

УДК 681.51

*С.В. Минухин, А.В. Коровин*

## РАЗРАБОТКА, РЕАЛИЗАЦИЯ И ОЦЕНКА ЭФФЕКТИВНОСТИ МЕТОДОВ ПЛАНИРОВАНИЯ РЕСУРСОВ GRID В ПАКЕТЕ GRIDSIM

Рассмотрены задачи моделирования процессов планирования ресурсов в пакете GridSim. Предложены модификации метода FCFS и эффективный метод планирования ресурсов, использующий подход на основе решения задачи о наименьшем покрытии. Разработаны алгоритмы и программное обеспечение на языке Java. Проведены моделирование в пакете GridSim и сравнительный анализ полученных результатов на основе метрик эффективности, обосновывающий эффективность предложенных в работе методов планирования.

### Введение

Современные технологии создания информационных систем и расширение возможностей доступа к общим вычислительным ресурсам многих пользователей поэтапно привели к появлению двух современных концепций высокопроизводительных вычислений – созданию технологий GRID и параллельных архитектур.

Исследования в области GRID [1] представляют собой развитие современной парадигмы вычислений – метакомпьютинга, использующей ресурсы, принадлежащие различным географически распределенным организациям. Одним из видов ресурсов, используемых в GRID, являются многопроцессорные системы (кластеры), которые могут состоять из сотен (тысяч) процессоров. Суперкомпьютеры (и кластеры как подкласс) используются для высокопроизводительных вычислений при решении вычислительно трудоемких задач: моделировании климата, финансовых расчетов, молекулярной биологии, моделировании быстропротекающих физических процессов и т. д. При этом требуется не только найти подходящий ресурс для назначения на него задания, но и построить оптимальное расписание порядка выполнения заданий на каждом из ресурсов. Кроме составления расписания, имеется еще ряд задач, которые требуют их решения системой управления выполнением заданий – подписка пользователя на выпол-

нение задания, предоставление удаленного доступа к вычислительным ресурсам, администрирование и мониторинг состояния узлов кластеров, доставка результатов выполненного задания пользователю. При этом оптимизация стратегий управления распределенной средой на непосредственно функционирующей GRID-инфраструктуре связано с многочисленными проблемами – масштабируемостью, синхронизацией процессов планирования, априори неизвестной динамикой потоков заданий, добавлением новых ресурсов и т. д. Поэтому представляется актуальным проведение исследований, моделирующих GRID-среду. Выделим следующие направления моделирования GRID: создание имитационных моделей крупномасштабных систем [2], разработка моделей GRID для уже существующих систем [3], разработка специализированных пакетов – GridSim, SimGrid, OptorSim, Bricks, наиболее полные обзоры функциональности и специализации которых приведены в работах [4, 5].

Важным фактором, влияющим на адекватность результатов моделирования и их использование в проектах GRID, является проведение экспериментов на реальных данных. Они отображают практические возможности моделирования, позволяют определить границы применимости моделируемых методов для различных параметров GRID-среды. Результаты моде-

лирования планирования ресурсов реального GRID с помощью GridSim в работе [6] показали, что в случае невысокой интенсивности потока заданий и большого количества ресурсов применение распространённого метода FCFS не позволяет равномерно распределять вычислительные ресурсы для решения заданий в GRID-системе. Учитывая это, представляется актуальным разработка методов планирования ресурсов и их валидация средствами симулятора среды GRID.

Цель данной работы – разработка эффективных методов планирования ресурсов GRID на основе модификаций метода FCFS и новых встраиваемых методов планирования ресурсов средствами расширения и реконфигурации пакета GridSim и оценка их эффективности.

### Принципы работы и возможности пакета GridSim

Данная платформа позволяет пользователям моделировать работу GRID-системы с возможностью симулирования характеристик ресурсов и вычислительных сетей для ее различных конфигураций (последняя версия GridSim размещена по адресу <http://www.gridbus.org/gridsim/>, лицензия GPL). Она предоставляет возможность моделировать поведение пользователей GRID-системы, вычислительных ресурсов и брокеров ресурсов (планировщиков) в средах распределённых вычислений. С помощью GridSim можно проводить воспроизводимые эксперименты, которые невозможны в реальных GRID-системах.

К основным возможностям пакета GridSim относятся:

- моделирование различных характеристик ресурсов GRID;
- моделирование различных политик (стратегий) планирования выполняемых заданий на ресурсах (кластерах), как уже в нем реализованных (FCFS, Easy Backfill, Conservative Backfill), так и разработанных пользователями методов алгоритмов;
- использование данных о загрузке реальных кластеров GRID для проведения экспериментов (тестирование планиров-

щиков) на реальных данных;

- моделирование параллельного выполнения заданий;
- поддержка механизма аукциона (экономических моделей) для планирования заданий и ресурсов;
- моделирование различных конфигураций вычислительной сети GRID-системы с различными топологиями;
- моделирование региональных компонентов GRID информационных сервисов.

