

ной эмпирической матрице парных сравнений, а по каждой из эмпирических матриц парных сравнений.

При этом необходимо учесть следующее:

1. Для всех модификаций ММАИ на каждом этапе должна быть проведена соответствующая нормировка оценок весов по каждой эмпирической матрице парных сравнений.
2. Статистическая эффективность ММАИ всех модификаций для общего случая в целом ниже, чем для двухуровневой структуры, поскольку сказывается эффект наложения ошибок определения весов по всем эмпирическим матрицам парных сравнений.
3. ММАИ (вторая версия), эффективно решает задачу, если единая наилучшая альтернатива  $A_i$  при реализации второго этапа оказалась в числе первых  $m_3$  альтернатив, а на третьем этапе дерево иерархии оказалось интегрально хорошо согласованным (с эмпирическими матрицами парных сравнений размерности  $m_3 \times m_3$  на  $s$ -том уровне иерархии).
4. Должна быть создана методика экспертного опроса, позволяющая обоснованно определить закон (законы) искажения эмпирических  $\gamma_{ij}$  (элементов эмпирической матрицы парных сравнений), что позволит эффектив-

но использовать ММАИ соответствующих модификаций.

5. Формальная модель каждой эмпирической матрицы парных сравнений последнего уровня иерархии с точностью до задания параметрического вероятностного распределения искажения ее элементов является одной и той же.

### Выводы

В данной статье были предложены модификации МАИ для двух случаев:

1) Искажение идеальных значений  $\gamma_{ij}$  элементов эмпирической матрицы парных сравнений  $\gamma_{ij}^*$  не зависит от ее размеров. Искажение  $\gamma_{ij}^*$  истинного значения  $\gamma_{ij}$  ( $\forall \gamma_{ij} \geq 1, i \neq j$ ) тем больше, чем больше  $\gamma_{ij}$  отличается от единицы.

2) Искажение  $\gamma_{ij}$  зависит от двух факторов: размерности эмпирической матрицы парных сравнений и величины значения  $\gamma_{ij} - 1$ .

Модификации МАИ приведены для двухуровневой иерархии, затем полученные результаты адаптированы для общего случая.

В статье приведены результаты статистических исследований, доказывающих эффективность данных модификаций по сравнению с классическим МАИ.

### Список литературы

1. Павлов А.А., Кут В.И. Математические модели оптимизации для обоснования и нахождения весов объектов по неоднородным матрицам парных сравнений. Системні дослідження та інформаційні технології. №3, 2007р.
2. Павлов А.А., Кут В.И., Штанкевич А.С. Нахождение весов по матрице парных сравнений с односторонними ограничениями. Вісник НТУУ „КПІ” Інформатика, управління та обчислювальна техніка, Київ 2008р. №48.-122 с. – С.29-32.
3. Павлов А.А., Лишук Е.И., Кут В.И. Математические модели оптимизации для обоснования и нахождения весов в методе парных сравнений” Системні дослідження та інформаційні технології. №2, 2007р.
4. Павлов А.А., Лишук Е.И., Кут В.И. Многокритериальный выбор в задаче обработки данных матрицы парных сравнений. Вісник НТУУ „КПІ” Інформатика, управління та обчислювальна техніка, Київ 2007р. №46.
5. Згуровский М.З., Павлов А.А., Штанкевич А.С. Модифицированный метод анализа иерархий. Інформаційні дослідження та інформаційні технології. №2. 2010 р. (принято к печати).
6. Андрейчиков А.В., Андрейчикова О.Н. Анализ, синтез, планирование решений в экономике.- Москва: Финансы и статистика.-2001.
7. Саати Т. Принятие решений. Метод анализа иерархий: Tomas Saaty. The Analytic Hierarchy Process. –Пер. с англ. Р.Г. Вачнадзе. – М.: Радио и связь, 1993. – 315 с.
8. Саати Т., Кернс К. Аналитическое планирование. Организация систем: Пер. с англ. Р.Г. Вачнадзе: Под ред. И.А. Ушакова. – М.: Радио и связь, 1991. – 223 с.
9. Saaty T.L. Multicriteria Decision Making. The Analytic Hierarchy Process.,-New York:McGraw Hill International,1990.p.437.

Поступила в редакцию 30.11.2009

КУЛАКОВ Ю.А.,  
РУСАНОВА О.В.,  
ШЕВЕЛО А.П.

