

Т.В. Филимончук, М.А. Волк

Харьковский национальный университет радиоэлектроники, Харьков

РАЗРАБОТКА МОДИФИЦИРОВАННОГО МЕТОДА ОБРАТНОГО ЗАПОЛНЕНИЯ BACKFILL ДЛЯ КОНСЕРВАТИВНОГО РЕЗЕРВИРОВАНИЯ

В работе предложен усовершенствованный метод распределения заданий Backfill с учетом трафика внутри задания, который учитывает интенсивность и объем потоков данных между задачами в задании, что позволяет повысить эффективность использования GRID-системы за счет уменьшения времени выполнения пула заданий. Предложенный метод внедрен в среду имитационного моделирования GRASS для последующего анализа планов распределения и поиска наилучшего распределения для пулов заданий.

Ключевые слова: GRID-система, задания, вычислительные ресурсы, алгоритмы распределения, планировщик заданий, среда моделирования GRASS.

Введение

Постановка проблемы и ее актуальность. В последнее время появление новых информационных технологий, таких как облачные вычисления, GRID-вычисления, гибридные кластерные системы, обусловило качественное развитие технологий решения задач большой вычислительной размерности. К наиболее перспективным направлениям, сегодня можно отнести технологии параллельных распределенных вычислений, которые являются основой GRID-вычислений.

GRID-система – это географически распределенная инфраструктура, построенная на множестве разнородных сетевых ресурсов и используемая для решения научных задач, требующих больших вычислительных мощностей.

Важное место в GRID-системах при распределении отводится планировщикам заданий, в обязанности которых ставится разработка плана распределения, который бы удовлетворял требованиям, выставляемым поставщиками заданий. Разработка плана распределения направлена на оптимизацию ряда параметров целевой функции, которая сокращает время выполнения заданий в системе, что приводит к сокращению простоя вычислительных ресурсов за счет их более эффективного использования.

В настоящее время насчитывается более двадцати известных планировщиков для GRID-систем, однако они обладают рядом недостатков:

- не все предоставляют открытый код;
- некоторые версии платные;
- не все корректно работают с операционной системой поставщика заданий;
- большинство не восстанавливают задания после сбоя и др.

Но главным недостатком выступает их жесткая привязанность к решению конкретного класса задач.

В связи с этим возникают сложности при распределении заданий на вычислительные ресурсы, т.к. на данный момент отсутствует универсальный по эффективности метод, который позволяет распределять любые классы заданий, учитывая при этом все особенности имеющихся доступных ресурсов.

Анализ последних достижений и публикаций. Проведем анализ существующих в настоящее время алгоритмов распределения, которые занимаются распределением заданий на вычислительные ресурсы.

При работе большинства алгоритмов распределения запуск задания осуществляется на первый подходящий вычислительный ресурс без анализа остальных [1–3]. В работе [4] в качестве параметра отбора предложена функция полезности, однако оптимизация выбора ресурса проводится только на основании доступных на данный момент вычислительных ресурсов. Большинство алгоритмов распределения при построении плана не ориентируются на предварительное резервирование, которое может сократить время нахождения задания в системе за счет уменьшения простоя вычислительных ресурсов [1; 2; 5].

Существуют подходы, которые ориентированы на использовании при распределении заданий методов целочисленного линейного или смешанного программирования [6; 7], которые позволяют поставщикам устанавливать предпочтительные временные рамки использования отдельно взятых вычислительных ресурсов. В работе [8] предложен подход использования метода целочисленного линейного программирования в связке с генетическим алгоритмом, который при подборе ресурсов учитывает стоимость их использования для отдельных групп поставщиков заданий. В работе [9] предлагается модель алгоритма распределения заданий, анализирующая динамику загрузки системы и

осуществляющая планировку «на лету».

Существует ряд методов, рассмотренных в работах [10–13], которые при распределении учитывают предпочтения поставщиков заданий, ресурсов, а также администраторов виртуальных организаций. В работе [13] рассмотрен критерий оптимизации на основании предпочтения не только поставщиков заданий, но и поставщиков ресурсов. В данном случае владельцы ресурсов могут регулировать загруженность системы за счет возможности разбиения задания на независимые части. Отдельные части задания направляются на ресурсы, которые в данный момент не загружены, что повышает эффективность использования системы за счет уменьшения простоя вычислительных ресурсов.

