Исследование алгоритма планирования POS для проблемно-ориентированных вычислительных сред*

А.В. Шамакина, Л.Б. Соколинский

ФГБОУ ВПО «Южно-Уральский государственный университет» (НИУ)

В работе исследуется новый алгоритм POS (Problem-Oriented Scheduling) для планирования вычислительных заданий с потоковой структурой. Вычислительное задание может быть представлено в виде ориентированного ациклического графа, узлами которого являются задачи, являющиеся составными частями задания, а дуги соответствуют потокам данных, передаваемых между отдельными задачами. Разработанный алгоритм позволяет масштабировать отдельные задачи в задании и указывать количество процессорных ядер для каждого из вычислительных узлов. Исследована эффективность алгоритма POS в сравнении с двумя известными алгоритмами планирования: DSC и Min-Min.

1. Введение

Развитие технологий распределенных вычислений в конце 1990-х годов позволило объединить географически-распределенные по всему миру гетерогенные ресурсы. Появились технические возможности для решения масштабных задач в области науки, техники и коммерции на территориально-распределенных ресурсах, принадлежащих разным владельцам [1]. Исследования данной тематики привело к возникновению концепции грид вычислений, и затем к новой концепции облачных вычислений. Для раскрытия всех потенциальных возможностей использования распределенных вычислительных ресурсов принципиально важно наличие результативных и эффективных алгоритмов планирования, используемых менеджерами ресурсов. На сегодняшний день предложено большое количество алгоритмов планирования, ориентированных на их использование в распределенных вычислительных средах [2]. Некоторые из алгоритмов производят планирование с учетом потоков работ в сложных приложениях. Однако разработанные алгоритмы часто не учитывают специфику области задачи и не используют дополнительные знания о ней, алгоритмы планирования рассматривают информацию о прикладных сервисах, но не используют знания о производственном процессе, описываемом потоком работ.

В работах [3–4] описан новый алгоритм планирования ресурсов POS (Problem-Oriented Scheduling) для распределенных многоядерных вычислительных сред, который позволяет планировать запуск одной задачи на нескольких процессорных ядрах с учетом ограничений по масштабируемости данной задачи. Вычислительное задание, представляющее собой поток работ, моделируется в виде нагруженного, помеченного ориентированного графа задания. Каждая задача-вершина помечается парой натуральных чисел, первое из которых задает время выполнения задачи на одном процессорном ядре, а второе – максимальное количество ядер, до которого задача демонстрирует ускорение близкое к линейному. Веса дуг задают объем данных, передаваемый между задачами. Предложенный алгоритм планирования ресурсов предусматривает построение последовательности конфигураций графа задания с целью минимизации критического пути в данном графе. В начальной конфигурации граф задания разбивается в каноническую ярусно-параллельную форму (ЯПФ). Задаются начальные кластеризация задач по вычислительным кластерам и расписание их выполнения. На следующем шаге происходит переход от начальной конфигурации графа к новой конфигурации с помощью перемещения одной из задач по ярусам ЯПФ и ее присоединения к определенному кластеру. Таким образом, изменяются кластеризация и расписание. Переход к новой конфигурации происходит только в том случае, когда сложность критического пути не увеличивается. Процесс построения последовательности конфигураций завершается, когда все вершины зафиксированы. Данный алго-

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 14-07-31159.

ритм реализован в виде грид-сервиса на платформе UNICORE на языке программирования Java [5].

В данной статье отражены результаты вычислительных экспериментов по исследованию адекватности и эффективности разработанного алгоритма планирования ресурсов POS для проблемно-ориентированных вычислительных сред. В разделе 1. описывается методика проведения экспериментов и указываются методы генерации тестовых графов заданий. Раздел 2. посвящен экспериментам, подтверждающим адекватность модели вычислительной среды и эффективность алгоритма планирования ресурсами POS на различных графах вычислительных заданий. Приводятся результаты вычислительных экспериментов по сравнению алгоритмов POS, DSC и Min-Min.

