

В.В. Топорков¹, А.С. Топоркова², А.В. Бобченков¹,
Д.М. Емельянов¹, А.С. Целищев¹

АЛГОРИТМЫ ОТБОРА СЛОТОВ В ЗАДАЧЕ ПЛАНИРОВАНИЯ ПАКЕТА ЗАДАНИЙ В РАСПРЕДЕЛЕННЫХ СРЕДАХ*

Введение

Экономические модели выделения ресурсов и планирования являются весьма эффективными в распределенных вычислениях с неотчуждаемыми ресурсами, включая грид, мультиагентные системы и облачные вычисления [Garg et al., 2009; Bredin et al., 1999].

В [Ernemann et al., 2002] рассматриваются эвристические алгоритмы поиска наборов слотов на основе задаваемых пользователем функций полезности.

Среди различных подходов к организации вычислений в распределенных системах и средах можно выявить две устойчивых тенденции.

Одна из них основывается на использовании брокеров ресурсов [Vuuya et al., 2002; Ernemann et al., 2002], которые, как правило, позволяют оптимизировать выполнение конкретного приложения [Toporkov, 2009].

Другая тенденция связана с образованием виртуальных организаций и ориентирована, прежде всего, на грид-системы. При образовании виртуальных организаций [Kurowski et al., 2003] осуществляется оптимизация планирования на уровне потоков заданий. В рамках предложенной модели планирования потоков заданий [Топорков и др., 2009], важным этапом является предварительный поиск альтернативных вариантов назначения выполнения каждого из заданий.

Новизна подхода, состоит в применении экономических механизмов как на этапе отбора альтернативных наборов слотов, так и на этапе планирования выполнения пакета заданий.

¹ 111250, Москва, ул. Красноказарменная, 14, МЭИ (ТУ), ToporkovVV@mpei.ru.

² 109028, Москва, Б. Трехсвятительский пер., д. 1-3/12, стр. 8, МГИЭМ(ТУ).

* Работа выполнена при содействии Совета по грантам Президента РФ для поддержки ведущих научных школ (грант НШ-7239.2010.9), РФФИ (проект № 09-01-00095), Минобрнауки в рамках аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы» (проекты № 2.1.2/6718; 2.1.2/13283) и федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы (государственные контракты № П2227; № 16.740.11.0038; № 16.740.11.0516).

2. Схема планирования

В рамках рассматриваемой модели, в основу которой заложена иерархическая схема управления потоками заданий [Toporkov, 2009; Kuroki et al., 2003], фигурируют пользователи, запускающие задания, и собственники вычислительных ресурсов. Планирование выполнения совокупности (пакета) независимых заданий осуществляется циклично, на основе динамично обновляемых расписаний выполнения заданий в локальных процессорных узлах (рис. 1).

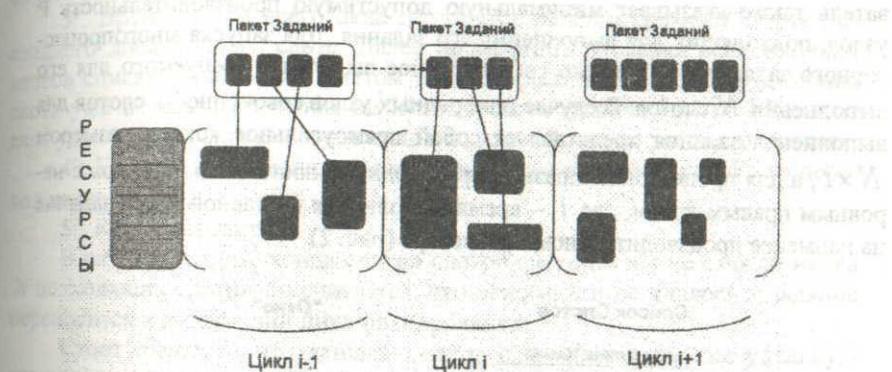


Рис. 1. Планирования потока заданий на основе циклов

В каждом цикле планирования запуска пакета заданий требуется решение двух задач. Во-первых, отбор подходящих (по ресурсу, времени, цене) слотов для каждого задания пакета [Ernemann et al., 2002]. В общем случае у каждого задания может оказаться несколько альтернативных наборов слотов для выполнения – альтернатив. Во-вторых, необходим выбор комбинации альтернатив, являющейся эффективной или оптимальной с точки зрения прохождения всего пакета заданий в текущем цикле планирования.