В работе [14] предложен логико-вероятностный алгоритм, который ориентирован на многоуровневое планирование с учетом заданных критериев качества (надежность, стоимость, общее время выполнения). Распределение ресурсов проходит в четыре этапа с учетом спроса и предложения вычислительных ресурсов.

В работе [15] предлагается надстройка для планировщика заданий, которая повышает эффективность использования вычислительных ресурсов за счет введения классификации поступающих заданий в систему на основе ряда критериев.

Большинство методов, которые используют заданные критерии отбора [10; 11], имеют существенный недостаток, связанный с линейной временной сложностью, т.е. время построения предварительного плана напрямую зависит от числа доступных на данный момент времени вычислительных ресурсов.

Проведенный анализ показывает, что всем планировщикам GRID-систем присущ существенный недостаток, который ориентирован на использовании единого брокера, который, как правило, используется для определенного класса задач. Поэтому разработка технологии распределения заданий на вычислительные ресурсы является актуальной задачей, т.к. выбор вычислительных ресурсов определяет эффективность использования всей системы в целом. Правильный подбор вычислительных ресурсов под задания влияет на время их простоя, сокращает время и объемы передаваемой между устройствами информации.

Формулирование цели статьи. Целью работы является модификация метода Backfill, который при распределении заданий на вычислительные ресурсы будет учитывать интенсивность и объем потоков данных между задачами в задании, а также внедрение его в среду моделирования GRASS с целью проведения экспериментов, показывающих целесообразность его использования.

Изложение основного материала

Распределение заданий на вычислительные ресурсы GRID-системы – это сложная задача, т.к. в ходе распределения необходимо учитывать влияние множества входных требований, которые выставляют поставщики задания.

Задания, поступающие в GRID-систему, образуют поток $\{Z_i, i = 1, 2, \dots, M\}$, где каждое задание приведено к заданному заранее формату и имеет вид $Z_i = \{ar_i^z, os_i^z, pc_i^z, ps_i^z, ms_i^z, dc_i^z, pt_i^z, ca_i^z, rt_i^z\}$,

$\forall i = 1..M$ [16]. Вычислительные ресурсы также образуют поток $\{R_j, j = 1, 2, \dots, N\}$ и представлены

кортежем $R_j = \{ar_j^r, os_j^r, pc_j^r, ps_j^r, ms_j^r, dc_j^r, bw_j^r, d_j^r\}$,

$\forall j = 1..N$ [16]. Однако, на основании поступивших

данных для распределения, перед планировщиком возникает вопрос, по какому параметру осуществлять оптимизацию. В настоящее время существует ряд задач, в которых необходимо анализировать множество параметров, что, в свою очередь, накладывает трудности на процесс распределения. Для таких случаев рационально использовать методы решения многокритериальных задач оптимизации, которые позволяют объединять ряд параметров в единый критерий. Однако данное объединение осуществляется с помощью эвристических методов, следовательно, с математической точки зрения не существует идеального решения таких задач, т.к. каждое из них имеет преимущества и недостатки.

С целью улучшения качества работы планировщика заданий, был введен обобщенный критерий оценки задания относительно возможности его выполнения на вычислительных ресурсах GRID-системы, позволяющий учитывать все необходимые параметры ресурсов и заданий [17]. Свертывание кортежа упрощает процедуру подбора вычислительных ресурсов для запуска задания, т.к. результатом является получение единственного оптимального варианта распределения.

В настоящее время большой популярностью в системах распределения пользуется метод обратного заполнения Backfill, который учитывает не только информацию о текущем состоянии очереди, но и состояние ресурсов (занят/свободен). Однако при распределении ресурсов с помощью данного метода нет рычагов влияния на ход распределения конкретного задания, т.к. распределение осуществляется по принципу «первый подходящий». В данном случае не учитывается связность задач в задании, за счет которой можно получать выигрыш по времени для получения наилучшего распределения вычислительных ресурсов под требования задания.

