УДК 004.451.42

ИНТЕГРАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ ИНТЕГРИРОВАННОЙ СРЕДОЙ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ МЕТАКЛАСТЕР С ПОДСИСТЕМОЙ ПЛАНИРОВАНИЯ MAUI

В.П. Гергель, В.Д. Кустикова, А.В. Сенин

Нижегородский госуниверситет им. Н.И. Лобачевского

SeninAndrew@gmail.com

Поступила в редакцию 27.01.2011

Рассмотрен вопрос интеграции системы управления высокопроизводительными вычислениями Метакластер с подсистемой планирования Maui. Приведен краткий обзор наиболее широко используемых алгоритмов планирования на кластерных системах, описаны основные модификации алгоритма обратного заполнения. Описана схема интеграции с Maui. Приведено описание тестовой системы, обзор критериев эффективности алгоритмов планирования, сравнение эффективности собственного алгоритма Метакластера и алгоритмов, реализованных в Maui.

Ключевые слова: Метакластер, система управления кластером, планирование параллельных задач, алгоритм обратного заполнения, подсистема планирования Maui.

Введение

Высокопроизводительные системы кластерного типа в настоящее время стали незаменимым инструментом для решения широкого круга вычислительно трудоемких задач науки, промышленности, сферы развлечений и других сфер. Статистика рейтингов производительности суперкомпьютеров показывает постоянный рост общего числа узлов и ядер [1]. С увеличением количества вычислительных элементов возрастает и сложность управления такими установками. Поэтому большое значение приобретает правильный выбор управляющего программного обеспечения, способного гарантировать быстрое выполнение пользовательских заданий. Центральным компонентом в системе управления кластером является подсистема планирования, которая отвечает за распределение задач по узлам многопроцессорной системы и определение времени их запуска.

В данной работе рассматриваются два подхода к решению задачи планирования, реализованные в системе управления кластерами Метакластер. Приводится описание встроенного алгоритма планирования и этапов реализации альтернативного подхода, который предполагает интеграцию с компонентой планирования Maui. Проводится сравнительный анализ эффективности собственной реализации алгоритма со стратегиями, которые поддерживаются разработчиками Maui, на основании наиболее распространенных критериев эффективности.

1. Система управления кластерами Метакластер

Метакластер – система управления высокопроизводительными вычислениями, разрабатываемая в ННГУ [2]. К основным особенностям системы относятся возможность управления несколькими кластерами, поддержка операционных систем семейств Windows и Linux, возможность интеграции с системами управления сторонних разработчиков. В процессе работы над проектом большое внимание уделяется вопросам планирования, так как это ключевой фактор в достижении высоких показателей эффективности использования вычислительной установки. Исследования ведутся в двух направлениях: адаптация существующих алгоритмов планирования и разработка новых стратегий. Более подробное описание архитектуры Метакластера приводится в [2].

Планирование в системе управления Метакластер осуществляется на двух уровнях: сначала выбирается наиболее подходящий для исполнения задачи кластер, затем выполняется распределение задачи по вычислительным узлам. При выборе кластера учитываются требования, указанные пользователем (технические характеристики ресурсов, операционная система, необходимые программы и пакеты). Если указанным требованиям соответствуют несколько кластеров, то выбирается наименее загруженный вычислительный ресурс.

В качестве алгоритмов планирования на кластере используются стратегия FCFS (запуск задач осуществляется строго в порядке поступления на кластер) и простейшая модификация алгоритма обратного заполнения (backfilling). Поддержка более сложных модификаций обеспечивается за счет интеграции с планировщиком Maui. В настоящее время также ведутся исследования по использованию переборных техник выбора оптимального распределения задач (см., например, [3]).

2. Обзор алгоритмов планирования

Планирование распределения параллельных заданий по узлам высокопроизводительной системы в общем случае – NP-трудная задача [4], что делает невозможным нахождение оптимального расписания за разумное время на большинстве реальных установок. На практике для планирования вычислений используются те или иные эвристики, предоставляющие решение, близкое к оптимальному. В основном используются следующие подходы для получения приближенного решения задачи планирования:

- Списочные алгоритмы. В данных алгоритмах задачи, подлежащие планированию, выстраиваются в очередь согласно некоторому правилу. В качестве такого правила может выступать, например, взвешенная свертка параметров задачи: приоритеты пользователя и задачи, время нахождения в очереди, время работы программы и др. После составления такого списка задачи ставятся на выполнение строго в данном порядке по наличию необходимых свободных ресурсов. Самым популярным вариантом списочного алгоритма является FCFS (First Come First Served – упорядочивание задач в порядке поступления),
- Алгоритм обратного заполнения (backfilling). Алгоритм является модификацией списочных алгоритмов, применимой при наличии информации о предполагаемом времени работы задач. В этом случае порядок запуска может частично нарушаться за счет более раннего выполнения небольших менее приоритетных задач, если они не замедлят старт более приоритетных ресурсоемких задач. Данный алгоритм является наиболее популярным в настоящее время,
- Переборные эвристики. В том случае, когда известна целевая функция, которую оптимизирует алгоритм планирования, задачу планирования можно свести к стандартной задаче поиска минимума функции, которая может решаться с применением широкого круга различных эвристик: генетические алгоритмы [5], алгоритм имитации отжига [6], переборные алгоритмы с ограничением числа перебираемых точек [3].

