

---

УДК 621.396

**С.В. Листровой**<sup>1</sup>, д-р техн. наук,  
**Е.С. Листровая**<sup>2</sup>, канд. техн. наук., **М.С. Курцев**<sup>1</sup>, аспирант

<sup>1</sup> Украинский государственный университет железнодорожного транспорта  
(Украина, 61050, Харьков, пл. Фейербаха, 7,  
тел. (050) 3029912, e-mail: om1@yandex.ru; kurtsev\_m@ukr.net),

<sup>2</sup> Национальный аэрокосмический университет им. Н.Е. Жуковского  
(Украина, 61070, Харьков, ул. Чкалова, 17,  
e-mail: listravkina@gmail.com)

## **Метод и модель планирования распределения пакетов заданий в кластере Grid системы**

Представлена имитационная модель работы Grid системы, которая позволяет 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, подтверждающие преимущества метода планирования выполнения заданий на основе решения задач булевого нелинейного программирования.

Представлено імітаційну модель роботи Grid системи, яка дозволяє порівнювати існуючі методи планування виконання завдань, і дано математичний опис досліджуваних в моделі характеристик. Наведено приклад розв'язку задачі на основі методу групової вибірки. Отримано експериментальні результати, які доводять переваги методу планування виконання завдань на основі розв'язку задач булевого нелінійного програмування.

*К л ю ч е в ы е с л о в а:* Grid, планирование, ресурс, кластер, пакетная обработка, групповая выборка.

Системы планирования выполнения задач обеспечивают решение проблемы эффективного и гибкого назначения поступивших задач обработки данных на доступные вычислительные ресурсы распределенных систем обработки данных (РСОД). При использовании РСОД основной проблемой является трудоемкость настройки программного обеспечения, выполняющего назначение задач на вычислительные ресурсы, связанная со следующими свойствами РСОД:

1) разнородность задач согласно ресурсным требованиям, аппаратная гетерогенность вычислительных узлов и различная загрузка узлов РСОД требуют специального учета, что ведет к созданию сложных политик планирования;

© С.В. Листровой, Е.С. Листровая, М.С. Курцев, 2016

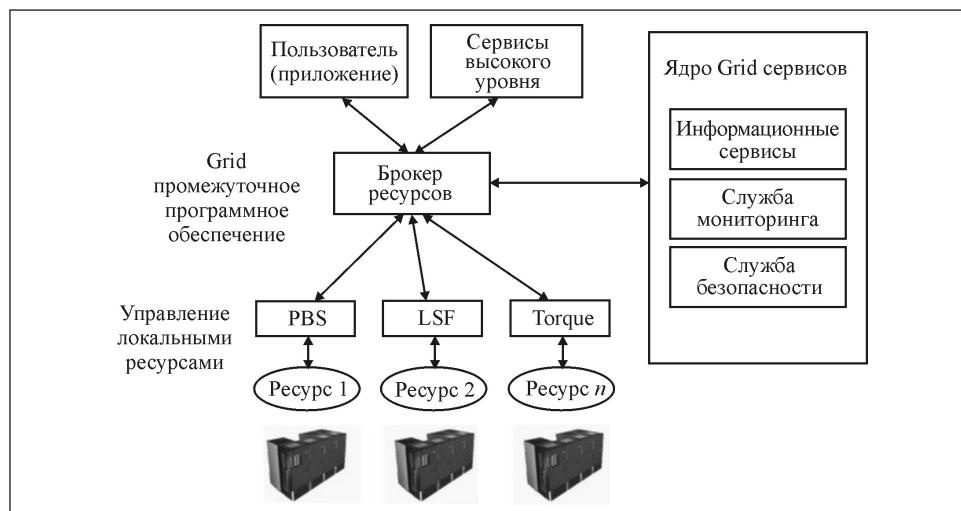


Рис. 1. Система управления пакетной обработкой в Grid

2) отсутствие полной информации о ресурсных требованиях задач усложняет принятие интеллектуальных решений по их планированию.

Широкое распространение кластерных систем, Grid систем и облачных систем связано с увеличением числа решаемых прикладных задач и значительным возрастанием нагрузок на вычислительные системы. Поставщики сетевых сервисов, опираясь на крупные консолидированные центры обмена данными, особое внимание стали уделять совершенствованию методов планирования задач обработки данных. В общем случае схема управления ресурсами в Grid имеет вид, представленный на рис. 1.

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. Эта функция может быть реализована непосредственно разработчиками конкретной СУПО либо с помощью внешних планировщиков — отдельно разработанных программных средств (например, планировщика Maui). В основе каждого планировщика лежит алгоритм, от которого существенно зависит эффективность управления заданиями в целом.

В работах [1, 2] приведена концептуальная схема планирования пакетов заданий в распределенной вычислительной системе (РВС) (рис. 2), в которой в качестве алгоритма планирования использовано решение задачи о наименьшем покрытии (ЗНП). Как показано в [1, 3], данная концепция планирования позволяет существенно сократить суммарное время выпол-