2. Методика проведения экспериментов

Исследование алгоритма планирования ресурсов POS проводилось на следующих двух классах заданий:

1) многокритериальная оптимизация (МКО);
2) случайные задания (СЗ).

Класс МКО представляет вычислительные задания по многокритериальной оптимизации, составляющие большой процент загрузки современных суперкомпьютерных и распределенных вычислительных систем. Графы заданий класса МКО имеют следующую структуру: ярусно-параллельная форма состоит из трех ярусов. Первый и третий ярусы параллельной формы содержат по одной вершине. Второй ярус содержит w вершин. Число w указывается при генерации графа. Вершина первого яруса соединена дугами со всеми вершинами второго яруса. Вершина третьего яруса также соединена дугами со всеми вершинами второго яруса. Каждой задаче-вершине сопоставляется пара чисел: максимальная масштабируемость задачи m_v и время выполнения задачи на одном процессорном ядре t_v. Числа m_v и t_v являются константами и имеют для всех задач одинаковые значения. Аналогично, каждой дуге сопоставлялась константа δ – объем передаваемых данных, одинаковая для всех дуг. На рис. 1 приведен пример графа задания класса МКО для ширины $w = 100$.

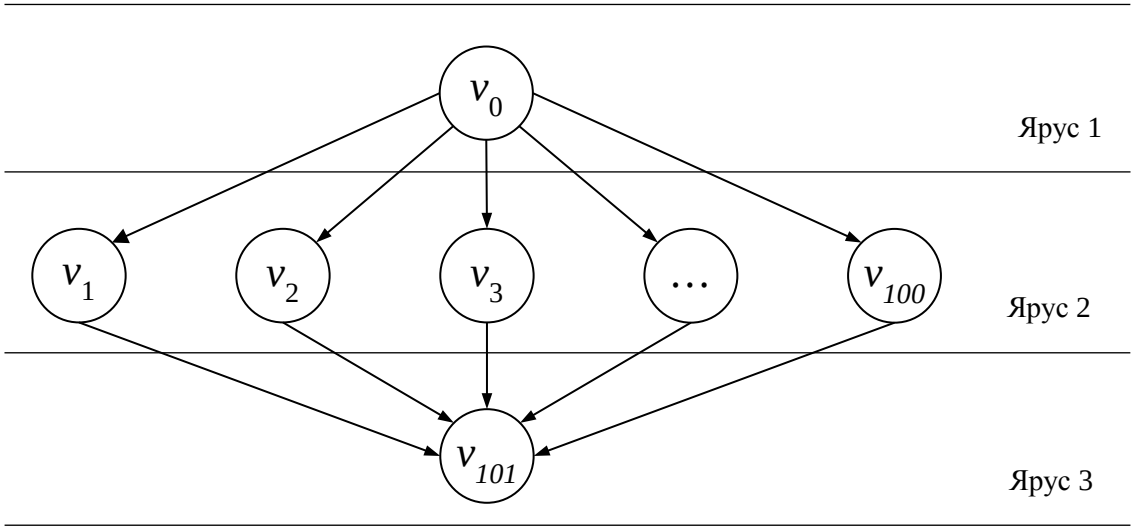

Рис. 1. Пример графа задания класса МКО.

Класс СЗ представляет *случайные* задания с различным числом вершин и дуг. Качественной характеристикой графа задания в классе СЗ является отношение T/Δ, где T – среднее время выполнения задачи на одном ядре, Δ – средний вес дуги. По этому параметру в классе СЗ могут быть выделены следующие три важные группы.

1) M1 – сбалансированные графы заданий, у которых $T/\Delta \in (0.8,1.2)$. В таких заданиях время на передачу данных сопоставимо с временем вычислений.

2) M2 – крупнозернистые графы заданий, у которых $T/\Delta \in (3,10)$. В таких заданиях большая часть времени тратится на вычисления.

