МАСШТАБИРУЕМЫЕ МОДЕЛИ ПЛАНИРОВАНИЯ И УПРАВЛЕНИЯ ПОТОКАМИ ЗАДАНИЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЯХ

В.В. Топорков, А.С. Топоркова, А.С. Целищев, А.В. Бобченков, Д.М. Емельянов

1. Введение и мотивация

В широком спектре различных подходов к организации вычислений в распределенных средах можно выделить две устойчивых тенденции. Одна из них основывается на использовании доступных ресурсов. Роль посредников между пользователями и вычислительными узлами выполняют агенты приложений – брокеры ресурсов. Ряд проектов в этом направлении часто ассоциируется с планированием вычислений на уровне приложений (application-level scheduling). Это – AppLeS, APST, Legion, DRM, Condor-G, Nimrod-G и ряд других. Зачастую не предполагается наличия какого-либо регламента в предоставлении ресурсов, как например, в системе метакомпьютинга X-Com (НИВЦ МГУ). Другая тенденция связана с образованием виртуальных организаций и ориентирована, прежде всего, на грид-системы. И тот, и другой подходы имеют свои достоинства и недостатки.

В рамках первого из направлений системы управления ресурсами являются хорошо масштабируемыми и адаптируемыми к особенностям приложений. Однако использование независимыми пользователями различных критериев для оптимизации планов выполнения своих заданий (в условиях возможной конкуренции с другими заданиями) может ухудшать такие интегральные характеристики, как время выполнения пакета заданий и загрузка ресурсов [1]. Образование виртуальных организаций естественным образом ограничивает масштабируемость систем управления заданиями. Наличие определенных правил предоставления и потребления ресурсов позволяет повысить эффективность планирования и распределения ресурсов на уровне потоков заданий (job-flow scheduling). При этом время выполнения отдельных приложений может и увеличиваться, поскольку при планировании не удается учесть важных особенностей структуры заданий и пользовательских предпочтений, касающихся ресурсов.

Необходимость контроля над потоками независимых заданий приводит к тому, что в качестве посредника между пользователями и локальными системами выступают различного рода метапланировщики, менеджеры, грид-диспетчеры и т. п. Здесь можно упомянуть программный комплекс GrAS (ИПМ им. М.В. Келдыша РАН), сетевую среду распределенных вычислений, разработанную в МСЦ РАН, НИИ «Квант», ИПМ им. М.В. Келдыша, проект GrADS (Суперкомпьютерный центр Сан Диего, Аргоннская национальная лаборатория США). Заметим, что альтернативной является такая виртуализация ресурсов грид, когда осуществляется полный контроль над потоком заданий единым управляющим центром, а в вычислительных узлах не используются локальные планировщики и системы пакетной обработки заданий: коммерческие платформы DCGrid, LiveCluster, GridMP, Frontier; волонтерские проекты серии @Home; система CCS, поддерживающая управление отчуждаемыми ресурсами [2].

Принципиальная новизна подхода к организации масштабных вычислений в распределенных средах, предлагаемого в данной работе, заключается в следующих положениях:

1. стратегии диспетчеризации ориентированы на динамичное перераспределение потоков заданий между доменами узлов на основе экономических моделей поддержки политики предоставления и потребления ресурсов в виртуальных организациях, а также анализе свойств заданий (ресурсных запросов, структуры, статистики загрузки процессоров и потребности в данных);
2. планирование и коаллокация заданий реализуются с помощью обновляемых стратегий (списков возможных времен запусков и назначений задач) в соответствии с динамикой загрузки, освобождения, резервирования неотчуждаемых ресурсов и локальными политиками обработки заданий в доменах узлов.

По сути это означает интеграцию методов планирования на уровне потоков заданий и уровне приложений с целью эффективного использования вычислительных ресурсов распределенных сред.