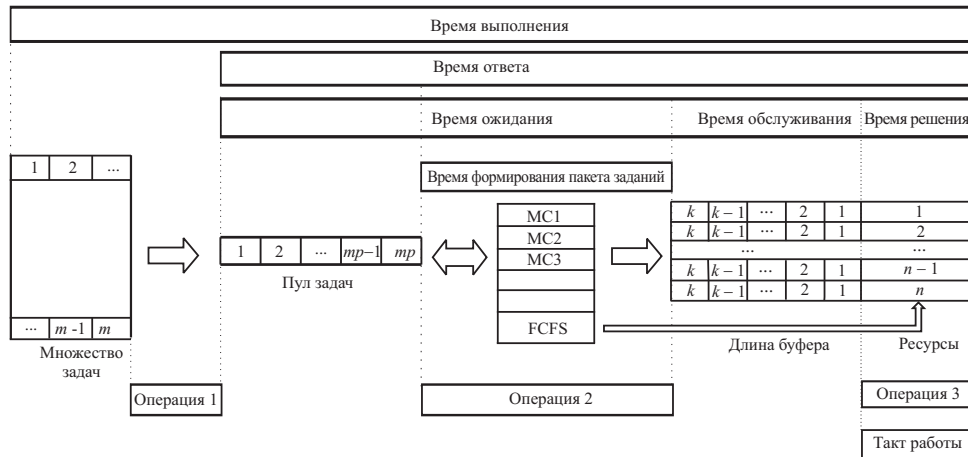


Рис. 2. Структурная схема модели кластера Grid системы:  $m$  — число задач в очереди;  $mp$  — длина пула (очереди) задач;  $k$  — длина буфера ресурсов;  $n$  — количество ресурсов; операция 1 — помещение задач в пул (вход системы); операция 2 — распределение задач между ресурсами и возврат непоместившихся в очередь; операция 3 — имитация процесса решения задач; MC1— MC3, FCFS — методы формирования пакетов заданий

нения заданий и увеличить коэффициент использования ресурсов системы по сравнению с алгоритмом FCFS (First Come First Serve). Это списочный алгоритм и алгоритм обратного заполнения (Backfill), в котором задания выстраиваются в список согласно их поступлению, и как только необходимое число ресурсов становится доступным заданию, находящемуся в начале списка, задание поступает на ресурс. Однако к недостаткам данного подхода можно отнести тот факт, что в этом случае нет возможности обеспечить максимальный суммарный приоритет выполняемых заданий на отдельных этапах планирования в РВС.

Предлагается разработанная процедура планирования, позволяющая преодолеть указанный недостаток, при этом уменьшить суммарное время обработки заданий в системе и увеличить коэффициент использования ресурсов по сравнению с процедурой планирования на основе решения ЗНП.

**Постановка задачи и модель планирования. Процесс обработки запросов в кластере.** Задание Grid представляет собой обычный исполняемый файл (скрипт, программный код). Службами Grid задание доставляется на исполнительные ресурсы, а выполнение происходит в среде операционной системы (ОС) этих ресурсов. Как правило, программа, подготовленная на определенном компьютере, не требует каких-либо модификаций для использования в Grid. Однако любая программа рассчитана

на определенную среду выполнения — ОС, архитектуру компьютера, объемы и характеристики его ресурсов.

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

Следует обратить внимание на параметр ресурсного запроса, определяющий время использования ресурсов, т.е. время выполнения задания. Известно, что пользовательская оценка этого времени редко бывает точной, однако наличие этого параметра представляется весьма важным по двум причинам. Во-первых, даже приблизительная оценка времени выполнения позволяет использовать более эффективные алгоритмы планирования. Во-вторых, существует общепринятая практика работы в системах с разделяемыми ресурсами, согласно которой время выполнения (как и ресурсный запрос в целом) служит защитой от программных ошибок, представляя собой ограничения (по объемам и по времени) на потребляемые ресурсы, т.е. при превышении указанных в запросе лимитов задание принудительно завершается.

Заметим, что параметр времени исполнения заказывается в расчете на определенную производительность ресурсов. Как в процессе планирования, так и при запуске задания необходимо выполнить пересчет времени в соответствии с конкретными исполнительными ресурсами. На практике получили распространение несколько 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 работах [1—3] предложены существенные расширения формализма языка запросов, направленные на спецификацию связанных заданий и цепочек заданий.

При планировании отбор ресурсов выполняется по информационной базе, содержащей сведения о составе и характеристиках ресурсов Grid. Поставка этих данных в информационную базу осуществляется в оперативном режиме специализированными распределенными системами мониторинга ресурсов, из которых наиболее распространены MDS и R-GMA.

Рассмотрим планирование на основе приоритетов, когда предполагается, 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 для самого приоритетного задания из очереди, которое может на них разместиться. При этом большая часть процессоров будет всегда занята мелкими заданиями, т.е. возникает фрагментация ресурсов. Даже если задание имеет самый высокий приоритет, необходимый ему объем ресурсов может никогда не образоваться и, следовательно, задание может никогда не стартовать. Для среды, обслуживающей однопроцессорные задания, этой проблемы нет. Она возникает, 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 Grid рассматривается как циклический процесс обработки фиксированного на момент планирования множества заданий, находящихся в очереди, и распределения их по ресурсам в соответствии с определенным временем выделения ресурсов и их адресом. При выполнении этого процесса осуществляется координация разделения ресурсов между заданиями пользователей. В среде такого масштаба, как Grid, планирование является важнейшим механизмом обеспечения качества обслуживания, прежде всего, обеспечения 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 Grid, называют неотчуждаемыми. Достоинство Grid с неотчуждаемыми ресурсами состоит в том, что не требуются затраты на формирование специальной ресурсной базы, и кластер Grid может создаваться динамически на ограниченный период времени с целью решения какой-либо крупной проблемы, для которой ресурсов отдельных членов кооперации не хватает. С точки зрения планирования неотчуждаемость ресурсов усложняет ситуацию, так как в этом случае на них поступает два потока заданий: 1) поток из Grid (глобальный), управляемый планировщиком; 2) локальный поток заданий, которые запускаются средствами, отличными от пользовательских интерфейсов Grid. Например, в условиях кластеризованных ресурсов локальные задания вводятся непосредственно через интерфейсы специального программного обеспечения. Таким образом, локальные задания неподконтрольны планировщику, хотя создаваемая ими загрузка ресурсов должна учитываться планировщиком при распределении глобальных заданий.

