

ИССЛЕДОВАНИЕ РАБОТЫ МОДУЛЯ УПРАВЛЕНИЯ ЖЕЛЕЗНОДОРОЖНЫМИ РЕСУРСАМИ НА ОБЛАЧНОМ КЛАСТЕРЕ

П.С. Морозов

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет) (СГАУ), Самара, Россия

Рассматривается исследование работы модуля оптимального управления ресурсами на облачном, вычислительном кластере. Приводится краткое описание исследуемого модуля. В качестве вычислительного кластера используется облачная платформа Amazon Web Services от корпорации Amazon. Анализируются и поясняются полученные в ходе исследования результаты.

Ключевые слова: управление ресурсами, облачный кластер, алгоритм оптимизации, Amazon Web Services.

Введение

Благодаря бурному развитию цивилизации, многие задачи по управлению ресурсами переходят в разряд тех, где человеку становится весьма проблематично, а то и не возможно эффективно находить их решение.

Доставка грузов на международную космическую станцию, организация грузоперевозок на автомобильном транспорте, планирование пассажирских перевозок на железнодорожном транспорте и многие другие области сегодня остро ставят вопрос о сокращении влияния человеческого фактора за счет автоматизации процесса принятия решения и внедрения интеллектуальных систем, обеспечивающих быстрое и эффективное урегулирование в случае возникновения внештатных ситуаций.

В данной работе анализируется одна из вариаций решения класса задач по составлению оптимального плана, реализованного в виде программного модуля. Сам модуль легко модифицируется для работы с предметной областью железнодорожной тематики (управление движением локомотивами, управление назначением вагонов к составам и т.п.). Алгоритм работы модуля представляет собой модифицированный «жадный» алгоритм, который осуществляет процесс планирования с помощью мультиагентного взаимодействия планируемых объектов посредством обмена задачами в рамках аукциона с двойным подтверждением [1]. Алгоритм [2] снабжается эвристиками, что обеспечивает получение квазиоптимального решения, но обеспечивает хорошее быстродействие, а так же возможность использования ограничений в виде нелинейных функций, что часто требуется на практике, но невозможно при использовании, например, методов линейного программирования.

Используемый кластер

В качестве кластера используется облачная платформа от корпорации Amazon под названием Amazon Web Services (AWS). AWS предлагает огромный спектр решений для задач самого широкого круга. В нашем случае нас интересуют облачные вычисления, поэтому нас интересует продукт AWS Elastic Beanstalk, который позволяет запускать свои веб-

приложения на «облаке», а так же управлять ими. AWS Elastic Beanstalk позволят выбрать нужный по техническим характеристикам instance (узел), а так же осуществляет настраиваемое автомасштабирование, которое распределяет нагрузку на развернутое приложение по различным узлам. В качестве узлов будут использоваться процессоры семейства Intel® Xeon®.

Исследование модуля

Для проведения экспериментов будем использовать три набора данных с 327, 873, 2639 планируемых объектов. Так же для сравнения ускорения будем запускать модуль на 2, 4 и 8 CPU. Каждый эксперимент будет состоять из 5 серий и для анализа будет браться их среднее значение.

В таблице 1 приводятся результаты измерений скорости работы для набора данных из 327 планируемых объектов.

Табл. 1. Результаты измерений для 327 планируемых объектов

Кол-во CPU	Время выполнения, с					
	T1	T2	T3	T4	T5	\bar{T}
2	5,06	4,69	4,86	4,72	5,02	4,87
4	1,84	1,51	1,61	1,40	1,46	1,56
8	0,78	0,84	0,87	0,86	0,83	0,84

В таблице 2 приводятся результаты измерений скорости работы для набора данных из 873 планируемых объектов.

Табл. 2. Результаты измерений для 873 планируемых объектов

Кол-во CPU	Время выполнения, с					
	T1	T2	T3	T4	T5	\bar{T}
2	11,56	11,40	11,53	11,23	10,18	11,18
4	4,91	4,72	4,17	4,21	4,33	4,47
8	2,55	3,07	2,59	2,63	3,05	2,78

В таблице 3 приводятся результаты измерений скорости работы для набора данных из 2639 планируемых объектов.

Табл. 3. Результаты измерений для 2639 планируемых объектов

Кол-во CPU	Время выполнения, с					
	T1	T2	T3	T4	T5	\bar{T}
2	11,56	11,40	11,53	11,23	10,18	11,18
4	4,91	4,72	4,17	4,21	4,33	4,47
8	2,55	3,07	2,59	2,63	3,05	2,78

На рисунке 1 изображены результаты измерений для 327 планируемых объектов.