2. Иерархическая среда диспетчеризации заданий

В основу реализации стратегий коаллокаций закладывается иерархическая структура (рис. 1), состоящая из метапланировщика потоков заданий, подконтрольных ему менеджеров заданий, которые, в свою очередь, взаимодействуют с локальными менеджерами управления ресурсами, например, системами пакетной обработки заданий. Преимущества иерархически организованных структур управления заданиями хорошо известны. Так строится комплекс GrAS, реализующий централизованную диспетчеризацию заданий на основе прогноза загрузки локальных узлов и метода опережающего планирования. Иерархия промежуточных серверов в системе X-Com [3] позволяет снизить простой узлов из-за заметных задержек в передаче данных и занятости управляющего сервера обслуживанием других узлов. Древовидная структура менеджеров в сетевой среде распределенных вычислений на базе ресурсов МСЦ РАН позволяет избежать тупиков при доступе к разделяемым ресурсам. Еще один немаловажный аспект при организации вычислений в неоднородных средах –

объединение под управлением одного менеджера тех вычислительных узлов, которые являются схожими по архитектуре, составу, политике администрирования и т.д. Иерархическая централизованная схема контроля над

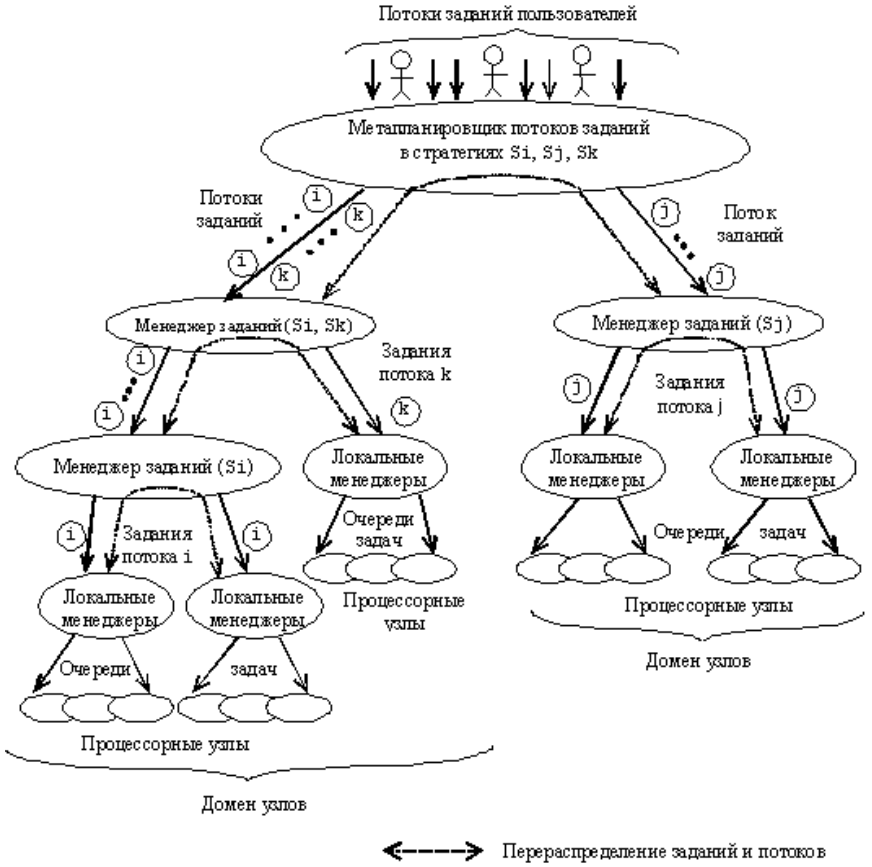

Рис. 1. Иерархическая среда диспетчеризации потоков заданий

глобальной очередью заданий со стороны метапланировщика применяется в проекте GrADS.