Стандартизированные запросы на выполнение вычислений, обслуживаются как задания для общераспространенных ОС. Выполнение этих заданий осуществляется на ресурсах, выбранных из общего пула. Основные этапы обработки задания следующие:

- 1) присвоение приоритетов заданиям и мониторинг состояния ресурсов системы и состояния выполнения заданий на основе данных системы наблюдения;
- 2) планирование выборки заданий из очереди;
- 3) выделение из общего пула исполнительных ресурсов, на которых задание будет выполняться;
- 4) доставка выполняемых файлов и входных файлов на исполнительные ресурсы;
- 5) выполнение задания;
- 6) по окончании выполнения задания доставка результирующих файлов на серверы хранения (на рабочее место пользователя).

Все перечисленные этапы обработки задания выполняются автоматически без участия субъекта, выдавшего запрос (в частности, пользователя, но это может быть и программа). Поэтому кластеры Grid систем действительно представляют собой единую операционную среду.

**Характеристики процесса обработки запросов в кластере.** При наличии ограничений на размер хранимой очереди в системе может возникнуть ситуация, когда 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.

Необходимо учесть дополнительное требование к системе управления — она должна иметь одновременный доступ к любому числу свободных ресурсов на данный момент времени и на одном ресурсе системы может выполняться только одно задание.

Под разрешением очереди (РО) будем понимать процесс поэтапной выборки заданий из очереди заданий, которые необходимо и возможно выполнить на данном этапе. Задания в очередь поступают в любой момент времени, но состояние очереди определяется только между этапами. При выполнении РО заданий необходимо выбрать наиболее приемлемый способ выборки заданий и наилучший метод реализации выбранного способа.

Для оценки эффективности выборки заданий будем использовать коэффициент использования ресурсов  $K_{и.р}$ , позволяющий определить, какая часть из общего количества ресурсов, к которым обращаются стоящие в очереди задания, будет использована. Если  $K_{и.р} = 0$ , то ни один ресурс не

будет использован, если  $K_{и,р} = 1$ , то будут использованы все ресурсы, для которых существуют задания на данный момент. Однако не всегда можно выбрать такие задания, для которых все ресурсы были бы задействованы. Поэтому возникает задача выбора такого способа РО заданий, при котором  $K_{и,р} \rightarrow 1$ . В общем виде можно записать

$$K_{и,р} = N_{и} / N_{о}, \quad (1)$$

где  $N_{и}$  — число ресурсов, которые будут задействованы при реализации определенной выборки;  $N_{о}$  — число ресурсов, для которых существуют задания из очереди. Но в (1) не учтены необходимые приоритеты заданий. Если их учесть, то значение коэффициента  $K_{и,р}$  будет зависеть от способа выборки (числитель в (1)).

Пусть  $Y_i$  — максимальная величина приоритета заданий из очереди заданий, обращающихся к ресурсу  $R_i$ , а  $M$  — число ресурсов. Тогда знаменатель в (1) примет вид

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

Наиболее перспективными способами обслуживания заданий являются способы групповой выборки. Способ групповой выборки — это такой способ, при реализации которого из очереди заданий обслуживается несколько заданий одновременно. Выбираются задания, для которых требуется обработка на различных ресурсах, и сумма приоритетов которых максимальна. В случае наличия равнозначных заданий выбирают более «старые».

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

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

где  $N$  — число заданий в очереди;  $x_p$  — булева переменная. Если задание  $Z_p$  выбрано в этом варианте, то  $x_p = 1$ , если нет, то  $x_p = 0$ .

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

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

С учетом (2) и (4) получаем

$$K_{и,р} = \sum_{p=1}^N \beta_p x_p / \sum_{i=1}^M Y_i. \quad (5)$$

Для того чтобы коэффициент  $K_{и,р}$  принял единичное значение, необходимо, чтобы в (5) числитель был равен знаменателю (в лучшем случае). В общем случае числитель должен стремиться к знаменателю. Поскольку знаменатель для конкретного момента времени является константой, необходимо сделать такую выборку заданий  $X$  из очереди, чтобы числитель принял максимальное значение, которое не должно превысить знаменатель. Это означает, что необходимо выбрать из очереди как можно большее число заданий, обращающихся к различным ресурсам, сумма приоритетов которых была бы максимальной. Стремление к максимальной сумме приоритетов выбранных заданий является главным критерием при выборе заданий из очереди.

**Математическая модель процедуры планирования.** В соответствии с (3) для описания суммы приоритетов  $\beta_k$  выбранных  $x_k$  заданий используем функционал (1):

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

Пусть  $A_{kg}$  — булева переменная, равная единице, если задание  $Z_k$  использует ресурс  $R_g$ , и равная нулю, если нет;  $B_g$  — число ресурсов типа  $R_g$ . Тогда, исходя из условия, что в любой момент времени любой ресурс может быть использован для выполнения задания, получаем  $M$  ограничений вида

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

Следовательно, необходимо найти такую выборку  $X$  из множества  $\{X\}$ , 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 нахождения оптимальной выборки  $X$ , ее обслуживания и изменения функционала (6) и ограничений (7) с учетом изменений в очереди после обслуживания выборки.

