СТРАТЕГИИ КОАЛЛОКАЦИЙ ДЛЯ РЕШЕНИЯ БОЛЬШИХ ЗАДАЧ В РАСПРЕДЕЛЕННЫХ СРЕДАХ

В.В. Топорков, А.С. Топоркова, А.С. Целищев

Серьезные трудности при организации вычислений в распределенных средах обусловлены разнородностью состава и большим количеством наступающих событий, связанных с динамикой доступности ресурсов. Это заставляет использовать методы прогнозирования их освобождения и занятия, предварительного резервирования и опережающего планирования (проект диспетчера GrAS, ИПМ им. М.В. Келдыша РАН). В известных решениях проблемы планирования распределенных вычислений строится один вариант расписания, которое к моменту фактического назначения ресурсов заданию может стать неактуальным из-за изменения состояния очереди заданий и состава узлов, значительного времени передачи задания на удаленный узел и т.д. Поэтому планирование приходится выполнять заново, с учетом изменившегося состояния ресурсов. Отсюда возникает необходимость построения многофакторных и многокритериальных стратегий согласованного планирования и выделения ресурсов (коаллокаций) в распределенных вычислительных средах.

Сейчас можно, по-видимому, выделить два различных подхода к организации распределенных вычислений, связанных с решением больших задач.

Один из них рассчитан на использование доступных распределенных неоднородных ресурсов, когда не предполагается наличия какого-либо регламента в их предоставлении [1]. Пример реализации – система X-Com (НИВЦ МГУ, http://X-Com.parallel.ru).

Другой подход ориентирован на Грид-системы и образование виртуальных организаций, в рамках которых устанавливаются определенные правила предоставления и использования ресурсов [2] .

Чаще всего и в том, и другом подходах информационная структура задания и неоднородность составляющих его задач не учитываются. Так, в X-Com [1] модель вычислений строится в рамках парадигмы Master/Slaves, когда процессы-рабочие обмениваются данными лишь с процессом-мастером. Что касается Грид-систем, то ряд современных программных платформ позволяет работать со структурированными заданиями [2]. Например, система управления заданиями WLMS в составе комплекса gLite (http://www.glite.org) оперирует с заданиями разных типов: простыми и составными. Модель составного задания – ориентированный бесконтурный граф (DAG), вершины которого соответствуют отдельным заданиям, а дуги – связям между ними. При этом, однако, не оговаривается ни сложность, ни разнотипность составляющих заданий по требуемым вычислительным ресурсам. В проблеме коаллокации неоднородных распределенных ресурсов для сложноструктурированных заданий пренебречь этими аспектами зачастую невозможно [3].

Одна из первых реализаций планирования многопроцессорных заданий с учетом их структуры применительно к параллельным вычислениям была осуществлена в планировщике Maui (http://www. clusterresources.com): здесь допускается, что задание (job) может состоять из параллельных задач (tasks), однако все задачи считаются однотипными в смысле потребностей в вычислительном ресурсе. Заметим, что и система WLMS поддерживает выполнение многопроцессорных параллельных заданий в среде MPICH, однако структура задания и сложность составляющих его задач не принимаются во внимание.

В настоящей работе предлагается подход для формирования различных типов стратегий коаллокаций с учетом структуры задания, разнотипности и сложности составляющих его задач, а также особенностей политики репликации и хранения данных в распределенной неоднородной среде с динамично изменяющимся составом вычислительных узлов. В качестве базовой концепции выбирается схема «метапланировщик – локальные системы пакетной обработки», ориентированная на образование виртуальных организаций пользователей [4].

Согласованное выделение распределенных вычислительных ресурсов для сложноструктурированных заданий связано с необходимостью учета информационной связности составляющих задач, поскольку распределенность среды создает заметные задержки во взаимодействии удаленных узлов. При этом части задания (задачи) могут быть существенно неоднородными по вычислительной нагрузке, что также необходимо учитывать при выборе состава процессорных узлов для выполнения задания.

Предложены и исследованы три типа стратегий коаллокаций для выполнения составных заданий:

- с высокой степенью распределенности вычислений в сочетании с политикой активной репликации данных (**S1**);
- с высокой степенью распределенности вычислений при удаленном доступе к данным, отсутствием динамичной репликации (**S2**);
- с крупноблочным разбиением заданий и статичным хранением данных на удаленных узлах (**S3**).

Изучено влияние стратегий коаллокаций на показатели качества обслуживания:

- среднюю загрузку и «цену» использования неоднородных процессорных узлов;
- время выполнения неодинаковых по вычислительной сложности задач;
- время «жизни» стратегии в условиях динамично изменяющегося состава вычислительной среды;
- отклонение запрашиваемого времени запуска задания (при предварительном резервировании) от фактического времени доступа к ресурсам.

Результаты исследования стратегий приведены на рис. 1, 2, 3.