Оригинальность предлагаемого нами решения заключается в динамическом перераспределении потоков заданий в среде на основе стратегий. Стратегия выполнения составного задания представляет собой список возможных назначений на ресурсы и планов для каждой из задач, входящих в задание, и строится с использованием метода критических работ [4]. Для порождения стратегий используется метод критических работ, суть которого состоит в планировании очередной критической работы (цепочки неназначенных задач наибольшей длительности или стоимости при использовании комбинации ресурсов в наилучшими априорными оценками задач) и разрешении возможных конфликтов за разделяемый ресурс с ранее назначенными задачами.

Задача метапланировщика – распределение потоков заданий между доменами узлов в соответствии с выбранной стратегией коаллокации. На менеджеры заданий возлагается функция поддержки и актуализации стратегий на основании взаимодействия с локальными менеджерами (системами пакетной обработки) и имитационного моделирования прохождения локальных заданий в узлах.

3. Многокритериальные и многофакторные стратегии управления заданиями

Стратегии коаллокации заданий в динамично организованных распределенных средах должны учитывать множество факторов, связанных с загрузкой и доступностью ресурсов, их неоднородностью, т.е. быть многофакторными [5]. Использование методов приоритетного планирования на основе очередей не всегда эффективно для решения проблемы коаллокации многопроцессорных заданий. Хорошо известны побочные эффекты применения этих методов из опыта эксплуатации кластерных систем, таких как LL, NQE, LSF, PBS и др. Например, реализация традиционной дисциплины обслуживания в порядке поступления (FCFS) ведет к простаиванию части ресурса. Упорядочивание заданий в очереди по приоритетам в соответствии с их трудоемкостью, например, первым обслуживается наименее трудоемкое (LWF) либо самое большое задание, ведет к фрагментации ресурсов и делает невозможным запуск некоторых заданий из-за отсутствия требуемого уровня ресурсов. В распределенных средах эти эффекты могут привести к непредсказуемому времени выполнения задания, а, значит, и обусловить неприемлемое качество обслуживания. Поэтому в ряде проектов внимание сосредоточено на построении расписания на будущее, поддерживаемого механизмами предварительного резервирования ресурсов. Здесь необходимо упомянуть кластерный планировщик Maui, реализующий алгоритм обратного заполнения (backfilling). Метод опережающего планирования для грид-

приложений использован в отечественном проекте GrAS (ИПМ им. М.В. Келдыша РАН). Механизм удаленного резервирования ресурсов грид поддерживается также проектами GARA, Ursala, Silver. Однако при этом строится один вариант расписания, которое может стать неактуальным из-за изменения состояния локальной очереди заданий, значительного времени доставки задания на удаленный узел и т.д.

Разработанный ранее метод критических работ [4] развит для планирования составных заданий с целью порождения многофакторных и многокритериальных стратегий распределения потоков заданий. Этот метод может быть отнесен к классу методов приоритетного планирования, поскольку он позволяет строить список распределений ресурсов и планов запуска задач на будущее. При этом он основан на использовании методов динамического программирования и, следовательно, использует некие интегральные характеристики, например, суммарную стоимость использования ресурсов системой задач, входящих в задание. Здесь нет никакого противоречия, поскольку метод ориентирован на коаллокацию составных заданий. Новизна нашего подхода по сравнению с известными решениями проблемы планирования в распределенных средах, включая грид, заключается в том, что строится совокупность вариантов планирования и коаллокаций на будущее, по сути и составляющих стратегию. Стратегия формируется на основе формализованных критериев эффективности, позволяющих адекватно представить экономические принципы распределения ресурсов при использовании соответствующих функций стоимости, а также решать задачу балансировки загрузки разнородных процессорных узлов. Стратегия строится с использованием методов динамического программирования таким образом, что позволяет оптимизировать планирование и распределение ресурсов для совокупности задач, входящих в составное задание [4].

Стратегия представляет собой своего рода заготовку возможных действий метапланировщика и его реакций на события, связанные с загрузкой и предварительным резервированием ресурсов. Чем больше факторов в виде формализованных критериев при ее формировании учитывается, тем более полной она является в смысле охвата возможных событий в вычислительной среде.