GridSim проектировался как многоуровневая система моделирования [7, 8], возможности которой могут быть достаточно легко и значительно расширены. GridSim основан на SimJava [9] – пакете для симулирования дискретно-событийных процессов, реализованном на языке Java. Для реализации собственных сценариев работы и конфигурации системы (extensions) для тестирования собственных политик (стратегий) и алгоритмов пользователями используется верхний уровень симулятора.

*Визуализация.* Начиная с версии 5.0, в GridSim используется пакет parallel.gui, в котором содержится интерфейсы Visualizer и AbstractVisualizer для визуализации процесса моделирования. Согласно официальной документации GridSim [10] эта реализация может использоваться в отладочных целях, например, для пошаговой проверки новых политик (стратегий) или алгоритмов планирования. Для повышения эффективности отображения и визуализации результатов планирования в GridSim используется Alea 2 [11]. Он позволяет наблюдать изменение состояния моделирования GRID-инфраструктуры непосредственно во время проведения эксперимента, т. е. в реальном времени, и отображать результаты в удобной форме – в виде гистограмм, графиков и спектральных диаграмм, отображающих процессы планирования загрузки кластеров.

*Анализ* проведения модельного эксперимента можно осуществить по метрикам:

- количество процессорных элементов – требуемых (по заданиям), используемых и доступных;

– загрузка кластеров, в процентном соотношении по отношению к общему количеству кластеров в GRID-системе, за каждый час, день;

– количество заданий, которые ждут выполнения, и выполняющиеся в данный момент времени;

– среднее использование (загруженность) процессорных элементов кластеров за каждый час.

*Хранение информации.* Все результаты экспериментов, помимо отображения на графиках, сохраняются в текстовые файлы в формате CSV. После проведения экспериментов эти данные можно загружать в программные продукты, которые позволяют проводить их последующий статистический анализ. Данные, сохраненные в формате CSV, могут быть преобразованы в другие форматы: XLS (Microsoft Excel), XML, STA (Statistica) для использования в расчетах, построения графиков и т. д. Эксперименты можно проводить на записях загрузки реальных GRID-инфраструктур в следующих форматах: Grid Workloads Format (GWF) и Standart Workloads Format (SWF).

*Планирование.* В Alea 2 реализован *централизованный планировщик*, который использует известные алгоритмы планирования – FCFS, Easy Backfill, Earliest Suitable Gap и другие [12–15]. Планировщик способен обрабатывать ситуацию, когда задания поступают *динамически во время проведения симуляции*. В этом случае генерируемые графики будут менять свой вид, отображая текущее состояние экспериментальной инфраструктуры GRID. Архитектура Alea 2 позволяет проводить эксперименты на одних и тех же исходных данных инфраструктуры системы – поступающих заданиях и доступных ресурсах – с использованием различных методов планирования. Таким образом, рассмотренные возможности GridSim позволяют разрабатывать достаточно полные с точки зрения анализа и адекватные модели

планирования ресурсов и проверять их эффективность на реальных данных.

## **Моделирование методов планирования загрузки вычислительных кластеров GRID**

Выбор метода (политики, стратегии) и алгоритма планирования выполнения заданий на вычислительном кластере влияет на загрузку ресурсов кластера, время решения задач и, как результат, – на эффективность использования этих ресурсов. Существует ряд довольно простых алгоритмов планирования, которые позволяют выбрать задание для назначения на освободившийся ресурс из списка подготовленных для выполнения заданий [16]: выбор первого из очереди задания (First Come First Served, FCFS), задание с максимальным приоритетом (Highest Priority First, HPF), задания, которое выполнится за кратчайшее время и т. д. Метод RoundRobin [17] – упреждающий планировщик, который разработан специально для систем с разделением времени: планировщик выбирает для задания ресурс, на который указывает счетчик из списка, после чего значение счетчика увеличивается на 1. Основным преимуществом метода RoundRobin по сравнению с другими (например, EDF, работа с наименьшим директивным сроком – первая, и алгоритмом Earliest Suitable Gap – первый приемлемый интервал времени) является то, что он не требует предварительной информации о времени выполнения работы (задания).

Алгоритм FB (feedback) очередей с обратной связью использует  $n$  очередей, каждая из которых обслуживается в порядке поступления. Новая работа поступает в очередь с номером 0, затем после получения кванта времени она переходит в очередь со следующим номером и т. д. после очередного кванта времени. Для повышения эффективности планирования ресурсов в данной работе предлагаются модификации метода FCFS.

