

СПОСОБ ИЕРАРХИЧЕСКОГО ПЛАНИРОВАНИЯ ЗАДАЧ В GRID-СИСТЕМАХ

Предложен усовершенствованный способ иерархического планирования. Повышение эффективности планирования достигается за счет физических каналов связи у процессоров вычислительного узла и, возможности поддержки дуплексного режима передачи информации.

The improved method for hierarchical planning is proposed. Efficiency increasing of planning is achieved by physical links of processor's compute node and ability to support duplex communication.

Введение

В связи с расширением сферы использования и увеличением нагрузки на Grid-системы [1] актуальным становится задача повышения эффективности их функционирования. Специфическая проблема, которая лежит в основе концепции Grid-системы, – это скоординированное совместное использование разноронных, динамически меняющихся ресурсов [2]. Эффективность решения данной задачи в значительной степени зависит от способа планирования и распределения задач в системе [3].

В зависимости от характера решаемых задач и типа Grid-систем в них используются как статические [4,5], так и динамические алгоритмы планирования [6,7,8]. При этом используются три основных способа планирования: *централизованный, децентрализованный и иерархический* [9].

Централизованный способ планирования плохо масштабируется с увеличением количества ресурсов, поэтому этот способ пригоден лишь для Grid-систем с ограниченным числом узлов.

Децентрализованный способ планирования обеспечивает лучшую отказоустойчивость и надежность, по сравнению с централизованным способом, однако, отсутствие метапланировщика, который обладает информацией обо всех приложениях и ресурсах, снижает эффективность процесса планирования.

Следует отметить, что задача планирования задач в Grid-системе является NP-полной [10], сложность которой является нелинейной функцией от размерности системы, то есть от количества ее узлов. С целью уменьшения сложности задачи планирования в Grid системах боль-

шой размерности используются иерархические схемы планирования [11].

Модифицированный способ планирования задач

В работе [12] предложен иерархический способ планирования, особенностью которого является улучшенная возможность адаптации к определенным условиям, таким как структура пользовательского приложения и состав вычислительных ресурсов. К недостаткам данного подхода следует отнести то, что не учитывается наличие у процессоров узла множества физических каналов связи с другими процессорами. Это естественно сказывается на эффективности процесса планирования задач.

В общем случае задача планирования заключается в следующем. Пусть имеется Grid система, состоящая из K узлов. Каждый i -й узел состоит из P_i процессоров (компьютеров). Задано приложение из m подзадач с помощью DAG графа. Известны также топологии каждого из K узлов системы, описанные с помощью неориентированных графов. Требуется найти i -й узел Grid системы, который обеспечивает минимальное общее время выполнения заданного приложения $\{T_i | i=1, \dots, K\}$.

$$T_i = \sum_{j=1}^m \sum_{l=1}^{P_i} t_{jli} \times X_{jli} + S_i + \max\{Tr, Tf_i\},$$

где: t_{jli} – время выполнения j -й подзадачи в l -м процессоре i -го узла Grid системы;

S_i – время доставки входных данных и результатов приложения в (из) i -го узла Grid системы;

Tr – время готовности приложения к выполнению в узлах системы;

Tf_i – время освобождения i -го узла Grid системы для выполнения заданного приложения в эксклюзивном режиме;

$X_{ji} = 1$, если j -й подзадача выполняется в l -м процессоре i -го узла;

$X_{ji} = 0$ в противном случае.

В данной работе в качестве основного показателя эффективности используется коэффициент ускорения выполнения задачи

$$K_y = T_l / T_n,$$

где: T_l – время выполнения задачи на одном процессоре узла Grid системы, T_n – время выполнения задачи на процессорах узла Grid системы.

При этом коэффициент эффективности функционирования системы равен:

$$K_{эс} = \frac{K_y}{N},$$

где: K_y – коэффициент ускорения, N – количество процессоров в узле Grid системы.

Эффективность работы планировщика определяется по формуле:

$$K_{эа} = \frac{\max(T_{кр}; \frac{\sum w_i}{N})}{T_n},$$

где: N – количество процессоров, $T_{кр}$ – критическое время задачи, w_i – вес i -й вершины, T_n – время выполнения задачи на заданной вычислительной системе.