4. Результаты имитационного моделирования стратегий диспетчеризации

Нами разработана среда имитационного моделирования [5] и проведены исследования показателей эффективности различных типов стратегий коаллокации в зависимости от структуры заданий и различных политик размещения данных в среде:

– с высокой степенью распределенности вычислений (задач) в среде и политикой активного перемещения (репликации) данных (РВ-РД), а также модификация (МРВ-РД) этого типа стратегии для наилучшей и наихудшей оценок времени выполнения задач;
– с высокой степенью распределенности вычислений в сочетании с удаленным доступом к данным (РВ-УД);
– с крупноблочным разбиением задач и статичным хранением данных в удаленных узлах (КВ-УД).

Стратегия РВ-РД, в отличие от модификации, строится в полном диапазоне возможных оценок времени выполнения задач для отобранных базовых узлов. Поэтому она является более полной по сравнению с модификацией МРВ-РД с точки зрения охвата возможных событий в среде, связанных с загрузкой ресурсов. Однако порождение такого рода стратегий метапланировщиком является весьма трудоемкой процедурой с точки зрения вычислительных затрат.

Процессорные узлы отбирались и группировались в соответствии с их производительностью. В первую группу попадали «быстрые» узлы с относительной производительностью 0.66 ... 1, ко второй и третьей группам были отнесены узлы с производительностью 0.33 ... 0.66 и 0.33 («медленные» процессоры). Число отбираемых узлов соответствовало степени параллелизма задания.

Исследованы стратегии для 12000 заданий при одинаковом предельном времени завершения со случайными оценками длительности выполнения задач, объемов вычислений, времени передачи и объемов данных.

Выделялись следующие параметры заданий в различных стратегиях:

- среднее число задач: 26.9 (РВ-РД), 25.2 (РВ-УД), 10.0 (КВ-УД);
- среднее число работ: 33.4 (РВ-РД), 29.0 (РВ-УД), 13.1 (КВ-УД);
- среднее число работ, содержащих более трех задач: 6.2 (РВ-РД), 5.3 (РВ-УД), 2.3 (КВ-УД);
- максимальное число задач в работах: 5.3 (РВ-РД), 4.4 (РВ-УД), 3.7 (КВ-УД).

Было проведено статистическое исследование эффективности планирования в стратегиях РВ-РД, РВ-УД, КВ-УД на уровне приложений с использованием метода критических работ. Цель исследования заключалась в том, чтобы оценить способность метода к «предвидению» допустимых распределений без имитационного моделирования той или иной дисциплины прохождения заданий в локальных узлах. Для 12000 случайным образом сгенерированных заданий метод позволял построить допустимые планы в 38% экспериментов для стратегии РВ-РД, 37% – для РВ-УД, 33% – для КР-УД (рис. 2, а). Результат вполне объясним: на уровне планирования приложений метод критических работ позволяет строить стратегии лишь для доступных, не занятых другими заданиями, процессорных узлов. При этом весьма характерна доля конфликтов за «быстрые» и «медленные» узлы: 32% – за «быстрые», 68% – за «медленные» в стратегии РВ-РД, 56% и 44%

соответственно в РВ-УД, 74% и 26% соответственно в КВ-УД (рис. 2, б). Таким образом, чем выше степень распределенности задач, тем меньше вероятность конфликтов за наиболее производительный ресурс.

