ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ № 3(17), 2015, с.??—??

УДК 004.382.2:004.451.42:.44:004.023

А.В. Ефимов, С.Н. Мамойленко, Е.Н. Перышкова

Обработка масштабируемых задач на вычислительных системах с помощью менеджера ресурсов PBS/Torque и планировщика Maui[1]

Аннотация. Рассмотрена проблема формирования расписаний решения масштабируемых задач на распределённых вычислительных системах (ВС). Предложены эвристики планирования решения масштабируемых задач на ВС. Произведена модернизация системы управления ресурсами (СУР) PBS/Torque и планировщика Maui. В программном обеспечении реализованы предложенные эвристики для обслуживания масштабируемых задач. Предложен способ описания паспорта масштабируемых задач. Исследованы показатели процесса планирования при решении наборов масштабируемых задач. Результаты исследований показали что, использование свойства масштабируемости задач позволяет уменьшить среднее время ожидания задач в очереди и суммарное время решения всех задач набора.

Ключевые слова и фразы: вычислительные системы, управление ресурсами, масштабируемые задачи

Введение

Распределённые вычислительные системы (ВС) относятся к перспективным средствами обработки информации и используются при решении сложных задач науки и техники [1].

В архитектурном плане ВС представляет собой композицию множества элементарных машин (ЭМ) и сети связей между ними. Структура ЭМ допускает варьирование от процессорного ядра до

[1] (Работа выполнена при поддержке Совета по грантам Президента Российской Федерации (проект МД-2620-2014.9) и Российского фонда фундаментальных исследований (гранты № 15-07-00048, 15-07-00653, 13-07-00160))

конфигураций, включающих универсальные процессоры и специализированные ускорители (например, GPGPU).

По статистике, 86.8 % ВС из списка top500 [7] являются кластерными. Кластерные ВС функционируют под управлением системного программного обеспечения, включающего, в том числе, систему управления ресурсами (СУР), например, PBS/Torque [2], Altair PBS Pro, SLURM, HTCondor и др.

Основным назначением ВС является обработка (решение) задач. Под задачей понимается требование выполнить параллельную программу на ресурсах ВС. Требование формируется пользователем в виде паспорта задачи, содержащим ресурсный запрос и параллельную программу.

Как правило, ВС функционируют в мультипрограммном режиме. При этом СУР формирует расписание решения задач и распределяет таким образом ресурсы ВС между несколькими одновременно решаемыми задачами.

Согласно классификации, предложенной в работе [6], задачи можно разделить на 4 типа (rigid, moldable, evolving и malleable). Rigid – обычные задачи с одним ресурсным запросом. Moldable – масштабируемые задачи обладающие набором ресурсных запросов, один из которых выбирается системой перед началом решения, при этом в процессе решения не изменяется. Evolving и malleable задачи допускают изменение, соответственно пользователем или системой, количества используемых ресурсов в процессе решения задачи.

Исследования показывают [3], что 98 % задач, решаемых на высокопроизводительных ВС являются масштабируемыми. При этом количество ресурсов для решения задачи масштабируется пользователем на этапе заполнения ресурсного запроса в паспорте задачи.

Повысить эффективности эксплуатации ресурсов ВС и снизить время нахождения задач в очереди СУР возможно за счёт разработки алгоритмического и программного инструментария позволяющего менеджеру ресурсов ВС учитывать свойство масштабируемости задач.

Обзор существующих СУР показывает, что большинство из них не учитывает свойства масштабируемости задач при управлении ресурсами. Коллективом авторов предложена модификация СУР PBS/Torque и планировщика Maui, реализующая базовые функции по обслуживанию масштабируемых задач.

1. Обзор предыдущих работ

Известно, что проблема формирования расписаний, в том числе и при управлении ресурсами ВС, является NP-полной [8, 9, 10, 11]. При планировании решения масштабируемых задач проблема осложняется необходимостью выбора одного из ресурсных запросов предоставленных пользователем. В работах [12, 13, 14] представлен широкий обзор публикаций предыдущих достижений, а так же предложены и промоделированы алгоритмы планирования решения масштабируемых задач на высокопроизводительных ВС.

