

## МОДЕЛЬ РАБОТЫ ЛОКАЛЬНОГО ПЛАНИРОВЩИКА НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО БУЛЕВОГО ПРОГРАММИРОВАНИЯ

**Аннотация.** Рассмотрена имитационная модель работы грид-системы, позволяющая сравнивать существующие методы планирования выполнения заданий, и дано математическое описание исследуемых в модели характеристик. Приведен пример решения задачи на основе метода групповой выборки. Получены экспериментальные результаты, доказывающие преимущества метода планирования выполнения заданий на основе решения задач булевого нелинейного программирования.

**Ключевые слова:** грид-системы, планирование, ресурс, кластер, пакетная обработка, групповая выборка.

### ВВЕДЕНИЕ

Системы планирования выполнения задач разработаны для эффективного и гибкого назначения поступивших в очередь задач на доступные вычислительные ресурсы распределенных систем обработки данных (РСОД). При формировании очереди заданий в РСОД основной проблемой является трудоемкость настройки программного обеспечения, выполняющего их назначение на вычислительные ресурсы. Трудоемкость зависит от следующих факторов: 1) разнородность задач по ресурсным требованиям, аппаратная гетерогенность вычислительных узлов и различная загрузка узлов РСОД требуют специального учета, что обуславливает формирование сложных политик планирования; 2) отсутствие полной информации о ресурсных требованиях задач затрудняет принятие интеллектуальных решений по их планированию.

Широкое применение кластерных, облачных и грид-систем связано с увеличением числа решаемых прикладных задач и значительным возрастанием нагрузок на вычислительные системы.

Действующими версиями систем управления пакетной обработкой (СУПО) являются: Platform LSF, Windows Compute Cluster Server, Condor, PBS, SGE, TORQUE, LoadLever, MOSIX [1]. Одна из наиболее важных функций СУПО — обеспечение механизма планирования распределения заданий. Данная функция может быть реализована непосредственно разработчиками конкретной СУПО либо с помощью внешних планировщиков — отдельно разработанных программных средств (например, Maui). Основа каждого планировщика — алгоритм, от которого зависит эффективность управления заданиями в целом. В [2–4] приведена концептуальная схема планирования пакетов заданий в распределенной вычислительной среде, использующая в качестве алгоритма планирования решение задачи о наименьшем покрытии (ЗНП). Цель данной статьи — разработка процедуры планирования, позволяющей уменьшить суммарное время обработки заданий в очереди и увеличить суммарный коэффициент важности использования ресурсов по сравнению с процедурой планирования на основе решения ЗНП.

### ПОСТАНОВКА ЗАДАЧИ И МОДЕЛЬ ПЛАНИРОВАНИЯ

**Процесс обработки запросов в кластере.** Задачи в грид-системе представляют собой обычный исполняемый файл (скрипт, программный код). Службами грид они доставляются на исполнительные ресурсы, а собственно выпол-

нение происходит в среде их операционной системы (ОС). Как правило, программа, разработанная на конкретном компьютере, не требует каких-либо модификаций для использования в грид. Однако любая программа рассчитана на определенную среду выполнения: ОС, архитектуру компьютера, объемы и характеристики его ресурсов.

Планировщик должен учитывать, что кластер грид-системы является гетерогенной инфраструктурой, в состав которой включены компьютеры с различной архитектурой и комплектацией. Гетерогенность проявляется в том, что разные классы ресурсов (процессор, основная память, кэш-память, дисковая память) различаются типом (процессоры — архитектурой) и характеристиками (процессоры — производительностью, память — объемом). В связи с этим исполнительные ресурсы не могут быть произвольными, а должны соответствовать требованиям задания.

Рассмотрим параметр ресурсного запроса, определяющий время использования ресурсов, т.е. выполнения задания. Известно, что оценка его пользователями редко бывает точной, однако наличие данного параметра важно по двум причинам. Во-первых, даже приблизительная оценка времени выполнения позволяет использовать более эффективные алгоритмы планирования; во-вторых, согласно общепринятой практике работы в системах с разделяемыми ресурсами время выполнения (как и ресурсный запрос в целом) служит защитой от программных ошибок, представляя собой ограничения (по объемам и времени) на потребляемые ресурсы, т.е. при превышении указанных в запросе лимитов задание принудительно завершается. Отметим, что параметр времени исполнения задается в расчете на определенную производительность ресурсов. Как в процессе планирования, так и при запуске задания должен проводиться пересчет времени в соответствии с конкретными исполнительными ресурсами. На практике получили распространение несколько разных языков ресурсных запросов [2]. Язык RSL, применяемый в системе Globus Toolkit, ориентирован на запуск как однопроцессорных, так и многопроцессорных MPI-заданий. Языки ClassAd системы Condor и JDL (WMS) позволяют определять альтернативные варианты ресурсного запроса и специфицировать пользовательские предпочтения при выборе ресурсов. В ряде работ [1–3] предлагаются существенные расширения формализма языка запросов, направленные на спецификацию связанных между собой заданий и цепочек заданий.