В их основе лежат процедуры выбора ресурсов, описание которых приведе-

но в табл., псевдокоды реализации алгоритмов – на рис. 1 – 3.

Таблица. Модификации метода FCFS на основе процедур выбора ресурсов

№ п.п.	Название метода	Описание процедуры выбора ресурсов	Псевдокод
1	FCFS+Random	Перед планированием всех заданий из очереди изменяется порядок выбора доступных и свободных ресурсов по равномерному закону	Рис. 1
2	FCFS+Ring	Перед планированием всех заданий из очереди производится циклический сдвиг списка доступных и свободных ресурсов на 1	Рис. 2
3	FCFS+RoundRobin	Во время планирования – для каждого задания из очереди – производится циклический сдвиг списка доступных ресурсов (кластеров) на 1. В случае, если следующий выбранный ресурс занят, выбирается следующий по порядку и т. д.	Рис. 3

```

procedure FCFSRANDOM(queue, resources)
for i ← 0 to length|resources|
  do {
    left ← Random % length|resources|
    right ← Random % length|resources|
  }
  swap(resourcesleft, resourcesright)

for each r ∈ resources
  do {
    for each t ∈ queue
    do {
      if r accepts t
      then {assign t to r
  
```

Рис. 1. Псевдокод реализации алгоритма FCFS + Random

```

procedure FCFSRING(queue, resources)
  last ← resourceslength|resources|-1
  for i ← 1 to length|resources|
    do resourceslength|resources|-i ← resourceslength|resources|-i-1

  resources0 ← last

  for each r ∈ resources
    do {
      for each t ∈ queue
      do {
        if r accepts t
        then {assign t to r
  
```

Рис. 2. Псевдокод реализации алгоритма FCFS + Ring

```

procedure FCFSROUNDROBIN(queue, resources)
for each t ∈ queue
  do {
    for each r ∈ resources
    do {
      if r accepts t
      then {assign t to r
  }
  do {
    last ← resourceslength|resources|-1
    for i ← 1 to length|resources|
      do resourceslength|resources|-i ← resourceslength|resources|-i-1

    resources0 ← last
  }
  
```

Рис. 3. Псевдокод реализации алгоритма FCFS + RoundRobin

## Моделирование планирования ресурсов GRID на основе решения задачи о наименьшем покрытии

Рассмотрим задачу планирования заданий в GRID при следующих условиях. Предполагается наличие системы GRID с  $n$  ресурсами (кластерами), на которых необходимо решить  $m$  заданий. Каждое задание описывается набором требований к ресурсам, а ресурсы, в свою очередь, имеют характеристики, удовлетворяющие требованиям заданий. Математическая постановка задачи о наименьшем покрытии и эффективные ранговые методы ее решения приведены в монографии [1]. Ранговый метод, исследованный в [1], имеет временную сложность  $O(mn^3)$  и погрешность не более 5 %, что позволяет эффективно использовать его при планировании ресурсов.

Существенным отличием данного метода от представленных в Alea 2 является использование для распределения заданий промежуточного пула фиксированного размера, пакета заданий на каждый ресурс (кластер) и *иерархическая двухуровневая архитектура* GRID. Алгоритм планирования запускается при выполнении одного из двух условий: размер пула достиг установленного размера или прошло заданное время с момента последнего распределения заданий.

Задания, находящиеся в пуле, представляются в виде прямоугольной матрицы соответствий, в которой строки определяют задания, а столбцы – ресурсы, на которых эти задания могут выполняться. Если  $i$ -я задача может быть решена на  $j$ -ом ресурсе, то в ячейке  $ij$  матрицы стоит значение 1, если нет – 0. Метод находит наименьшее количество ресурсов (столбцов матрицы) которые могут решить все задания (строки матрицы), находящиеся в пуле. После этого задания распределяются на те ресурсы, которые вошли в наименьшее покрытие, и, если они загружены полностью, а задания в пуле еще остались, то они распределяются на другие свободные