## ИЕРАРХИЧЕСКИЙ СПОСОБ ПЛАНИРОВАНИЯ ДЛЯ GRID

Статья посвящена повышению пользовательской производительности Grid систем за счет более эффективного способа планирования вычислений. Предложен усовершенствованный иерархический способ планирования, ориентированный на один из трех критериев оптимизации по выбору пользователя. Эффективность данного способа обеспечивается за счет разработанных авторами списочных алгоритмов для однородных и неоднородных узлов Grid систем, а также предложенных способов трансформации графов задач.

Paper is devoted to increasing of Grid systems real performance by effective scheduling method using. We propose an improved hierarchical scheduling method that uses one of the three optimization criterion determined by user. This method efficiency provides by creation of the new list scheduling algorithms for homogeneous and heterogeneous Grid system nodes and creation of task graph transformation.

### Введение

В настоящее время современные Grid системы все чаще используются при решении крупномасштабных проблем в науке, технике и бизнесе. Для обеспечения практической применимости Grid систем должна быть решена проблема обеспечения качества обслуживания. Качество обслуживания многоаспектное понятие, включающее: безопасность, надежность и доступность ресурсов; а также гибкость политики распределения ресурсов, которая непосредственно влияет на приемлемое и предсказуемое время ответа. Таким образом, для достижения высокого качества обслуживания и высокой пользовательской производительности Grid систем существенно важной является эффективная система планирования вычислений. К сожалению, способы планирования в традиционных параллельных и кластерных системах, не всегда могут быть использованы в Grid системах. Планирование в этих системах значительно сложнее из-за уникальных особенностей Grid систем [1-3]. Задача планирования в данных системах представляет интерес как самостоятельное исследовательское направление, которому посвящено достаточно большое количество публикаций.

В то же время, известно, что задачи планирования относятся к классу NP-полных и точного решения в общем случае не имеют. Это и определяет целесообразность разработки и исследования новых эффективных подходов к вопросам планирования в Grid системах.

Данная работа посвящена проблеме планирования в вычислительных интер Grid сетях (НТС) [1], которые состоят из множества суперкомпьютерных центров (кластеров и параллельных компьютерных систем различной сложности), рассредоточенных по всему миру и объединенных посредством Интернета.

### Основные способы планирования в Grid системах и их анализ

В настоящее время существует три основных способа планирования в Grid системе: *централизованный, децентрализованный и иерархический* [4].

При *централизованном* способе все пользовательские приложения посылаются централизованному планировщику. Существует единая очередь в централизованном планировщике для поступающих приложений. Как правило, локальный узел не имеет своей очереди, и не выполняет функцию планирования. Узел только получает задачи от центрального планировщика и выполняет их. Преимуществом такого способа является достаточно высокая эффективность планирования, связанная с тем, что планировщик в данном случае имеет информацию о всех доступных ресурсах и поступающих приложениях. С другой стороны, централизованный планировщик может оказаться узким местом с точки зрения как надежности, так и производительности. Таким образом, централизованный способ планирования пригоден

лишь для Grid систем с ограниченным числом узлов.

При *децентрализованном* способе, функция планирования выполняется на каждом узле системы. В этом случае любой узел Grid системы работает и как планировщик, и как вычислительный ресурс. Пользовательские приложения передаются локальному планировщику Grid системы, где выполняются приложения. Локальный планировщик отвечает за планирование локальных приложений и обслуживает локальную очередь собственных приложений. Кроме того, он должен быть в состоянии ответить на запросы внешних приложений, принимая или отклоняя их. Так как ответственность планирования распределена, отказ отдельного планировщика не затрагивает работу других. Таким образом, децентрализованный способ обеспечивает лучшую отказоустойчивость и надежность, по сравнению с централизованным. Но нехватка глобального планировщика, который знает информацию обо всех приложениях и ресурсах, обычно приводит к низкой эффективности. Данный способ может быть применен для Grid систем с неограниченной масштабируемостью, однако эффективность планирования может быть достаточно низкой.