При планировании отбор ресурсов выполняется по информационной базе, содержащей сведения о составе и характеристиках ресурсов грид. Передача этих данных в информационную базу осуществляется в оперативном режиме специализированными распределенными системами мониторинга ресурсов, из которых наиболее применяемые — MDS и R-GMA. Ресурсный запрос используется как формальный критерий отбора, однако он только сужает множество возможных исполнительных ресурсов.

Рассмотрим планирование на основе приоритетов, предполагающее, что ресурсы для более приоритетных заданий выделяются раньше, чем для менее приоритетных. В этих условиях широко используемые алгоритмы типа FCFS (FIFO) работают по принципу выделения освободившихся ресурсов самому приоритетному заданию из очереди, которое может на них разместиться. При этом большая часть процессоров будет занята мелкими заданиями, т.е. возникает фрагментация ресурсов. Даже если задание имеет самый высокий приоритет, необходимый ему объем ресурсов может не образоваться и, следовательно, задание никогда не стартует. Для среды, обслуживающей однопроцессорные задания, такой проблемы нет. Она появляется, когда имеются разделяемые ресурсы, например при обслуживании многопроцессорных заданий. Аналогичная ситуация возникает на машинах с общей памятью, в среде с общим файловым пространством и в других случаях, когда ресурсы делятся между заданиями, а не выделяются под задание целиком.

Планирование в грид рассматриваем как циклический процесс, во время которого фиксированное на момент планирования множество заданий, находящихся в очереди, обрабатывается и распределяется по ресурсам, при этом определяются время выделения ресурсов и их адреса. В данном процессе, который выполняется в контексте диспетчеризации, осуществляется координация разделения ресурсов между заданиями пользователей. В среде такого масштаба, как грид, планирование является важнейшим механизмом обеспечения качества обслуживания, которое выражается, прежде всего, в обеспечении приемлемого и предсказуемого времени выполнения заданий пользователя, а также гибкости политики распределения ресурсов в соответствии с приоритетами.

Ресурсы, используемые в режиме разделения между владельцами и пользователями грид, называют неотчуждаемыми. Преимущество грид с неотчуждаемыми ресурсами в том, что не требуется затрат на формирование специальной ресурсной базы и кластер грид может создаваться динамически на ограниченный период времени в целях решения какой-либо проблемы, для которой не хватает ресурсов отдельных членов кооперации. С точки зрения планирования неотчуждаемость ресурсов усложняет ситуацию, поскольку в этом случае на них поступает два потока заданий: поток из грид (глобальный), управляемый планировщиком; локальный поток заданий, запускаемых средствами, отличными от пользовательских интерфейсов грид. Например, в условиях кластеризованных ресурсов локальные задания вводятся непосредственно через интерфейсы системы пакетной обработки. Таким образом, локальные задания неподконтрольны планировщику, хотя создаваемая ими загрузка ресурсов должна учитываться планировщиком при распределении глобальных заданий.

Функционирование грид-систем вычислительного типа можно рассматривать как процесс обслуживания стандартизированных запросов на выполнение вычислений, оформленных в виде заданий для общераспространенных ОС, причем выполняются эти задания на ресурсах, выбираемых из общего пула. Перечислим основные этапы обработки задания: присвоение приоритетов заданиям, мониторинг состояния ресурсов системы и выполнения заданий на основе данных системы наблюдения; планирование выборки заданий из очереди; выделение из общего пула исполнительных ресурсов, на которых задание будет выполняться; доставка исполняемых и входных файлов на исполнительные ресурсы; выполнение задания; по окончании обработки задания доставка результирующих файлов на серверы хранения (в частности, на рабочее место пользователя).

Перечисленные этапы обработки задания выполняются автоматически, без участия субъекта, выдавшего запрос, поэтому кластеры грид-систем представляют собой единую операционную среду.

**Характеристики процесса обработки запросов в кластере.** При наличии ограничений на размер хранимой очереди в системе может возникнуть ситуация, при которой очередной запрос на выполнение задания, поступивший на вход системы, не будет принят к обслуживанию, что приведет к его потере или задержке в обслуживании. Поэтому актуальной является разработка системы с приемлемой (требуемой) скоростью обслуживания заданий, поступающих с заданной интенсивностью.

Отметим дополнительные требования к системе управления: одновременный доступ к любому количеству свободных ресурсов на текущий момент времени; на одном ресурсе системы может выполняться только одно задание.

Под разрешением очереди заданий будем понимать процесс поэтапной выборки заданий из очереди, которые необходимо и возможно выполнить на данном этапе. Задания в очередь поступают в любой момент времени, но ее состояние определяется только между этапами.

Для разрешения очереди необходимо определить наиболее приемлемый способ выборки заданий и наилучший метод его реализации.