ресурсы. Оставшиеся после этого нераспределенными задания возвращаются обратно в пул, после чего пул вновь заполняется заданиями из глобальной очереди и приведенная процедура опять повторяется. Псевдокод алгоритма Minimal Cover (MC), реализующего данный метод, показан на рис. 4.

```

procedure TASKPLANNINGSMC(pool, resources)
for each  $r \in resources$ 
  do { for each  $t \in pool$ 
    do { if  $r$  accepts  $t$ 
      then {  $vsetR_r \leftarrow t$ 
         $vsetT_t \leftarrow r$ 
      }
    }
  }

   $cover \leftarrow SEARCHCOVER(vsetR, vsetT)$ 

for each  $t \in vsetT$ 
  do { for each  $r \in cover$ 
    do { if  $vsetR_r$  accepts  $t$ 
      then assign  $t$  to  $vsetR_r$ 
    }
  }

procedure SEARCHCOVER(vsetR, vsetT)
for each  $r \in vsetR$ 
  do  $cols \leftarrow r$ 

for each  $t \in vsetT$ 
  do if  $cols_t > 0$ 
    then  $rows \leftarrow t$ 

 $coverIsFound \leftarrow false$ 
while true
  repeat
    for each  $r \in rows$ 
      do { if  $length(r) == 1$ 
        then {  $cover \leftarrow r$ 
           $rows \leftarrow r \times rows$ 
          if  $length(rows) == 0$ 
            then return ( $cover$ )
          }
        }
      until  $coverIsFound$ 

    repeat
       $coverIsFound \leftarrow false$ 
      for each  $c \in cols$ 
        do { if  $length(c) == 1$ 
          then {  $cover \leftarrow c$ 
             $cols \leftarrow c \times cols$ 
            if  $length(cols) == 0$ 
              then return ( $cover$ )
            }
          }
        else { erase  $rows_c$ 
          }
        until  $coverIsFound$ 
      }
      if  $length(rows) == 0$ 
        then return ( $cover$ )
       $maxCover \leftarrow Max(rows, vsetR)$ 
       $cover \leftarrow maxCover$ 
       $rows \leftarrow maxCover \times rows$ 
    }

```

Рис. 4. Псевдокод алгоритма Minimal Cover, основанного на решении задачи о наименьшем покрытии

Для сравнения показателей работы предложенных модификаций FCFS (см. табл.) и данного метода были выбраны

предоставляемые системой Alea 2 [11] метод FCFS и метод Earliest Suitable Gap (ESG), показавший лучший результат при моделировании на реальных данных [6].

*Процедура расширения.* После реализации алгоритма Minimal Cover на языке Java он был интегрирован в пакет Alea 2 с использованием возможностей пакета.

1. *Рефакторинг кода* [18]. Целочисленные константы, соответствующие алгоритму планирования, заменены на перечисления; конкретные имена файлов с исходными данными изъяты из кода и указаны в качестве переменных окружения виртуальной машины Java [19].

2. *Добавление* новых алгоритмов планирования реализовано на основе интерфейса DispatchingAlgorithm с единственным методом dispatch, который принимает два параметра: список заданий и список доступных ресурсов.

3. *Для возможности разработки алгоритмов планирования независимо от среды моделирования* разработаны интерфейсы Task и Resource, представляющие собой абстракции задания и ресурса, соответственно. В пакете impl находятся реализации следующих интерфейсов:

- MinimalCover – реализация алгоритма планирования по методу нахождения наименьшего покрытия;

- GridSimResource – реализация абстракции ресурса, транслирующей все вызовы классу ресурсов из GridSim;

- GridSimTask – реализация абстракции задачи, транслирующей все вызовы классу заданий из GridSim. Выбор алгоритма планирования изменен согласно методу рефакторинга «Замена условного оператора полиморфизмом (Replace Conditional with Polymorphism)» [18] и вынесен в класс AlgorithmFactory, реализованный согласно шаблону проектирования Фабрика [20].

### Анализ результатов моделирования и оценка эффективности разработанных методов

Для проведения сравнительного анализа модифицированных и нового ме-

тодов планирования ресурсов с уже реализованными выбраны следующие показатели эффективности работы системы: коэффициент использования ресурса (средняя загрузка ресурса за выбранный период моделирования) и коэффициент межкластерной загрузки.

В процессе моделирования в качестве исходных данных использованы реальные данные (по неделям) о загрузке французского Grid – AuverGrid [21]).

В приведенных далее результатах использованы следующие сокращения анализируемых методов: FCFS – First Come First Served; модифицированные FCFS (табл.), MC – Minimal Cover (метод решения задачи о наименьшем покрытии); ESG – Earliest Suitable Gap.

На рис. 5, а – г показаны результаты моделирования со следующими исходными данными: количество кластеров GRID – 20; количество процессорных элементов в каждом кластере – 10; размер промежуточного пула – 10; количество заданий – 10000; период планирования – 1-я, 2-я, 3-я, 4-я недели, соответственно.

Анализ графиков на рис. 5, а – г показал, что за все периоды планирования наиболее равномерно загружает ресурсы (кластеры) GRID метод наименьшего покрытия (MC), далее – предложенные в данной работе методы, использующие FCFS совместно с процедурами выбора ресурсов (см. табл.), а наименее эффективными являются методы ESG и FCFS.

