

ФОРМИРОВАНИЕ И ПЛАНИРОВАНИЕ ПАКЕТОВ ЗАДАНИЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ¹

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

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.

Ключевые слова: распределенные вычисления, планирование, слот, циклическая схема планирования, пакет заданий, формирование пакета.

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 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]. 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), бэкфиллинг, механизмы вычисления приоритетов пользователей и заданий, и разделения ресурсов [12–14]. Важным аспектом в этих подходах является преимущественное соблюдение очереди и пользова-

¹ Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии – 2015».

тельских 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. Таким 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. 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.

1. Формирование системы заданий

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 пакеты 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].

1.1. 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.
- 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.

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 рассчи-

тать 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.

1.2. Показатель совместимости задания и вычислительной среды

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 $\frac{c}{p}$. Для домена ресурсов берется среднее Q_0 значение Q_0 по всем узлам. Для отдельного задания показатель формируется аналогично: $Q = \frac{s}{ntp}$.
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 $t \cdot p \cdot n$.

Коэффициент Dq состоит из четырех 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 определяется как сумма следующих слагаемых.

$$Dq_1 = \frac{K_q}{c_q} \left(\frac{Q}{Q_0} - C_q \right) \quad (1)$$

Слагаемое нормализует соотношение коэффициентов Q задания и Q_0 среды.

$$Dq_2 = \frac{K_n}{c_n} \left(C_n - \frac{n}{n_0} \right) \quad (2)$$

Нормализует соотношение количества подходящих вычислительных узлов в домене ресурсов с количеством узлов, необходимых для выполнения задания $\frac{n}{n_0}$.

$$Dq_3 = \frac{K_l}{c_l} \left(C_l - \frac{t \cdot p}{l_s} \right) \quad (3)$$

Данное слагаемое нормализует отношение требуемого заданию времени резервирования ресурсов и средней длины слотов системы.

$$Dq_4 = \frac{K_v}{c_v} \left(C_v - \frac{t \cdot p \cdot n}{v_s} \right) \quad (4)$$

Определяет отношение необходимого для выполнения задания процессорного времени к суммарному процессорному времени в текущем цикле планирования.

Выбирать задания в пакет при помощи коэффициента Dq , представляющего собой сумму (1) – (4), возможно различными способами. Например, отбирать задания с максимальным значением Dq . Однако в таком случае возможна ситуация, когда в первых циклах будут распланированы самые «ценные» задания, и на последующих циклах эффективность планирования резко упадет. Поэтому 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.

1.3. 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.

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. 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. 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».

Другой подход, рассматриваемый в работе, предполагает формирование пакета последовательно по одному заданию. Подход использует коэффициент Dq , но данный коэффициент изменен, так чтобы учитывать задания, уже находящиеся в пакете. Например, в случае использования ограничения по времени, слагаемое Dq_4 вычисляется следующим образом:

$$Dq_4 = \frac{K_v}{C_v} \left(C_v - \frac{t \cdot p \cdot n + \sum_{i=1}^N t_i \cdot p_i \cdot n_i}{V'_s} \right) \quad (5)$$

Сумма в (5) берется по всем N заданиям, уже отобранному в пакет, а V'_s – суммарное процессорное время для всех задний. 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. Отметим, 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.

2. Экспериментальные исследования

Эффективность предложенных методик исследуется с использованием симулятора PBC [19]. 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. 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 приведены значения ограничений, введенных при моделировании.