С целью повышения эффективности планирования приложений в вычислительных Grid системах предлагается модификация иерархического способа за счет учета:

- количества физических каналов связи процессора узла, соединяющих его с другими процессорами;
- режима дуплекс/полудуплекс канала процессора при выполнении планирования.

Рассмотрим основные этапы предлагаемого иерархического способа планирования для вычислительных Grid систем.

1. Пользователь представляет приложение в виде графа задачи (рис.1) и формирует ресурсный запрос, с учетом заданного критерия оптимизации.

2. **Метадиспетчер имеет следующую информацию об узлах: стоимость работы каждого узла за единицу времени, а также скорость доставки данных в каждый узел.**

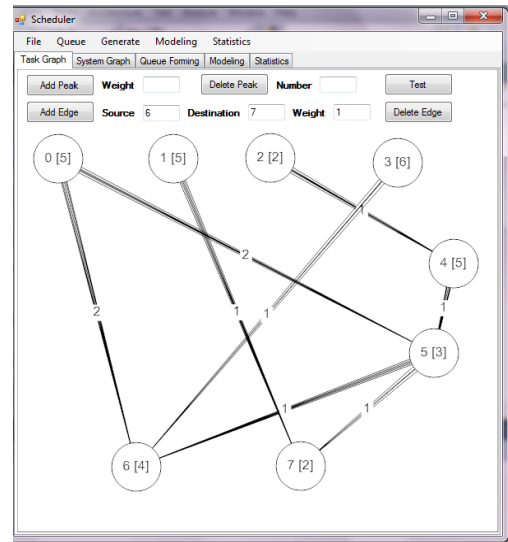


Рис.1

3. Метадиспетчер посылает локальным менеджерам граф задачи и получает в ответ возможное время запуска на каждом из них. Для определения этого времени на каждом из локальных ресурсов выполняется планирование заданного приложения.

4. Локальные менеджеры сообщают возможное время освобождения своих узлов для определения начального времени выполнения приложения.

5. В зависимости от заданного пользователем критерия оптимизации и имеющихся данных, метадиспетчер выбирает оптимальный узел Grid системы и посылает туда запрос на резервирование ресурсов, причем резервирование должно быть создано таким образом, чтобы воспользоваться зарезервированными ресурсами могло только это приложение.

6. Если резервирование выполнено, приложение посылается в очередь выбранного узла. В том случае если ресурсы, необходимые для запуска задания освободятся раньше времени начала резервирования, необходимо обеспечить разрешение запуска приложения.

В рамках данной работы был проведен ряд экспериментов, направленных на сравнение эффективности предложенного алгоритма планирования для однородных узлов (ПА) с эффективностью базового алгоритма (БА), описанного в работе [12].

Для сбора статистики задавались такие переменные:

- Дуплексность.
- Количество физических линков (PL).
- Количество вершин графа задачи (Nв).

Моделирование выполнялось при таких параметрах:

- Веса вершин в диапазоне $1 \dots N_{в}/2$.
- Веса пересылок в диапазоне $1 \dots N_{в}/2$.
- Связность в диапазоне $10 \dots 90\%$ с шагом 10% (C).
- Отношение количества вершин графа задачи к количеству вершин графа системы $5 : 1, 10 : 1$. (D).

- Топологии: полносвязная на 4 узла, звезда на 4 узла.

- Количество физических линков: 3, 1 (PL)

На рис. 2. приведен пример формирования очереди из графа задачи.

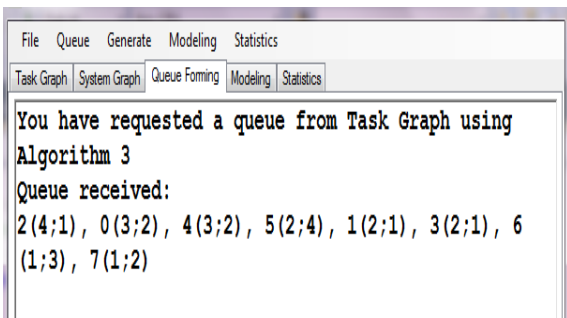


Рис. 2

Результатом погружения является модифицированная диаграмма Ганта (рис.3), в которой кроме расписаний работы каждого процессора отображено также расписание работы каждого канала связи.