3) M3 – мелкозернистые графы заданий, у которых $T/\Delta \in (0.1,0.3)$. В таких заданиях значительное время затрачивается на передачу данных, а вычисления требуют незначительного времени.

Заголовок раздела оформляется без отрыва от первого абзаца с выравниванием по левому краю. Сверху и снизу заголовок отделяется одной пустой строкой. Примеры заголовков и их стили представлены в таблице 1.

3. Результаты экспериментов

В первой серии экспериментов исследовалась плотность расписания, генерируемого алгоритмом POS. Под плотностью здесь понимается величина, обратная количеству вычислительных узлов, задействованных для выполнения задания.

Сначала плотность расписания была исследована для задач класса МКО. Эксперименты проводились для трех значений параметра w, определяющего ширину графа задания: 100, 200, 300. Для всех вершин и дуг графа использовались одинаковые значения весов и маркировки, приведенные в табл. 1.

Табл. 1. Параметры для построения графов заданий класса МКО

Параметр	Семантика	Значение
m_v	Масштабируемость задач	10
t_v	Время выполнения задачи на 1 ядре	100
δ	Объем данных, передаваемых по дуге	50

Результаты экспериментов приведены на рис. 2 в виде зависимости количества задействованных вычислительных узлов от количества процессорных ядер в одном вычислительном узле. Графики показывают, что при увеличении количества ядер в вычислительном узле плотность расписания для заданий МКО возрастает и стремится к максимальному значению 1 во всех рассмотренных случаях. При этом плотность расписания существенно возрастает при увеличении ширины графа задания.

Затем плотность расписания была исследована для задач класса СЗ. Эксперименты проводились для трех значений параметра w, определяющего ширину графа задания: 30, 50, 70. Для всех вершин и дуг графа использовались одинаковые значения весов и маркировки, приведенные в табл. 2.

Табл. 2. Параметры для построения графов заданий класса СЗ

Параметр	Семантика	Значение
l	Высота графа задания	10
m_v	Масштабируемость задач	10
t_v	Время выполнения задачи на 1 ядре	100
δ	Объем данных, передаваемых по дуге	50

Результаты экспериментов приведены на рис. 3. Графики показывают, что при увеличении количества ядер в вычислительном узле плотность расписания для заданий СЗ также возрастает и стремится к максимальному значению 1 во всех рассмотренных случаях. При этом плотность расписания существенно возрастает при увеличении ширины графа задания.

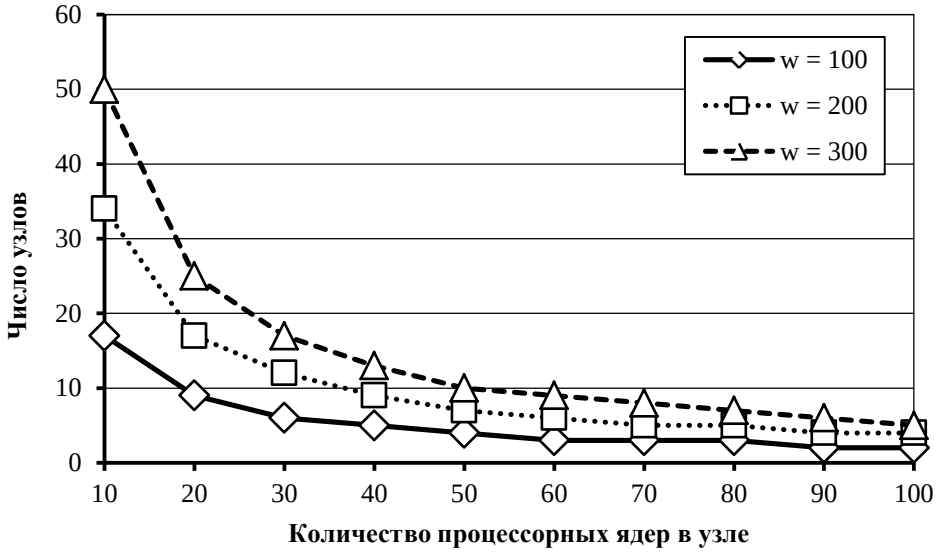

Рис. 2. Зависимость количества задействованных вычислительных узлов от количества процессорных ядер в вычислительном кластере для задания МКО.

