

СИМОНЕНКО А.В.,
 ПИХ С.В.,
 СЛУЦКИЙ Н.В.,
 ВОРОБИЙОВ В.В.

СИСТЕМА ПРОСТРАНСТВЕННОГО РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

В статье рассматриваются основные подходы распределения (планирования) заданий в глобальных GRID-системах. Показано, что в одноуровневых неоднородных системах более эффективно использование планировщиков пространственного типа, основанных на принципах исключющего планирования с учетом пространственного отношения заявка – ресурс.

This article is devoted to the main approaches of the distribution (planning) tasks in distributed systems. It is shown that siblings of inhomogeneous systems more effectively use planners-based on the principles of exclusionary planning.

1. Введение

Весь процесс планирования в параллельной (распределенной) ВС можно представить в виде многоуровневой уровневой схемы, где каждый уровень (являющийся составной частью полной схемы планирования) представляет собой отдельную задачу. Составными частями полной задачи планирования являются: ввод потока заданий, мониторинг среды исполнения, структурный анализ соответствия входного потока заявок имеющимся ресурсам, адаптация распределения работ соответственно особенностям ВС, планирование выполнения задач в ПВС в соответствии с функцией цели, адресное распределение процессов по ресурсам, перераспределение процессов при изменении состояния ВС.

В настоящий момент, наиболее распространенными являются системы управления пакетной обработки заданий. Они используются в таких системах как Platform LSF, Windows Compute Cluster Server, PBS, Condor, SGE, LoadLever. С развитием grid-технологий вопрос управления распределенными вычислительными системами уделяется огромное внимание. На данный момент одним из наиболее распространенных алгоритмов, позволяющих производить планирование нагрузки вычислительной системы, является алгоритм Backfill. Данный алгоритм реализован в составе внешнего планировщика Maui. Maui – планировщик заданий с открытым кодом для использования в кластерах и суперкомпьютерах. Для определения ресурсов, готовых для

приема задания, можно использовать локальный менеджер ресурсов (LMP или LRM – Local Resource Manager) [1, 4, 5, 6, 7]. Таким образом исходной информацией для работы планировщика является множество имеющихся, готовых ресурсов (вычислительных узлов) и множество заданий (задач) для распределения. Ввиду того, что в данной статье рассматривается строго неоднородное отношение задача – ресурс, то претендование заявки на ресурс отображается, вектором $R = \{R_1, R_2, R_3, \dots, R_n\}$ (где R_i – булево значение, указывающее на то, может ли данный ресурс использоваться для обработки данной заявки, 1 – может, 0 – не может). Вектор R определяет множество ресурсов, на которых может быть выполнена заявка.

Менеджер ресурсов периодически через время Δt , заданное администратором системы, выполняет мониторинг системы и опрашивает ресурсы (свободны они или нет). После опроса система отправляет каждую заявку на соответствующий свободный ресурс, руководствуясь определённым алгоритмом обработки очереди [5, 6], который может обслужить данную заявку (естественно в векторе R_i данного ресурса равен 1), или задерживает её в очереди, если все возможные ресурсы заняты.

Существующие решения, использующие потоковое планирование, не учитывают влияния любого назначения на последующие. Т.е. когда заявка, погруженная в определённый ресурс, но имеющая возможность выбора, может заблокировать назначение на него дру-

гих заявок, которые не имеют других претендентов. В этом случае заявка будет ожидать освобождения ресурса. В этом случае система планирования не будет давать эффективное решение по показателю – среднее время ожидания.

Если учесть что за время Δt система набирает пул задач, то анализируя это множество векторов, отображающих претендование заявок на ресурсы, можно определить, какие именно ресурсы и каким заявкам назначить, при этом самые требовательные получают свои ресурсы первыми, чтобы уменьшить время ожидания.

В данной работе выполнен сравнительный анализ среднего времени ожидания в системах, работающих по обычному потоковому принципу планирования и предлагаемому алгоритму, основанному на принципе пошагового, исключающего планирования.

2. Основные критерии выбора принципа планирования

Если учитывать, что в системе формируется пул задач, то с учётом того, что в ЛМР поступает информация в виде битового вектора (1 – ресурс можно назначить этой заявке, 0 – ресурс не назначать), то за время Δt наберётся определённое количество заявок и свободных ресурсов, удовлетворяющее условию $JN = RN$, где JN – количество заявок, RN – общее количество ресурсов.