К сожалению, применение на практике находят не все разрабатываемые алгоритмы планирования [15]. В работах [16, 17] представлены результаты сравнения алгоритмов планирования масштабируемых задач с наиболее популярным алгоритмом планирования (FCFS + Backfill). При этом предложенные алгоритмы реализуются в специализированных планировщиках, а при сравнении с планировщиком SLURM в работе [17] масштабируемые задачи используются только с одним вариантом ресурсного запроса.

В последнее время наблюдается повышение интереса к moldable и malleable задачам в связи с развитием концепции высокопроизводительных вычислений в облаке (high-performance computing as a service). Например в работе [18].

2. Описание масштабируемой задачи

Задача представляется пользователем в виде паспорта (скрипта) для оболочки (shell), содержащего требования к ресурсам, атрибуты задания и набор команд, которые необходимо выполнить.

Для представления в паспорте масштабируемой задачи необходимо использовать атрибут *–L список_ запросов* команды *qsub* или в скрипте задания.

Аргумент *список_ запросов* записывается в виде:

$$request_1[,request_2][,request_i][,request_n]$$

здесь $request_i$ – один из вариантов конфигурации ресурсов для решения масштабируемой задачи. Предусмотрены следующие параметры для описания конфигураций ресурсов каждого запроса:

$$nodes = value[@\,ppn = value][@\,weight = value][@\,walltime = value],$$

где *nodes* – количество узлов, *ppn* – количество процессорных ядер на каждом узле, *weight* – приоритет данного запроса относительно других запросов (натуральное число), *walltime* – максимальное время использования ресурсов запроса в формате *[[часы:]минуты:]секунды]*. Пример паспорта масштабируемой задачи представлен на Рис. 1.

```
#PBS -N moldable_job
#PBS -L nodes=2@ppn=8@weigth=10@walltime=00:25:00,nodes=5@ppn=4@weight=20@walltime=00:20:00
#PBS -j oe

cd $PBS_O_WORKDIR

mpiexec ./test
```

Рис. 1. Пример паспорта масштабируемой задачи

3. Модификация программного обеспечения

Модификации системы управления ресурсами PBS/Torque и планировщика Maui оформлены в виде патчей содержащих изменения относительно файлов соответствующих проектов.

3.1. PBS/Torque

В PBS/Torque реализована возможность запуска масштабируемых задач. Для активации режима обработки масштабируемых задач достаточно в паспорте задачи предлагается использовать новый атрибут «*–L*».

При постановке масштабируемой задачи в очередь, планировщик выбирает тот запрос $request_i$, который обладает максимальный

приоритетом (*weight*). Дальнейший алгоритм планирования службы *pbs_ sched* не изменен.

3.2. Maui

3.2.1. Настройка

Для активации режима обработки масштабируемых задач в планировщике *Maui* необходимо в конфигурационном файле *maui.cfg* указать опцию *EnableMoldableJobs TRUE*. В этом случае планировщик начинает воспринимать атрибут *«–L»* в паспорте задачи.

Добавлена новая политика рассмотрения масштабируемых задач на этапе поиска подходящей конфигурации ресурсов. Для этого в конфигурационный файл нужно добавить опцию *MoldableJobSchedulingPolicy Time, Rank, Weight* или *Worth*.

3.2.2. Политики планирования

Логика работы планировщика Maui, а также этапы и политики планирования сохранены. Реализован перебор вариантов запросов для масштабируемых задач на этапах планирования и обратного заполнения (по алгоритму backfill) в соответствии с выбранной политикой.

Политика *Time*. В данном случае запросы ресурсов для задачи рассматриваются в порядке увеличения времени решения (*walltime*).

Политика *Rank* практически соответствует политике ShortJobFirst, когда первой выбирается вариант с минимальным количеством процессорных ядер (*nodes * ppn*).

Политика *Weight* предполагает рассмотрение запросов в порядке уменьшения пользовательского приоритета (*weight*).