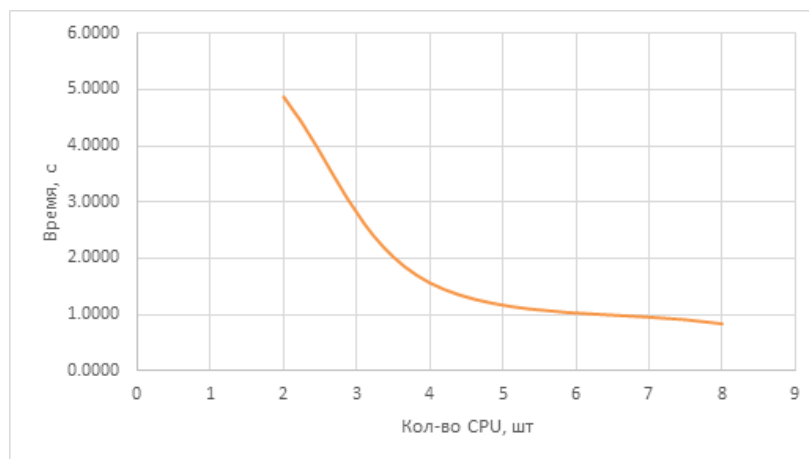


Рис. 1. Результаты измерений для 327 планируемых объектов

Для остальных наборов данных графики аналогичны.

На рисунке 2 изображены результаты измерений на 8 CPU.

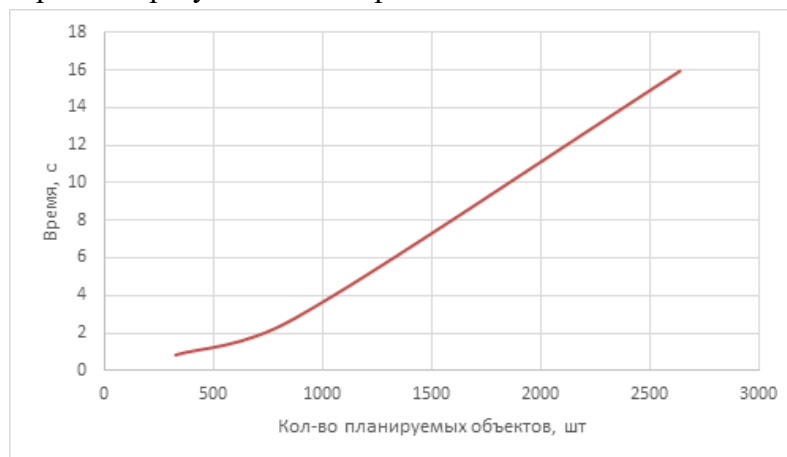


Рис. 2. Результаты измерений на 8 CPU

Для остальных конфигураций кластера графики выглядят аналогично.

Анализ полученных результатов

На рисунке 2 видно, что на время растет линейно в зависимости от количества планируемых объектов, комбинаторного взрыва не происходит, а значит алгоритм в исследуемом модуле является полиномиальным.

На рисунке 1 видна зависимость времени выполнения от количества CPU, при одинаковом наборе входных данных. Резкое увеличение скорости работы, при увеличении количества CPU с 2 до 4 можно объяснить тем, что при 2 CPU потоки, вычисляющие промежуточные этапы решения, скапливаются в большую очередь, ожидая своего процессорного времени. При этом выигрыш времени вычисления при увеличении размера входных данных, уменьшается (для 327 объектов ускорение примерно равно 3, для 873 объектов ускорение примерно равно 2,5, для 2639 – 2,4) по этой же причине – потоков становится больше, очереди увеличиваются.

Дальнейшее замедление ускорения с увеличением кол-ва CPU до 8, объясняется тем, что промежуточные результаты вычисления собирает мастер агент, т.е. все потоки обраща-

ются к одному центральному и при увеличении количества CPU увеличивается количество потоков, обращающихся к центральному потоку, поэтому образуется очередь из этих обращающихся, что увеличивает общее время выполнения. Из-за этого дальнейшее увеличение количества CPU не принесет значительной выгоды в временном ускорении.

Заключение

Постановка задачи для разработки модуля (железнодорожная тематика) имеет предельные значения на уровне пары десятков тысяч объектов для единовременного планирования, поэтому реализованный модуль будет эффективно справляться со своей предметной областью, не смотря на выявленные в ходе анализа экспериментальных данных особенности работы алгоритма в виде плохой масштабируемости.

Литература

1. Mobach, D.G.A. Train Driver Rescheduling using Task-Exchange Teams / Mobach D.G.A., Abbink E.J.W. , Fioole P.J., Lentink R.M., Kroon L.G., van der Heijden E.H.T., Wijngaards N.J.E. // Proceedings of the Second International Workshop on: Optimisation in Multi-Agent Systems (OPTMAS) at AAMAS 2009. – 2009. URL: <http://users.ecs.soton.ac.uk/sdr/optmas2/submissions/mobach.pdf>(дата обращения 28.03.2016).
2. Морозов, П.С. Решение задачи оптимального управления ресурсами на примере оперативного планирования локомотивов / Морозов П.С., Белоусов А.А. // Информационные технологии и нанотехнологии (ИТНТ-2015). [Электронный ресурс]: материалы Международной конференции и молодежной школы. - Электрон. текстовые и граф. дан. (4,2 Мбайт). - Самара, Изд-во СамНЦ РАН, 2015. - 1 эл. опт. диск (CD-ROM).