Для оценки эффективности выборки заданий применим коэффициент использования ресурсов  $K_{ИР}$ . Данный коэффициент показывает, какая часть ресурсов из общего количества, к которым обращаются размещенные в очереди задания, будет использована. Если  $K_{ИР} = 0$ , то ни один ресурс не задействован, а если  $K_{ИР} = 1$ , то используются все ресурсы, для которых существуют задания на текущий момент. Однако не всегда возможно выбрать такие задания, чтобы были задействованы все ресурсы. Поэтому ставится задача выбора такого способа разрешения очереди заданий, при котором  $K_{ИР}$  стремилось бы к 1. В общем виде  $K_{ИР}$  определяется следующим образом:

$$K_{ИР} = \frac{N_{И}}{N_{О}}, \quad (1)$$

где  $N_{И}$  — количество ресурсов, задействованных при реализации определенной выборки;  $N_{О}$  — количество ресурсов, на которых может быть решена выборка задач из очереди.

В данном виде  $K_{ИР}$  не учитывает приоритетов заданий, поэтому их необходимо учесть в коэффициенте (1). При этом представление коэффициента  $K_{ИР}$  будет зависеть от выбранного способа выборки, а точнее, только числитель в (1). Пусть  $Y_i$  — максимальная величина приоритета заданий из очереди, обращающихся к ресурсу  $R_i$ , тогда знаменатель примет вид

$$N_{О} = \sum_{i=1}^M Y_i. \quad (2)$$

Для обслуживания заданий наиболее перспективны способы групповой выборки, при реализации которых из очереди обслуживается несколько заданий одновременно. Выбираются задания, требующие обработки на разных ресурсах, с максимальной суммой их приоритетов. В случае наличия равнозначных заданий выбирают более «старые».

Пусть  $\{\vec{X}\}$  — множество всех вариантов выборки заданий из очереди,  $\vec{X}$  — один из вариантов выборки заданий, причем

$$\vec{X} = \{x_1, x_2, \dots, x_p, \dots, x_N\}, \quad p = \overline{1, N}, \quad (3)$$

где  $N$  — количество заданий в очереди,  $x_p$  — булева переменная, равная 1, если задание  $Z_p$  выбрано в данном варианте, и 0, если нет.

Пусть  $\beta_p$  — приоритет задания  $Z_p$ . Тогда числитель из (1) примет вид

$$N_{И} = \sum_{p=1}^N \beta_p x_p. \quad (4)$$

С учетом (2) и (4) получаем  $K_{ИР}$  следующего вида:

$$K_{ИР} = \frac{\sum_{p=1}^N \beta_p x_p}{\sum_{i=1}^M Y_i}. \quad (5)$$

Для того чтобы  $K_{ИР}$  принял единичное значение, необходимо, чтобы числитель (4) из (5) был равен знаменателю (2) из (5). Но это предполагается в лучшем варианте, а в общем случае числитель должен стремиться к знаменателю. Поскольку знаменатель для конкретного момента времени — константа, необходимо выполнить такую выборку заданий  $\vec{X}$  из очереди, чтобы числитель принял максимальное значение, причем оно не должно превысить знаменатель. Это

означает, что нужно выбрать из очереди как можно большее количество заданий, которые обращаются к разным ресурсам, и чтобы сумма их приоритетов была максимальна. Стремление к максимуму суммы приоритетов заданий является главным критерием при выборе заданий из очереди.

**МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРОЦЕДУРЫ ПЛАНИРОВАНИЯ  
НА ОСНОВЕ МЕТОДА ГРУППОВОЙ ВЫБОРКИ**

В соответствии с (3) для описания суммы приоритетов  $\beta_k$  выбранных заданий  $x_k$  используем функционал (1)

$$F = \sum_{k=1}^p \beta_k x_k \rightarrow \max. \quad (6)$$

Пусть  $A_{kg}$  — булева переменная, равная 1, если задание  $Z_k$  использует ресурс  $R_g$ , и 0, если нет;  $B_g$  — количество ресурсов данного типа  $R_g$ . Тогда исходя из условия, что в любой момент времени любой ресурс можно использовать для выполнения задания, получаем  $M$  ограничений вида

$$\sum_{k=1}^p A_{kg} x_k \leq B_g, \quad g = \overline{1, M}. \quad (7)$$

Следовательно, необходимо найти такую выборку  $\vec{X}$  из множества  $\{\vec{X}\}$ , для которой функционал (6) примет максимальное значение при выполнении всех ограничений (7). Получена задача линейного программирования с булевыми переменными.

Разрешение очереди запросов при такой формализации происходит поэтапно. Каждый этап состоит из определения оптимальной выборки  $\vec{X}$ , ее обслуживания и изменения функционала (6) и ограничений (7) с учетом изменений в очереди после обслуживания выборки.

Рассмотрим метод групповой выборки на следующей задаче.

Имеется очередь из семи заданий, каждое из которых имеет свой приоритет и требует использования ресурсов определенного типа (табл. 1). По данным табл. 1 составим табл. 2, в которой для каждого ресурса отметим задания, которые к нему обращаются.

Рассмотрим этапы решения данной задачи.