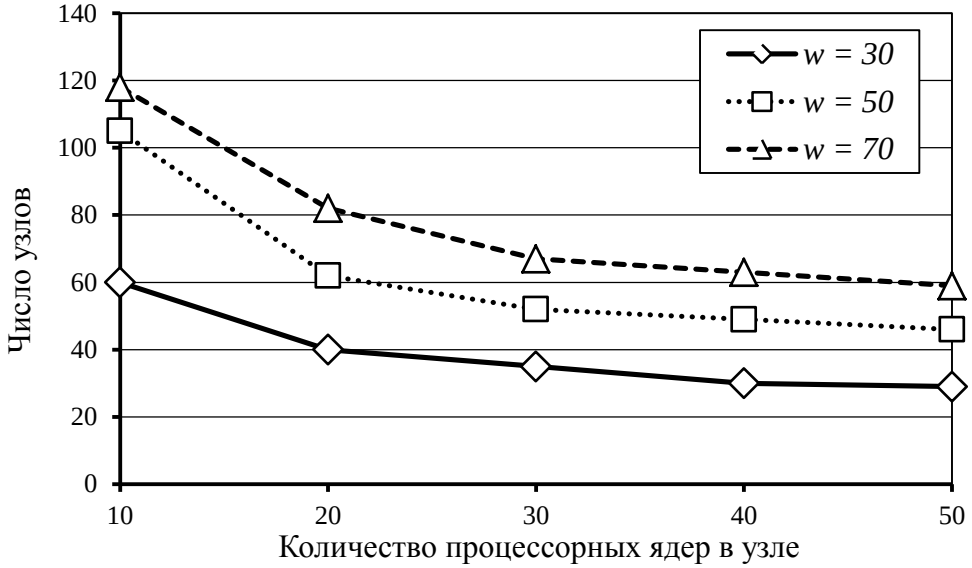

Рис. 3. Зависимость количества задействованных вычислительных узлов от количества процессорных ядер в вычислительном кластере для класса С3.

Во второй серии экспериментов была исследована эффективность алгоритма POS в сравнении с двумя другими алгоритмами планирования: DSC [6] и Min-Min [7]. Для этого были подготовлены три группы графов заданий $M1$, $M2$ и $M3$ с параметрами, приведенными в табл. 3.

Табл. 3. Параметры построения графов заданий

Параметр	Семантика	Значение
m_v	Масштабируемость задачи	20
t_v	Время выполнения задачи на 1 ядре	40
δ_{M1}	Средний вес дуги для группы $M1$	40
δ_{M2}	Средний вес дуги для группы $M2$	10
δ_{M3}	Средний вес дуги для группы $M3$	400
d	Число ядер на вычислительном узле	100

Табл. 4. Сравнение алгоритмов POS, DSC и Min-Min для группы M1

№ п/п	l	w	$\|V\|$	$\|E\|$	Количество задействованных вычислительных узлов			Относительная эффективность	
					POS	DSC	MinMin	POS/DSC	POS/MinMin
1	5	10-20	51	126	7	21	4	73,57%	88,72%
2	10	10-20	117	296	4	36	4	72,76%	89,51%
3	10	20-30	211	505	16	68	6	73,45%	88,04%
4	20	5-10	137	252	2	46	2	90,18%	94,36%
Среднее значение								77,49%	90,16%

Табл. 5. Сравнение алгоритмов POS, DSC и Min-Min для группы M2

№ п/п	l	w	$\|V\|$	$\|E\|$	Количество задействованных вычислительных узлов			Относительная эффективность	
					POS	DSC	MinMin	POS/DSC	POS/MinMin
1	5	10-20	49	113	3	25	4	85,00%	82,35%
2	10	10-20	130	390	12	42	4	79,14%	60,16%
3	10	20-30	206	464	17	73	6	75,32%	71,14%
4	20	5-10	141	290	13	41	2	79,14%	32,17%
Среднее значение								79,65%	61,46%