В данной работе мы рассматриваем два типа критериев и две задачи однокритериальной оптимизации выполнения пакета заданий. Это минимизация стоимости выполнения пакета заданий при ограничении на суммарное время занятия слотов, либо минимизация времени прохождения пакета заданий при фиксированном бюджете виртуальной организации.

3. Алгоритмы отбора слотов

В начале каждого цикла динамично обновляются наборы доступных слотов на основе информации, поступающей от локальных менеджеров ресурсов. Каждый из слотов соответствует временному отрезку, который

может быть использован для выполнения задачи, входящей в состав много-процессорного задания, на том или ином типе ресурса. Требования задания к ресурсам оформляются в виде *ресурсного запроса*, содержащего тип, количество N и характеристики узлов (тактовую частоту процессора, емкость оперативной памяти, дисковое пространство, операционную систему и т.д.), а также оценку времени t их использования. Кроме того, пользователь указывает максимальную удельную стоимость C за использование отдельного узла, которую он готов заплатить. В рассматриваемой нами модели пользователь также указывает минимальную допустимую производительность p узлов, подходящих для выполнения его задания. Для запуска многопроцессорного задания необходимо согласованное выделение требуемого для его выполнения N слотов. В случае однородных узлов совокупность слотов для выполнения задания представляет собой прямоугольное «окно» размером $N \times t$, а для процессоров с разной производительностью это – «окно» с неровным правым краем, где t – время выполнения составной части задания на наименее производительном процессоре (рис. 2).

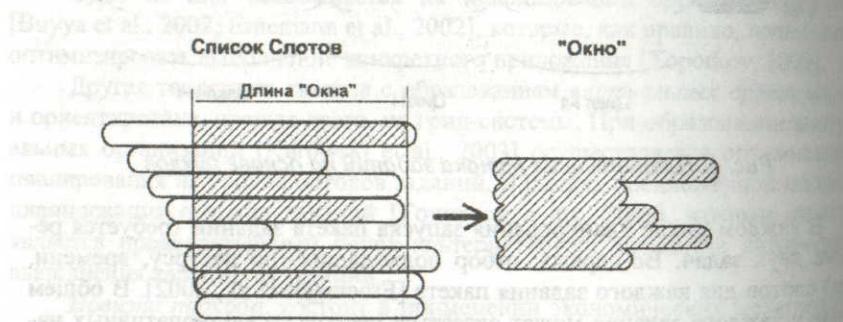


Рис. 2. Формирование окна с неровным правым краем

Нами рассматриваются два алгоритма поиска альтернативных наборов слотов. Первый из них ALP, производит поиск окна на упорядоченном по неубыванию времени старта списке слотов системы. Его особенность состоит в том, что он учитывает ограничение C на максимальную удельную стоимость каждого отдельного слота.

Приведем описание алгоритма ALP:

- 1°. Доступные слоты системы упорядочиваются по неубыванию времени старта.
- 2°. Из полученного списка выбирается следующий подходящий слот s_k . Слот считается подходящим, если удовлетворяет следующим требованиям:

Секция 4

- a) производительность узла, на котором выделен слот $P(s_k) \geq P$;
 b) длины слота достаточно, чтобы выполнить часть многопроцессорного задания (с учетом актуальной производительности узла)
 $L(s_k) \geq tP(s_k)/P$;
 c) удельная стоимость использования слота не выше ограничения
 $C(s_k) \leq C$.

Если условия а), б) и с) выполняются, то слот добавляется в список слотов окна.

3°. Время старта всего окна полагается равным времени старта последнего добавленного слота. Далее происходит проверка всех текущих слотов списка окна. Если с учетом нового общего времени старта длины слота уже не хватает, чтобы выполнить часть задания (пункт 2°б), этот слот удаляется из списка слотов окна и в дальнейшем не рассматривается.

4°. Если число оставшихся слотов в списке окна меньше N , то переход на шаг 2°.

5°. Конец алгоритма.

В случае, если до конца списка слотов системы мы не смогли найти N подходящих слотов, то считается, что окно найти не удалось и задание переносится в следующий цикл планирования.

Стоит отметить, что данный подход с ограничением на удельную стоимость отдельных слотов сужает область поиска и не позволяет использовать для составления окна более дорогие слоты. Отличие следующего алгоритма AMP состоит в том, что он оперирует с ограничением на максимальный бюджет выполнения задания. Максимальный бюджет формируется как $S = CtN$, где t – необходимое время резервирования, N – запрашиваемое количество слотов, а C – указанное в ресурсном запросе ограничение на удельную стоимость отдельного слота. Таким образом, алгоритм AMP производит поиск окна, суммарная стоимость слотов которого не превышает максимальный бюджет S .