**Этап 1.** Запишем функционал (6), подставив значения из табл. 1:

$$F = 3_1 + 3_3 3_2 + 23_3 + 23_4 + 43_5 + 3_6 + 3_7 \rightarrow \max. \quad (8)$$

Ограничения (7) примут вид

$$3_1 + 3_3 + 3_5 \leq 1, \quad (9)$$

$$3_1 + 3_7 \leq 1, \quad (10)$$

$$3_3 + 3_4 + 3_7 \leq 1. \quad (11)$$

Ограничение (9) сформировано для ресурса  $R_1$ , (10) — для  $R_3$ , а (11) — для  $R_4$ .

**Таблица 1.** Исходные данные

Задания	$3_1$	$3_2$	$3_3$	$3_4$	$3_5$	$3_6$	$3_7$
Приоритеты	1	3	2	2	4	1	1
Ресурсы	$R_1 R_3$	$R_2$	$R_1 R_4$	$R_4$	$R_1$	$R_5$	$R_3 R_4$

**Таблица 2.** Обратные исходные данные

Ресурсы	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$
Задания	$3_1 3_3 3_5$	$3_2$	$3_1 3_7$	$3_3 3_4 3_7$	$3_6$

На основании решения задачи линейного программирования с булевыми переменными получаем, что на этапе 1 необходимо обслужить задания  $Z_2, Z_4, Z_5, Z_6$ .

**Этап 2.** Запишем функционал (6) с учетом результатов этапа 1:

$$F = Z_1 + Z_3 + Z_7 \rightarrow \max.$$

Ограничения (7) примут вид

$$Z_1 + Z_3 \leq 1; Z_1 + Z_7 \leq 1; Z_3 + Z_7 \leq 1.$$

Из результатов решения задачи следует, что необходимо обслужить задание  $Z_3$ .

**Этап 3.** Аналогично получаем, что на данном этапе необходимо обслужить задание  $Z_1$ .

**Этап 4.** На этом этапе необходимо обслужить задание  $Z_7$ .

Таким образом, рассмотренная задача методом групповой выборки решается за четыре этапа.

#### ПРОЦЕДУРА ПЛАНИРОВАНИЯ ПАКЕТОВ ЗАДАНИЙ В ГРИД-СИСТЕМЕ НА ОСНОВЕ МЕТОДА ГРУППОВОЙ ВЫБОРКИ С ИНДИВИДУАЛЬНОЙ СЕГМЕНТАЦИЕЙ

При реализации данного метода задания, находящиеся в очереди, разбиваются на подзадачи. Из очереди выбираются подзадачи, требующие для реализации ресурсы различных типов, с максимальной суммой их приоритетов. В случае наличия равнозначных подзапросов выбирают более «старые».

Подзадача — часть задания, требующая для реализации использования ресурса конкретного типа. Если задание требует  $K$  различных ресурсов, то оно разбивается на  $K$  подзадач. В таком варианте нужно выбрать из очереди как можно большее количество подзадач, обращающихся к разным ресурсам, и сумма приоритетов этих подзадач должна быть максимальной. Стремление к максимуму суммы приоритетов подзадач является главным критерием при выборе подзадач из очереди.

Пусть  $Z_{kg}$  — подзадача задания  $Z_k$ , обращающегося к ресурсу  $R_g$ ;  $C_{kg}$  — приоритет подзадачи  $Z_{kg}$ ;  $\{\bar{X}\}$  — множество всех вариантов выбора подзадач из очереди,  $\bar{X}$  — один из вариантов выбора подзадач. Причем  $\bar{X} = \{x_{11}, x_{12}, \dots, x_{kg}, \dots, x_s\}$ , где  $k = \overline{1, p}$ ,  $g = \overline{1, M}$ ,  $p$  — количество заданий в очереди,  $M$  — количество ресурсов,  $x_{kg}$  — булева переменная, равная 1, если соответствующая подзадача  $Z_{kg}$  выбрана в данном варианте, и 0, если нет. Тогда функционал (6) примет вид

$$F(x) = \sum_{j=1}^{p_1} C_{1j} S_1(C_n^1) + \sum_{j=1}^{p_2} C_{2j} S_2(C_n^2) + \dots \\ \dots + \sum_{j=1}^{p_k} C_{kj} S_k(C_n^k) + \dots + \sum_{j=1}^{p_n} C_{nj} S_n(C_n^n) \rightarrow \max, \quad (12)$$

где  $S_r(C_n^r) = S_1 + S_2 + \dots + S_{p_r}$  — сумма всех возможных сочетаний произведений переменных, содержащих  $r$  различных переменных в каждом произведении  $S_r = X_p X_k \dots X_m$ ;  $p_r = \frac{n!}{r!(n-r)!}$ ;  $C_{rj}$  — коэффициенты в произведениях  $S_r$ , содержащих  $r$  переменных.