В том случае, когда задачи допускают приостановку выполнения (preemption), могут применяться специальные алгоритмы: алгоритм управления группами заданий с прерываниями (gang scheduling [7]), алгоритм, основанный на множестве очередей (feedback [8]).

В табл. 1 приведены алгоритмы и техники планирования, реализованные в Метакластере, в некоторых наиболее известных системах управления кластерами, а также в автономном планировщике Maui. Необходимо отметить, что в большинстве систем поддерживается в качестве базовой стратегии распределения заданий алгоритм FCFS, при этом дополнительно реализуется алгоритм обратного заполнения в простейшей модификации FIRSTFIT, которая будет подробнее описана ниже.

Таблица 1

Алгоритмы и техники планирования, реализованные в некоторых компонентах планирования распределения заданий

	Алгоритмы планирования			**Особенности**	
	Обратное заполнение	**Переборные эвристики**	**Алгоритмы с приостановкой**	**Резервирование**	**Несколько кластеров**
MS HPC Server 2008	+	–	–	–	–
LoadLeveler	+	–	+	+	+
Platform LSF	+	–	+	+	+
PBS-Pro	+	–	+	+	–
Метакластер	+	+[1]	–	–	+
Maui	+	–	+	+	–

[1] Переборные эвристики в системе управления Метакластер используются, в основном, для исследовательских целей.

3. Описание алгоритма обратного заполнения

Процедура обратного заполнения фактически представляет собой оптимизированный процесс планирования распределения задач, позволяющий эффективно использовать имеющиеся вычислительные ресурсы. Каждая итерация алгоритма обратного заполнения состоит из трех этапов:

1. Распределение задач по принципу FCFS (First Come First Served) (рис. 1а). Отказ от этой стратегии происходит в момент, когда для исполнения следующей задачи недостаточно ресурсов.
2. Резервирование ресурсов для нескольких наиболее приоритетных задач (рис. 1б).
3. Заполнение областей простаивания ресурсов («окон», рис. 2) – определение задачи или набора задач, которые могут выполняться, не вызывая задержку запуска задач, для которых выполнено резервирование. Правило выбора задач (на рис. 2а и рис. 2б показаны примеры 2-х вариантов выбора) определяет модификацию алгоритма обратного заполнения.

4. Интеграция планировщика сторонних разработчиков

Интеграция планировщика сторонних разработчиков – это один из возможных подходов к решению задачи планирования в системе управления кластером. Данный подход не требует непосредственной реализации алгоритма распределения заданий по узлам многопроцессорной системы. Интеграция предполагает обеспечение корректного взаимодействия с внешней компонентой планирования.

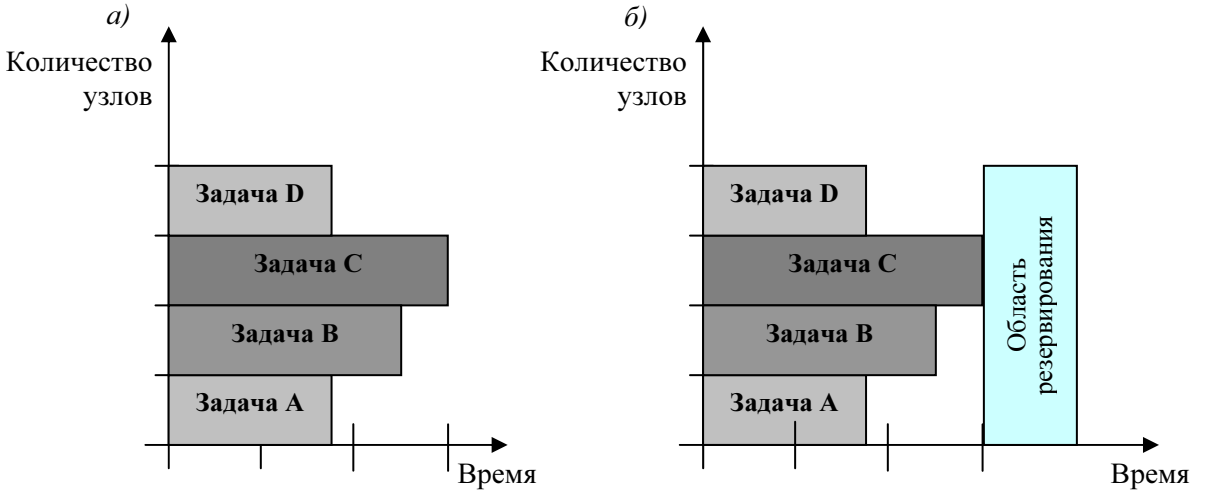

Рис. 1. Пример работы алгоритма обратного заполнения (а – распределение задач по принципу FCFS, б – резервирование ресурсов)

