J. Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management // 32nd Australasian Computer Science Conference, Wellington, New Zealand, Proceedings. 2009. P. 151-159.
3. Buyya R., Abramson D., Giddy J. Economic Models for Resource Management and Scheduling in Grid Computing // J. Concurrency and Computation. 2002. Vol. 14, No. 5. P. 1507–1542.
4. Топорков В.В., Емельянов Д.М. Экономическая модель планирования и справедливого распределения ресурсов в распределенных вычислениях // Программирование. 2014. № 1. С. 54-65.

5. 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. ТиСУ. 2014. № 5. С. 95-105.
6. Mutz A.; Wolski R., Brevik J. Eliciting Honest Value Information in a Batch-queue Environment // 2007 8th IEEE/ACM International Conference on Grid Computing. 2007. P. 291–297.
7. Berman F., Wolski R., Casanova H. et al. Adaptive Computing on the Grid Using AppLeS // IEEE Trans. On Parallel and Distributed Systems (TPDS). 2003. Vol.14, No.4. P. 369-382.
8. Cirne W., Brasileiro F., Costa L. et al. Scheduling in Bag-of-task Grids: The PAUÁ Case // 16th Symposium on Computer Architecture and High Performance Computing, 2004. P. 124–131.
9. Воеводин Вл.В., Жолудев Ю.А., Соболев С.И., Стефанов К.С. Эволюция системы метакомпьютинга X-Com // Вестник Нижегородского университета им. Н.И. Лобачевского. 2009. № 4. С. 157–164.
10. Dail H., Sievert O., Berman F. et al. Scheduling in the Grid Application Development Software Project // Grid resource management. State of the Art and Future Trends. Eds J. Nabrzyski, J.M. Schopf and J. Weglarz. Kluwer Acad. Publ. 2003. P. 73 – 98.
11. Kurowski K., Oleksiak A., Nabrzyski J. et al. Multi-criteria Grid Resource Management Using Performance Prediction Techniques // In: Integrated Research in GRID Computing. Eds. S. Gorlatch and M. Danelutto. Springer. 2007. P. 215-225.
12. Moab Adaptive Computing Suite.
URL: <http://www.adaptivecomputing.com/products/moab-adaptive-computing-suite.php> (дата обращения: 27.11.2014).
13. Kannan S., Roberts M., Mayes P. et al. Workload Management with LoadLeveler. IBM, First ed., November 2001. 210 p.
14. Tsafir D., Etsion Y., Feitelson D. Backfilling Using System-generated Predictions Rather than User Runtime Estimates // IEEE Transactions on Parallel and Distributed Systems. 2007. Vol. 18, No. 6. P. 789–803.
15. Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D., Potekhin P. Preference-Based Fair Resource Sharing and Scheduling Optimization in Grid VOs // Procedia Computer Science. 2014. Vol. 29. P. 831–843.
16. Zhou Z., Lan Z., Tang W., Desai N. Reducing Energy Costs for IBM Blue Gene/P via Power-Aware Job Scheduling // Seventeenth Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), May 24, 2013, Boston, Massachusetts, USA. 2013. P. 96-115.
17. Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D. Slot Selection Algorithms in Distributed Computing // Journal of Supercomputing. 2014. Vol. 69, No. 1. P. 53–60.
18. Soner S., Özturan C. Integer Programming Based Heterogeneous CPU-GPU Cluster Scheduler for SLURM Resource Manager // Fourteenth IEEE International Conference on High Performance Computing and Communication & Ninth IEEE International Conference on Embedded Software and Systems, June 25 - 27, 2012, Liverpool, United Kingdom. IEEE. 2012. P. 418-424.
19. 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.