УДК 681.326

В.Г.КОТУХ, канд. техн. наук, М.А. МИРОШНИК, канд. техн. наук,
С.Н. СЕЛЕВКО, канд. техн. наук

МЕТОДЫ ПЛАНИРОВАНИЯ РЕСУРСОВ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ

Введение

Анализ современного состояния и тенденций развития компьютерных систем (КС) с использованием новых информационных и субмикронных технологий определяет новые требования к производительности и отказоустойчивости мультипроцессорных систем.

С точки зрения отказоустойчивости, функционирование КС представляет собой последовательность чередующихся интервалов работоспособного состояния и восстановления работоспособности путем реконфигурации структуры. Основанием процедуры реконфигурации КС и восстановления работоспособности является идентификация состояния системы и обнаружение неисправных модулей в процессе профилактического обслуживания и тестового диагностирования внешними или встроенными средствами диагностирования. С ростом сложности КС возрастает время и стоимость процедуры тестового диагностирования, что является причиной уменьшения производительности всей системы и эффективности её использования. Идеальным решением этой проблемы является использование методов функционального (онлайнового) диагностирования КС, когда в качестве тестов используются рабочие воздействия, поступающие в процессе функционирования системы. При этом исключаются или сводятся к минимуму интервалы времени профилактических тестовых проверок работоспособности системы.

Использование в составе КС процессорных модулей с интеллектуальными свойствами (IP-modules) определяет необходимость построения внутренней инфраструктуры КС, управляющей процессом оптимизации производительности и функциональной эффективности КС, а также онлайновым процессом тестового диагностирования, локализации неисправных модулей и реконфигурации её структуры. В ряде исследований показано, что потребление энергии в режиме тестового диагностирования в полтора раза превышает энергию, потребляемую в процессе нормального диагностирования, и этот разрыв имеет тенденцию увеличиваться с ростом сложности компьютерных сетей.

Для построения диагностической инфраструктуры с интеллектуальными свойствами (ДИ-ИС) можно использовать диагностические процессоры (модули), которые осуществляют онлайновое тестовое диагностирование КС. Однако при этом необходимо управлять процессом переключения процессорных модулей КС с режима функционирования в режим тестового диагностирования с учетом ограничений на допустимое время выполнения функциональных программ управления и потребляемой энергии на выполнение тестового диагностирования.

Известно, что для многих технологических процессов характерны непрерывный процесс протекания в пределах суточных интервалов и длительные циклы повторяемости (до нескольких десятков лет). Кроме того, к ним предъявляются жесткие требования, регламентируемые отраслевыми и государственными стандартами, связанные с высоким уровнем ответственности и серьёзными техногенными рисками. Поэтому диагностическое обслуживание подобных систем может осуществляться либо в режимах микродиагностики (в промежутках между вычислениями управляющих воздействий), либо по окончании технологического процесса. В настоящей статье поставлена задача разработки концепции диагностической инфраструктуры, обеспечивающей совмещение процессов функционального и тестового диагностирования КС в процессе управления сложными динамическими объектами, разработки программно-технических средств реализации отказоустойчивых КС.

GRID-технологии

Технологии *GRID* используются для создания географически распределенной вычислительной инфраструктуры, объединяющей ресурсы различных типов с коллективным доступом к ним в рамках виртуальных организаций, состоящих из предприятий и специалистов, совместно использующих эти ресурсы. Развитие и внедрение технологий *GRID* носят стратегический характер. В ближайшей перспективе эти технологии позволят создать принципиально новый вычислительный инструмент для развития высоких технологий в различных сферах человеческой деятельности. Идейной основой технологии *GRID* является объединение ресурсов путем создания компьютерной инфраструктуры нового типа, обеспечивающей глобальную интеграцию информационных и вычислительных ресурсов на основе сетевых технологий и специального программного обеспечения промежуточного уровня (между базовым и прикладным ПО), а также набора стандартизованных служб для обеспечения надежного совместного доступа к географически распределенным информационным и вычислительным ресурсам: отдельным компьютерам, кластерам, хранилищам информации и сетям. Отметим высокую актуальность исследований, проводимых в данном направлении, которая подтверждается многочисленными научными работами отечественных и зарубежных авторов [1 – 9].