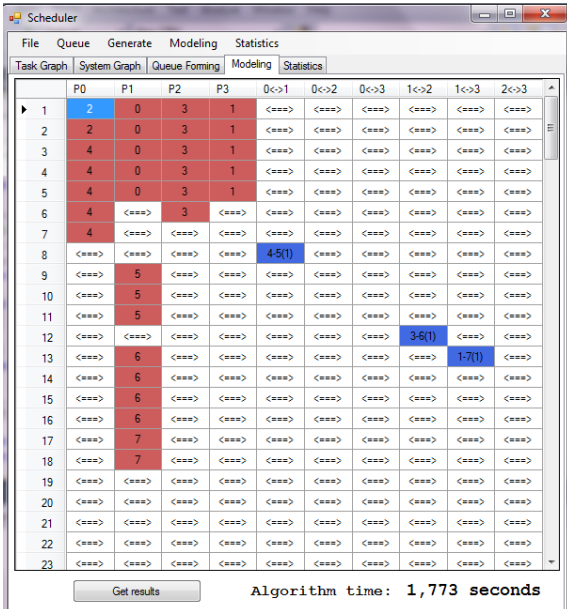


Рис. 3

В результате моделирования были получены результаты, позволяющие оценить влияние количества физических линков процессора (рис.4) и режима дуплекс на коэффициент ускорения (рис.5).

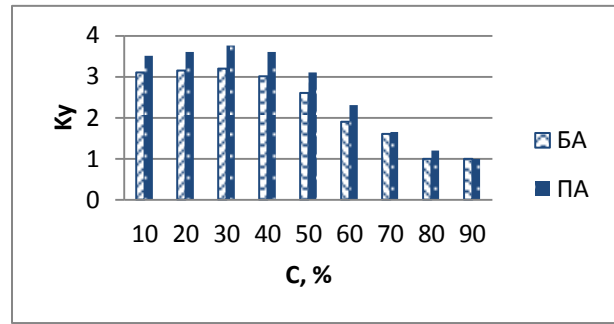


Рис. 4

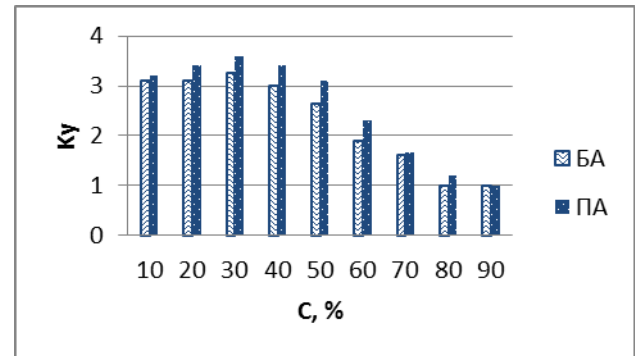


Рис. 5

На рис.6 представлена зависимость коэффициента ускорения от связности задачи с учетом физических линков и дуплексного режима передачи информации.

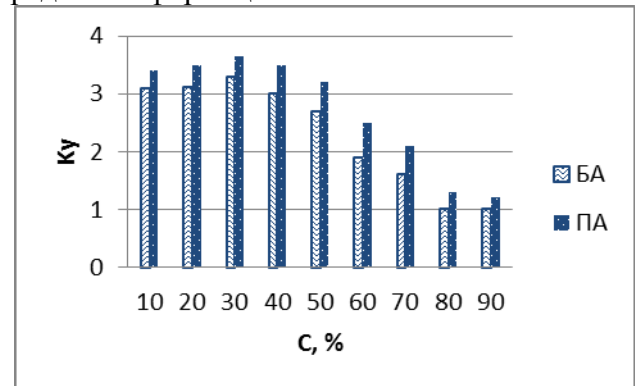


Рис. 6

Очевидно, что учет наличия трех физических линков у каждого процессора узла позволяет увеличить коэффициент ускорения в среднем на 10-20%. Такой результат достигается за счет одновременного выполнения пересылок данных по нескольким каналам.