**Пример.** Рассмотрим метод групповой выборки при решении следующей задачи. Очередь состоит из семи заданий, каждое из которых имеет свой приоритет и требует использования ресурсов определенного типа. Исходные данные задачи представим в следующем виде:

Задание . . . . .	$Z_1$	$Z_2$	$Z_3$	$Z_4$	$Z_5$	$Z_6$	$Z_7$
Приоритет . . . . .	1	3	2	2	4	1	1
Ресурс . . . . .	$R_1 R_3$	$R_2$	$R_1 R_4$	$R_4$	$R_1$	$R_5$	$R_3 R_4$



По этим данным определим, какие задания обращаются к каждому ресурсу:

Ресурс . . . . .	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$		
Задание . . . . .	$z_1$	$z_3$	$z_5$	$z_2$	$z_1 z_7$	$z_3 z_4 z_7$	$z_6$

Э т а п ы р е ш е н и я з а д а ч и.

1. Запишем функционал (6), подставив указанные выше значения приоритетов:

$$F = z_1 + 3 z_2 + 2 z_3 + 2 z_4 + 4 z_5 + z_6 + z_7 \rightarrow \max.$$

Ограничения (7) примут вид

$$z_1 + z_3 + z_5 \leq 1, \quad (8)$$

$$z_1 + z_7 \leq 1, \quad (9)$$

$$z_3 + z_4 + z_7 \leq 1. \quad (10)$$

Ограничение (8) составлено для ресурса  $R_1$ , (9) — для  $R_3$ , а (10) — для  $R_4$ . Решив задачу линейного программирования с булевыми переменными, определим, что на первом этапе могут быть выполнены задания  $z_2, z_4, z_5, z_6$ .

2. Запишем функционал (6) с учетом результатов первого этапа:

$$F = z_1 + z_3 + z_7 \rightarrow \max.$$

Ограничения (7) примут вид  $z_1 + z_3 \leq 1, z_1 + z_7 \leq 1, z_3 + z_7 \leq 1$ . Решив данную задачу, определим, что необходимо обслужить задание  $z_3$ .

3. Аналогично определяем, что на третьем этапе необходимо обслужить задание  $z_1$ .

4. На четвертом этапе необходимо обслужить задание  $z_7$ .

Следовательно, в данном случае РО будет выполнено в четыре этапа.

**Метод групповой выборки с индивидуальной сегментацией** — это такой метод, при реализации которого задания, находящиеся в очереди, разбиваются на подзадачи. Из очереди полученных подзадач выбирают те, для реализации которых требуются ресурсы различного типа и сумма приоритетов которых максимальна. В случае наличия равнозначных подзапросов выбирают более «старые».

Подзадача — это часть задания, для реализации которой требуется использование ресурса конкретного типа. Если задание требует  $K$  различных ресурсов, то его разбивают на  $K$  подзадач. В этом случае надо выбрать из очереди как можно большее число подзадач, обращающихся к различным ресурсам, сумма приоритетов которых была бы максимальна. Стремление к максимуму суммы приоритетов выбранных 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  $Z_{kg}$ ;  $\{X\}$  — множество всех вариантов выбора подзадач из очереди;  $X$  — один из вариантов выбора подзадач,  $X = \{x_{11}, x_{12}, \dots, x_{kg}, \dots, x_s\}$ , где  $k = \overline{1, \dots, p}$ ,  $g = \overline{1, \dots, M}$ ;  $p$  — число заданий в очереди;  $x_{kg}$  — булева переменная, равная единице, если выбрана соответствующая подзадача  $Z_{kg}$ , и равная нулю, если нет. Тогда функционал (6) примет вид

$$F(x) = \sum_{j=1}^{p_1} C_{1j} S_1(C_n^1) + \sum_{j=1}^{p_2} C_{2j} S_2(C_n^2) + \dots \\ \dots + \sum_{j=1}^{p_k} C_{kj} S_k(C_n^k) + \dots + \sum_{j=1}^{p_n} C_{nj} S_n(C_n^n) \rightarrow \max, \quad (11)$$

где  $S_r(C_n^r) = S_1 + S_2 + \dots + S_{p_r}$  — сумма всех возможных сочетаний произведений переменных, в каждом из которых содержится  $S_r = X_p X_k, \dots, X_m$  различных переменных;  $p_r = \frac{n!}{r!(n-r)!}$ ;  $C_{ij}$  — коэффициенты в произведениях  $S_r$ , содержащих  $r$  переменных. Ограничения (7) примут вид

$$\sum_{k=1}^p A_{kg} x_{kg} \leq B_g, \quad g = \overline{1, \dots, M}. \quad (12)$$

Получена задача нелинейного программирования с булевыми переменными. Для рассмотренного выше примера функционал (11) примет вид

$$F = Z_{11} + Z_{13} + 3Z_2 + 2 Z_{31} + 2 Z_{34} + 2 Z_4 + 4 Z_5 + Z_6 + Z_{73} + Z_{74} + \\ + Z_{11} Z_{13} + 2 Z_{31} Z_{34} + Z_{73} + Z_{74} \rightarrow \max,$$

а ограничения (7) — следующий вид:

$$Z_{11} + Z_{31} + Z_5 \leq 1; \quad Z_{13} + Z_{73} \leq 1; \quad Z_{34} + Z_4 + Z_{74} \leq 1.$$

При использовании метода групповой выборки с индивидуальной сегментацией данное РО будет выполнено в три этапа.

Из рассмотренного примера можно сделать следующие выводы.