Коэффициент межкластерной загрузки рассчитывался по формуле:

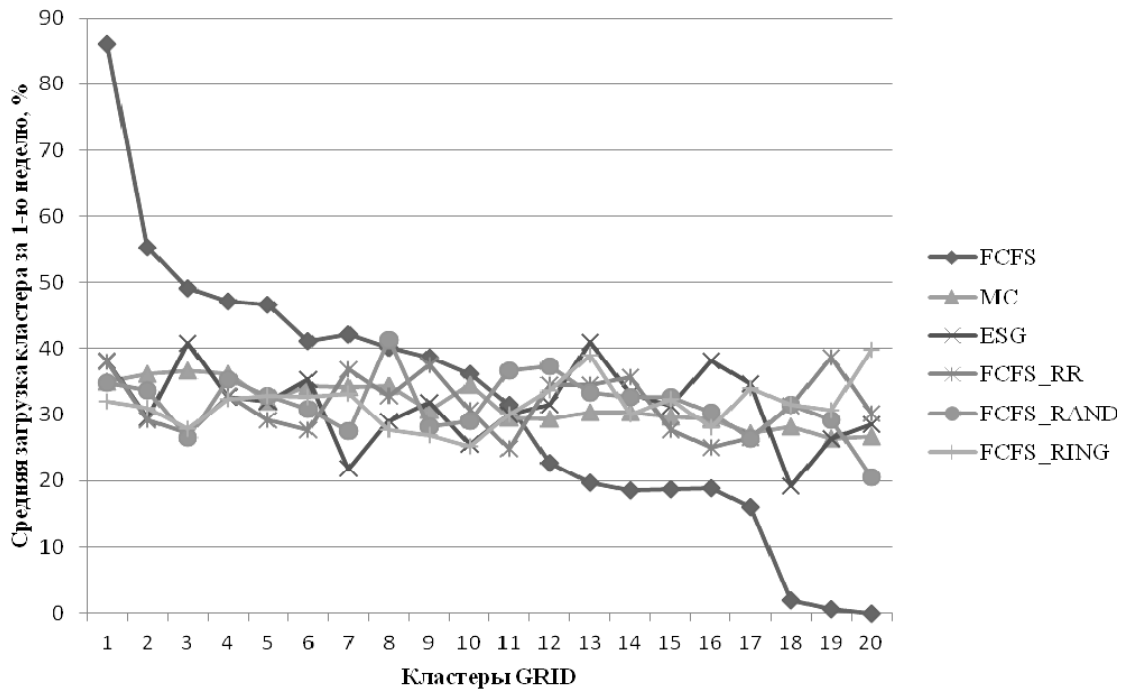
$$v_{cp} = \frac{\sum_{i=0}^{i=0} v_i}{n},$$

$$d = \sqrt{\frac{\sum_{i=0}^{i=0} (v_{cp} - v_i)^2}{n}}, \quad \beta = 1 - \frac{d}{v_{cp}},$$

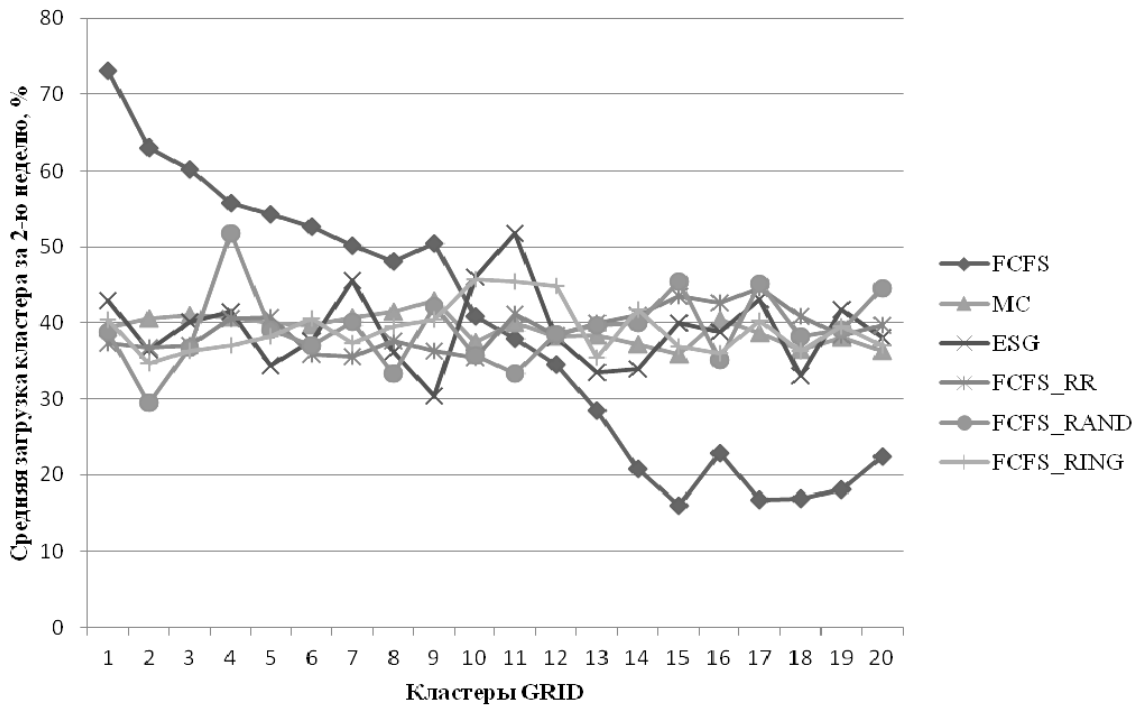
где  $n$  – общее количество ресурсов;  $\beta$  – коэффициент межкластерной загрузки;