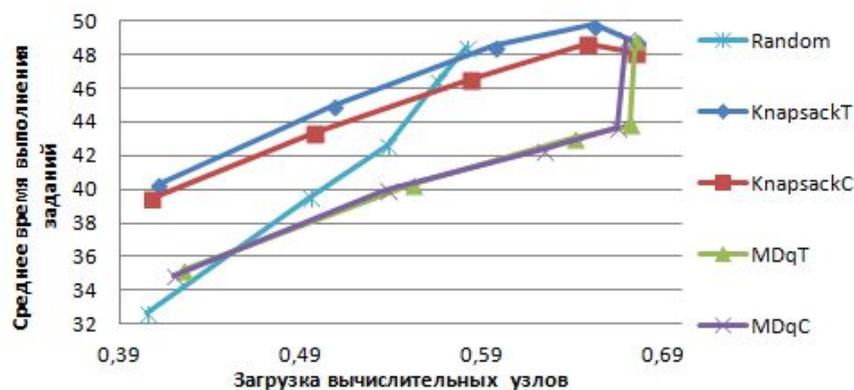
Таблица

Параметры схем формирования пакета в зависимости от серии экспериментов

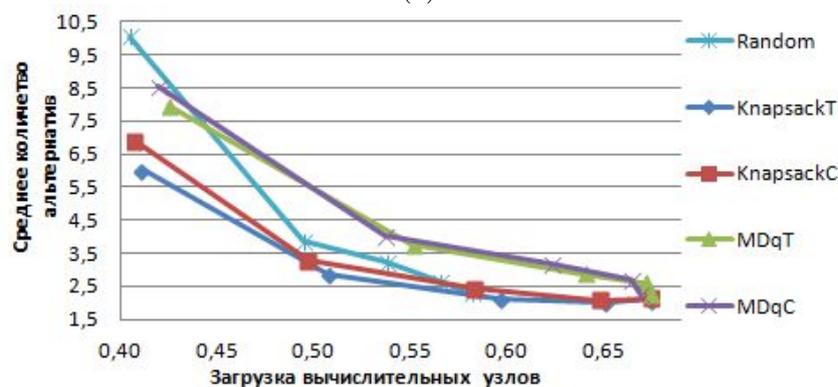
№ серии экспериментов	1	2	3	4	5
Размер пакета для схемы Random	6	20	30	40	50
Значение <code>limitCoefficient</code> для схем KnapsackT, KnapsackC, MDqT, MDqC	0,1	0,3	0,5	0,7	0,9

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.

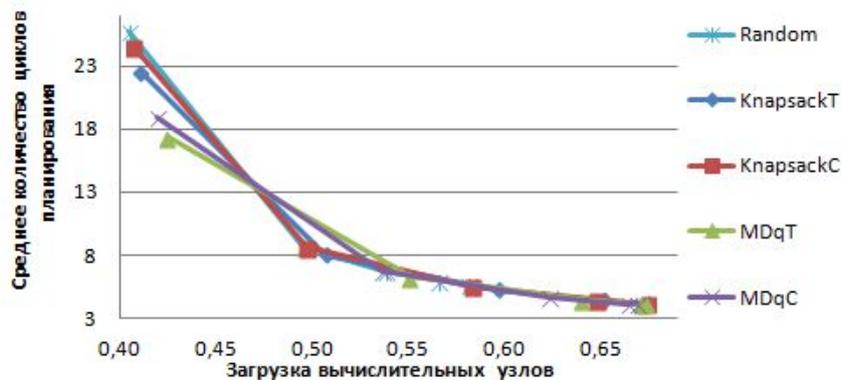
На рис. 1 (а, б, в) представлены соответственно время выполнения, количество альтернатив и среднее число циклов планирования в зависимости от загрузки узлов. За счет наличия нескольких слагаемых в показателе Dq даже при превышении ограничения в слагаемом Dq_4 (по стоимости или по времени) из-за заполненности 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. 1 (а) видно, что схемы MDqT и MDqC обеспечивают преимущество над KnapsackT и KnapsackC по времени выполнения заданий (критерий ВО) в среднем на 15% при примерно равном количестве циклов планирования.



(а)



(б)



(в)

Рис. 1. Средние значения времени выполнения (а), количества альтернатив (б), числа циклов планирования (в) в зависимости от средней загрузки узлов

Как можно видеть из рис. 1 (а), почти на всем рассматриваемом интервале по оси загрузки ресурсов лучший результат обеспечивают схемы MDqT и MDqC. Графики на рис. 1 (б) подтверждают результаты, представленные на рис. 1 (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. 1 (в)) в этих схемах примерно одинаковы.

На рис. 2 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 положи-

тельным значением этого коэффициента. При этом «вес» задания (необходимое ему процессорное время или бюджет выполнения) не принимается в расчет: задания отбираются из потока равным образом в соответствии со значением коэффициента Dq , пока не будет превышено ограничение, заданное слагаемым (4).