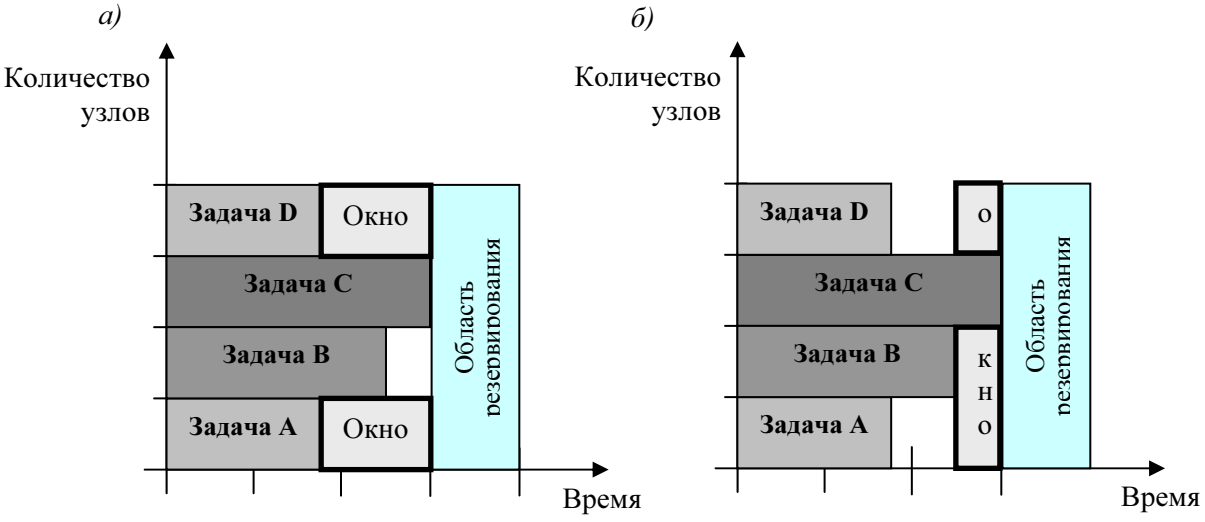

Рис. 2. Определение областей простаивания ресурсов

Типичная схема взаимодействия системы управления кластерами с планировщиком сторонних разработчиков показана на рис. 3 и состоит из нескольких этапов:

- планировщик с некоторой периодичностью (интервал опроса, как правило, устанавливается в конфигурационном файле компонента планирования) отправляет запросы на получение информации, которая требуется для выполнения итерации планирования,
- внутренний компонент системы управления обрабатывает принятые запросы и формирует ответные сообщения, передаваемые планировщику,
- планировщик на основании полученной информации определяет узлы для запуска заданий и оповещает систему управления о необходимости активации процесса задачи, отправляя соответствующее сообщение.

Таким образом, разработчики системы управления должны предоставлять следующую функциональность:

- формирование списка пользовательских задач,

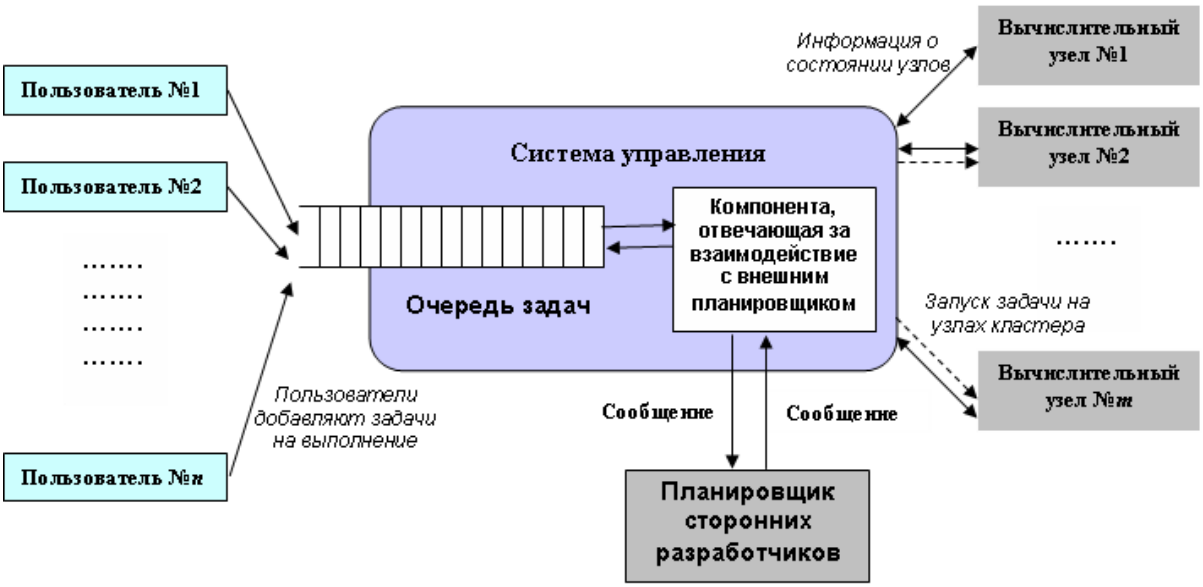

Рис. 3. Общая схема взаимодействия системы управления кластером с внешней компонентой планирования

- формирование списка разрешенных вычислительных ресурсов, а также предоставление информации о загруженности узлов, объеме оперативной памяти, размере свободного дискового пространства и проч.,
- организация обработки запросов внешнего планировщика и предоставление ответных сообщений в требуемом формате.

При этом на стороне внешнего планировщика осуществляется:

- получение информации об изменении состояния очереди задач и множества доступных вычислительных узлов,
- выполнение итерации планирования, в процессе которой определяется порядок запуска заданий и множество ресурсов для запуска,
- отправление запроса о необходимости запуска задачи на определенном наборе узлов.

Для реализации рассмотренной схемы в системе управления необходимо создать внутренний компонент, отвечающий за взаимодействие с планировщиком сторонних разработчиков.

5. Планировщик Maui

Планировщик Maui – компонент планирования распределения заданий по узлам многопроцессорной системы, разрабатываемый в Maui High Performance Center. Данный продукт является свободно распространяемым, исходные коды можно загрузить с официального сайта проекта ([9]). Maui разрабатывается как самостоятельный компонент планирования, который встраивается в наиболее известные системы управления, функционирующие на UNIX-платформах (в частности, Torque).

Характерные особенности планировщика Maui:

- в состав планировщика входит подсистема автоматического определения приоритета заданий, который формируется как линейная свертка нескольких факторов (права пользователя, количество запрашиваемых ресурсов и другая служебная информация о задании, в частности, время пребывания задания в очереди),

• в качестве базовой стратегии планирования реализован алгоритм обратного заполнения,

• поддерживаются механизмы заблаговременного резервирования ресурсов (advanced reservation) и оценки времени запуска заданий,

• наличие командного интерфейса администратора и библиотеки внешних интерфейсов взаимодействия с некоторыми системами управления кластерами.

Основное достоинство рассматриваемого компонента планирования перед аналогами состоит в том, что в ней реализовано несколько модификаций алгоритма обратного заполнения, которые позволяют получить эффективное решение задачи планирования на различных наборах заданий. Используемая модификация определяется в конфигурационном файле планировщика (параметр BACKFILLPOLICY):

• FIRSTFIT предполагает, что в окно простаивания ресурсов попадает задание с наибольшим приоритетом,

• BESTFIT состоит из нескольких этапов. Сначала определяется набор задач, которые потенциально могут заполнить окно простаивания ресурсов. Затем область простаивания заполняется тем заданием из полученного множества, которое наиболее эффективно использует ресурсы. В качестве критерия эффективности (параметр BACKFILLMETRIC) может быть выбрано количество занимаемых процессоров, либо время, в течение которого используются необходимые процессоры, либо сочетание предшествующих критериев,

• модификация GREEDY предполагает, что в окно простаивания ресурсов попадает набор заданий, который среди всех возможных комбинаций заполнения обеспечивает максимально эффективное использование ресурсов (критерий эффективности может быть выбран аналогично модификации BESTFIT).

Для выполнения итерации планирования Maui необходима информация о доступных вычислительных ресурсах и заданиях, поступающих на исполнение на кластер. Получение таких данных обеспечивается посредством взаимодействия с системой управления. Разработчики планировщика Maui предоставляют широкий спектр средств коммуникации с различными системами управления кластерами. Множество всех форматов взаимодействия условно можно разделить на две группы:

• «закрытые» интерфейсы – интерфейсы взаимодействия с такими системами управления, как PBS, LoadLeveler, Sun Grid Engine. Необходимые для коммуникации библиотеки, как правило, входят в состав стандартного пакета поставки системы управления,

• «открытые» интерфейсы: SSS-интерфейс обмена сообщениями в формате xml и WIKI-интерфейс, предполагающий передачу строк фиксированного формата. Данная группа интерфейсов предполагает возможность их использования в процессе интеграции компонента планирования в системы управления, для которых не реализованы стандартные библиотеки взаимодействия.