1. Использование метода групповой выборки с индивидуальной сегментацией позволяет сократить число этапов РО и уменьшить частоту отказов в обслуживании заданий на входе системы при пиковой нагрузке.

2. Для обеспечения эффективности предложенного метода необходимо использовать в качестве математического аппарата методы с малой временной сложностью для решения задач линейного и нелинейного программирования с булевыми переменными.

Таким образом, в качестве метода планирования предлагается использовать ранговый алгоритм решения задач нелинейного булевого программирования [2].

**Модель кластера Grid системы.** Для моделирования работы кластера Grid системы (см. рис. 2) формируется множество задач, которые будут поступать в систему в течение всего процесса моделирования. Устанавливаются параметры системы (длина пула входных задач, частота планирования, время задержки и др.) и задают метод планирования (предусмотрено несколько методов решения ЗНП (МС, FCFS)) и метод, основанный на решении задачи нелинейного булевого программирования (NLP).

Процесс моделирования работы кластера состоит из пошагового выполнения трех операций:

- 1) имитация поступления задач на вход системы; из общего числа задач в пул загружаются первые  $mp$  задач;
- 2) распределение задач из очереди пула между ресурсами и возвращение непоместившихся задач обратно в пул;
- 3) имитация решения задач ресурсами.

Имитацией решения задачи является вычитание из числа тактов, определяющих сложность решения задачи, числа тактов, определяющих производительность ресурса, на котором задача решается. Данная операция принята за условную единицу времени — такт. Все временные процессы при моделировании работы кластера измеряются в тактах.

Блок-схема алгоритма моделирования процесса работы типового кластера Grid системы, представлена на рис. 3.

А л г о р и т м моделирования работы кластера следующий:

1. Выполнение операции 1: на вход системы подается установленное число задач.
2. Выполнение операции 2: формируется пакет заданий установленным методом.
3. Выполнение операции 3: имитируется процесс решения для всех задач, которые на текущий момент времени находятся в ресурсе; выполняется такое число тактов, которое было использовано для формирования пакета заданий.
4. Выполнение операции 4: имитируется процесс решения задач, пока не наступает следующий период планирования, или пока не будут решены все задачи.
5. Переход к выполнению шага 1.

Представленная модель включает в себя следующие элементы:

- 1) множество задач — набор задач, подаваемых на Grid систему для моделирования процесса ее работы;