Ограничения (7) примут вид

$$\sum_{k=1}^p A_{kg} x_{kg} \leq B_g, \quad g = \overline{1, M}. \quad (13)$$

Получили задачу нелинейного программирования с булевыми переменными. Для рассмотренного ранее примера функционал (12) примет вид

$$F = z_{11} + z_{13} + 3z_2 + 2z_{31} + 2z_{34} + 2z_4 + 4z_5 + z_6 + \\ + z_{73} + z_{74} + z_{11}z_{13} + 2z_{31}z_{34} + z_{73}z_{74} \rightarrow \max,$$

а ограничения (7) примут вид

$$z_{11} + z_{31} + z_5 \leq 1; z_{13} + z_{73} \leq 1; z_{34} + z_4 + z_{74} \leq 1.$$

При методе групповой выборки с индивидуальной сегментацией сформированная очередь разрешится за три этапа.

Исходя из рассмотренного примера, отметим: использование метода групповой выборки с индивидуальной сегментацией позволяет сократить количество этапов, за которые разрешается очередь, что сокращает частоту отказов в обслуживании заданий на входе системы при пиковой нагрузке; в целях обеспечения эффективной работы указанных методов необходимо в качестве математического аппарата применять методы с малой временной сложностью для решения задач линейного и нелинейного программирования с булевыми переменными.

Предлагается использовать в качестве метода планирования ранговый алгоритм решения задач нелинейного булевого программирования, приведенный в [5].

#### РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ КЛАСТЕРА ГРИД-СИСТЕМЫ

Для моделирования функционирования кластера грид-системы, структурная схема которого представлена на рис. 1, формируется множество задач, которые подаются в систему в процессе моделирования работы. Устанавливаются параметры системы (длина пула входных задач, частота планирования, время задержки и другие) и задается метод планирования (предусмотрено несколько алгоритмов решения ЗНП, обозначенных МС, FCFS, и метод, основанный на решении задачи нелинейного булевого программирования, обозначенный NLP).

Процесс моделирования работы кластера состоит из пошагового выполнения трех операций: 1) имитация поступления задач на вход системы; из общего количества задач в пул загружаются первые  $tr$  задач; 2) распределение задач из очереди пула между ресурсами и возврат в пул не поместившихся задач; 3) имитация решения задач с использованием ресурсов; имитация решения задачи заключается в вычитании от значения сложности решения задачи значения производительности ресурса, на котором решается задача (данная операция принята за условную единицу времени — такт; все временные процессы в моделировании работы кластера измеряются в этих единицах).

Моделирование работы кластера осуществляется пошагово: 1) операция № 1 — на вход системы подается заданное количество задач; 2) операция № 2 — формируется пакет заданий выбранным методом; 3) операция № 3 — имитируется процесс решения для всех задач, находящихся на текущий момент в ресурсе; выполняется такое количество тактов, которое было потрачено на формирование пакета заданий; 4) операция № 3 — имитируется процесс решения задач до момента наступления следующего периода планирования или пока не будут решены все задачи; 5) переход к выполнению шага 1.

**Элементы модели.** Модель включает следующие элементы: множество задач — набор задач, которые подаются в грид-систему для моделирования процесса ее работы; множество ресурсов — набор ресурсов, выполняющих решение множества задач, поданных в грид-систему; параметры моделирования — набор параметров, определяющих режим работы и характеристики модели грид-системы; методы планирования — набор методов, с помощью которых грид-система распределяет входящие вызовы между ресурсами.

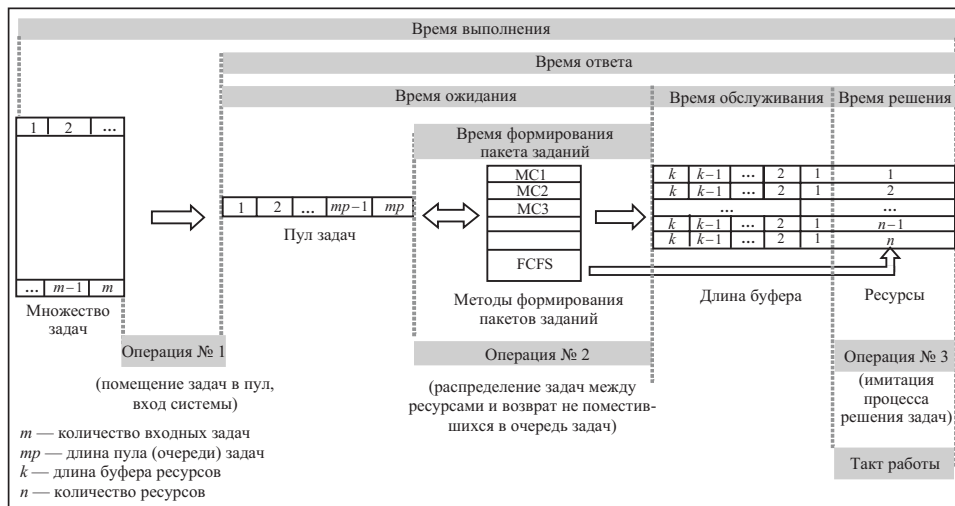


Рис. 1. Структурная схема модели кластера грид-системы

Под термином задача подразумевается структура с набором следующих характеристик: универсальность; типы ресурсов, задействованных для решения задачи; сложность решения; приоритет. Данные характеристики задач определяются параметрами множества. В модели для процесса работы грид-системы необходимо установить следующие параметры множества задач: количество задач — общее количество задач в множестве; количество типов ресурсов — максимальное число типов ресурсов, которые можно использовать для решения задачи. Для совместимости параметров количество типов ресурсов в множестве задач должно совпадать с аналогичным параметром в множестве ресурсов, на которых планируется решать данное множество задач.