$v_i$  – коефіцієнт використання ресурса (середня загрузка  $i$ -ого ресурса за період);  
 $v_{cp}$  – середня загрузка всіх ресурсів за пе-

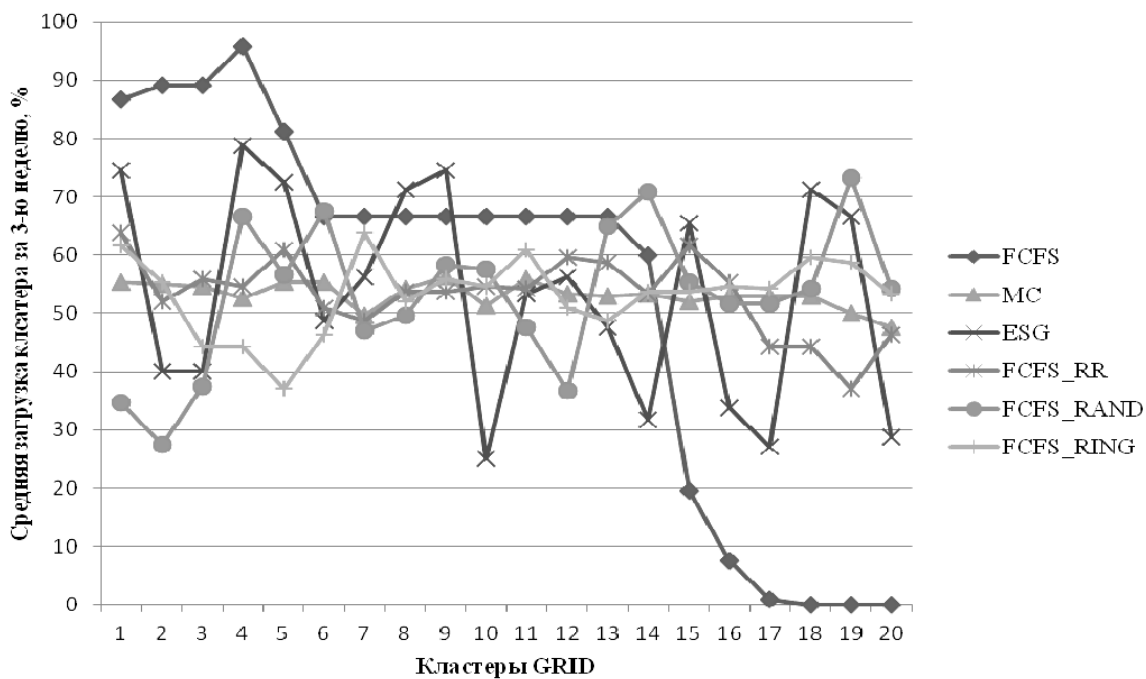
ріод;  $d$  – середньоквадратическое відхилення середньої загрузки кластерів за період.