Эффективность функционирования *GRID*-систем как среды с коллективной формой обслуживания пользователей определяется, в первую очередь, согласованностью распределения имеющихся ресурсов, которое должно происходить автоматически, опираясь на планирование вычислительных процессов в *GRID* в целом. Поэтому одной из ключевых функций, требуемых от программного обеспечения *GRID*-систем, является функция диспетчеризации, которая обеспечивает распределение ресурсов из общего ресурсного пула между заданиями, доставку программ и данных для их последующего выполнения [6 – 9]. В архитектуре *GRID*-системы функция диспетчеризации реализуется специальными программными службами, обеспечивающими такой уровень интеграции распределённых ресурсов, при котором *GRID*-система представляется в виде единой операционной среды обработки запросов (заданий, приложений, требований). Совокупность таких служб составляет систему диспетчеризации. Большинство существующих на сегодня систем диспетчеризации предназначено для обслуживания вычислительных *GRID*-систем, состоящих из кластеров (кластеризованных ресурсов) – традиционной формы организации распределённых вычислительных ресурсов. Используемым на практике системам диспетчеризации присущи довольно жёсткие ограничения на их применение: они не способны исключить такие нежелательные эффекты, как задержки времени обработки заданий, задержка обработки в ситуациях, когда имеются простаивающие вычислительные ресурсы, неравномерность загрузки ресурсов кластерах *GRID*-систем. Диспетчеризация включает в себя систему планирования [8, 9], которая обеспечивает накопление и затем гарантированное выделение ресурсов на кластерах (коаллокацию ресурсов): это предотвращает «зависание» заданий, которое является следствием фрагментации ресурсного пула.

Одной из основных задач для эффективного использования ресурсов *GRID* является задача планирования ресурсов для назначения на них заданий внешних и внутренних потоков *GRID.* Отметим, что использование, например, централизованных методов планирования, несмотря на их достаточно высокую популярность в современных промышленных *GRID,* не всегда позволяет эффективно решать проблемы загруженности имеющихся ресурсов. С этой точки зрения более производительными являются иерархические схемы построения *GRID*-систем. Основная идея, положенная в основу этих исследований, заключается в том, что на первом уровне планирования определяются те задания, для которых наиболее оптимальным образом определяются ресурсы, на которые они распределяются для решения (при этом используется математические модели упаковки в контейнеры). Результатом практического использования такого подхода является эмуляция в реально работающую систему, которая показала лучшие результаты обеспечения равномерной загруженности используе-

мых ресурсов. Вместе с тем, разработка формализованных моделей первого уровня планирования иерархической *GRID*-системы, на котором решаются и задачи метапланирования, на наш взгляд, требует проведения теоретических и практических исследований.

Цель данной работы – разработка нового подхода к планированию распределения ресурсов в гетерогенных *GRID*-системах, позволяющего повысить качество обслуживания заданий пользователей, а именно планирования, позволяющего обеспечить равномерную загрузку вычислительных ресурсов, повысить оперативность обработки поступающих на выполнение заданий, уменьшить общее время решения заданий на основе математической модели задачи о наименьшем покрытии и предложить вариант практической реализации метода. Реализацию нового подхода предполагается осуществить на основе имитационной модели, в основе которой лежит двухуровневая архитектура *GRID*.

Базовая конструкция и принципы работы модели

Для описания конструкции и работы предлагаемой модели используем следующие понятия и определения. В качестве базовой архитектуры модели выбирается архитектура вычислительной *GRID*-системы, использующая иерархическую структуру: на первом уровне находится система планирования ресурсов всей *GRID*-системы (планировщик первого уровня), на втором – локальном – уровне осуществляется локальное планирование заданий на ресурсе (локальный планировщик), предназначенное для планирования решения назначенных на данный ресурс заданий. Такая система, в соответствии с классификацией *GRID*-систем, называется двухуровневой иерархической системой – с планировщиками верхнего уровня и локального уровней планирования заданий [3 – 8].

В двухуровневой архитектуре предлагается использовать следующие компоненты:

- глобальная очередь, которая формируется по мере осуществления подписки пользователями своих заданий на их решение в *GRID*; глобальная очередь характеризуется размером L_{GlQuer}, количеством задач, поступивших на их решение в *GRID* за время формирования очереди T_{GlQuer};

- промежуточный пул заданий, выбранных из глобальной очереди, имеющий размер L_{pool} ($L_{pool} \leq L_{GlQuer}$). Размер пула может определяться, например, периодичностью процесса планирования: днями, неделями и т.д.;

- пакет заданий, формируемый как результат обработки заданий пула на основе решения задачи о наименьшем покрытии, имеющий определенный размер L_{buffer} и представляющий собой очередь на локальный ресурс.

- ресурсы R_i , представляющие собой кластеры *GRID* системы, на которых будут решаться спланированные планировщиком верхнего уровня задания.