Универсальность — максимальное значение универсальности задач во всем множестве (определяет количество типов ресурсов, на которых можно решить задачу). Закон распределения универсальности — случайный закон, по которому задачам назначается универсальность. Максимально возможное значение определяется параметром «универсальность», минимальное значение соответствует единице.

Количество уникальных задач — процент задач, которые могут решаться только одним ресурсом (процентное отношение к общему количеству задач; принимает значения от 0 до 100 %). Сложность решения — максимальное значение сложности решения задач во всем множестве (определяет количество тактов, потраченных на решение задачи ресурсом с производительностью 1 такт). Закон распределения сложности решения — случайный закон, по которому задачам назначается сложность решения. Максимально возможное значение сложности определяется параметром «сложность решения», минимальное значение соответствует одному такту. Суммарная сложность решения — сумма сложностей решения всех задач множества. Данный параметр не задается пользователем, а подсчитывается при формировании множества задач.

Приоритет — максимальное значение приоритета задач во всем множестве (определяет важность решения задачи в условных единицах). Закон распределения приоритета — случайный закон, по которому задачам назначается приоритет. В модели предусмотрена возможность изменения всех параметров по различным случайным законам: равномерному, нормальному, Эрланга, экспоненциальному.

В модели возможно разбиение исходных задач на подзадачи, при этом суммарная сложность решения задач остается прежней, увеличивается только их количество. Ресурс представляет структуру с набором таких характеристик: тип; производительность. В модели возможно задавать следующие параметры для множества ре-



ресурсов: количество ресурсов — общее количество ресурсов в множестве; количество типов ресурсов — максимальное число типов ресурсов. Производительность ресурса определяется количеством тактов сложности решения задачи, которое будет выполнено ресурсом за 1 такт. Имеется также возможность задавать закон распределения производительности, по которому ресурсам назначается производительность. Кроме того, в модели автоматически рассчитывается суммарная производительность (общая производительность всех ресурсов множества).

Размер буферов всех ресурсов множества определяет количество задач, которые могут быть поставлены в очередь на решение к ресурсу. В модели предусмотрено два типа буфера: последовательный — задачи последовательно, по одной, решаются ресурсом; параллельный — все задачи в буфере решаются ресурсом одновременно. Сформирован набор параметров, определяющих режим работы и характеристики модели грид-системы. В модели задаются следующие параметры: задачи — множество задач, которое подается в систему в процессе моделирования; ресурсы — множество ресурсов системы, на которых решаются задачи при моделировании; интенсивность — максимальное количество задач, поступающих на вход системы при очередном такте планирования. Закон распределения интенсивности — случайный закон, по которому задачи распределяются для последовательного поступления на вход системы.

Максимально возможное значение интенсивности определяется параметром «интенсивность», минимальное значение — ноль; отчужденность — максимальное количество вышедших из строя ресурсов на очередном такте планирования (задается от 0 до 100 % относительно количества ресурсов). Закон распределения отчужденности — случайный закон, по которому распределяется отчужденность для каждого такта планирования. Максимальное значение отчужденности определяется параметром «отчужденность», минимальное значение — ноль; длина пула — максимальное количество задач, которые может поместить система для дальнейшей обработки; задержка — количество тактов, затрачиваемых системой на передачу задачи из буфера в ресурс на решение; частота планирования — количество тактов между планированиями, а также между поступлениями новых задач на вход системы; коэффициент планирования — коэффициент, на который уменьшается количество операций, выполненных планировщиком для перевода операций, использованных на планирование в затраченное время, которое в системе выражается в условных единицах — тактах.

Рассмотрим набор методов, с помощью которых грид-система распределяет входящие задачи между ресурсами. В данной модели существует возможность проводить исследования с применением следующих методов планирования выполнения заданий.

**Метод планирования МС.** Метод основан на поиске наименьшего числа ресурсов, которыми можно решить все задачи, т.е. ЗНП. Для поиска наименьшего числа ресурсов используется приближенный частотный метод [1, 5].

Алгоритм работы МС: 1) определяются соответствия между задачами и ресурсами, которые они используют; 2) решается задача поиска минимального покрытия частотным методом; 3) задачи помещаются в буферы ресурсов, вошедших в минимальное покрытие; 4) процедура планирования заканчивается, если поместились все задачи, иначе задачи возвращаются в пул и выполняется переход к «операции № 2».

В процессе планирования задачи, для решения которых не осталось свободных ресурсов, возвращаются обратно в пул.