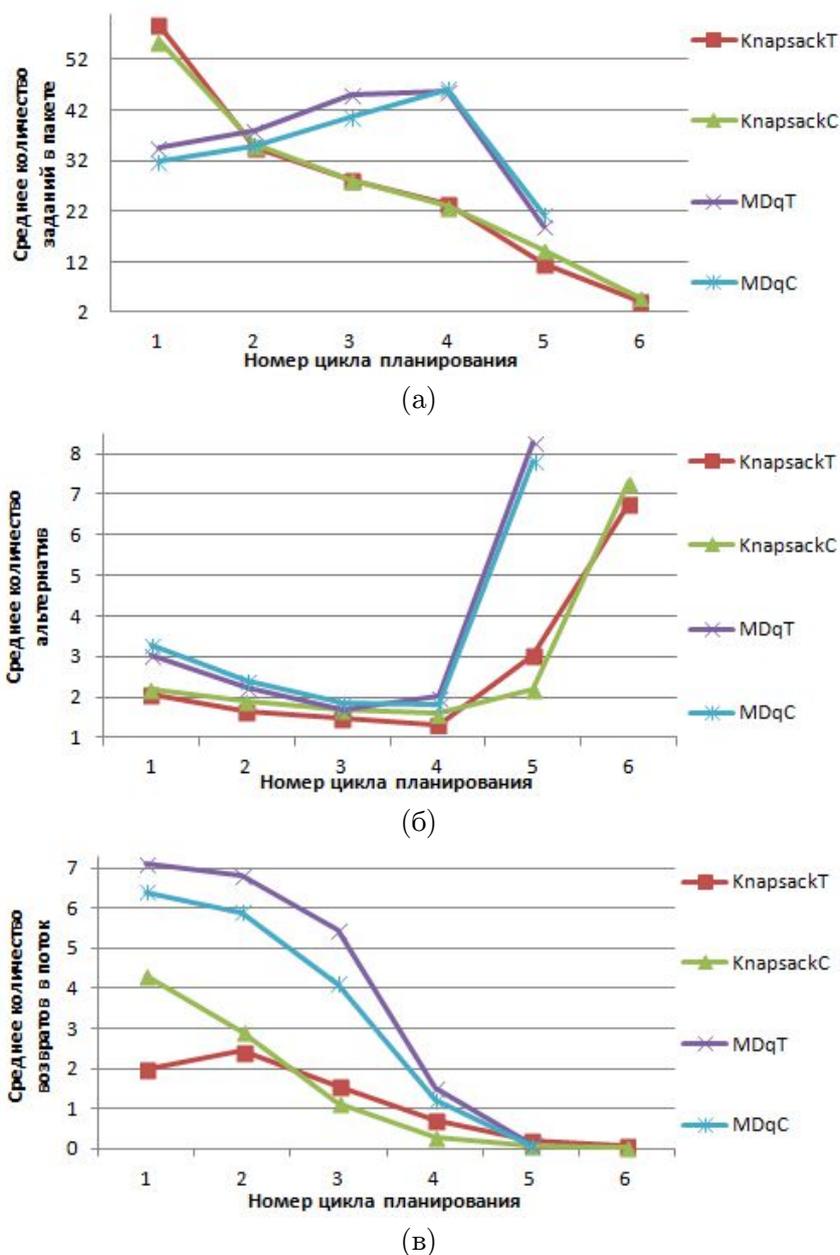


Рис. 2. Средние значения количества альтернатив на задание (а), заданий в пакете (б) и возвратов в поток (в) в зависимости от цикла планирования

С другой стороны, подход KnapsackT(C) максимизирует сумму значений Dq , используя строгое весовое ограничение. Оказывается, что данная политика выбирает задания, требующие относительно меньшее количество ресурсов, т.к. такие задания вносят меньший вклад в общий вес пакета. Поэтому планирование потока происходит неравномерно: на первых циклах планирования отбираются относительно небольшие задания, тогда как задания, требующие больше ресурсов, остаются в потоке и планируются последними (что можно видеть на рис. 2). На первых циклах KnapsackT(C) формирует пакеты большего размера, а на последующих циклах с тем же весовым ограничением размер