Последовательность обработки заданий, находящихся в глобальной очереди, следующая:

- задание выбирается из глобальной очереди и помещается в пул по определенному принципу (приоритету, стоимости и т.д.);

- планировщик верхнего уровня в соответствии с режимом (политикой) обслуживания очереди планирует задания на свободные и доступные на данный момент времени ресурсы на основе решения задачи о покрытии;

- после выбора ресурса задание поступает в пакет заданий, который формируется для ресурса R_i, и находится в нем до тех пор, пока он не будет полностью сформирован. По окончании его формирования все задания пакета поступают на выполнение на соответствующий ресурс;

- после того, как задания на ресурсе выполнятся, пул опять полностью заполняется заданиями и т.д. до тех пор, пока все задания из глобальной очереди не будут выбраны, размещены на ресурсы и решены.

Таким образом, новизна предлагаемого подхода заключается:

- в использовании процесса планирования распределения ресурсов на основе предлагаемого метода планирования (в отличие от работы *GRID-систем*, работающих непосредственно с очередями, в которых реализован централизованный планировщик (например, FCFS, SJF). Обсуждения преимуществ такого подхода изложены в работах [3, 4, 6, 8]. Учитывая, что начало следующего действия обработки каждого задания непосредственно начинается после окончания предыдущего и происходит только в заданной последовательности, можно сделать вывод о том, что в условиях нестационарности поступающих заданий предлагаемая модель относится к классу дискретно-событийных имитационных моделей. Отметим, что такие модели рассмотрены в работах [6, 8] и показали свою высокую эффективность при использовании в *GRID*-системах.

Отметим, что пакетный режим обработки на основе буфера достаточно широко используется в устройствах телекоммуникационных и компьютерных систем для повышения эффективности планирования их работы (коммутаторы, маршрутизаторы, сетевые адаптеры). Преимущества использования буфера проявляется в условиях высокой интенсивности данных, поступающих на устройства. Вопросы применения буферов в *GRID* и компьютерных сетях исследовались в работах [6 – 8].

Введем следующие определения, позволяющие уточнить и упростить понимание сущности моделируемых процессов.

Под жизненным циклом задания будем понимать последовательность этапов его обработки – начиная с момента времени помещения его в глобальную очередь, и заканчивая временем его решения на выбранном ресурсе.

Под жизненным циклом глобальной очереди заданий будем понимать совокупность жизненных циклов всех заданий глобальной очереди, для него необходимо определить все характеристики производительности работы модели *GRID.* Отметим, что введение этих понятий обусловлено тем, что анализ производительности дискретно-событийной модели системы необходимо осуществить на разрешении заданий глобальной очереди – сформированной и обслуженной *за определенный промежуток времени работы системы*.

Будем рассматривать гетерогенную *GRID*-систему как среду, состоящую из различных типов ресурсов $\{R_i\}$, и в которую поступает поток различных типов заданий $\{З_i\}$ (рис. 1).

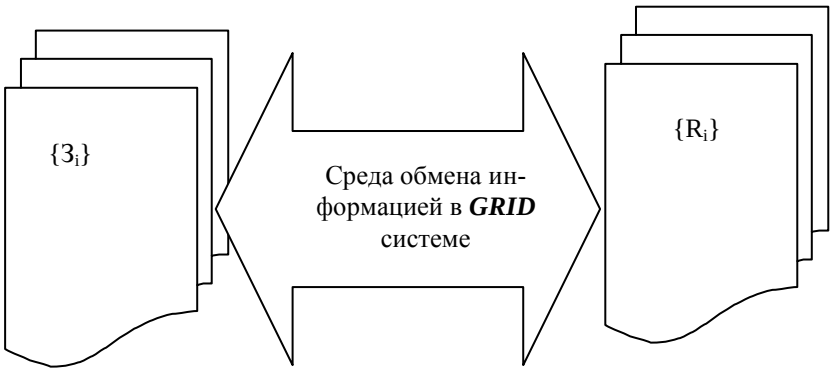

Рис. 1. *GRID* как гетерогенная среда

В общей очереди заданий каждое задание характеризуется вектором характеристик $(\alpha_1^3, \alpha_2^3, ..., \alpha_m^3)$, в качестве которых могут быть тип процессора, объем требуемой для его выполнения памяти, требования к времени решения и т.д. Гетерогенность системы проявляется в том, что одно задание из множества $\{З_i\}$ может выполниться на нескольких из множества $\{R_i\}$ ресурсах, что позволяет использовать это свойство системы для формализации предлагаемого в данной работе подхода. Распределение заданий осуществляется на основе принципа раздельного распределения заданий на ресурсы (рис. 2).