Существует ряд задач, которые оперируют

большим объемом входных (выходных) данных: например, задачи рендеринга видеопотока, задачи обработки статистической информации, полученной от астрономических наблюдений. При распределении такого класса заданий необходимо анализировать объем входных и выходных данных, что позволит уменьшить время простоя вычислительных ресурсов GRID-системы и позволит уменьшить время, связанное с пересылкой данных между задачами.

Задачи в задании могут иметь связность трех типов: не связные, малосвязные и сильносвязные. Коэффициент связности задач в задании (coefficient of association – ca) указывает поставщик задания при добавлении его в GRID-систему.

Если задачи в задании не связаны между собой (ca=0), то при подборе вычислительных ресурсов перед планировщиком не возникает никаких проблем, т.е. множество вычислительных ресурсов, которые пришли на вход «Модуля анализа связности» [16] из «Модуля свертки кортежа» [16] остается неизменным и передается на вход планировщика для дальнейшего распределения в зависимости от выбранного алгоритма распределения. Планировщик, анализируя полученную информацию о связности задач в задании, а также данные из БД о предыдущих распределениях подобных заданий, производит подбор вычислительных ресурсов для каждого задания.

Если задачи в задании слабосвязные (ca≤0.3), то множество вычислительных ресурсов также остается неизменным, однако ресурсам, которые находятся в одном месте (кластере), отдается предпочтение (устанавливается метка) и в случае, если при распределении данные ресурсы свободные, то брокер будет ориентироваться на распределение данного задания на них.

Если же задачи в задании сильносвязные (0.3<ca≤1), то из множества вычислительных ресурсов удаляются ресурсы, которые были набраны для запуска задания из разных ресурсов (например, R₁ + R₅), т.к. распределять такие задания необходимо только на одном ресурсе (кластере). В случае, если на момент распределения такого ресурса нет в системе, то задание получает отказ в распределении и возвращается поставщику, либо для изменения характеристик вычислительных ресурсов, либо для разрешения распараллеливания задач. Если ресурс в системе присутствует, но на данный момент занят, задание будет отложено на время, которое требуется для полного освобождения ресурса.

На основании предложенных моделей [16], введения обобщенного критерия оценки задания [17], а также анализе задач задания на связность предлагается модифицировать алгоритм Backfill с консервативным резервированием, за счет введения двух дополнительных параметров: суммарной за-

держки времени передачи пакета по каналу и пропускной способности канала. Эти два параметра в связке позволяют сократить время на пересылку результатов между отдельно взятыми ресурсами и уменьшить трафик в сети за счет разгрузки каналов связи.

Входными данными для предложенного алгоритма (Backfill_mod) являются:

– список заданий в очереди с указанием требований к ресурсам $\{Z_i, i = 1, 2, \dots, M\}$;

– перечень запущенных заданий с указанием использования вычислительных узлов и ожидаемым временем их окончания $Z_i^{run} \in \{Z_i, i = 1, 2, \dots, M\}$;

– рекомендуемый список ресурсов, которые были отобраны для запуска каждого задания $\{R_j^r, j = 1, 2, \dots, K\}$ по обобщенному критерию оценки задания.

Результатом работы данного алгоритма является получение для каждого задания Z_i «окна для запуска».

На первом шаге резервирования происходит сортировка списка запущенных заданий Z_i^{run} в соответствии с их ожидаемым временем окончания rt , $\{Z | \text{sort}(rt_i, \forall i = 1 \dots M)\}$;

– время выполнения заданий из списка t разбивается на периоды, соответствующие времени окончания заданий τ . Для каждого периода записывается количество используемых процессоров $t \subset \tau, \forall \tau := \text{pr} | R_j, \forall j = 1 \dots N$, что формирует профиль использования $\text{Pr of} \subset \{\tau\}$.