6. Описание взаимодействия с использованием WIKI-интерфейса

Для организации взаимодействия при интеграции планировщика Maui в систему Метакластер было принято решение использовать WIKI-интерфейс, т.к. SSS-интерфейс находится на стадии разработки.

В рамках WIKI-интерфейса коммуникация с системой управления предполагает передачу сообщений следующего формата:

<SIZE><CHAR>CK=<CKSUM><WS>TS=
=<TS><WS>AUTH=<AUTH><WS>DT=
=<DATA>,

где <SIZE> – размер передаваемого сообщения; <CHAR> – любой разделительный символ, <CKSUM> – контрольная сумма; <TS> – момент времени, когда было передано сообщение; <AUTH> – идентификатор пользователя, от имени которого отправлено сообщение; <DATA> – передаваемое сообщение (информативная часть – конкретный запрос планировщика или ответное сообщение системы управления фиксированного формата, содержащие определенный набор обязательных параметров), <WS> – пробельный символ (или символ табуляции).

Перед выполнением итерации планирования Maui отправляет два стандартных запроса GETNODES и GETJOBS для получения информации о доступных вычислительных узлах и состоянии очереди задач, соответственно. В момент принятия решения о запуске задачи планировщик отправляет запрос STARTJOB, где указывается набор узлов, на которых необходимо выполнить активацию процесса задачи.

Наряду с приведенными группами команд Maui может отправлять и другие типы запросов (CANCELJOB – заблаговременная остановка задачи, SUSPENDJOB – временная остановка задачи, RESUMEJOB – возобновление работы задачи). Обработка этих запросов происходит в соответствии с возможностями системы управ-

ления, в частности, приостановка и возобновление работы задачи в системе Метакластер не предусмотрена.

7. Интеграция Maui с системой управления Метакластер

Решение задачи интеграции Maui с системой управления кластерами Метакластер включает несколько этапов (рис. 4).

Чтобы интегрировать Maui и Метакластер согласно предложенной схеме (рис. 3), сначала необходимо скомпилировать и сконфигурировать планировщик (в ОС Windows использовался cygwin). Затем реализовать прототип компонента, отвечающего за взаимодействие с Maui в соответствии с принятым протоколом. Для обеспечения безопасности коммуникации необходимо использовать модуль подсчёта контрольных сумм, содержащий алгоритм, реализованный в Maui. Для контроля правильности совместного функционирования можно использовать log-файлы или командный интерфейс планировщика.

8. Оценка эффективности алгоритмов планирования

Перед непосредственным использованием алгоритмов планирования необходимо оценить эффективность поддерживаемых стратегий, чтобы в зависимости от загрузки кластера выбрать оптимальную стратегию распределения ресурсов. Решение данной задачи включает выбор критериев оценки эффективности, подготовку тестовых наборов заданий и проведение экспериментов.

Решение задачи планирования предполагает поиск оптимального расписания запуска заданий. При этом под оптимальностью понимается минимизация или максимизация некоторого целевого критерия эффективности. Существуют различные метрики, усредненные значения которых используются для оценки эффективности компонент планирования. В данной работе мы будем использовать следующие критерии:

- время ожидания запуска (waiting time),
- время замедления (slowdown) – отношение времени отклика (сумма времени ожидания и времени работы задачи) ко времени работы задачи $\frac{T_w + T_r}{T_r}$, где T_w – времени ожидания запуска и T_r – время работы задачи.

Первый критерий является наиболее наглядным с точки зрения пользователя, т.к. показывает, сколько в среднем придется ждать пользователю момента запуска своего приложения. Второй критерий характеризует вклад времени ожидания старта задачи в суммарное время ожидания получения результатов работы задачи.

Один из возможных подходов к получению репрезентативной рабочей нагрузки кластера