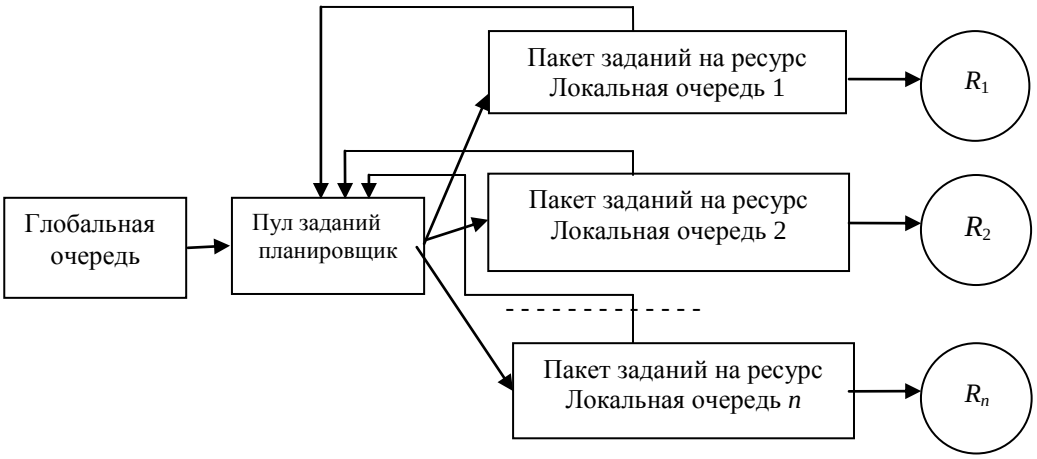

Рис. 2. Раздельное распределение заданий в гетерогенной среде

В этих условиях распространенный метод типа *FCFS* и его модификации приводят к существенной неравномерности загрузки ресурсов [6, 8].

Рассмотрим процесс формирования пакетов – очередей на локальные ресурсы – из глобальной очереди, имеющих ограниченные размеры. Формирование очередей возможно реализовать с помощью таблицы соответствия, показывающей, какие задания на каком ресурсе могут быть решены (выполнены) с учетом того, что на одном ресурсе может выполняться несколько независимых заданий (см. табл. 1, 2).

В ячейках $\{i,j\}$ таблицы соответствия проставляется значение «1» в случае, если задание $З_i$ может быть выполнено на ресурсе R_i В случае использования для планирования алгоритмов типа *FCFS* очередь формируется последовательно, задания отправляются в соответствии с их приоритетом на имеющийся на данный момент времени свободный ресурс непосредственно после нахождения первого на данный момент свободного ресурса из списка доступных ресурсов. *Свободным ресурсом* для данного подхода является такой, для которого в локальной очереди на данный ресурс есть свободное место в пакете, т.е. пакет для этого ресурса не сформирован полностью. При этом сначала в очередь помещаются все задания, которые могут быть решены на первом ресурсе, далее случайным образом выбирается следующий ресурс и аналогично формируется следующая локальная очередь на ресурс и т.д. до тех пор, пока пакеты для формирования локальных очередей на ресурсы будут заполнены (сформированы).

Для упорядочивания формирования локальных очередей положим, что сначала формируется первая локальная очередь (очередь на первый ресурс), далее вторая и т.д. пока все очереди на ресурсы не будут сформированы. Необходимо выбрать некоторую стратегию формирования очередей, обеспечивающую эффективное использование ресурсов системы. Особенностью работы гетерогенной системы с большим количеством ресурсов является то, что некоторые задания могут быть выполнены только на определенных типах ресурсов, а другие – на этих же типах ресурсов и еще дополнительно на нескольких типах ресурсов. Поэтому при произвольном размещении заданий в очереди по таблице соответствия может возникать конкуренция некоторых заданий к одному ресурсу. Это может привести к неравномерности загрузки ресурсов и существенной задержке выполнения конкурирующих заданий. Для устранения этих проблем предлагается распределять задания в очереди на предыдущем шаге таким образом, чтобы подготовить на последующем шаге максимально большее количество свободных ресурсов для выполнения на них следующего подмножества заданий из пула.

Для этого предлагается метод отображения процесса решения заданий глобальной очереди на локальные очереди на ресурсы на основе непрерывного решения задачи о наименьшем покрытии (ЗНП).