**Метод планирования FCFS.** Метод основан на принципе «первым пришел, первым обслужился».

Алгоритм работы FCFS: 1) определяются соответствия между задачами и ресурсами, на которых они могут решаться; 2) выбирается первая по порядку

задача из пула и помещается в первый свободный ресурс, на котором она может быть решена; если таких ресурсов не оказалось, задача возвращается в пул; 3) процедура планирования прекращается после обработки в пуле последней задачи. В данном методе не используется буфер, задачи сразу посылаются на решение ресурсу, если он свободен.

**Метод планирования NLP.** Метод основан на поиске решения задачи нелинейного программирования: алгоритм находит максимальное количество задач с суммарным максимальным приоритетом, которые могут быть помещены в буфер свободных ресурсов.

Алгоритм работы NLP: 1) определяются соответствия между задачами и ресурсами, на которых они могут решаться; 2) решается задача нелинейного программирования; 3) задачи помещаются в буферы ресурсов; если остались задачи, для которых есть свободные ресурсы, выполняется переход к «операции № 2»; если таких задач нет, процедура планирования заканчивается. В процессе планирования задачи, не имеющие свободных ресурсов, которые могут их решить, возвращаются в пул.

**Исследуемые характеристики работы грид-системы.** В представленной модели в процессе работы грид-системы рассматриваются следующие характеристики.

- Время выполнения — количество тактов работы системы, за которое решено все множество задач, поданных на вход грид-системы.
- Время ответа — среднее количество тактов за которое решена одна задача в режиме моделирования грид-системы.
- Максимальное время ответа — максимальное количество тактов, за которое решена одна задача.
- Время ожидания — среднее количество тактов, которое ожидали задачи на входе системы до последующей обработки.
- Время обслуживания — среднее количество тактов, которое задачи находились в буфере назначенного ресурса до начала их решения данным ресурсом.
- Время планирования — среднее количество тактов, которое затрачено системой на планирование.
- Коэффициент использования — среднее геометрическое значение коэффициентов использования всех ресурсов (коэффициент использования ресурса — отношение суммы приоритетов всех решенных задач этим ресурсом к сумме приоритетов всех задач из входного множества, которые могли быть решены на данном ресурсе).
- Коэффициент загрузки — среднее геометрическое значение коэффициентов загрузки всех ресурсов. Коэффициент производительности ресурса — отношение количества решенных задач этим ресурсом к количеству всех задач из входного множества, которые могли быть решены на данном ресурсе.
- Коэффициент важности — среднее геометрическое отношения суммарного приоритета задач, помещенных в буферы в результате планирования, к суммарному приоритету задач, находившихся в пуле.
- Коэффициент сохранения важности — среднее геометрическое отношения суммарного приоритета задач, помещенных в буферы в результате планирования после отказа случайного количества ресурсов, к суммарному приоритету задач, помещенных в буферы в результате планирования до отказа ресурсов.
- Коэффициент ускорения — отношение расчетного времени выполнения задач, решаемых последовательно одним ресурсом с производительностью, равной средней производительности всех ресурсов грид, ко времени выполнения этих задач грид-системой.

**Пример моделирования.** При заданных значениях (рис. 2 и 3) и параметрах моделирования (рис. 4) полученные результаты приведены в табл. 3.

Задание входных задач и ресурсов		
Входные задачи		
Параметр	Значение	T1000_70_34676
Количество задач	5000	T1000_30_34806
Количество типов ресурсов	50	T1000_30_35125
Универсальность	10	T2000_30_35125
Закон распределения универсальности	Эрланга	T5000_50_87746
Количество уникальных задач в %	15	
Сложность решения	50	
Закон распределения сложности решения	Экспоненциальный	
Суммарная сложность решения	87746	
Приоритет задач	25	
Закон распределения приоритета	Нормальный	

Рис. 2. Параметры множества задач

Ресурсы		
Параметр	Значение	R100_454
Количество ресурсов	70	R70_309
Количество типов ресурсов	50	R70_329
Закон распределения типов	Равномерный	R70_320
Производительность	10	R70_314
Закон распределения производительности	Экспоненциальный	R70_326
Суммарная производительность	310	R70_269
Длина буфера	70	R70_310
Тип буфера	Последовательный	

Рис. 3. Параметры множества ресурсов

Параметры моделирования работы GRID системы		
Задачи	T5000_50_87746	<input checked="" type="checkbox"/> Метод планирования на основе поиска минимального покрытия MC
Интенсивность	50	<input type="checkbox"/> Оптимизированный метод планирования на основе поиска минимального покрытия MCO
Закон распределения интенсивности	Нормальный	<input type="checkbox"/> Метод планирования на основе точного метода поиска минимального покрытия MCA
Ресурсы	R70_310	<input type="checkbox"/> Метод планирования на основе поиска минимального покрытия жадным методом GREEDE
Отчужденность	10	<input checked="" type="checkbox"/> Метод планирования FCFS
Закон распределения отчужденности	Равномерный	<input checked="" type="checkbox"/> Метод планирования на основе решения задачи нелинейного программирования NLP
Длина пула	70	
Задержка	50	
Частота планирования	50	
Коэффициент планирования	10000	