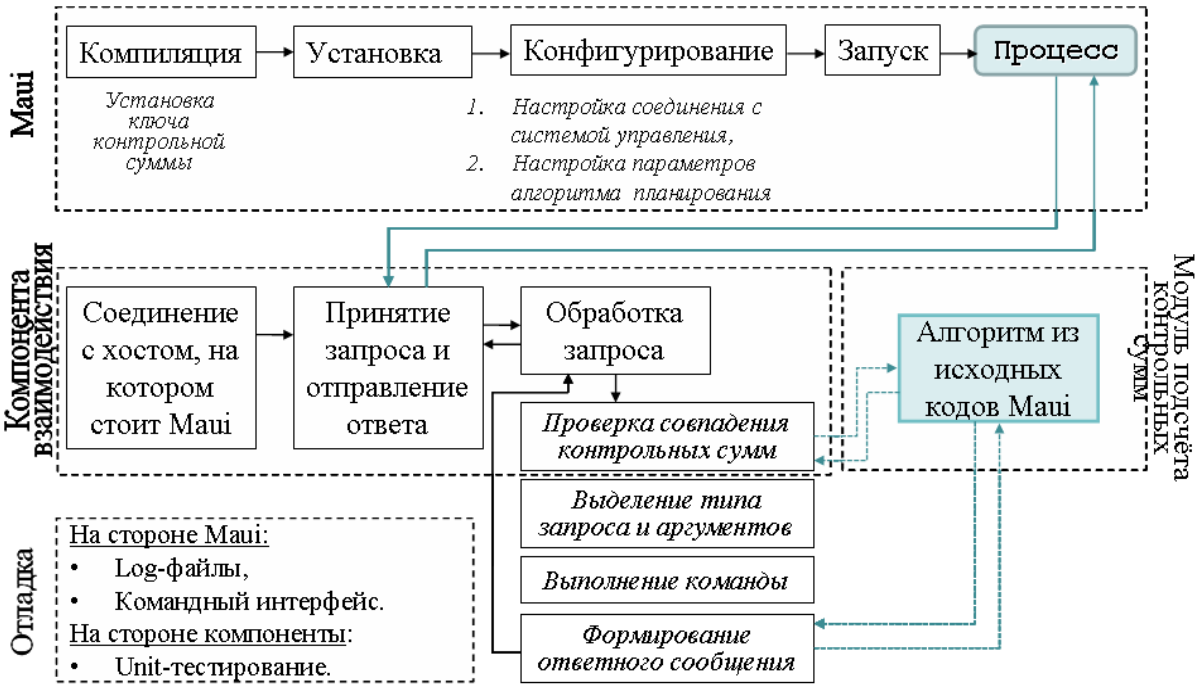

Рис. 4. Этапы решения задачи интеграции планировщика Maui в систему управления Метакластер

состоит в использовании архивов загрузки реальных систем, например, из Parallel Workloads Archive [10]. Набор задач (трасса загрузки) представляется файлом в формате SWF (Standard Workload Format), каждая строка которого содержит информацию о задании, запущенном на кластере. В каждом файле трассы приведена информация о ресурсах (общее количество узлов и ядер).

Другой подход предполагает построение моделей рабочей нагрузки с помощью статистических методов (см. подробнее, например, в [11]).

При выборе тестовых наборов задач для оценки эффективности алгоритмов планирования большинство авторов руководствуется первым подходом, поэтому для проведения экспериментов было принято решение использовать некоторые трассы из Parallel Workloads Archive.

9. Тестовая инфраструктура

В настоящий момент эмуляция является основным инструментом для оценки эффективности стратегий планирования, т.к. не требует значительных вычислительных и временных затрат. Проведение однократного эксперимента с некоторым тестовым набором задач, для которого предположительное время завершения всего запланированного множества составляет около года, занимает всего несколько часов.

Для получения оценок эффективности алгоритма планирования, поддерживаемого Maui, удобно использовать встроенный режим эмуляции [9]. Для этого необходимо конвертировать файлы трасс загрузки и информации о ресурсах в формат, обрабатываемый планировщиком (использовалось вспомогательное приложение). Режим эмуляции предполагает изменение конфигурации компонента планирования – установку параметров, определяющих имена файлов трасс (SIMRESOURCETRACEFILE и SIMWORKLOADTRACEFILE) и коэффициент ускорения времени счета задач (SIMTIMERATIO). Если установить этот коэффициент равным 10, то задание, которое считалось, например, 10 часов в реальном времени, в режиме эмуляции будет выполнено за 1 час. По окончании планирования Maui формирует файл статистики, в котором содержится список задач и информация о реальном времени попадания этих задач в очередь, реальном времени их запуска и завершения. На основании данных статистики инфраструктурное приложение вычисляет интересующие метрики.

Для оценки эффективности встроенного алгоритма планирования системы Метакластер использовался собственный имитатор, поддерживающий входные файлы в формате SWF. Имитатор подменяет объекты, используемые планировщиком, создавая у последнего иллюзию работы на вычислительной системе, описанной в файле трассы. Кроме того, имитатор переводит системное время (за счет специфической реализации интерфейса, используемой планировщиком), заставляя планировщик проводить перепланирование только тогда, когда изменилось состояние системы, т.е. добавилась новая задача или закончила выполняться активная. Таким образом, время работы имитатора совпадает со временем работы планировщика. На выходе имитатор вычисляет и выводит на экран все интересующие метрики.

10. Сравнение эффективности стратегий планирования

Для анализа эффективности стратегий планирования, поддерживаемых разработчиками Maui [12] и реализованных в Метакластере, использовались пять наборов задач из PWA, которые содержат различное количество задач и соотношение числа коротких (время работы менее 1000 секунд) и длительных задач (табл. 2). На остальных наборах сохраняются общие тенденции, справедливые для рассматриваемых наборов.