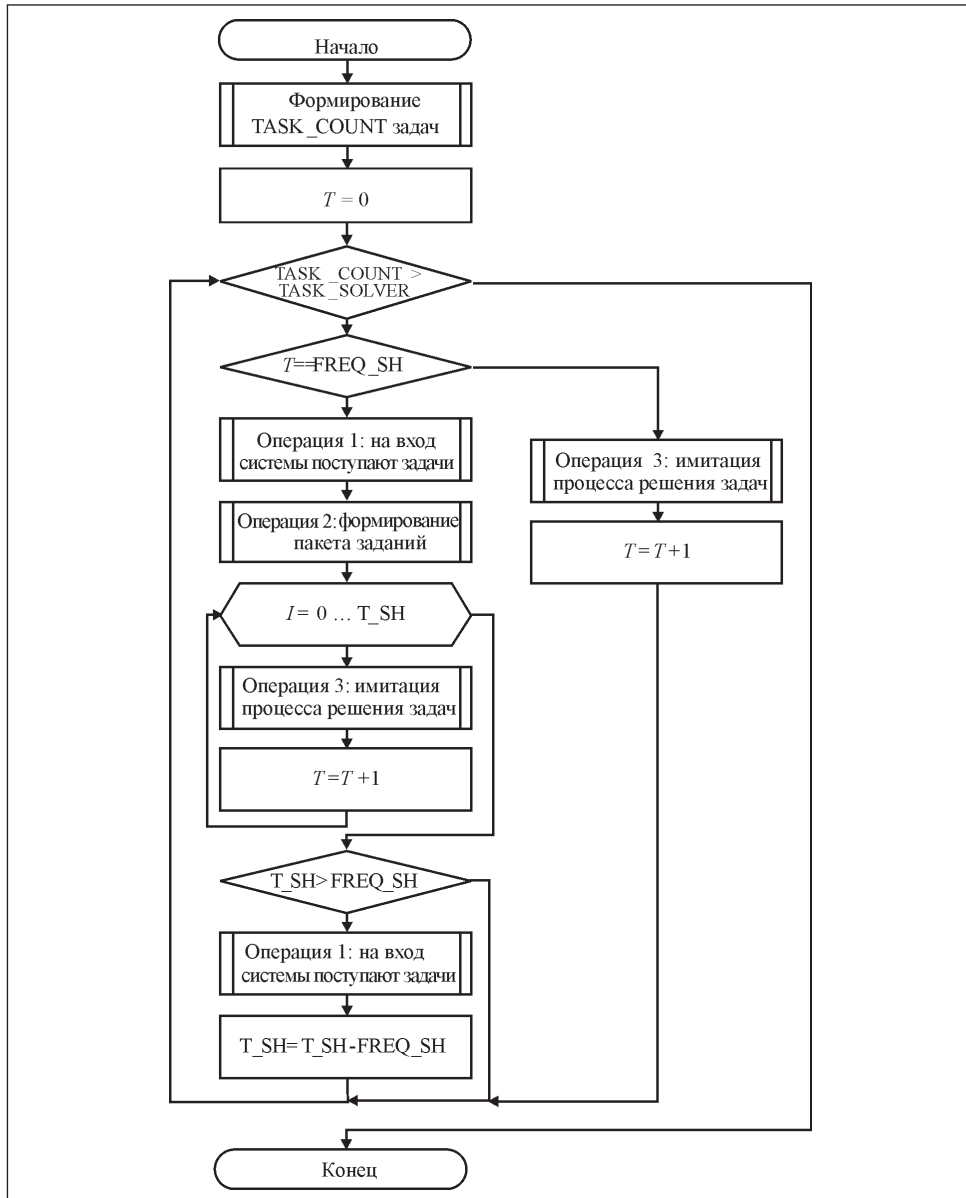


Рис. 3. Алгоритм моделирования процесса работы типовой кластера Grid системы: TASK\_COUNT — число задач, подаваемых на вход системы в течение всего времени моделирования процесса работы кластера; TASK\_SOLVER — число решенных задач; T — такт работы системы; T\_SH — время, потраченное на формирование пакета заданий; FREQ\_SH — период формирования пакета заданий (период планирования); выходы, показанные стрелкой, соответствуют логическому «да», а выходы без стрелок — логическому «нет»

2) множество ресурсов — набор ресурсов, выполняющих решения множества задач, поступивших на Grid систему;

3) параметры моделирования — набор параметров, определяющих режим работы и характеристики модели Grid системы;

4) методы планирования — набор методов, используемых Grid системой для распределения входящих вызовов между ресурсами.

**Множество задач и их параметры.** Под термином «задача» подразумевается структура с набором следующих характеристик: универсальность; типы ресурсов, на которых задача может быть решена; сложность решения; приоритет. Характеристики задач определяются параметрами множества. В модели процесса работы Grid системы необходимо установить такие параметры множества задач:

число задач — общее число задач в множестве;

число типов ресурсов — максимальное число типов ресурсов, на которых могут быть решены задачи (для совместимости число типов ресурсов в множестве задач должно совпадать с одноименным параметром в множестве ресурсов, на которых планируется решать данное множество задач);

универсальность — максимальное значение универсальности задач во всем множестве (определяет число типов ресурсов, на которых может быть решена задача);

закон распределения универсальности — случайный закон, по которому задачам будет назначена универсальность (максимально возможное значение определяется параметром «универсальность», минимальное значение равно единице);

число уникальных задач — процент задач (от 0 до 100 %), которые могут быть решены только одним ресурсом, относительно общего числа задач;

сложность решения — максимальное значение сложности решения задач во всем множестве (определяется числом тактов, выполненных для решения задачи ресурсом с производительностью один такт);

закон распределения сложности решения — случайный закон, по которому задачам назначается сложность решения (максимально возможное значение определяется параметром «сложность решения», минимальное значение — 1 такт);

суммарная сложность решения — сумма сложностей решения всех задач множества (параметр не задается пользователем, а подсчитывается при создании 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 будет назначен приоритет.

В модели предусмотрена возможность изменения всех параметров по различным случайным законам: равномерному, нормальному, Эрланга, экспоненциальному. Предусмотрена также возможность разбиения исходных задач на подзадачи, при этом суммарная сложность решения задач остается прежней, увеличивается только их число. Ресурс в модели является структурой с набором следующих характеристик: тип, производительность. В модели есть возможность задавать следующие параметры для множества ресурсов: число ресурсов — общее число ресурсов в множестве; число типов ресурсов — максимальное число типов ресурсов.

Производительность ресурса за один такт его работы характеризуется реализацией текущего числа тактов задачи, определяющих ее сложность. Имеется возможность задавать закон распределения производительности, по которому ресурсам будет назначена производительность. Кроме того, в модели автоматически рассчитывается суммарная производительность (сумма производительностей всех ресурсов множества). Размер буферов всех ресурсов множества определяется числом задач, которые могут быть поставлены в очередь к ресурсу на решение. В модели предусмотрено два типа буфера: последовательный — задачи последовательно решаются ресурсом; параллельный — все задачи в буфере решаются ресурсом одновременно.

Набор параметров, определяющих режим работы и характеристики модели Grid системы, следующий:

задачи — множество задач, поступивших в систему в процессе моделирования;

ресурсы — множество ресурсов системы, на которых будут решаться задачи при моделировании;

интенсивность — максимальное число задач, которое поступит на вход системы при очередном такте планирования;

закон распределения интенсивности — случайный закон, по которому задачи будут распределены для последовательного поступления на вход системы (максимально возможное значение определяется параметром «интенсивность», минимальное значение — 0);

отчужденность — максимальное число вышедших из строя ресурсов на очередном такте планирования (от 0 до 100% относительно числа ресурсов);

закон распределения отчужденности — случайный закон, по которому распределяется отчужденность для каждого такта планирования (максимальное значение определяется параметром «отчужденность», минимальное — 0);

длина пула — максимальное число задач, которые можно поместить в систему для дальнейшей обработки;

задержка — число тактов, которое затрачивается системой на передачу задачи из буфера в ресурс на решение;

частота планирования — число тактов между планированиями, а также число тактов между поступлениями новых задач на вход системы;

коэффициент планирования — коэффициент уменьшения числа операций, выполненных планировщиком для перевода операций, затраченных на планирование в затраченное время, в системе, выражаемое в тактах.

**Методы планирования** — это набор методов, которыми Grid система распределяет входящие задачи между ресурсами. В данной модели есть возможность проводить исследования на основании следующих методов планирования выполнения заданий.

*Метод МС* основан на поиске наименьшего числа ресурсов, которыми можно решить все задачи, т.е. ЗНП. Для поиска наименьшего числа ресурсов используется приближенный частотный метод [2, 3].