пакета уменьшается в несколько раз. Аналогично можно объяснить количество альтернатив, найденных подходом $\text{KnapsackT}(C)$. На первых циклах планирования большое число относительно небольших заданий конкурируют за ограниченные ресурсы, что приводит к фрагментации ресурсов. Фрагментация усложняет задачу поиска альтернатив на первых циклах планирования, а на последующих циклах становится сложным выделить даже небольшое число альтернатив для заданий, т.к. оставшиеся задания требуют больше ресурсов. С другой стороны, планирование на основе подхода $\text{MDqT}(C)$ происходит более равномерно: на рис. 2 (а) видно, что значения графиков $\text{MDqT}(C)$ относительно меньше изменяются от цикла к циклу. Резкое уменьшение числа возвратов в последних циклах планирования для данного подхода можно объяснить тем, что в потоке остаются задания с относительно более высоким значением Dq (т.е. более «благоприятные» задания), и шанс удачного планирования для таких заданий выше.

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. С другой стороны, для подходов $\text{MDqT}(C)$ 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 $\text{KnapsackT}(C)$. 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.

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 $\text{KnapsackT}(C)$ 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 $\text{MDqT}(C)$ 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 эксперимента демонстрируют значительное преимущество эвристического подхода $\text{MDqT}(C)$ над подходом на основе решения задачи о рюкзаке $\text{KnapsackT}(C)$. Объясняется это, прежде всего, тем, что в условиях динамически изме-

няющегося 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.

Работа выполнена при содействии Совета по грантам Президента Российской Федерации для поддержки молодых российских ученых и ведущих научных школ (шифры НШ-362.2014.9, МК-4148.2015.9), 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).

Литература

1. Foster I. The Anatomy of the Grid: Enabling Scalable Virtual Organizations / I. Foster, C. Kesselman, S. Tuecke // International J. Supercomputer Applications. — 2001. — Vol. 15, No. 3. — P. 200–220. DOI: 10.1177/109434200101500302
2. Garg S.K. Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management / S.K. Garg, R. Buyya, H.J. Siegel // 32nd Australasian Computer Science Conference, Wellington, New Zealand, Proceedings. — 2009. — Vol. 91. — P. 151–159.
3. Buyya R. Economic Models for Resource Management and Scheduling in Grid Computing / R. Buyya, D. Abramson, J. Giddy // J. Concurrency and Computation. — 2002. — Vol. 14, No. 5. — P. 1507–1542. DOI: 10.1002/cpe.690
4. Топорков В.В. Экономическая модель планирования и справедливого разделения ресурсов в распределенных вычислениях / В.В. Топорков, Д.М. Емельянов // Программирование. — 2014. — № 1. — С. 54–65.
5. Мультиагентный подход к управлению распределенными вычислениями в кластерной GRID-системе / В.Г. Богданова, И.В. Бычков, А.С. Корсуков // Известия РАН. ТИСУ. — 2014. — № 5. — С. 95–105. DOI: 10.7868/s0002338814040039
6. Mutz A. Eliciting Honest Value Information in a Batch-queue Environment / A. Mutz, R. Wolski, J. Brevik // 2007 8th IEEE/ACM International Conference on Grid Computing. — 2007. — P. 291–297. DOI: 10.1109/GRID.2007.4354145
7. Adaptive Computing on the Grid Using AppLeS / F. Berman, R. Wolski, H. Casanova et al. // IEEE Trans. On Parallel and Distributed Systems. — 2003. — Vol. 14, No. 4. — P. 369–382. DOI: 10.1109/TPDS.2003.1195409
8. Scheduling in Bag-of-task Grids: The PAUÁ Case / W. Cirne, F. Brasileiro, L. Costa et al. // 16th Symposium on Computer Architecture and High Performance Computing. — 2004. — P. 124–131. DOI: 10.1109/SBAC-PAD.2004.37