Таблица 2

Некоторые качественные характеристики тестовых наборов задач

Название набора	Среднее время работы задач (сек.)	Общее количество задач	Количество небольших задач (менее 1000 сек.)	Процент небольших задач (%)
CTC-SP2-1995-1	9778.27	70918	40403	57
NASA-iPSC-1993-2.1-cln	764.9	18239	15936	87.4
SDSC-DS-2004-1	6696.23	96089	58470	60.8
DAS2-fs3-2003-1	716.07	66112	64395	97.4
DAS2-fs4-2003-1	2819.097	32953	24797	75.2

Сравнение производилось по критериям среднее время ожидания (рис. 5) и среднее время замедления (рис. 6). По первому критерию все 3 алгоритма демонстрируют близкие результаты: различия менее 13%. По второму критерию различия более существенны: Метакластер заметно проигрывает Maui на 2х последних наборах. Характерная особенность этих 2х наборов – наличие большого количества недлительных задач (более 70% задач, см. табл. 2). Третий тест, на котором Метакластер демонстрирует лучший результат по среднему времени замедления, напротив, содержит больше крупных заданий. Различия объясняются тем, что в Метакластере и Maui по-разному строятся приоритеты задач (в обоих случаях использовались параметры по умолчанию).

Таким образом, алгоритм Метакластера эффективнее использовать, когда значителен поток длительных задач, а Maui, напротив, показывает лучшие результаты при большом проценте задач коротких.

Интересным также является тот факт, что модификация BESTFIT алгоритма обратного заполнения, реализованная в Maui, показала результаты хуже, чем FIRSTFIT. Данный результат можно объяснить тем, что метод BESTFIT требует более тонкой настройки под конкретный поток задач. В частности, в настройках системы необходимо выбрать параметр BACKFILLMETRIC, определяющий наилучших кандидатов для обратного заполнения методом BESTFIT. В данной работе использовалось значение PROCSECONDS.

Результаты сравнения также показывают, что правильный выбор алгоритма планирования, а также целевого критерия, соответствующих характеру использования данной вычислительной установки, может обеспечить существенный выигрыш (например, на наборе DAS2-fs3-2003-1 разрыв между лучшим и худшим результатами по критерию среднее время замедления составил более 3.5 раз).