Политика *Worth* предполагает вычисление ценности запросов по формуле:

$$\frac{weight_i}{nodes_i * ppn_i * walltime_i}$$

после чего запросы рассматриваются в порядке уменьшения ценности.

Если в *maui.cfg* опция *EnableMoldableJobs* инициализирована значением *TRUE* или опция *MoldableJobSchedulingPolicy* не задана или инициализирована *NONE*, то запросы рассматриваются в порядке их перечисления в паспорте задачи.

4. Экспериментальное исследование

Тестирование модифицированного программного обеспечения проводилось на ресурсах пространственно-распределённой мультикластерной вычислительной системы созданной совместно Лабораторией вычислительных систем ИФП СО РАН и Центром параллельных вычислительных технологий ФГОБУ ВПО «СибГУТИ» [4].

Для проведения исследования использовался кластер из 12 вычислительных узлов на базе процессора Intel Core i3 по 4 процессорных ядра на каждом.

4.1. Описание процесса

Для поведения экспериментов набор задач генерировался на основе модели рабочей загрузки, предложенной в работе [5]. Программа генерирующая наборы задач создаёт их в виде XML-файла с уникальным идентификатором. В программе предусмотрена возможность указать долю масштабируемых задач в наборе. Рассматривались наборы из 50, 100, 200, 400 и 800 задач. Генератором задавались все необходимые параметры ресурсного запроса (*nodes*, *ppn*, *walltime и weight*).

Управление процессом запуска задач и сбором статистики осуществляется с помощью специально разработанного скрипта на языке ruby.

Скрипт из XML-файла с набором задач формирует соответствующие паспорта и ставит задачи в очередь. После того, как все задачи установлены в очередь скрипт начинает анализировать лог-файлы PBS/Torque для вычисления времени ожидания в очереди и времени решения задачи. Время решения всех задач набора определяется как время окончания решения последней задачи из набора.

Загрузка вычислительных ресурсов определяется по формуле:

$$U = \frac{\sum_{i=1}^{L} t_i * r_i}{T * N},$$

где T – время решения задач набора, N – количество процессорных ядер в системе (12*4 = 48 шт.), t_i – время решения i-ой задачи набора, r_i – количество процессорных ядер, выделенных задаче i, L – количество задач в наборе.

4.2. Результаты

Результаты экспериментов показывают, что использование масштабируемых задач в модифицированной СУР Torque и планировщике MAUI позволяет сократить среднее время решения задач набора в пределах 27 %, а среднее время ожидания задач в очереди уменьшить на 22 %. При этом количество задач в наборе должно не менее 100, а масштабируемыми из них должны быть не менее 50 %.

Лучшими из предложенных для планировщика Maui политик являются Rank и Worth в сочетании с алгоритмом BackFill.

Загрузка ресурсов системы при использовании модифицированного программного обеспечения составила примерно 77%, что соответствует загрузке ресурсов при использовании стандартного планировщика Maui с алгоритмом BackFill.

Заключение

Предложенные в работе политики позволили сократить суммарное время решения всех задач набора, а так же сократить среднее время ожидания задач в очереди.

Список литературы

[1] Хорошевский В.Г. Распределённые вычислительные системы с программируемой структурой// Вестник СибГУТИ. 2010. №2 (10). С. 3-41.

[2] Torque Resource Manager [Электронныйресурс]. Адрес доступа:

http://www.adaptivecomputing.com/products/opensource/torque (дата обращения 12.10.2015).

[3] Lifka D. The ANL/IBM SP scheduling system // Job Scheduling Strategies for Parallel Proc. LNCS. Springer-Verlag, 1995. Vol. 949. P. 295-303.

[4] Ресурсы центра параллельных вычислительных технологий ФГОБУ ВПО "СибГУТИ" [Электронныйресурс]. Адрес доступа: http://cpct.sibsutis.ru/index.php/Main/Resources (дата обращения 12.10.2015)

[5] Cirne W., Berman F. A model for moldable supercomputer jobs. 15th Intl. Parallel & Distributed Processing Symp. 2001.