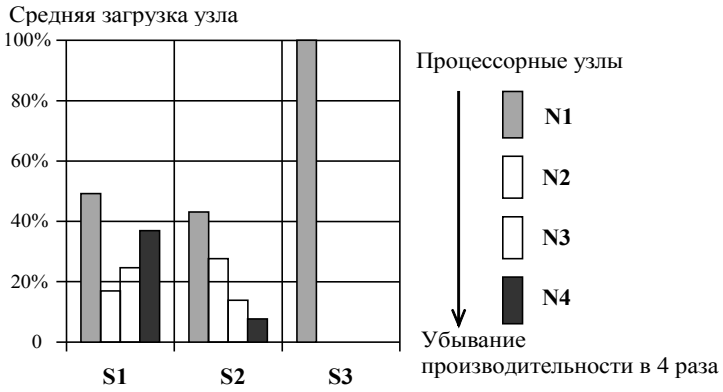

Рис. 1. Сравнение стратегий по загрузке вычислительных узлов

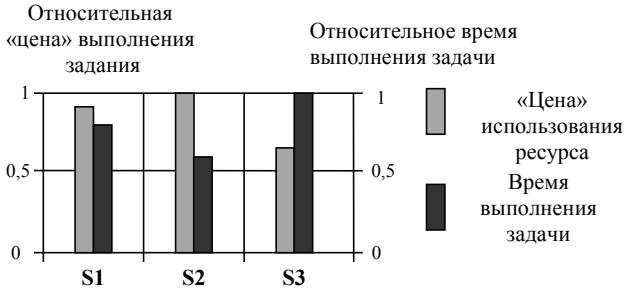

Рис. 2. «Цена» и время решения задачи

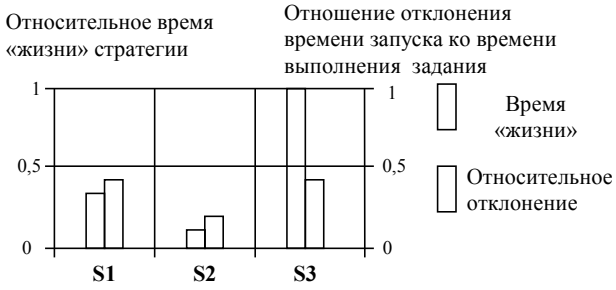

Рис. 3. Время «жизни» и погрешность стратегии

Для исследования эффективности стратегий коаллокации разработана имитационная модель метапланировщика [4], позволяющего генерировать сценарии распределенной обработки, формировать стратегии планирования и выбирать опорный план вычислений в зависимости от состояния загрузки узлов и заданных показателей качества обслуживания (средней загрузки узлов, времени исполнения задачи, стоимости использования ресурса). Имитационная модель позволяет формировать поток заданий от пользователей. Каждое из заданий представляется соответствующим параметризованным графом. Степень информационной связности вершин (задач) может варьироваться. Число базовых процессорных узлов принимается кратным максимальной ширине яруса графа. Оценки длительности выполнения задач, объемов вычислений, времени доступа к данным – целые случайные числа, равномерно распределенные в заданном диапазоне. Конфликты между задачами, конкурирующими за один и тот же узел, разреша-

ются за счет незадействованных базовых узлов так, что обеспечивается минимум функции штрафа – суммы отношений объемов вычислений к оценкам длительности выполнения задач.

При построении стратегии **S1** для каждой из задач бралась наихудшая и наилучшая из возможных оценок времени исполнения (на самом «медленном» и самом производительном узлах), для стратегий **S2**, **S3** планы формировались во всем диапазоне оценок для отобранных базовых узлов. Число узлов в распределенной среде – 20...30. Базовые процессорные узлы, отбираемые метапланировщиком, разделялись на четыре типа **N1**, ... , **N4** по производительности: относительная производительность узла **N1** – 1, узлов **N2**, **N3** и **N4** – соответственно 0.5, 0.33 и 0.25. Исследовано более 3000 заданий с различной степенью информационной связности задач. Время решения задач и соответствующие относительные объемы вычислений различались в 2...3 раза. При формировании стратегий с укрупнением задач обработки соответствующие оценки параметров задач исходной модели задания суммировались. При моделировании очереди локальной системы пакетной обработки заданий принималась дисциплина обслуживания в порядке поступления (FCFS).

Стратегия **S1** (с активным перемещением данных в среде) обеспечивает в целом сбалансированную загрузку процессорных узлов различной производительности, в то время как в стратегиях **S2**, **S3** с удаленным доступом к данным имеется явный перекос в сторону загрузки наиболее производительных узлов из-за заметных задержек передачи данных (см. рис. 1). Особенно характерно это проявляется в стратегии **S3** с крупноблочным разбиением задач. Подобные стратегии практически во всех случаях пытаются «захватить» самые производительные узлы, не загружая более медленные, с тем чтобы избежать временных потерь, связанных с удаленным доступом к данным.