Помимо моделирования на уровне приложений, было проведено исследование характеристик стратегий в условиях, когда выполнялось имитационное моделирование прохождения независимых заданий в локальных узлах. Для этого был использован режим SIMULATION планировщика Maui. Известно, что в нем предусмотрены возможности указания приоритетов и политики выбора заданий, выделения узлов и резервирования. Ряд утилит позволяет управлять ходом имитации (shedctl), получать информацию о выполняющихся и простаивающих заданиях и задачах (showq), ресурсах (checknode) и т.д. Особенности формирования приоритетов заданий определяются в соответствующем конфигурационном файле. Известно, что Maui поддерживает механизм гарантированного резервирования ресурсов. Расчет времени, на которое они выделяются, осуществляется исходя из значения WallClockLimit – ожидаемого максимального времени выполнения задания, определяемого пользователем. Один из ключевых алгоритмов планирования в Maui – обратное заполнение (бэкфилинг). Его реализация не блокирует принятую дисциплину очереди, запуская на выполнение задания, для которых достаточно незанятых ресурсов. Таким образом, концепция Maui адекватна методу критических работ, а режим SIMULATION позволяет строить прогноз загрузки ресурсов и выбрать из стратегии соответствующий план.

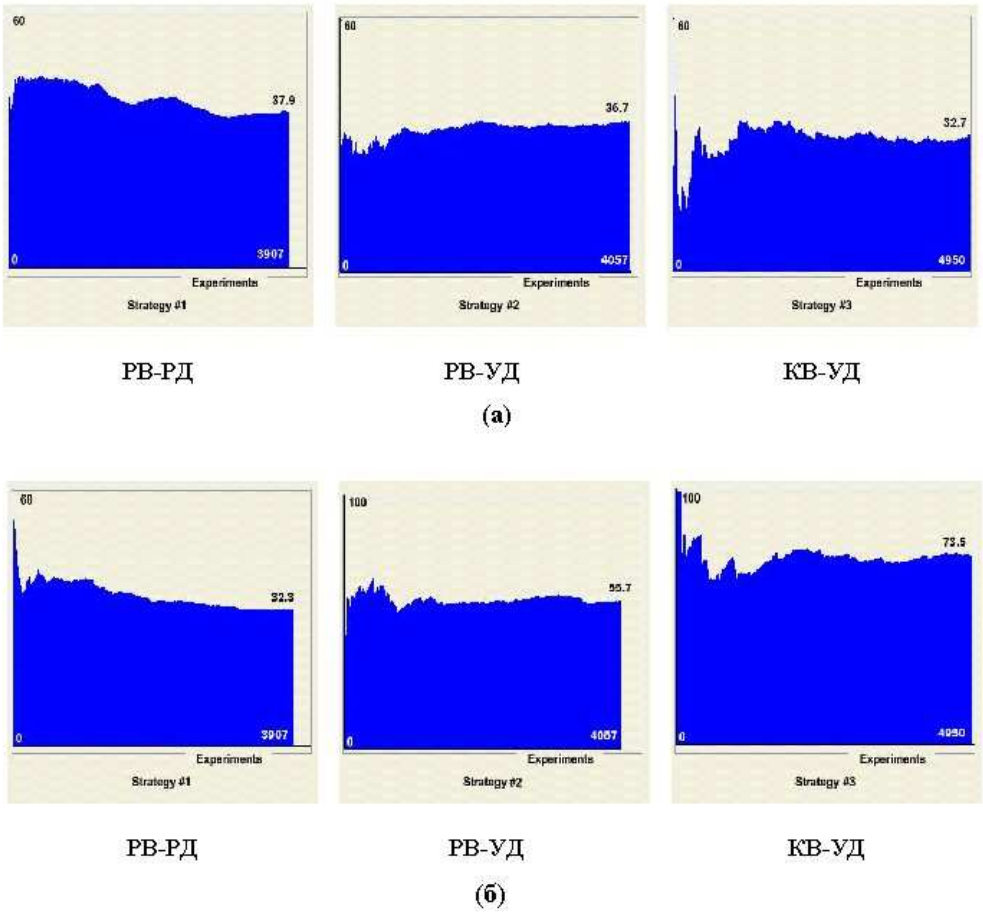

Рис. 2. Результаты имитационного моделирования на уровне приложений: процент допустимых планов (а), процент конфликтов за «быстрые» процессоры (б)

Ниже приведены результаты сравнительного анализа показателей качества стратегий РВ-УД, КВ-УД и модификации МРВ-РД (в стратегиях РВ-УД, КВ-УД планы формировались во всем диапазоне оценок для отобранных узлов). Исследованы такие показатели качества стратегий, как «цена» выполнения задания, время решения задачи, погрешность планирования (оценка времени запуска) и время жизни стратегии, в течение которого планы вычислений остаются приемлемыми с учетом динамики состояния среды. «Цена» измерялась как сумма отношений объемов вычислений отдельных задач ко времени их решения, округляемых до ближайшего не меньшего целого числа.