[6] Feitelson, D.G. Toward convergence in job schedulers for parallel supercomputers / D.G.Feitelson, L. Rudolph // Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science. – 1996. – Vol. 1162. – P. 1-26.

[7] Top500 supercomputing sites [Электронный ресурс]. Адрес доступа: http://www.top500.org/lists/2015/06/ (дата обращения 12.10.2015).

[8] Конвей, Р. В. Теория расписаний / Р. В. Конвей, В. Л. Максвелл, Л. В. Миллер. – М.: Наука, 1975. – 360 с.

[9] Гери, М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон ; пер. с англ. – М. : Мир, 1982. – 416 с.

[10] Коффман, Э. Г. Теория расписаний и вычислительные машины / Л. Дж. Бруно, Р. Л. Грэхем, В. Г. Коглер [и др.] ; под ред. Б.А. Головкина, пер. с англ. В.М. Амочкина, М.: Изд-во «Наука», 1984. – 336 С.

[11] D. Bernstein, M. Rodeh and I. Gertner, “On the Complexity of Scheduling Problems for Parallel/Pipelined Machines“, IEEE Transactions on Computers, 1998, vol. 38, pp. 1308

[12] G. Sabin, M. Lang, and P. Sadayappan Moldable Parallel Job Scheduling Using Job Efficiency_An iterative approach

[13] R. Khandekary, B. Schieber, H. Shachnaix, T. Tamir Real-time Scheduling to Minimize Machine Busy Times

[14] K-C. Huang, W. Hsieh, C-H. Hung Online Scheduling of Moldable Jobs with Deadline

[15] I. Grudenić Scheduling Algorithms and Support Tools for Parallel Systems

[16] G. Utrera, J. Corbalán, J. Labarta Another approach to backfilled jobs applying Virtual Malleability to expired windows

[17] O. Sarood, A. Langer, A. Gupta, L. Kale Maximizing Throughput of Overprovisioned HPC Data Centers Under a Strict Power Budget

[18] K-C. Huang, T-C. Huang, Y-H Tung Moldable Job Scheduling for HPC as a Service with Application Speedup Model and Execution Time Information

Об авторах:

Александр Владимирович Ефимов к.т.н.

Сергей Николаевич Мамойленко д.т.н доцент

Евгения Николаевна Перышкова

А.В. Ефимов,
С.Н. Мамойленко,
Е.Н. перышкова
Центр параллельных вычислительных технологий Федерального государственного бюджетного образовательного учреждения высшего образования «Сибирский государственный университет телекоммуникаций и информатики»
Лаборатория вычислительных систем Федерального государственного бюджетного учреждения науки
Институт физики полупроводников им. А.В. Ржанова Сибирского отделения Российской академии наук
e-mail:efimov,e_perishkova,sergey@csc.sibsutis.ru

Образец ссылки на публикацию:

А.В. Ефимов, С.Н. Мамойленко, Е.Н. Перышкова. Обработка масштабируемых задач на вычислительных системах с помощью менеджера ресурсов PBS/Torque и планировщика Maui // Программные системы: теория и приложения: электрон. научн. журн. 2013. Т. 4, № 3(17), с. ??–??.

URL: http://psta.psiras.ru/read/???

A.V. Efimov, S.N. Mamojlenko, E.N. Peryshkova. Moldable jobs servicing on high-performance computing systems with resource manager PBS/Torque and Maui scheduler.

Abstract. This study contains a review of the moldable jobs scheduling problem on dis-tributed high-performance computing system sand proposes heuristic solutions to this problem. We have also modernised the resource manager PBS/Torque as well as the Maui scheduler. The modified software is capable of implementing the proposed heuristical solutions to moldable jobs scheduling problem. The paper further pro-poses a method for describing moldable jobs scripts for pbs/torque. Further we study the queue waiting time and makespan during the servicing sets of moldable jobs. The results of the study show that via the use of moldable jobs allows us to reduce the average job queue waiting time and makespan.

Key Words and Phrases: high-performance computing, resource management, moldable jobs.