Учет дуплексного режима канала процессора увеличивает коэффициент ускорения в среднем на 5-15%. Это обусловлено тем, что при высоких связностях задачи в графе возникает много перекрестных связей, в результате чего у процессоров возникает необходимость одновременно передавать данные друг другу. Если же учесть обе характеристики системы, то, как следует из рис. 6, прирост коэффициента ускорения по сравнению с базовым алгоритмом,

представленным в работе [12], составит в среднем 20-25%. Производительность узла убывает по мере увеличения связности задачи. Это связано с большим количеством пересылок. Наиболее близкий к оптимальному коэффициент ускорения наблюдается на отметке 30% связности задачи.

Заключение

Предложенный в данной работе усовершенствованный иерархический способ планирова-

ния задач позволяет достигнуть более высокой эффективности планирования за счет учета количества физических каналов связи у процессоров конкретного вычислительного узла, а также учета возможности поддержки каналами связи процессоров дуплексного режима.

Выполнена практическая реализация предложенного алгоритма в виде программного продукта, который может быть использован для моделирования планирования задач в GRID-системах.

Список литературы

1. I. Foster, C. Kesselman. Computational Grids. Chapter 2 of "The Grid: Blueprint for a New Computing Infrastructure", Morgan-Kaufman, 1999.
2. I. Foster, C. Kesselman and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, in the International J. Supercomputer Applications, 15(3), pp. 200-220, fall 2001.
3. Y. Zhu, A Survey on Grid Scheduling Systems, Department of Computer Science, Hong Kong University of Science and Technology, 2003.
4. S.Y. You, H.Y. Kim, D. H. Hwang, S. C. Kim, Task Scheduling Algorithm in GRID Considering Heterogeneous Environment, in Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA '04, pp. 240-245, Nevada, USA, June 2004.
5. Симоненко В.П., Куреньов А.С. Алгоритм статического планирования для GRID систем. - Вісник НТУУ "КПІ". Сер. Інформатика, управління та обчислювальна техніка. - 2011. Випуск 53. - С.10 – 17.
6. A. Takefusa, S. Matsuoka, H. Casanova and F. Berman, A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid, in Proc. of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01), pp. 406-415, San Francisco, California USA, August 2001.
7. N. Muthuvelu, J. Liu, N. L. Soe, S. Venugopal, A. Sulistio and R. Buyya, A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids, Proceedings of the 3rd Australasian Workshop on Grid Computing and e-Research (AusGrid 2005), Newcastle, Australia, January 30 – February 4, 2005.
8. K. Kurowski, B. Ludwiczak, J. Nabrzycki, A. Oleksiak and J. Pukacki, Improving Grid Level Throughput Using Job Migration And Rescheduling, Scientific Programming vol.12, No.4, pp. 263-273, 2004
9. V. Hamscher, U. Schwiegelshohn, A. Streit, R. Yahyapour, Evaluation of Job-Scheduling Strategies for Grid Computing, in Proc. of GRID 2000 GRID 2000, First IEEE/ACM International Workshop, pp. 191-202, Bangalore, India, December 2000.
10. H. El-Rewini, T. Lewis, and H. Ali, Task Scheduling in Parallel and Distributed Systems, ISBN: 0130992356, PTR Prentice Hall, 1994.
11. T. Casavant, and J. Kuhl, A Taxonomy of Scheduling in General-purpose Distributed Computing Systems, in IEEE Trans. on Software Engineering Vol. 14, No.2, pp. 141-154, February 1988.
12. Кулаков Ю.А., Русанова О.В., Шевело А.П. Иерархический способ планирования для GRID-систем - Вісник НТУУ "КПІ". Сер. Інформатика, управління та обчислювальна техніка. - 2009. Випуск 51. - С.57– 66.