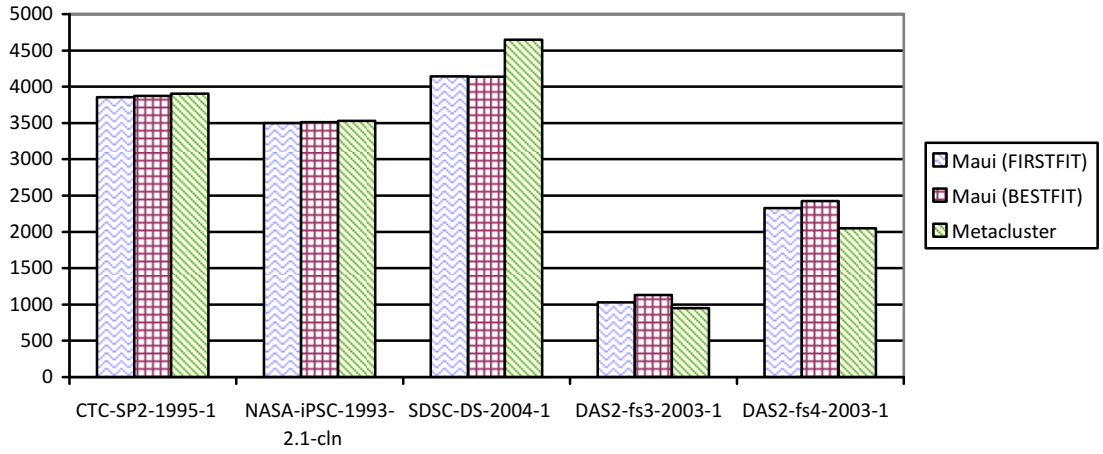

Рис. 5. Среднее время ожидания (в секундах) для некоторых наборов задач PWA

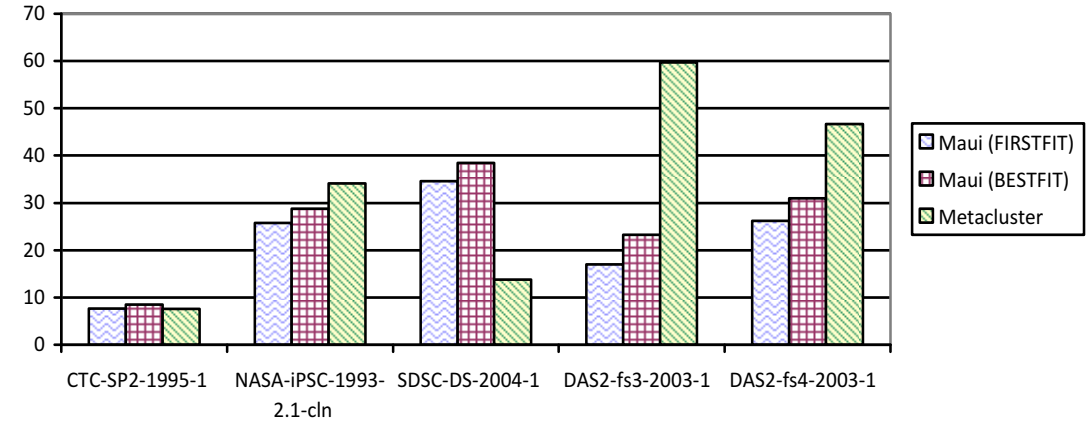

Рис. 6. Среднее время замедления для некоторых наборов задач PWA

Заключение

Интеграция системы управления вычислительной инфраструктурой Метакластер и подсистемы планирования Maui существенно расширяет возможности администратора по настройке алгоритма планирования под конкретную вычислительную установку, что позволяет значительно повысить эффективность ее использования. Результаты вычислительной симуляции показали потенциальный положительный эффект от использования Maui с системой Метакластер. В настоящее время в Центре суперкомпьютерного моделирования ННГУ проводится эксперимент по использованию планировщика Maui на вычислительном кластере ННГУ под управлением Метакластера.

Исследования выполнены в рамках НИР по теме «Экзафлопсный инструментарий симуляции процессов роста и физических свойств кремниевых наноструктур для современной наноиндустрии».

Список литературы

1. TOP500 Supercomputing Sites [Электронный ресурс]. URL: http://www.top500.org (дата обращения: 12.09.2010).

2. Гергель В.П., Сенин А.В. Разработка системы управления интегрированной средой высокопроизводительных вычислений Метакластер // Вестник ННГУ (принято к печати).

3. Vasupongayya S., Hui S., Massey B. Search-based Job Scheduling for Parallel Computer Workloads // Cluster 2005 Conference, Boston, MA, USA, September 26–30, 2005.

4. Garey M.R., Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness // NY: W.H. Freeman & Co, 1979.

5. Holland J. H. Adaptation in Natural and Artificial Systems // The MIT Press, 1992.

6. Kirkpatrick S., Gelatt C., Vecchi M. Optimization by Simulated Annealing // Science. 1983. V. 220. № 4598. P. 671–680.

7. Corbalan J., Martorell X., Labarta J. Improving Gang scheduling through job performance analysis and malleability // Proc. 15th international conference on Supercomputing. Sorrento, Italy, 2001. P. 303–311.

8. Sename O., Simon D., Robert D. Feedback scheduling for real-time control of systems with communication delays // Proc. ETFA'03 9th IEEE International Conference on Emerging Technologies and Factory Automation, Lisbonne, 16–19 Sept. 2003. V. 2. P. 454–461.

9. Описание режима эмуляции планировщика Maui [Электронный ресурс]. URL: http://www. clusterresources.com/products/maui/docs/16.0simulations.shtml (дата обращения: 12.09.2010).

10. Parallel Workloads Archive [Электронный ресурс]. URL: http://www.cs.huji.ac.il/labs/parallel/ workload (дата обращения: 12.09.2010).

11. Lublin U., Feitelson D. The workload on parallel supercomputers: modeling the characteristics of rigid jobs // School of Computer Science and Engineering. 2001. P. 1105–1122.

12. Jackson D., Snell Q., Clement M. Core Algorithms of the Maui Scheduler // Lecture Notes in Computer Science Job Scheduling Strategies for Parallel Processing, 7th International Workshop, JSSPP 2001, Cambridge, MA, USA, 2001. V. 2221. P. 87–102.

INTEGRATING THE HIGH PERFORMANCE COMPUTING ENVIRONMENT MANAGEMENT SYSTEM METACLUSTER WITH MAUI SCHEDULER

V.P. Gergel, V.D. Kustikova, A.V. Senin

Integrating the high performance computing environment management system Metacluster with Maui Scheduler is considered. A short survey of the most widely used cluster scheduling algorithms and the main modifications of the backfill algorithms are given. A Metacluster-Maui Scheduler integration scheme is described together with the test system infrastructure. An overview is presented of the performance criteria for scheduling algorithms, as well as the performance comparison of the Metacluster own scheduling algorithm with those implemented in Maui Scheduler.

Keywords: Metacluster, cluster management system, parallel task scheduling, backfilling algorithm, Maui Scheduler.