9. Эволюция системы метакомпьютинга X-Com / Вл.В. Воеводин, Ю.А. Жолудев, С.И. Соболев, К.С. Стефанов // Вестник Нижегородского университета им. Н.И. Лобачевского. — 2009. — № 4. — С. 157–164.
10. Scheduling in the Grid Application Development Software Project / H. Dail, O. Sievert, F. Berman et al. // Grid resource management. State of the Art and Future Trends. / Eds J. Nabrzyski, J.M. Schopf, J. Weglarz. — Kluwer Acad. Publ., 2003. — P. 73–98.
11. Multi-criteria Grid Resource Management Using Performance Prediction Techniques / K. Kurowski, A. Oleksiak, J. Nabrzyski et al. // Integrated Research in GRID Computing. / Eds. S. Gorlatch, M. Danelutto. — Springer, 2007. — P. 215–225.
12. Moab HPC Suite Enterprise Edition. URL: <http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-suite-enterprise-edition> (дата обращения: 08.02.2014).
13. Workload Management with LoadLeveler. / S. Kannan, M. Roberts, P. Mayes et al. — IBM, First ed., 2001. — 210 p.
14. Tsafrir D. Backfilling Using System-generated Predictions Rather than User Runtime Estimates / D. Tsafrir, Y. Etsion, D. Feitelson // IEEE Transactions on Parallel and Distributed Systems. — 2007. — Vol. 18, No. 6. — P. 789–803.
15. Preference-Based Fair Resource Sharing and Scheduling Optimization in Grid VOs / V. Toporkov, A. Toporkova, A. Tselishchev et al. // Procedia Computer Science. — 2014. — Vol. 29. — P. 831–843. DOI: 10.1016/j.procs.2014.05.075
16. Reducing Energy Costs for IBM Blue Gene/P via Power-Aware Job Scheduling / Z. Zhou, Z. Lan, W. Tang, N. Desai // Seventeenth Workshop on Job Scheduling Strategies for Parallel Processing. — May 2013. — P. 96–115.
17. Slot Selection Algorithms in Distributed Computing / V. Toporkov, A. Toporkova, A. Tselishchev, D. Yemelyanov // Journal of Supercomputing. — 2014. — Vol. 69, No. 1. — P. 53–60. DOI: 10.1007/s11227-014-1210-1
18. Soner S. Integer Programming Based Heterogeneous CPU-GPU Cluster Scheduler for SLURM Resource Manager / S. Soner, C. Özturan // Fourteenth IEEE International Conference on High Performance Computing and Communication & Ninth IEEE International Conference on Embedded Software and Systems. — June 2012. — P. 418–424.
19. Методы и эвристики планирования в распределенных вычислениях с неотчуждаемыми ресурсами / В.В. Топорков, А.В. Бобченков, Д.М. Емельянов, А.С. Целищев // Вестник ЮУрГУ, серия «Вычислительная математика и информатика». — 2014. — Т. 3., № 2. — С. 43–62.

Топорков Виктор Васильевич, д.т.н., профессор, заведующий кафедрой Вычислительной техники, Национальный исследовательский университет «МЭИ» (Москва, Российская Федерация), ToporkovVV@mpei.ru.

Емельянов Дмитрий Михайлович, к.т.н., ассистент, кафедра Вычислительной техники, Национальный исследовательский университет «МЭИ» (Москва, Российская Федерация), YemelyanovDM@mpei.ru.

Потехин Петр Анатольевич, аспирант, кафедра Вычислительной техники, Национальный исследовательский университет «МЭИ» (Москва, Российская Федерация), PotekhinPA@mpei.ru.

Поступила в редакцию 17 марта 2015 г.

JOB BATCH GENERATION AND SCHEDULING IN DISTRIBUTED COMPUTING ENVIRONMENTS

V.V. Toporkov, National Research University "MPEI" (Moscow, Russian Federation)
ToporkovVV@mpei.ru,

D.M. Yemelyanov, National Research University "MPEI" (Moscow, Russian Federation)
YemelyanovDM@mpei.ru,

P.A. Potekhin, National Research University "MPEI" (Moscow, Russian Federation)
PotekhinPA@mpei.ru

The paper considers approaches to ranked jobs system generation in the model of cyclic scheduling in the virtual organizations of distributed computing environments. Two different methodologies of job selection for scheduling are proposed and compared: the first one is based on the solution of knapsack problem, the second one utilizes a heuristic "compatibility" indicator of jobs and computing domains. Experimental results that allow estimating of the proposed solutions efficiency are presented, the solutions are compared to random job selection.