Рис. 4. Параметры моделирования

Таблица 3. Результаты моделирования работы грид-системы при заданных параметрах

Методы планирования	Показатели качества планирования								
	Время выполнения (такт)	Время ожидания (такт)	Коэффициент важности	Коэффициент ускорения	Коэффициент сохранения важности	Коэффициент загрузки	Коэффициент использования	Максимальное время ответа (такт)	Время обслуживания (такт)
MC	16157	36	0.626	1.226	0.781	0.384	0.38	5347	2095
FCFC	18762	196	0.188	1.056	0.763	0.334	0.334	2075	50
NLP	14206	5	0.999	1.395	0.868	0.453	0.454	4168	833

## ЗАКЛЮЧЕНИЕ

Как показывают результаты моделирования, использование в качестве алгоритма планирования предложенного в [6, 7] метода решения задачи нелинейного булевого программирования позволяет существенно увеличить такие по-

казатели, как коэффициент важности, коэффициент сохранения важности, коэффициент ускорения, и уменьшить время выполнения заданий и время ожидания по сравнению с известными процедурами планирования.

#### СПИСОК ЛИТЕРАТУРЫ

1. Минухин С.В. Модели и методы решения задач планирования в распределенных вычислительных системах. Х.: Щедрая усадьба плюс, 2014. 323 с.
2. Пономаренко В.С., Листровой С.В., Минухин С.В., Знахур С.В. Методы и модели планирования ресурсов в GRID-системах. Х.: ИНЖЭК, 2008. 407 с.
3. Листровой С.В., Минухин С.В. Подход и модель планирования распределения ресурсов в Grid. *Международный научно-технический журнал «Проблемы управления и информатики»*. 2012. № 5. С. 65–82.
4. Шелестов А.Ю., Куссуль Н.Н., Скакун С.В. Grid-технологии в системах мониторинга на основе спутниковых данных. *Проблемы управления и информатики*. 2006. № 1–2. С. 259–270.
5. Коваленко В.Н., Коваленко Е.И., Корягин Д.А., Любимский Э.З., Хухлаев Е.В., Шорин О.Н. Грид-диспетчер: реализация службы диспетчеризации заданий в Грид. *Сб. докл. Междунар. конф. «Распределенные вычисления и Грид-технологии в науке и образовании»*. Дубна, 2004. С. 133–139.
6. Симоненко В.П. Теоретические основы проектирования динамических пространственных планировщиков неоднородных GRID систем. *Электронное моделирование*. 2011. Т. 33, № 5. С. 57–71.
7. Минухин С.В. Метод распределения учебной загрузки с использованием нелинейного булевого программирования. *Інформаційні технології в освіті*. 2010. Вип. 5. С. 104–108.

Надійшла до редакції 17.02.2017

#### **О.С. Лістрова, В.О. Бриксін, М.С. Курцев** **МОДЕЛЬ РОБОТИ ЛОКАЛЬНОГО ПЛАНУВАЛЬНИКА НА ОСНОВІ РОЗВ'ЯЗАННЯ ЗАДАЧ НЕЛІНІЙНОГО БУЛЕВОГО ПРОГРАМУВАННЯ**

**Анотація.** Розглянуто імітаційну модель роботи грід-системи, яка дозволяє порівнювати наявні методи планування виконання завдань, та дано математичний опис досліджуваних в моделі характеристик. Наведено приклад розв'язання задачі на основі методу групової вибірки. Отримано експериментальні результати, що доводять переваги методу планування виконання завдань на основі розв'язання задач булевого нелінійного програмування.

**Ключові слова:** грід-системи, планування, ресурс, кластер, пакетне оброблення, групова вибірка.

#### **E.S. Listrovaya, V.A. Bryksin, M.S. Kurtsev** **MODELING LOCAL PLANNER OPERATION BASED ON SOLUTION OF NONLINEAR BOOLEAN PROGRAMMING PROBLEMS**

**Abstract.** The simulation model of grid system operation is considered, which allows comparing available job scheduling methods. The mathematical description of the characteristics analyzed in the model is given. An example of problem solution based on cluster sampling method is presented. The experimental results are obtained, which demonstrate the advantages of scheduling method based on the solution of Boolean linear programming problems.

**Keywords:** grid systems, planning, resource, cluster, batch processing, sample group.

**Лістрова Елена Сергеевна,**  
кандидат техн. наук, доцент кафедры Национального аэрокосмического университета им. Н.Е. Жуковского, Харьков, e-mail: listravkina@gmail.com.

**Брыксин Владимир Александрович,**  
кандидат техн. наук, старший преподаватель Украинского государственного университета железнодорожного транспорта, Харьков, e-mail: vladimir.bryksin@gmail.com.

**Курцев Максим Сергеевич,**  
аспирант Украинского государственного университета железнодорожного транспорта, Харьков, e-mail: kurtsev\_m@ukr.net.