При описании алгоритма AMP мы воспользуемся дополнительными обозначениями: N_S – текущее количество слотов в списке окна; M_N – суммарная стоимость первых N слотов окна.

Рассмотрим алгоритм AMP для поиска отдельного окна:

1°. Находится самое раннее окно из N слотов, используя алгоритм ALP без учета условия 2°с.

2°. Слоты внутри списка окна упорядочиваются по возрастанию их стоимости.

Далее вычисляется суммарная стоимость M_N первых N слотов окна. Если $M_N \leq S$, то к шагу 4°, в этом случае результирующее окно формируется из первых N слотов, а остальные возвращаются в список слотов системы. Иначе к шагу 3°.

3°. Добавляется следующий подходящий слот из списка с учетом ограничений 2°**a** и 2°**b** алгоритма ALP. Время старта формируемого окна полагается равным времени старта этого слота. Далее проверяется, все ли слоты успевают выполнить задание с учетом нового времени старта, а те, которые не успевают, удаляются (шаг 3° алгоритма ALP).

Если $N_S < N$, то повтор шага 3°. Если $N_S \geq N$, то к шагу 2°.

Если список слотов системы закончен и $N_S < N$, то найти окно не удалось.

4°. Конец алгоритма.

Следует отметить следующие три особенности предлагаемых алгоритмов. Во-первых, оба алгоритма учитывают производительность и разнородность вычислительных узлов. Это позволяет формировать окна с неровным правым краем. Во-вторых, оба алгоритма учитывают экономические ограничения пользователя: ограничение на максимальную допустимую удельную стоимость использования ресурсов. В-третьих, оба алгоритма обладают линейной сложностью $O(m)$ относительно общего количества слотов системы m , алгоритмы двигаются по списку слотов в направлении неубывания времени старта, без возвращения назад и пересмотра предыдущих шагов.

4. Результаты моделирования циклов планирования

Эксперимент состоит в сравнении результатов планирования пакета заданий на одинаковом наборе ресурсов, но с использованием различных алгоритмов поиска альтернативных наборов слотов ALP и AMP. На каждом цикле планирования происходит генерация исходных данных: списка доступных слотов системы и пакета пользовательских заданий. Рассмотрим задачу поиска альтернативных наборов слотов при минимизации суммарного времени выполнения пакета заданий $\min_{\bar{s}_i} T(\bar{s})$ с ограничением на общий бюджет B^* .

В общей сложности было промоделировано 25000 циклов планирования. Для подсчета результатов использовались только те из них, в которых все задачи пакета имели хотя бы одну альтернативу выполнения. По результатам моделирования алгоритм AMP превзошел ALP по целевому критерию минимизации суммарного времени $T(\bar{s})$ выполнения пакета заданий на 35 %. Среднее время выполнения одного задания при использовании ALP составило 59.85, а при использовании AMP – 39.01 (рис. 3,а).

При этом средняя стоимость выполнения задания с использованием ALP составила 313.56, а для AMP на 15 % больше – 369.69 (рис. 3,б).

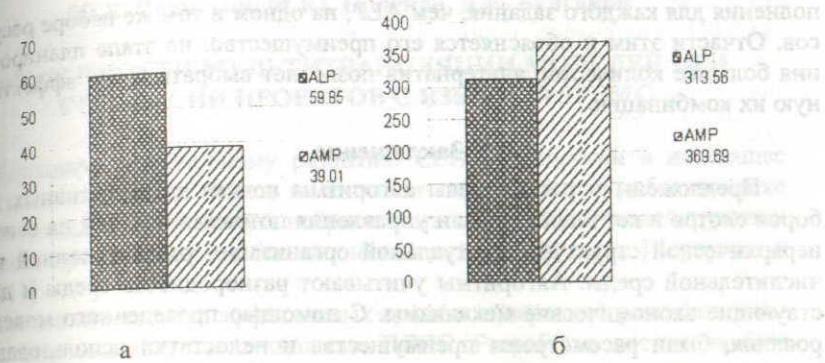


Рис. 3. Минимизация времени выполнения пакета заданий: среднее время выполнения задания (а); средняя стоимость выполнения задания (б)

При решении минимизации суммарной стоимости выполнения пакета заданий: $\min_{\bar{s}_j} C(\bar{s})$ с ограничением на суммарное время выполнения T^* ,

были собраны результаты 8571 отдельных циклов планирования, в каждом из которых все задания имели хотя бы по одной альтернативе выполнения.