А л г о р и т м работы *МС*:

1. Определение соответствия между задачами и ресурсами, на которых они могут решаться.

2. Решение ЗНП частотным методом.

3. Помещение задач в буферы ресурсов, вошедших в минимальное покрытие.

4. Если поместились все задачи, процедура планирования заканчивается, иначе — задачи возвращаются в пул и происходит переход к операции 2.

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

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

А л г о р и т м работы *FCFS*:

1. Определение соответствия между задачами и ресурсами, на которых они могут решаться.

2. Первая по порядку задача из пула помещается в первый свободный ресурс, которым она может быть решена; если таких ресурсов не оказалось, задача возвращается в пул.

3. Процедура планирования прекращается после обработки последней задачи в пуле. В данном методе не используется буфер, задачи сразу посылаются на решение ресурсу, если он свободен.

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

А л г о р и т м работы *NLP*:

1. Определение соответствия между задачами и ресурсами, на которых они могут решаться.

2. Решение задачи нелинейного программирования.

3. Размещение задач в буферы ресурсов; если остались задачи, для которых есть свободные ресурсы, происходит переход к операции 2, если таких задач нет, процедура планирования заканчивается. В процессе планирования задачи, для решения которых не осталось свободных ресурсов, помещаются обратно в пул.

**Исследуемые характеристики работы Grid системы.** В процессе работы Grid системы в представленной модели были исследованы такие характеристики:

- Время выполнения — число тактов работы системы, за которое было решено все множество задач, поступивших на вход Grid системы:  $t_{\text{вх}} = t_e - t_b$ , где  $t_b$  — такт поступления первой задачи в пул системы;  $t_e$  — такт решения системой последней задачи.

- Время ответа — среднее число тактов, за которое была решена одна задача. в режиме моделирования Grid системы:

$$t_r = \frac{1}{m} \sum_{i=1}^m t_{ri},$$

где  $m$  — число решенных системой задач;  $t_{ri}$  — число тактов  $i$ -й задачи с момента попадания ее в пул системы до момента решения системой.

- Максимальное время ответа — максимальное число тактов, за которое была решена одна задача:

$$t_{r \max} = \max_{0 < i \leq m} t_{ri}.$$

- Время ожидания — среднее число тактов, которое потребовалось для ожидания задачам на входе системы до последующей обработки:

$$t_w = \frac{1}{m} \sum_{i=1}^m t_{wi},$$

где  $t_{wi}$  — число тактов  $i$ -й задачи с момента попадания в пул системы до момента назначения ресурса, который будет ее решать, и отправки в буфер назначенного ресурса (время нахождения задачи в пуле системы).

- Время обслуживания — среднее число тактов, в продолжение которых задачи находились в буфере назначенного ресурса до начала их решения данным ресурсом:

$$t_s = \frac{1}{m} \sum_{i=1}^m t_{si},$$



где  $t_{si}$  — число тактов  $i$ -й задачи с момента попадания в буфер назначенного ресурса до начала ее решения данным ресурсом (время нахождения задачи в буфере ресурса).

- Время планирования — среднее число тактов, использованных системой на планирование:

$$t_p = \frac{1}{p} \sum_{i=1}^m t_{pi},$$

где  $p$  — число планирований, выполненных системой за время решения всех задач;  $t_{pi}$  — число тактов, использованных системой на  $i$ -е 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:

$$r_u = \sqrt[n]{\prod_{i=1}^n r_{ui}}, \quad r_{ui} = \frac{\sum_{j=1}^{m_{si}} \text{Pr}_j}{\sum_{j=1}^{m_i} \text{Pr}_j},$$

где  $n$  — число ресурсов в системе;  $r_{ui}$  — коэффициент использования  $i$ -го ресурса;  $m_{si}$  — число задач, решенных  $i$ -м ресурсом;  $m_i$  — число задач из общего множества входных задач, которые могли быть решены  $i$ -м ресурсом;  $\text{Pr}_j$  — приоритет  $j$ -й задачи.

- Коэффициент загрузки — среднее геометрическое значение коэффициентов загрузки всех ресурсов. Коэффициент производительности ресурса — отношение числа задач, решенных этим ресурсом, к числу всех задач из входного множества, которые могли быть решены данным ресурсом:

$$r_l = \sqrt[n]{\prod_{i=1}^n r_{li}}, \quad r_{li} = \frac{m_{si}}{m_i},$$

где  $r_{li}$  — коэффициент использования  $i$ -го ресурса.

- Коэффициент важности — среднее геометрическое отношения суммарного приоритета задач, помещенных в буферы в результате планирования, к суммарному приоритету задач, находившихся в пуле:

$$r_i = \sqrt[p]{\prod_{j=1}^p r_{ij}}, \quad r_{ij} = \frac{\sum_{k=1}^{m_{pj}} \text{Pr}_k}{\sum_{k=1}^{m_{wj}} \text{Pr}_k},$$

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

а

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

б

Параметры моделирования работы GRID системы		
Задачи	Значение	Методы
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 GREEDY
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	

в

Рис. 4. Параметры множества задач (а), множества ресурсов (б) и параметры моделирования (в)

где  $r_{ij}$  — отношение приоритета задач, помещенных в буфер ресурсов в результате планирования, к приоритету задач, находившихся в пуле системы на  $j$ -м планировании;  $m_{pj}$  — число задач, распланированных между ресурсами (помещенных в буфер ресурсов) при  $j$ -м планировании;  $m_{wj}$  — число задач, находившихся в пуле при  $j$ -м планировании;  $P\Gamma_k$  — приоритет  $k$ -й задачи.