На втором шаге резервирования происходит процесс распределения задания, для чего:

– происходит сортировка поступивших заданий из очереди в соответствии с критерием связности;

– на основании множества $\{R_j^r, j = 1, 2, \dots, K\}$

формируется список окон, которые могут быть использованы для запуска задания – W_i^{run} ;
 $i \in 1..M$

– для каждого полученного окна происходит формирование перечня доступных вычислительных ресурсов (CPU);

а) для каждой пары (CPU-брокер) определяется задержка времени передачи пакета по сети (δ_n);

б) для каждой пары (CPU-брокер) определяется пропускная способность канала (C_n);

– происходит набор необходимого числа CPU для запуска задач из задания ($a, z_a \in Z_i$) минимизируя один из критериев:

а) если задание имеет большой объем передаваемых входных данных и соответственно большой

объем выходных данных, то отбираются каналы связи с наибольшими значениями пропускной способности – C_n ;

б) если для запуска задания не требуется передача большого объема информации, то отбираются каналы связи с наименьшим значением δ_n ;

– по каждому полученному окну происходит суммирование значений попарных расстояний и определяется наименьшее из них – $D_{\min} = \min_{i \in 1..M} D_{W_i^{\text{run}}}$. Полученное значение является наилучшим результатом распределения.

После того как задание получило окно на выполнение, происходит обновление профиля с учетом того, что эти процессоры будут заняты.

Предложенный алгоритм Backfill_mod был внедрен в имитационную среду моделирования GRASS для последующего анализа и поиска наилучшего метода распределения пула заданий.

Для выбора эффективного плана распределения был разработан метод выбора наилучшего плана распределения, который включает в себя следующие этапы:

Этап 1: загрузка пулов заданий и ресурсов из реальных GRID-систем:

– $Z_i = \{ar_i^z, os_i^z, pc_i^z, ps_i^z, ms_i^z, dc_i^z, pr_i^z, ca_i^z, rt_i^z\}$, $\forall i = 1..M$ [17];

– $R_j = \{ar_j^r, os_j^r, pc_j^r, ps_j^r, ms_j^r, dc_j^r, dc_j^r, bw_j^r, d_j^r\}$, $\forall j = 1..N$ [17].

Этап 2: выбор алгоритма распределения из множества $Q_k = \{mn_k, lp_k\}$, $\forall k = 1..K$ [17].

Этап 3: запуск процесса моделирования в среде GRASS по каждому алгоритму распределения ($mn \in Q$):

а) на первом шаге в качестве результата получить множество планов распределения ($Plan_k$) $f : Q_k \rightarrow Plan_k$, $\forall k = 1..K$;

б) на втором шаге в качестве результата получить множество времен выполнения планов (T_k) по каждому алгоритму распределения ($mn \in Q$) $f : Plan_k \rightarrow T_k$, $\forall k = 1..K$.

Этап 4: анализ log-файлов для всех алгоритмов распределения ($mn \in Q$) $f : Q_k \rightarrow T_k$, $\forall k = 1..K$;

Этап 5: выбор наилучшего плана распределения на основании принятых правил и ограничений: $plan = \min t_k$, $\forall k = 1..K$.

Для подтверждения эффективности использования метода Backfill_mod в имитационную среду моделирования GRASS были загружены пулы заданий и вычислительных ресурсов из реальных GRID-систем. Используя различные алгоритмы распределения (реализованные в имитационной среде моде-

лирования GRASS), было проведено моделирование, в ходе которого было выявлено, что метод Backfill_mod дает существенный выигрыш по времени для заданий, которые характеризуются большим объемом входных или выходных данных. Это связано с тем, что предложенный метод учитывает интенсивность и объем потоков данных между задачами в задании. На рис. 1 показано время выполнения пула заданий для каждого алгоритма распределения.