Таблица 1

Соответствия, определяющая какие задания на каких ресурсах могут быть решены

	R_{1*}	R_2	R_{3*}	R_4	R_{5*}	R_6	R_7	R_8
$З_1$	1							
$З_2$		1	1	1				
$З_3$	1					1		
$З_4$			1					1
$З_5$		1	1	1				
$З_6$	1						1	
$З_7$					1			1
$З_8$					1		1	
$З_9$					1			
$З_{10}$		1	1					1
$З_{11}$		1			1			
$З_{12}$	1			1		1		

Таблица 2

Соответствия, определяющая какие задания на каких ресурсах могут быть решены

	R_{1*}	R_{2*}	R_3	R_{4*}	R_5	R_6	R_{7*}	R_8
$З_{13}$	1							
$З_{14}$		1						
$З_{15}$					1		1	1
$З_{16}$				1				
$З_{17}$			1				1	1
$З_{18}$		1						
$З_{19}$		1						
$З_{20}$			1			1	1	
$З_{21}$					1	1	1	1
$З_{22}$	1							
$З_{23}$			1	1			1	1
$З_{24}$		1						

Модель планирования на основе задачи о наименьшем покрытии, вычислительный эксперимент и анализ полученных результатов

Случайное формирование локальных очередей на основе метода *FCFS* в гетерогенных структурах может привести к большой неравномерности загрузки ресурсов и их неэффективному использованию.

Распределение заданий рассмотрим на основе принципа раздельного распределения заданий (см. рис.2), но при этом распределение с помощью диспетчера будет осуществляться не статично, как это делается в известных раздельных схемах распределения заданий, а динамично и непрерывно на основе следующей процедуры *D* .

Процедура D

1.На основе множества заданий пула, информации о заданиях и свободных ресурсах планировщиком первого уровня формируется таблица соответствия.

2. На основе таблицы соответствия, решения задачи о наименьшем покрытии (ЗНП) определяется *минимальное количество ресурсов*, на котором очередь заданий, вошедших в пул, может быть выполнена. При этом сначала заполняется очередь на ресурс, который вошел в минимальное покрытие и на нем возможно решение наибольшего количества заданий, далее очередь формируется к ресурсу из покрытия с наибольшими возможностями по решению задач среди оставшихся ресурсов, принадлежащих покрытию и т.д. Если очередь начинает превышать размер пакета заданий на ресурс (т.е. пакет заданий на данный ресурс

сформирован), задания в этом случае помещаются на свободный ресурс, который может его выполнить. Если таких ресурсов уже нет, то задание отправляется обратно в пул заданий.

3. Далее пул заданий заполняется новыми заданиями из глобальной очереди, и процедура повторяется. Благодаря применению в динамической процедуре *D* при планировании очередей ЗНП обеспечивается равномерная загрузка каждого ресурса. Далее реализация планирования выполнения заданий в локальных очередях к ресурсам может быть осуществлено программными средствами локальных планировщиков (PBS).

Для практической реализации процедуры *D* был выбран эффективный приближенный алгоритм решения ЗНП, рассмотренный в работе [6]. Для проверки эффективности такого подхода к планированию распределения ресурсов была разработана программная реализация разработанной модели *GRID_Scheduler_Model*, реализующая имитационную дискретно-событийную модель функционирования гетерогенной среды (см. рис. 3), включающую следующие функциональные элементы:

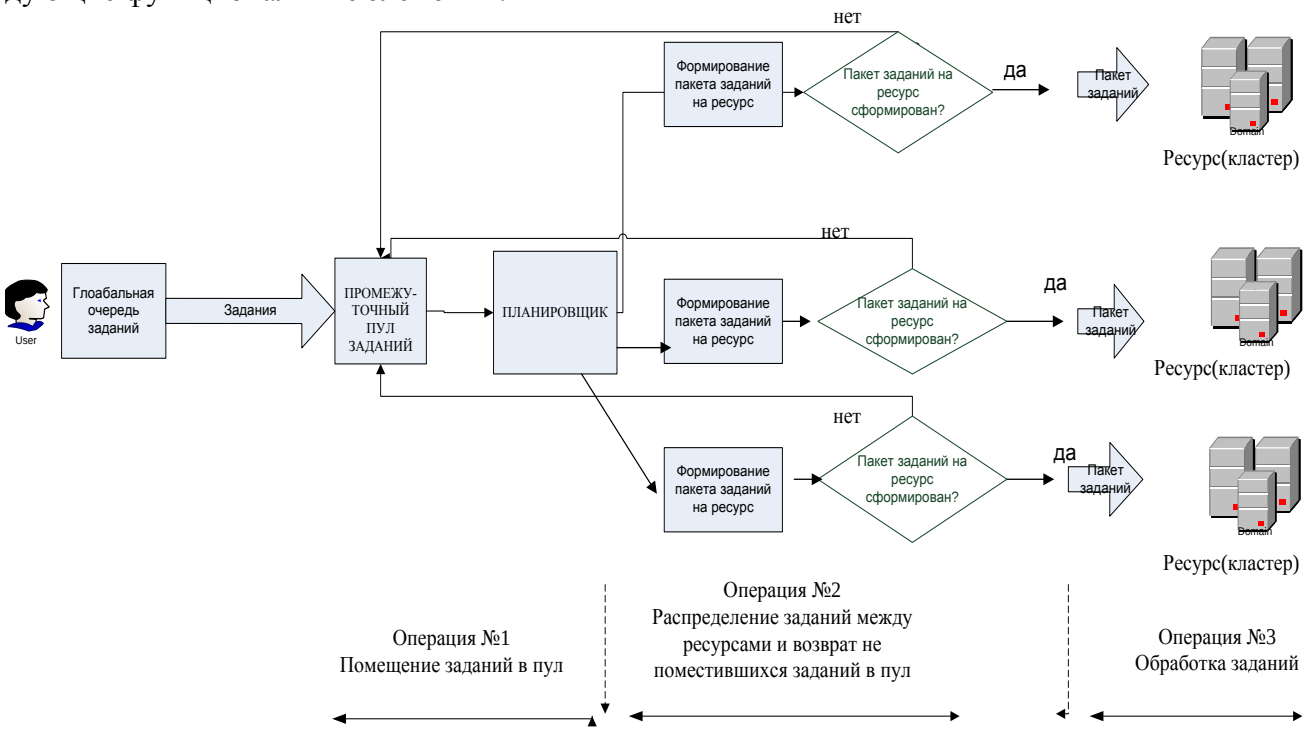

Рис. 3. Общая схема работы модели

Глобальная очередь заданий – задания (заявки, требования, приложения), которые подаются на систему для моделирования процесса работы гетерогенной среды.

Промежуточный пул заданий – буфер системы, в который в порядке поступления загружаются задания из глобальной очереди и *осуществляется планирование ресурсов*.

Планировщик – программный модуль, который распределяет на основе выбранного метода планирования задания, находящиеся в пуле, на ресурсы.

Модель заданий – определяется *сложностью* заданий, задается в тактах – более сложное задание требует для его выполнения большего количества тактов.

Ресурсы – элементы, моделирующие процесс выполнения заданий, которые были на них распределены, *производительность* которых тоже определяется числом тактов необходимых для освобождения ресурсов.

Модель времени – в качестве модели времени была выбрана условная единица времени работы системы – такт.

Один такт включает выполнение системой трех операций:

Операции № 1 – помещение входных заданий в пул;

Операция № 2 – распределение задач в очереди пула между ресурсами и возврат непоместившихся на ресурсы заданий обратно в пул;

Операция № 3 – имитация решения заданий ресурсами.

Для моделирования планирования использовалась матрица в виде прямоугольной таблицы, строки которой соответствуют номерам заданий, столбцы – номерам ресурсов, при этом, если значение ячейки таблицы $m_{ij} = 1$, то это означает, что *i*-е задание решается *j*-м ресурсом. Количество ресурсов, которыми может быть решено m_i-е задание, задается по одному из случайных законов (равномерному, Пуассона, Эрланга). Максимальное количество ресурсов, которым может быть решено m_i-е задание, определяется параметром «*универсальность задачи»*, который задается как определенный процент общего количества ресурсов, на которых может быть выполнено задание. Модель также позволяет оценивать количество заданий, попадающих на каждый ресурс системы, время нахождения заданий в системе, а также оценивать временные выигрыши по времени выполнения всех заданий в очереди при использовании различных процедур планирования и варьировать такими параметрами как сложность решаемых задач, производительность отдельных ресурсов, задержка заданий при их отправке на нижний уровень, частотой планирования выбранной процедурой планирования

В программе *GRID_Scheduler_Model* реализованы следующие методы планирования:

FCFS (First Come First Served) – метод распределяет задания по принципу «первый пришел первым обслужился»(*FIFO*), то есть в порядке их поступления (в порядке очереди), выбирая при этом для задания свободный и доступный ресурс, и централизованный метод планирования ресурсов в *GRID* [20]. Планировщик выбирает первую задачу из очереди пула и пробует поместить ее в пакет заданий на один из ресурсов, на которых она может быть решена. Если операция выполнена успешно, то планировщик переходит к новому заданию, если нет – возвращает его в пул, при этом оно остается первым в пуле и переходит к следующему заданию из очереди.