Значения относительных показателей качества, полученных в различных стратегиях:

- «цена» выполнения задания: 0.80 (МРВ-РД), 1 (РВ-УД), 0.60 (КВ-УД);
- время решения задачи: 0.75 (МРВ-РД), 0.60 (РВ-УД), 1 (КВ-УД);
- время жизни стратегии: 0.35 (МРВ-РД), 0.10 (РВ-УД), 1 (КВ-УД);

– отклонение старта / время выполнения задания: 0.45 (МРВ-РД), 0.20 (РВ-УД), 0.40 (КВ-УД).

Результаты исследований показывают, что самыми «дешевыми» оказываются самые «медленные» стратегии типа КВ-УД, они же являются наиболее «живучими». В то же время наименее «живучи» самые «быстрые», «дорогие» и наиболее «точные» стратегии типа РВ-УД. Менее «точные» стратегии типа МРВ-РД дают в среднем большее время решения задачи, чем «точные» стратегии типа РВ-УД, которые охватывают большее число возможных событий, связанных с динамикой загрузки узлов.

5. Заключение и направления дальнейшей работы

Проведенные исследования показывают, что эффективными являются стратегии коаллокаций, учитывающие как структуру заданий и реализующие планирование на уровне приложений, так и характеристики потоков заданий в среде для их группирования и распределения между вычислительными узлами в соответствии с динамикой загрузки узлов.

Вышеприведенные результаты исследования характеристик различных типов стратегий получены в условиях моделирования глобальных потоков заданий в виртуальной организации. Условие неотчуждаемости ресурсов требует глубоких дополнительных исследований, связанных с имитационным моделированием прохождения локальных заданий и разработкой методов прогнозирования загрузки локальных узлов. В них могут применяться различные дисциплины обслуживания очередей заданий (модификации FCFS, LWF), алгоритмы планирования (бэкфилинг, планирование в связке (gang scheduling) и т.д.) и реализовываться свои, локальные правила администрирования.

Очень важный аспект – влияние предварительного резервирования на качество обслуживания. Ряд факторов, характеризующих качество обслуживания и показатели балансировки потоков заданий, связан с динамическим изменением приоритетов заданий, когда пользователь виртуальной организации изменяет плату за выполнение.

Все это вопросы, требующие дополнительных исследований.

Работа выполнена при финансовой поддержке РФФИ (проект № 09-01-00095) и аналитической ведомственной целевой программы Федерального агентства по образованию «Развитие научного потенциала высшей школы» (проект № 2.1.2/6718).

ЛИТЕРАТУРА:

1. 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. P. 271-293.
2. Anderson D.P. and Fedak G. The Computational and Storage Potential of Volunteer Computing // Proc. of the IEEE/ACM International Symposium on Cluster Computing and Grid, New York: IEEE Press, 2006. P. 73-80.
3. Воеводин Вл. В. Решение больших задач в распределенных вычислительных средах // Автоматика и телемеханика. 2007. № 5. С. 32-45.
4. Топорков В.В. Опорные планы согласованного выделения ресурсов при организации распределенных вычислений на масштабируемых системах // Программирование. 2008. № 5. С. 50-64.
5. Toporkov V.V., Tselishchev A.S. Safety Strategies of Scheduling and Resource Co-allocation in Distributed Computing // Proc. of the 3rd International Conference on Dependability of Computer Systems. Los Alamitos: IEEE CS Press, 2008. P. 152-159.