При выборе принципа планирования следует также учитывать интенсивность входного потока заявок. В том случае, когда интенсивность мала, неоднородность системы слабо выражена и количество заявок, пришедших за время Δt значительно меньше числа освобождающихся ресурсов следует использовать обычные алгоритмы, основанные на принципе первого подходящего. Таким образом, алгоритм должен иметь определённую степень адаптивности и учитывать степень интенсивности поступления входных заявок, исходя из уровня загрузки системы.

Основным критерием, применения предложенного алгоритма, помимо интенсивности поступления входных заявок, является разреженность матрицы, образующейся из векторов пула задач.

3. Алгоритм планирования.

Предлагается алгоритм, основанный на следующих положениях:

- Использование принципа исключающего планирования,
- Использование принципа пошагового конструирования,
- Применение принципа min-max выбора.
- Применение принципа оптимальности Белмана,
- Применение принципа «надежного состояния системы выбора».

Алгоритм назначения:

1. Отмечается пул задач для распределения и формируется матрица выбора.

2. Поиск в пуле среди выделенных для распределения заявок, заявки, имеющей минимальное возможное мест размещения. (Входной фильтр менеджера ресурсов вычисляет сумму мест размещения заявки ещё на входе).

3. Для выделенной заявки (строки) ищется ресурс, имеющий минимальное число требований на захват ресурса от других заявок. Поиск ведется только для тех столбцов в которых имеется «1» в выбранной строке.

4. Найденные в пунктах 1, 2 строка и столбец удаляется из матрицы выбора.

5. Выбранная заявка из матрицы выбора может быть направлена на исполнение в выбранный ресурс.

6. Пункты 2-4 повторяются до полного размещения всех заявок.

4. Сравнительный анализ алгоритмов потокового и пространственного планирования

Сравнительный анализ алгоритмов потокового и пространственного планирования характеристик был выполнен с помощью разработанной системы моделирования работы распределенной системы. Сравнение выполнялось по различным критериям: интенсивности входного потока заявок, времени захвата задачей ресурса, вычислительной мощности ресурсов, степени неоднородности вычислительных ресурсов.

На рис. 1 – 4 изображены графики среднего времени ожидания задач и простоя ресурсов от степени неоднородности вычислительных ресурсов. Анализ полученных графики (1 – 4)

позволяет сделать вывод, что с уменьшением разреженности матрицы среднее время ожидания и среднее время простоя экспоненциально снижается, достигая некоторой константы, которое определяется временными характеристиками алгоритма планирования, тогда как количество решённых задач логарифмически увеличивается.

Исходя из полученных результатов нескольких тестов, анализируя их, можно прийти

к выводу, что лучше всего оценивать нужную степень разреженности матрицы стоит по времени ожидания заявок, а не по простоям ресурса.

При разреженности матрицы выбора (% наличия «0» в матрице выбора) более 30% получаем существенное уменьшение среднего времени ожидания заявок в системе обслуживания и простоя ресурсов вычислительной системы.

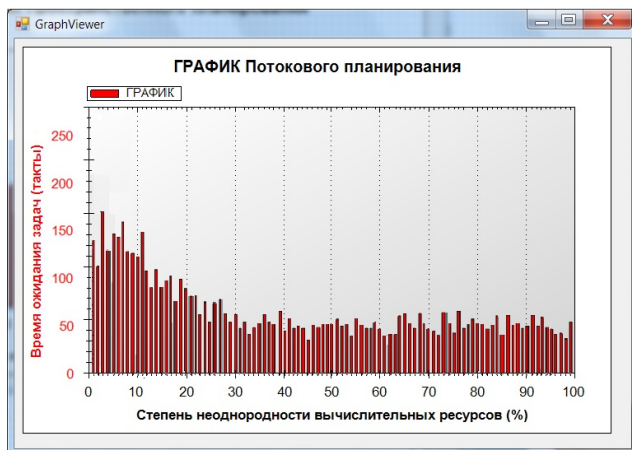


Рис. 1. Зависимость среднего времени ожидания задач при изменении степени неоднородности вычислительных ресурсов. (пространственный планировщик)