- Коэффициент сохранения важности — среднее геометрическое отношения суммарного приоритета задач, помещенных в буферы в результате планирования после отказа случайного числа ресурсов, к суммарному

приоритету задач, помещенных в буферы в результате планирования до отказа ресурсов:

$$r_s = \sqrt[p]{\prod_{j=1}^p r_{sj}}, \quad r_{si} = \frac{\sum_{k=1}^{m_{fi}} \text{Pr}_k}{\sum_{k=1}^{m_{pi}} \text{Pr}_k},$$

где  $r_{si}$  — отношение приоритета задач, помещенных в буферы ресурсов после планирования при случайном отказе ресурсов, к приоритету задач, помещенных в буферы ресурсов после планирования до отказа ресурсов на  $i$ -м планировании;  $m_{pj}$  — число задач, распланированных между ресурсами (помещенных в буфер ресурсов) до отказа случайного числа ресурсов на  $j$ -м планировании;  $m_{fi}$  — число задач, распланированных между ресурсами (помещенных в буфер ресурсов) после отказа случайного числа ресурсов на  $j$ -м планировании.

- Коэффициент ускорения — отношение расчетного времени выполнения задач, решаемых последовательно одним ресурсом с производительностью, равной средней производительности всех ресурсов Grid, к времени выполнения этих же задач Grid системой:  $r_a = t_{o,p} / t_b$ ,  $t_{o,p} = S / \bar{P}$ , где  $t_{o,p}$  — расчетное время последовательного решения задач одним ресурсом с производительностью, равной средней производительности ресурсов Grid;  $t_b$  — время выполнения задач Grid системой;  $S$  — суммарная

#### Результаты моделирования работы Grid системы при заданных параметрах

Параметр	Показатель качества планирования методом		
	MC	FCFS	NLP
Время			
выполнения	16 157	18 762	14 206
ожидания	36	196	5
Коэффициент			
важности	0,626	0,188	0,999
ускорения	1,226	1,056	1,395
сохранения важности	0,781	0,763	0,868
загрузки	0,384	0,334	0,453
использования	0,38	0,334	0,454
Максимальное время ответа	5347	2075	4168
Время обслуживания	2095	50	833

сложность решения всех задач;  $\bar{P}$  — средняя производительность всех ресурсов Grid.

Исследуемые характеристики выводятся на графики и в таблицу, которые находятся во вкладках «Временные характеристики» и «Коэффициенты». Результаты моделирования, полученные при заданных значениях и параметрах моделирования приведены на рис. 4 и в таблице, где время задано числом тактов.

### Выводы

Как видно из результатов моделирования, использование в качестве алгоритма планирования предложенного в [2] метода решения задачи нелинейного булевого программирования позволяет существенно увеличить такие показатели, 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. — Харьков: «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. — Харьков: ИНЖЭК, 2008. — 407 с.
3. Листровой С.В., Минухин С.В. Подход и модель планирования распределения ресурсов в Grid // Проблемы управления и информатика. — 2012. — № 5. — С. 65—82.

S.V. Listrovoy, E.S. Listrovaya, M.S. Kurtsev

### METHODS AND MODELS OF SCHEDULING THE ASSIGNMENT OF PACKAGES OF TASKS IN THE GRID SYSTEM CLUSTER

The simulation model of the Grid-system work is presented, which allows comparing the existing methods of performing job scheduling. The mathematical description of the test in the model of characteristics is given. An example of solving the problem based on cluster sampling method is presented. Experimental results confirm the advantages of performing scheduling based on the solution of the Boolean linear programming problems.

*Key words:* Grid, planning, resource, cluster, batch processing, the sample group.

### REFERENCES

1. Minukhin, S.V. (2014), *Modeli i metody resheniya zadach planirovaniya v raspredelyonnykh vychislitelnykh sistemakh, Monographiya* [Models and methods for solving scheduling problems in distributed computing systems, Monograph], Izdatelstvo OOO “Shchedraya usadba plus”, Kharkov, Ukraine.

2. Ponomarenko, V.S., Listrovoy, S.V. and Minuckhin, S.V. (2008), *Metody i modeli planirovaniya resursov v geterogennykh GRID-sistemakh, Monografiya* [Methods and models of resource scheduling in GRID-systems, Monograph], Izdatelskiy dom «INZhEK», Kharkov, Ukraine.
3. Listrovoy, S.V. and Minukhin, S.V. (2012), “Model and approach to the resource allocation planning in heterogeneous GRID-systems”, *Problems of control and informatics*, no. 5, pp. 120-133.

Поступила 05.04.16;  
после доработки 20.10.16

*ЛИСТРОВОЙ Сергей Владимирович, д-р техн. наук, профессор Украинского государственного университета железнодорожного транспорта (г. Харьков). В 1972 г. окончил Харьковское высшее военное командно-инженерное училище. Область научных исследований — задачи дискретной оптимизации и теории графов и их приложения к анализу вычислительных систем и сетей.*

*ЛИСТРОВАЯ Елена Сергеевна, канд. техн. наук, доцент кафедры экономики и маркетинга Национального аэрокосмического университета им. Н.Е. Жуковского (г. Харьков), который окончила в 1998 г. Область научных исследований — применение информационных систем в экономической сфере деятельности.*

*КУРЦЕВ Максим Сергеевич, аспирант Украинского государственного университета железнодорожного транспорта (г. Харьков). В 2010 г. окончил Украинскую государственную академию железнодорожного транспорта. Область научных исследований — задачи управления и планирования в распределенных вычислительных системах.*