Keywords: distributed computing, scheduling, slot, cyclic scheduling scheme, job batch, job generation.

References

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. DOI: 10.1177/109434200101500302
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. Vol. 91. 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. DOI: 10.1002/cpe.690
4. Toporkov V.V., Yemelyanov D.M. Ekonomicheskaya model' planirovaniya i spravedlivogo razdeleniya resursov v raspredelennykh vychisleniyakh [Economic model of scheduling and fair resource sharing in distributed computing]. Programmirovaniye [Programming and Computer Software]. 2014. No. 1. P. 54–65.
5. Bogdanova V.G., Bychkov I.V., Korsukov A.S. Mul'tiagentnyy podkhod k upravleniyu raspredelennymi vychisleniyami v klasternoy GRID-sisteme [Multiagent approach to distributed computing management in a cluster GRID system]. Izvestiya RAN. Teoriya i sistemy upravleniya [Journal of Computer and Systems Sciences International]. 2014. No. 5. P. 95–105. DOI: 10.7868/s0002338814040039
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. DOI: 10.1109/GRID.2007.4354145

7. Berman F., Wolski R., Casanova H., Cirne W., Dail H., Faerman M., Figueira S., Hayes J., Obertelli G., Schopf J., Shao G., Smallen S., Spring N., Su A., Zagorodnov D. Adaptive Computing on the Grid Using AppLeS // IEEE Trans. On Parallel and Distributed Systems. 2003. Vol. 14, No. 4. P. 369-382. DOI: 10.1109/TPDS.2003.1195409
8. Cirne W., Brasileiro F., Costa L., Paranhos D., Santos-Neto E., Andrade N., De Rose C., Ferreto T., Mowbray M., Scheer R., Jornada J. Scheduling in Bag-of-task Grids: The PAUÁ Case // 16th Symposium on Computer Architecture and High Performance Computing. 2004. P. 124–131. DOI: 10.1109/SBAC-PAD.2004.37
9. Voevodin V.I., Zholudev Y.A., Sobolev S.I., Stefanov K.S. Evolyutsiya sistemy metakomp'yutinga X-Com [The evolution of X-Com metacomputing system]. Vestnik Nizhegorodskogo universiteta im. N.I. Lobachevskogo [The bulletin of Lobachevsky State University of Nizhny Novgorod]. 2009. No. 4. P. 157–164.
10. Dail H., Sievert O., Berman F., Casanova H., YarKhan A., Vadhiyar S., Dongarra J., Liu C., Yang L., Angulo D., Foster I. Scheduling in the Grid Application Development Software Project // Grid resource management. State of the Art and Future Trends. Kluwer Acad. Publ., 2003. P. 73–98.
11. Kurowski K., Oleksiak A., Nabrzyski J., Kwiecień A., Wojtkiewicz M., Dyczkowski M., Guim F., Corbalan J., Labarta J. Multi-criteria Grid Resource Management Using Performance Prediction Techniques // Integrated Research in GRID Computing. Springer, 2007. P. 215–225.
12. Moab HPC Suite Enterprise Edition. URL:
13. <http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-suite-enterprise-edition> (accessed: 08.02.2014).
14. Kannan S., Roberts M., Mayes P., Brelsford D., Skovira J. F. Workload Management with LoadLeveler. IBM, First ed., 2001. 210 p.
15. 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.
16. 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. DOI: 10.1016/j.procs.2014.05.075
17. 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. May 2013. P. 96–115. DOI: 10.1007/s11227-014-1210-1
18. 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.
19. 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 2012. P. 418–424.
20. Toporkov V.V., Bobchenkov A.V., Yemelyanov D.M., Tselishchev A.S. Metody i evristiki planirovaniya v raspredelennykh vychisleniyakh s neotchuzhdae-mymi resursami [Scheduling methods and heuristics in distributed computing with non-dedicated resources]. Vestnik YuUrGU, seriya «Vychislitel'naya matematika i informatika» [Bulletin of the South Ural State University. Series “Computational Mathematics and Software Engineering”] 2014. Vol. 3, No. 2. P. 43–62.

Received March 17, 2015.