Табл. 6. Сравнение алгоритмов POS, DSC и Min-Min для группы M3

№ п/п	l	w	$\|V\|$	$\|E\|$	Количество задействованных вычислительных узлов			Относительная эффективность	
					POS	DSC	MinMin	POS/DSC	POS/MinMin
1	5	10-20	59	190	4	21	4	77,80%	92,40%
2	10	10-20	123	378	3	34	4	79,55%	97,64%
3	10	20-30	201	453	9	61	6	51,36%	90,72%
4	20	5-10	133	287	2	29	2	68,04%	87,31%
Среднее значение								69,19%	92,02%

Результаты представлены в табл. 4–6. Для сравнения алгоритма POS с алгоритмами DSC и Min-Min вычислялась *относительная эффективность* $\left(1 - \frac{T_{pos}}{T}\right) \cdot 100\%$, T_{dsc} — среднее время выполнения задания для алгоритма DSC, T — время выполнения задания алгоритмом DSC или Min-Min соответственно. Во всех случаях алгоритм планирования ресурсов POS демонстрирует существенное ускорение времени выполнения задания по сравнению с алгоритмами DSC и Min-Min. При этом алгоритм POS существенно превосходит DSC по плотности получаемого расписания и показывает результаты, сравнимые по этому показателю с алгоритмом Min-Min.

4. Заключение

Проведенные вычислительные эксперименты показывают, что для представительных классах задач МКО и СЗ алгоритм POS генерирует расписания, которые качественно превосходят по эффективности расписания, генерируемые известными алгоритмами планирования DSC и Min-Min. Это достигается за счет того, что алгоритм POS анализирует все зависимости задач по данным, как и алгоритм DSC, и обеспечивает возможность распределения задач по процессорным ядрам, подобно алгоритму Min-Min. Алгоритм планирования POS для распределенных многоядерных вычислительных сред позволяет планировать запуск одной задачи на нескольких процессорных ядрах с учетом ограничений по масштабируемости данной задачи. Данный алгоритм планирования предназначен для создания системы интеллектуального планирования и диспетчеризации приложений с потоковой структурой с целью повышения эффективности функционирования суперкомпьютерных комплексов с кластерной архитектурой.

Литература

1. Радченко Г.И., Соколинский Л.Б. Технология построения виртуальных испытательных стендов в распределенных вычислительных средах // Научно-технический вестник информационных технологий, механики и оптики. 2008. № 54. С. 134-139.
2. Шамакина А.В. Обзор технологий распределенных вычислений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2014. Т. 3, № 3. С. 51-85.
3. Шамакина А.В., Соколинский Л.Б. Формальная модель задания в распределенных вычислительных средах // Параллельные вычислительные технологии (ПаВТ'2014): труды международной научной конференции. Челябинск: Издательский центр ЮУрГУ, 2014. С. 343–354.
4. Шамакина А.В. Алгоритм планирования ресурсов POS для распределенных проблемно-ориентированных сред // Научный сервис в сети Интернет: многообразие суперкомпьютерных миров: Труды Международной суперкомпьютерной конференции. М.: Изд-во МГУ, 2014. С. 26-31.
5. Shamakina A., Sokolinsky L. Resource Scheduling Algorithm in Distributed Problem-Oriented Environments // UNICORE Summit 2014 Proceedings. IAS Series, 2014. Vol. 26. P. 49-60.
6. Yang T., Gerasoulis A. DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors // IEEE Transactions on Parallel and Distributed Systems. 1994. Vol. 5, No. 9. P. 951-967.
7. Maheswaran M., Ali S. et al. Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems // Journal of Parallel and Distributed Computing. 1999. Vol. 59, No. 2. P. 107-131.