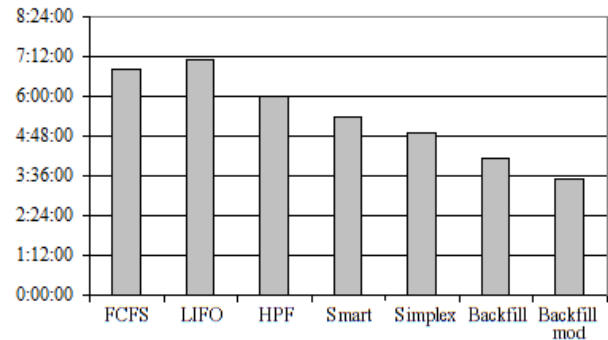


Рис. 1. Время выполнения пула заданий в имитационной среде моделирования GRASS

Выводы

Существующий алгоритм обратного заполнения Backfill не учитывает время, затраченное на передачу входных и выходных данных задания. Данное время напрямую зависит от пропускной способности канала связи и времени задержки. Предложенная модификация алгоритма Backfill с консервативным резервированием позволяет производить распределения заданий, поступающих в GRID-систему, в зависимости от связанности задач в задании. Использование предложенного алгоритма (Backfill_mod) позволяет повысить эффективность использования вычислительных ресурсов GRID-системы (до 22%) за счет уменьшения времени выполнения пула заданий (до 14%) в гетерогенных системах.

Предложенный метод выбора наилучшего плана распределения позволяет для каждого пула заданий определить тот алгоритм распределения, который позволит получить как выигрыш по времени, так и выигрыш по сокращению простоя вычислительных ресурсов и предложить его реальной GRID-системе. В ходе проведения ряда экспериментов была получена зависимость распределения от класса заданий. В настоящее время нет единого алгоритма, получающего оптимальный план распределения для любого пула заданий, но, если известны требования поставщиков задания и характеристики вычислительных ресурсов, имеется возможность проведения натурального эксперимента с целью получения наилучшего распределения в среде GRASS.

Список літератури

1. Aida K. Scheduling Mixed-parallel Applications with Advance Reservations / K. Aida, H. Casanova // In: 17th IEEE Int. Symposium on HPDC, IEEE CS Press, New York. – 2008. – P. 65-74.
2. Elmroth E. A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability / E. Elmroth, J. Tordsson // Concurrency and Computation. – 2009. – Vol. 25(18). – P. 2298-2335.
3. Cafaro M. Preference-Based Matchmaking of Grid Resources with CP-Nets / M. Cafaro, M. Mirto, G. Aloisio // Grid Computing. – 2013. – Vol. 11(2). – P. 211-237.
4. Ernemann C. Economic Scheduling in Grid Computing / C. Ernemann, V. Hamscher, R. Yahyapour // In: Feitelson, D.G., Rudolph L., Schwiegelshohn U. (eds.): JSSPP 2002. Springer, Heidelberg. – 2002. – Vol. 2537. – P. 128-152.
5. Castillo C. Resource Co-allocation for Large-scale Distributed Environments / C. Castillo, G.N. Rouskas, K. Harfoush // In: 18th ACM International Symposium on High Performance Distributed Computing, ACM, New York. – 2009. – P. 137-150.
6. An Advance Reservation-based Co-allocation Algorithm for Distributed Computers and Network Bandwidth on QoS-guaranteed Grids / A. Takefusa, H. Nakada, T. Kudoh, Y. Tanaka: // In: Frachtenberg E., Schwiegelshohn U.(eds.) JSSPP 2010. LNCS, Springer, Heidelberg. – 2010. – Vol. 6253. – P. 16-34.
7. MIP Model Scheduling for Multiclusters [Text] / H. Blanco, F. Guirado, J.L. L rida, V.M. Albornoz // In: EuroPar 2012. LNCS, Springer, Heidelberg. – 2012. – Vol. 7640. – P. 196-206.
8. Garg S.K. A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids / S.K. Garg, P. Konugurthi, R. Buyya // J. Par., Emergent and Distr. Systems. – 2011. – Vol. 26. – P. 493-517.
9. A Dynamic Rescheduling Algorithm for Resource Management in Large Scale Dependable Distributed Systems / A. Olteanu, F. Pop, C. Dobre, V. Cristea // Computers and Mathematics with Applications. – 2012. – Vol. 63(9). – P. 1409-1423.
10. Slot Selection Algorithms in Distributed Computing / V. Toporkov, A. Toporkova, A. Tselishchev, D. Yemelyanov // Journal of Supercomputing. – 2014. – Vol. 69(1). – P. 53-60.
11. Slot Selection Algorithms in Distributed Computing with Non-dedicated and Heterogeneous Resources / V. Toporkov, A. Toporkova, A. Tselishchev, D. Yemelyanov // In: Malyszhkin, V. (ed.) PaCT 2013. LNCS, Springer, Heidelberg. – 2013. – Vol. 7979. – P. 120-134.
12. Методы и эвристики планирования в распределенных вычислениях с неотчуждаемыми ресурсами / В.В. Топорков, А.В. Бобченко, Д.М. Емельянов, А.С. Целищев // Вестник ЮУрГУ, серия «Вычислительная математика и информатика». – 2014. – Т. 3. – №2. – С. 43-62.
13. Топорков В.В. Метопланирование вычислений в распределенных средах с неотчуждаемыми ресурсами / В.В. Топорков, Д.М. Емельянов, А.С. Топоркова // Информационные технологии в науке, образовании и управлении: труды международной конференции IT + S&E`16 / под. ред. проф. Е.Л. Глоризова. М.: ИНИТ. – 2016. – С. 22-31.
14. Костромин Р.О. Модели, методы и средства управления вычислениями в интегрированной кластерной системе / Р.О. Костромин // Фундаментальные исследования. – 2015. – № 6. – С. 35-38.
15. Феоктистов А.Г. Методология концептуализации и классификации потоков заданий масштабируемых приложений в разнородной распределенной вычислительной среде / А.Г. Феоктистов // Системы управления, связи и безопасности. – 2015. – №4. – С. 1-25.
16. Разработка информационной технологии распределения заданий для GRID-систем с использованием имитационной среды моделирования GRASS / Т.В. Филимончук, М.А. Волк, И.В. Рубан, В.Н. Ткачев // Восточно-европейский журнал передовых технологий. – Харьков: Технологический центр, 2016. – № 9 (81). – С. 45-53.
17. Волк М.А. Обобщенный критерий оценки задания для технологии планирования заданий в GRID / М.А. Волк, Т.В. Филимончук // Информатика, математическое моделирование, экономика: Сборник научных статей по итогам Третьей Международной научно-практической конференции, г. Смоленск, 24-26 апреля 2013 г. Том 2. – Смоленск: Смоленский филиал Российского университета кооперации, 2013. – С. 172-176.