MC (Minimal Cover) – метод основан на поиске наименьшего количества столбцов, которыми можно покрыть все строки в матрице (ЗНП). Задания, находящиеся в пуле, отображаются в виде прямоугольной матрицы соответствий, в которой строки определяют номера заданий, а столбцы – номера ресурсов. Если *i*-е задание может быть решено на *j*-м ресурсе, то в ячейке *ij* матрицы стоит значение «1» (см. табл. 1, 2). Метод находит наименьшее количество столбцов (ресурсов), которые могут покрыть (решить) все строки (задания), находящиеся в пуле. После этого задания распределяются на ресурсы, вошедшие в минимальное покрытие, и решаются на ресурсах. Если пакеты заданий на соответствующие ресурсы сформированы, а нераспределенные задания еще остались, то распределяем их на другие свободные ресурсы, а оставшиеся нераспределенными на ресурсы задания возвращаем обратно в пул и опять решаем задачу о покрытии.

Постановка эксперимента и метрики производительности работы системы

В качестве метрик производительности работы модели использовались коэффициент использования ресурсов и время ответа системы (время выполнения всех заданий глобальной очереди); коэффициент выигрыша по времени выполнения всех заданий в очереди:

$$K_{â} = \frac{T_{\sum}\ (FIFO)}{T_{\sum}\ (MC)},$$

где $T_{\sum}\ (FIFO)$ – время выполнения очереди заданий при использовании в качестве планировщика процедуры *FIFO*; $T_{\sum}\ (MC)$ – время выполнения той же очереди заданий при использовании в качестве планировщика процедуры *MC*. При моделировании исследовались зависимости времени выполнения всех заданий глобальной очереди в зависимости от числа

заданий в очереди при различных сложностях решаемых заданий и различной производительности ресурсов, а также зависимость коэффициента использования ресурсов гетерогенной *GRID*-системы от количества решаемых заданий. При этом коэффициент использования гетерогенной *GRID*-системы рассчитывался как отношение суммы количества тактов, выполненных ресурсами, работающими на каждом такте выполнения заданий, к общему количеству тактов, которое может быть выполнено всеми ресурсами системы за время выполнения всех заданий глобальной очереди. Анализ полученных зависимостей проведен для различных законов поступления и обслуживания заданий – равномерного закона, законов Пуассона и Эрланга.

Планировщик на основе процедуры *MC* планирует существенно лучше, чем планировщик на основе процедуры *FIFO*, что выражается в том, что он позволяет более равномерно загружать ресурсы системы.

Его применение позволяет уменьшить время выполнения всех заданий. На рис.6 приведена зависимость коэффициента использования 10 гетерогенных кластеров, имеющих по десять однородных ресурсов, при этом

$$K_{â} = \frac{T_{\sum}(FIFO)}{T_{\sum}(MC)} = \frac{65678}{20091} = 3{,}23 ,$$

где $T_{\sum}(FIFO)$ – время выполнения очереди заданий при использовании в качестве планировщика процедуры $FIFO$; $T_{\sum}(MC)$ – время выполнения той же очереди заданий при использовании в качестве планировщика процедуры *MC* заданы в тактах работы ресурсов. Экспериментальное исследование показало, что данный выигрыш может быть равен 10 и более. С увеличением числа ресурсов неравномерность их загрузки на основе процедуры планирования типа *FIFO* резко возрастает, в случае применения процедуры MC практически не изменяется.

Из исследования влияния различных законов распределения изменения параметров системы на характеристики планирования следует, что наиболее «тяжелым», с точки зрения планирования выполнения заданий, является равномерный закон распределения параметров системы.

Поэтому для этого закона распределения приведены результаты исследования взаимосвязи таких характеристик планирования, как время разрешения очереди, коэффициент использования ресурсов системы, размер очереди и размер буфера.

Сравнительный анализ планировщиков на основе процедуры *MC* и *FIFO* при их работе в гетерогенных структурах показал, что, подбирая размер буфера можно обеспечить выигрыш во времени работы выполнения заданий планировщиком на основе *MC* и при этом он обеспечивает равномерность загрузки всех ресурсов, а планировщик на основе процедуры планирования *FIFO* оставляет незагруженными от 30 до 40 % ресурсов.