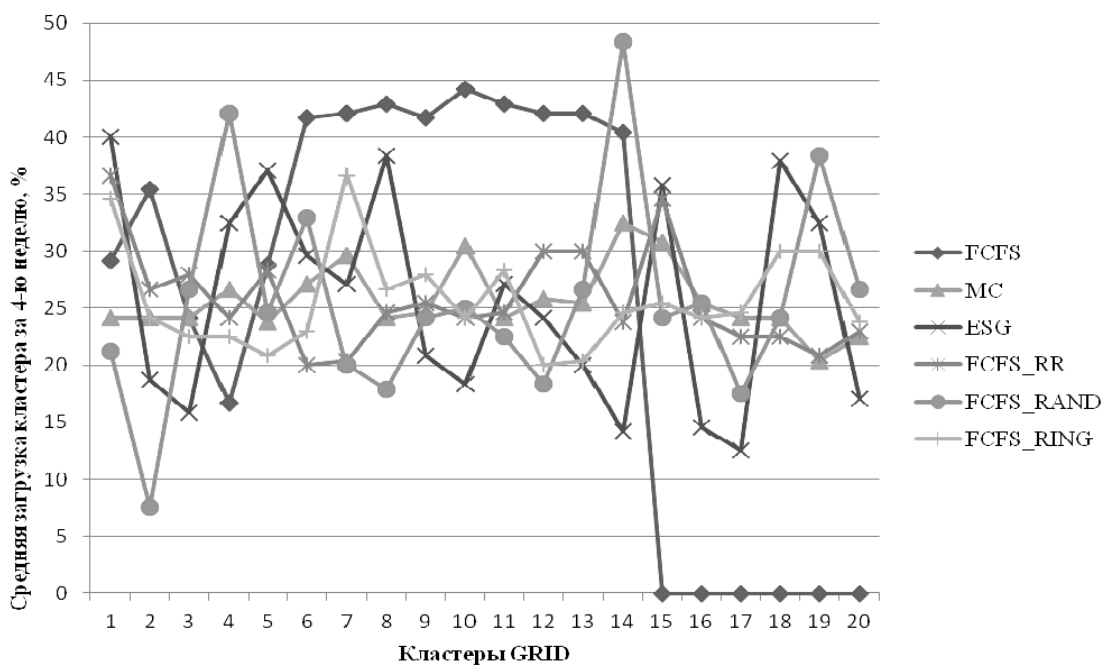
а



б



В



Г

Рис. 5. Результаты моделирования:  
 а – 1-я неделя; б – 2-я неделя;  
 в – 3-я неделя; г – 4-я неделя



На рис. 6 показаны результаты моделирования межкластерной балансировки загрузки кластеров для различных методов. Анализ графиков на рис. 6 показывает, что наиболее равномерно загружает ресурсы метод, использующий модель наименьшего покрытия, далее по эффективности следуют предложенные модификации метода FCFS, ESG и потом – FCFS.

### Заклучение

В работе предложены методы планирования ресурсов GRID-инфраструктуры, реализованные с помощью пакета моделирования GridSim, позволяющем анализировать поведение распределенных вычислительных систем при изменяющихся условиях. Разработаны модификации метода планирования ресурсов GRID FCFS и новый метод на основе решения задачи о наименьшем покрытии. Полученные на основе метрик эффективности планирования результаты компьютерного моделирования с помощью пакета GridSim на реальных данных загрузки кластеров GRID

показали, что использование предложенных методов дает преимущества в равномерности загрузки кластеров GRID по сравнению с такими методами планирования ресурсов GRID как FCFS и Earliest Suitable Gap.

В дальнейшем для исследования возможностей повышения эффективности планирования ресурсов предполагается провести модельные эксперименты на основе динамического изменения размеров пула и пакета заданий для метода о наименьшем покрытии в условиях изменения интенсивности потока заданий GRID.