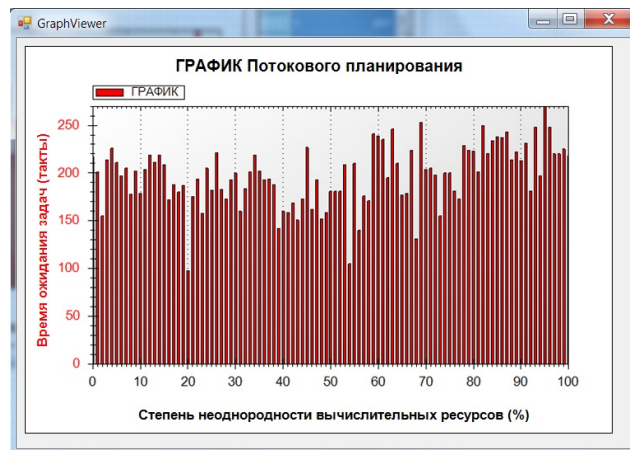


Рис. 2. Зависимость среднего времени ожидания задач при изменении степени неоднородности вычислительных ресурсов. (поточковый планировщик)



Рис. 3. Зависимость времени простоя ресурсов от степени неоднородности вычислительных ресурсов. (пространственный планировщик)

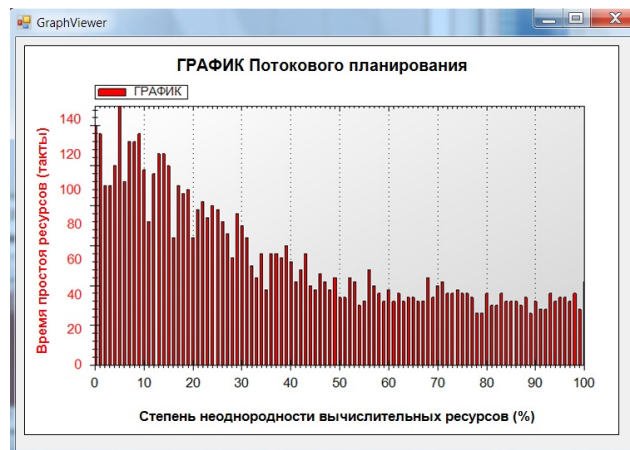


Рис. 4. Зависимость времени простоя ресурсов от степени неоднородности вычислительных ресурсов. (поточковый планировщик)

Выводы

Сравнительное моделирование двух типов планировщиков позволяет разработать систему планирования адаптивного типа, когда при малой степени интенсивности и малой степени неоднородности планировщик использует

поточковый принцип распределения заявок по ресурсам применяя либо выбор заявок по приоритету либо используя дисциплину FIFO. Наилучшее распределение заявок по ресурсам можно достичь в глобальных GRID системах в которых степень неоднородности выше 50%.

Список литературы

1. Коваленко В.Н., Корягин Д.А., Коваленко Д.А., Любимский Э.З. Метод опережающего планирования для GRID. Препринт ИПМ, 2005 г. – № 112.
2. Симоненко А.В. Выбор стратегии пространственного планирования в параллельных вычислительных системах. Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техніка, Київ 2001 р. №35.- с. 104-108.
3. Симоненко В.П. Математическая постановка задачи динамического распределения работ в GRID системах и оценки качества решения. Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техніка, Київ 2011 р. №53.- с. 37-41.
4. Fangpeng Dong, Selim G. Akl. Scheduling Algorithms for GRID Computing: State of the Art and Open Problems. Technical Report No. 2006-504. School of Computing. Queen's University, Kingston, Ontario, January 2006.
5. Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, Prachet Bhuyan A Survey on Grid Scheduling Systems. World Academy of Science, Engineering and Technology 64 2010.
6. Klaus Krauter, Rajkumar Buyya, Muthucumaru Maheswaran. A Taxonomy and Survey of Grid Resource Management Systems. Software – Practice & Experience, Volume 32, Issue 2 , February 2002.
7. F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J.Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, D.Zagorodnov. Adaptive Computing on the GRID using AppLeS, in IEEE. Trans On Parallel and Distributed Systems (TPDS), 2003.
8. K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S.Tuecke, A Resource Management Architecture for Metacomputing Systems, In D.G.Feitelson and L. Rudolph, editors, in Proc of the 4th Workshop on Job Scheduling Strategies for Parallel Processing, LNCS Vol. 1459, Orlando, Florida USA, March 1998.