Средняя стоимость выполнения задания в схеме с использованием алгоритма ALP составила 313.09, а при использовании AMP – 343.3, что демонстрирует преимущество алгоритма ALP в 9 % (рис. 4,а). Среднее же время выполнения задания для ALP составило 61.04, а с использованием AMP – 59.85, что на 15 % меньше (рис. 4,б).

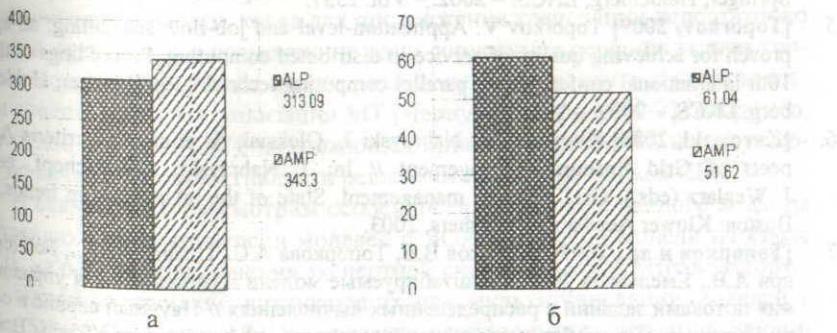


Рис. 4. Минимизация стоимости выполнения пакета заданий: средняя стоимость выполнения задания (а); среднее время выполнения задания (б)

Стоит отдельно отметить, что в ходе проведения эксперимента, алгоритм AMP позволял находить в среднем в 4-5 раз больше альтернатив вы-

полнения для каждого задания, чем ALP, на одном и том же наборе ресурсов. Отчасти этим и объясняется его преимущество: на этапе планирования большее количество альтернатив позволяет выбрать более эффективную их комбинацию.

Заключение

Предложены и рассмотрены алгоритмы поиска альтернативных наборов слотов в контексте модели управления потоками заданий на основе иерархической структуры виртуальной организации распределенной вычислительной среды. Алгоритмы учитывают разнородность среды и действующие экономические механизмы. С помощью проведенного моделирования, были рассмотрены преимущества и недостатки использования этих алгоритмов на этапе поиска подходящих наборов слотов.

Библиографический список

1. [Garg et al., 2009] Garg S.K., Buyya R., Siegel H.J. Scheduling parallel applications on utility Grids: time and cost trade-off management, Proceedings of the 32nd Australasian computer science conference (ACSC 2009), Wellington, New Zealand.
2. [Bredin et al., 1999] Bredin J., Kotz D., Rus D. Economic markets as a means of open mobile-agent systems, Proceedings of the workshop “Mobile agents in the context of competition and cooperation (mac3)”, 1999.
3. [Buyya et al., 2002] Buyya R., Abramson D., Giddy J. Economic models for resource management and scheduling in grid computing, *J. of concurrency and computation: practice and experience*. – 2002. – Vol. 14, № 5.
4. [Ernemann et al., 2002] Ernemann C., Hamscher V., Yahyapour R. Economic scheduling in grid computing. In: Proceedings of the 8th job scheduling strategies for parallel processing, D.G. Feitelson, L. Rudolph, U. Schliegelshohn (eds.), Springer, Heidelberg, LNCS. – 2002. – Vol. 2537.
5. [Toporkov, 2009] Toporkov V. Application-level and job-flow scheduling: an approach for achieving quality of service in distributed computing, Proceedings of the 10th international conference on parallel computing technologies, Springer, Heidelberg, LNCS. – 2009. – Vol. 5698.
6. [Kurowski, 2003] Kurowski K., Nabrzyski J., Oleksiak A. et al. Multicriteria Aspects of Grid Resource Management // In: J. Nabrzyski, J.M. Schopf, and J. Weglarz (eds.), Grid resource management. State of the art and future trends. - Boston: Kluwer Academic Publishers, 2003.
7. [Топорков и др., 2009] Топорков В.В., Топоркова А.С., Целищев А.С., Бобченков А.В., Емельянов Д.М. Масштабируемые модели планирования и управления потоками заданий в распределенных вычислениях // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Российской суперкомпьютерной конференции. – М.: Изд-во МГУ, 2009.