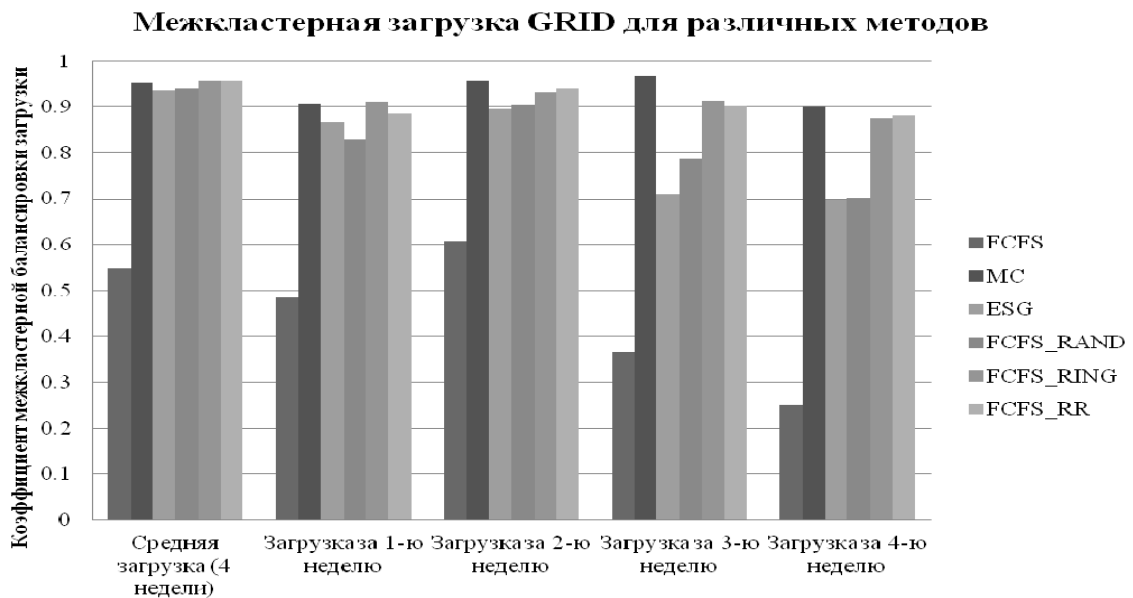


Рис. 6. Межкластерная загрузка GRID для различных методов

1. Пономаренко В.С., Лустровой С.В., Минухин С.В., Знахур С.В. Методы и модели планирования ресурсов в GRID-системах: монография. – Харьков: ИД «ИНЖЭК», 2008. – 408 с.
2. Xin Liu. Scalable Online Simulation for Modeling Grid Dynamics. – University Of California, San Diego, 2004. – 173 p.
3. Грушин Д.А., Поспелов А.И. Система моделирования Grid: реализация и возможности применения. [Электронный ресурс] – Режим доступа [http://www.ispras.ru/ru/proceedings/docs/2010/18/isp\\_18\\_2010\\_243.pdf](http://www.ispras.ru/ru/proceedings/docs/2010/18/isp_18_2010_243.pdf).
4. Кореньков В.В., Нечаевский А.В. Пакеты моделирования DataGrid // Электронный журнал «Системный анализ в науке и образовании». – 2009. – № 1. – С. 21. – 35.
5. Sulistio A., Yeo C.S., Buyya R. Simulation of Parallel and Distributed Systems: A Taxonomy and Survey of Tools. // International Journal of Software Practice and Experience, Wiley Press. – 2002. – P. 1 – 19.
6. Минухин С.В., Коровин А.В. Моделирование планирования ресурсов GRID средствами пакета GridSim // Системи обробки інформації. Інформаційні технології та комп'ютерна інженерія. – 2011. – Вип. 3(93). – С. 62 – 68.
7. Grid Scheduling Architecture Research Group (GSA-RG). [Электронный ресурс] – Режим доступа: <https://forge.gridforum.org/projects/gsa-rg/>.
8. Buyya R., Murshed M. Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing // Concurrency and computation: practice and experience. – 2002. – Vol. 14. – P. 1175–1220.
9. Howell F., McNab R. Simjava: A discrete event simulation library for Java. // In International Conference on Web-Based Modeling and Simulation, 1998. – P. 51–56.
10. Latest GridSim API. [Электронный ресурс] – Режим доступа: <http://www.buyya.com/gridsim/doc/api/>.
11. Klusáček D., Rudová H. Alea 2 - Job Scheduling Simulator // In proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST, 2010.
12. Hovestadt M., Kao O., Keller A., Streit A. Scheduling in HPC resource management systems: Queuing vs. planning. // In Job Scheduling Strategies for Parallel Processing, 2003. – P. 1 – 20.
13. Hong Nang, Tianwei Ni. PB-FCFS – A Task Scheduling Based on FCFS and Backfilling Strategy for Grid Computing // Pervasive Computing (JCPC), 2009 IEEE. – P. 507 – 510.
14. Klusáček D., Matyska L., Rudová H. Alea – Grid scheduling simulation environment // 7-th International Conference on Parallel Processing and Applied Mathematics, 2007. – P. 1029 – 1038.
15. Marco A., Netto S., Buyya R. A Flexible Resource Co-Allocation Model based on Advance Reservations with Rescheduling Support. [Электронный ресурс] – Режим доступа: <http://www.cloudbus.org/reports/FlexCoallocation2007.pdf>.
16. Li K. Job scheduling and processor allocation for grid computing on Metacomputers // Journal of Parallel and Distributed Computing, Elsevier, 2005. – V. 65. – Issue 11. – P. 1406 – 1418.
17. Li-Hsing Yen, Chi-Hung Liao. Round-Robin with FCFS Preemption: A Simple MAC Scheduling Scheme for Bluetooth Piconet. // Advanced Information Networking and Applications, 2005. – V. 1. – P. 761 – 766.
18. Фаулер М. Рефакторинг: улучшение существующего кода. – СПб: Символ-Плюс, 2004. – 430 с.
19. Хорстманн К. С., Корнелл Г. Java 2. Библиотека профессионала. Том 1. Основы. – М.: Вильямс, 2008. – 816 с.
20. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2007. – 366 с.
21. The Grid Workloads Archive. [Электронный ресурс] – Режим доступа: <http://gwa.ewi.tudelft.nl/pmwiki/pmwiki.php?n=Workloads.Gwa-t-4>.

Получено 04.06.2011

#### Об авторах:

*Минухин Сергей Владимирович*, кандидат технических наук, доцент, профессор кафедры информационных систем,  
*Коровин Антон Викторович*, магистр информационных управляющих систем и технологий.

#### Место работы авторов:

Харьковский национальный экономический университет, 61166, г. Харьков, проспект Ленина, 9-А.  
Тел.: 057 702 1831,  
E-mail: ms\_vl@mail.ru,  
korovin.anton@gmail.com