Заключение

В итоге можно сделать важные для последующих исследований выводы:

-аналитическое выражение поверхности зависимостей имеет квадратичный характер, что можно использовать при одновременном управлении и планировании системы;

-функции поверхностей – выпуклы и имеют локальный минимум, следовательно, можно определить оптимальные соотношения между моделируемыми параметрами на основе получения значений, доставляющих минимум приведенным функциям, решая систему уравнений для частных производных по отдельным параметрам;

-задавая требуемые ограничения на значения коэффициента загрузки и среднее время ответа системы, можно моделировать и прогнозировать значения отдельных параметров

моделируемой системы, например размеры буфера и пула, в зависимости от интенсивности входного потока заданий, поступающего в глобальную очередь *GRID*.

Таким образом, в работе рассмотрен подход к планированию распределением ресурсов вычислительных ресурсов в двухуровневой гетерогенной *GRID*-системе на основе решения ЗНП. Предложена базовая модель и общая схема работы двухуровневой гетерогенной *GRID*-системы, использующая таблицу соответствия назначения заданий на выбранные в результате планирования ресурсы, и показано, что для нее можно использовать подход к планированию выбора ресурсов на основе ЗНП. Основные преимущества рассмотренного подхода заключаются в реализации процесса планирования на основе новых функциональных элементов – пула заданий, выбираемых из глобальной очереди, и пакета заданий на ресурсы, в совокупности позволяющих повысить равномерность загрузки ресурсов и оптимизировать время выполнения всей очереди заданий. Вычислительные эксперименты, проведенные с использованием программной реализации модели, подтвердили эффективность предложенного подхода по сравнению с известным и используемым в современных ***GRID***-системах методом *FIFO* с учетом выбранных метрик производительности системы.

Список литературы: 1. *Загородній, А.Г.* Грід – нова інформаційна обчислювальна технологія для науки / А.Г. Загородній, Г.М. Зіновєв, Є.С. Мартинов, С.Я. Свістунов, В.М. Шадура // Вісник НАН України. – 2005. – №6. – С. 17–25., 2. *Zagorodny, A., Zgurovsky, M., Zinoviey, G., Petrenko, A., Martynov, E.* Integrated Ukraine into European Grid infrastructure // Системні дослідження і інформаційні технології. – 2009. – № 2. – С. 35–49. 3. *Мирошник, М.А.* Решение задач диспетчеризации в распределенных телекоммуникационных системах / М.А. Мирошник, В.Г. Котух, С.Н. Селевко // Радиотехника. – 2011. – Вып. 169. – С. 139–152., 4. *Мирошник, М.А.* Отказоустойчивость распределенных телекоммуникационных систем / М.А. Мирошник, В.Г. Котух, С.Н. Селевко // Радиотехника. – 2011. – Вып. 168. – С. 51–55. 5. *Шелестов, А.Ю.* GRID-технологии в системах мониторинга на основе спутниковых данных / А.Ю. Шелестов, Н.Н. Куссуль, С.В. Скакун // Проблемы управления и информатики. – 2006. – №1,2. – С. 259 – 270. 6. *Мирошник М.А.* Методы повышения отказоустойчивости телекоммуникационных систем / В.Г. Котух, М.А. Мирошник, С.Н.Селевко // Радиотехника. – 2011. – Вып. 166. – С. 259–268. 7. *Коваленко, В. Н.* Методы и алгоритмы управления параллельными заданиями в гриде с ресурсами в форме кластеров / В. Н. Коваленко, Д. А. Семячкин // Вестник Южного научного центра РАН. – 2008. – Т. 4. – № 3. – С. 23–34. 8. *Мирошник, М.А*. Развитие современных направлений цифровых телекоммуникационных систем и сетей / В.Г. Котух, М.А. Мирошник, Селевко С.Н. // Радиотехника. – 2011. – Вып. 165. – С. 254–258. 9. *Коваленко, В.Н.* Грид-диспетчер: реализация службы диспетчеризации заданий в Грид / В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин, Э.З. Любимский, Е.В. Хухлаев, О.Н. Шорин // Сб. докл. Междунар. конф. «Распределенные вычисления и Грид-технологии в науке и образовании», Дубна, 2004. 10. *Минухин, С.В.* Моделирование планирования ресурсов GRID средствами пакета GridSim / С.В. Минухин, А.В. Коровин // Системи обробки інформації. Інформаційні технології та комп'ютерна інженерія. – 2011. – Вип. 3(93). – С. 62 – 68.

Харьковская национальная академия
городского хозяйства
Харьковская государственная академия
железнодорожного транспорта
Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 21.09.2012