Поступила в редколлегию 22.11.2016

Рецензент: д-р техн. наук, проф. С.Г. Удовенко, Харьковский национальный экономический университет им. С. Кузнеця, Харьков.

РОЗРОБКА МОДИФІКОВАНОГО МЕТОДУ ЗВОРОТНОГО ЗАПОВНЕННЯ BACKFILL ДЛЯ КОНСЕРВАТИВНОГО РЕЗЕРВУВАННЯ

Т.В. Філімончук, М.О. Волк

У роботі запропоновано вдосконалений метод розподілу завдань Backfill з урахуванням трафіку всередині завдання, який враховує інтенсивність і обсяг потоків даних між задачами в завданні, це дозволяє підвищити ефективність використання GRID-системи за рахунок зменшення часу виконання пулу завдань. Запропонований метод впроваджений в середу імітаційного моделювання GRASS для подальшого аналізу планів розподілу і пошуку найкращого розподілу для конкретного пулу завдань.

Ключові слова: GRID-система, завдання, обчислювальні ресурси, алгоритми розподілення, планувальник завдань, середа моделювання GRASS.

DEVELOPMENT OF A MODIFIED BACKFILL METHOD FOR CONSERVATIVE RESERVATION

T.V. Filimonchuk, M.A. Volk

This paper proposes a modified Backfill method for job distribution with conservative reservation with traffic parameters, which takes into account the intensity and volume of data flow between tasks in the job, that allows to increase efficient use of the GRID-system by reducing the execution time of jobs in pool. Method is implemented in the GRASS simulation environment for further analysis of the distribution plans and find the best distribution for a particular job pool.

Keywords: GRID-system, tasks, computing resources, distribution algorithms, scheduler, GRASS simulation environment.