При этом стратегии типа **S3** оказываются и самыми «медленными»: время выполнения задачи в **S3** в 1.25… 1.7 раза больше чем в стратегиях **S1**, **S2** (см. рис. 2). Меньшее время решения задач в **S2** по сравнению с **S1** объясняется тем, что планы в **S1** строятся на наихудших и наилучших оценках времени выполнения задач, а в **S2** – на оценках во всем спектре отобранных базовых узлов. В этом смысле планы в **S2** являются более точными, а сама стратегия **S2** – более полной по отношению к **S1**, поскольку охватывает больше вариантов возможных ситуаций с состоянием загрузки узлов. Однако формирование подобных стратегий может быть весьма трудоемким и неоправданным при инструментальной реализации метапланировщика, поскольку необходимо учитывать затраты времени на планирование, отбор опорных планов, а также динамично изменяющиеся состояния узлов.

Относительная «цена» стратегии измеряется как сумма отношений объемов вычислений ко времени фактической загрузки узла задачей. Самой «дешевой» оказывается самая «медленная» стратегия **S3**, а затраты на **S1**, **S2** соизмеримы (см. рис. 2).

В экспериментах оценивался и такой показатель как среднее время «жизни» стратегии, т.е. тот интервал времени, на котором планы вычислений остаются актуальными (приемлемыми) из-за динамики загрузки ресурса и отклонений фактического времени запуска задачи от запрашиваемого метапланировщиком (см. рис. 3). Результаты экспериментов показывают, что наиболее «живучими» являются самые «медленные» стратегии **S3**, а наименее устойчивы к динамике изменений – полные и «быстрые» стратегии типа **S2**. Это вполне объяснимо. Стратегии **S3** всегда пытаются монопольно захватить самый производительный ресурс, минимизировать затраты на обмен данными, что, кстати, и обуславливает большее, чем в **S2**, отклонение времени старта задания от фактического предоставления ресурса. Более полный спектр оценок ожидаемого времени выполнения задач в **S2**, с одной стороны, позволяет учесть большее число возможных событий, связанных с загрузкой ресурсов, а с другой, делает их наиболее уязвимыми с позиций «живучести».

Сравнительный анализ трех типов стратегий позволяет заключить, что предпочтительными, в том числе, и с позиций инструментальной реализации, являются стратегии типа **S1**, поддерживающие политику репликации и активного перемещения данных в среде. Они обеспечивают сбалансированную загрузку процессорного ресурса, занимают промежуточное положение по «цене», времени выполнения задач, временной устойчивости между «быстрыми» и «дорогими» (**S2**) и «медленными», но «дешевыми» (**S3**) типами стратегий. Однако в условиях ограниченности доступного процессорного ресурса предпочтительнее оказываются стратегии **S3**, а активная динамика изменения состава узлов приводит к необходимости формирования полных и более точных стратегий типа **S2**, ориентированных на больший охват возможных событий в распределенной среде.

Новизна полученных результатов заключается в том, что стратегии коаллокаций строятся с учетом структуры задания, неоднородности и вычислительной сложности составляющих его задач, а также различных типов политики хранения и репликации данных в распределенной среде. Отличие этого подхода к организации распределенных вычислений от известных решений заключается в использовании семейства стратегий, состоящего из опорных планов в действиях метапланировщика. Конкретный план и согласованное распределение ресурсов между задачами составного задания осуществляются в зависимости от состояния удаленных ресурсов, информация о которых передается локальными системами пакетной обработки. Коаллокация направляется на удаленный узел в виде ресурсного запроса и выбирается из заранее сформированной стратегии, что делает затраты на планирование значительно ме-

нее трудоемкими в сравнении с построением расписаний алгоритмами имитации отжига и GDA. Важный акспект – генерация стратегий экономичными по времени процедурами. Подбор базового состава ресурсов и распределение задания на число узлов не более десятка реализуются жадными алгоритмами за время, несоизмеримо меньшее затрат на планирование в известных подходах [3].

Авторы выражают благодарность Российскому фонду фундаментальных исследований за финансовую поддержку данной работы в рамках проекта № 06-01-00027.

ЛИТЕРАТУРА:

1. Воеводин Вл. В. Решение больших задач в распределенных вычислительных средах // Автоматика и телемеханика. 2007. № 5. С. 32-45.
2. Коваленко В.Н. Комплексное программное обеспечение грида вычислительного типа. Препринт № 10. М.: ИПМ им. М.В. Келдыша РАН, 2007. 39 с.
3. Топорков В.В. Многоуровневые стратегии согласованного выделения ресурсов в распределенных вычислениях с контрольными сроками // Автоматика и телемеханика. 2007. № 12. С. 131-146.
4. Топорков В.В., Целищев А.С., Бобченков А.В., Рычкова П.В. Проект метапланировщика: генерация сценариев распределенной обработки // Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ: Труды Всероссийской научной конференции (24-29 сентября 2007 г., г. Новороссийск). - М.: Изд-во МГУ им. М.В. Ломоносова, 2007. С. 27-30.