При *иерархическом* способе процесс планирования заданий распределен на двух уровнях: глобальном и локальном. На глобальном уровне управление заданиями осуществляет метапланировщик Grid, а на локальном – менеджер ресурсов. В отличие от централизованных метапланировщика Grid системы обычно не могут непосредственно управлять ресурсами сети, но работают как брокеры или агенты [5]. Информация о состоянии доступных ресурсов очень важна для метапланировщика Grid системы для того, чтобы выполнить эффективное планирование, особенно с учетом неоднородной и динамичной природы Grid сети. Роль информационной службы Grid системы (Grid information service (GIS)) – обеспечить такой информацией планировщиков Grid. GIS ответственна за сбор и предсказание информации о состоянии ресурса, такую как производительность процессора (процессоров) узлов, размер памяти, пропускная способность сети, готовность программного обеспечения и загрузка узла в определенный период. GIS отвечает на запросы для получения информации о ре-

сурсе или передает информацию пользователям. Система мониторинга и контроля Globus (The Globus Monitoring and Discovery System (MDS)) [6] – пример GIS.

Помимо информации о ресурсе от GIS, свойства приложений (например, приблизительное количество команд, требования к памяти и хранению, зависимость подзадач в задании и объемы коммуникации) и производительность ресурса для различных видов приложений также необходимы для выполнения эффективного планирования. Получение указанных свойств может быть обеспечено компонентой профилирования приложения (Application profiling (AP)), а измерение производительности ресурса для данного типа задания – компонентой тестирования (analogical benchmarking (AB)) [7], [8]. На основе информации, полученной от AP и AB, а также используемой *модели производительности* [3] производится оценка стоимости планирования узлов-кандидатов на выполнение приложения, из которых планировщик выбирает оптимальный в соответствии с целевой функцией.

Модуль запуска и контроля (Launching and Monitoring LM) (также известный как "компоновщик" [9]) осуществляют передачу приложения на выбранные ресурсы, пересылая входные данные и исполняемые файлы, а в случае необходимости и контролирует выполнение приложений. LM – Globus GRAM (Grid Resource Allocation and Management) – пример такого модуля [10].

Локальный менеджер Ресурсов главным образом отвечает за выполнение внешних (полученных от метапланировщика) и локальных заданий, а также передает сообщение для GIS информацией о состоянии ресурсов. В пределах домена один или множество местных планировщиков работают с конкретной локальной политикой управления ресурсами. Примеры таких локальных планировщиков включают OpenPBS [11] и Condor [12]. Для сбора информации о локальном ресурсе LRM используют такие инструментальные средства, как Network Weather Service [13], Hawkeye [12] и Ganglia [14].

Таким образом иерархический способ планирования объединяет преимущества как централизованного, так и децентрализованного способов. Он обеспечивает высокую эффективность планирования, поддерживает высокую надежность и может быть приме-

ним для Grid систем с неограниченной масштабируемостью. Его основным недостатком является сложность. В данной работе предлагается именно иерархический способ планирования, поскольку это наиболее эффективный подход для вычислительных ин-тер Grid систем.

### Анализ основных характеристик наиболее известных Grid систем

Рассмотрим основные характеристики наиболее известных Grid систем, приведенные в табл.1.

**Табл. 1. Основные характеристики Grid систем**

Система	Разработчик	Ресурсы	Приложения	Политика планирования	Планирование
Condor	University of Wisconsin, Madison	Grid одного домена, не разделяемы во времени	Однозадачные Приложения	Оптимизация системной производительности, максимизирующая пропускную способность	Децентрализованная организация планировщика, динамическое
Condor/G	University of Wisconsin, Madison	Grid множества доменов, не разделяемы во времени	Однозадачные приложения	Оптимизация системной производительности, максимизирующая пропускную способность	Децентрализованная организация планировщика, динамическое
AppLeS	University of California, San Diego	Grid одного домена, разделяемы во времени	Приложения в виде параллельного потока подзадач	Максимизация пользовательской производительности	Децентрализованная организация планировщика, статическое
Legion	University of Virginia	Grid одного домена, разделяемы во времени	Приложения в виде параллельного потока подзадач	Максимизация пользовательской производительности, минимизирующая время выполнения	Децентрализованная организация планировщика, динамическое
Nimrod/G	Monash University, Australia	Grid множества доменов, разделяемы во времени	Приложение формируется в реальном времени	Оптимизация экономических затрат	Децентрализованная организация планировщика, статическое
EGEE	Европейский проект	Grid множества доменов, не разделяемы во времени	Приложения в виде параллельного потока подзадач	Оптимизация системной производительности, максимизирующая пропускную способность	Иерархическая организация планировщика, статическое
RDIG	Россия	Grid множества доменов, не разделяемы во времени	Приложения в виде параллельного потока подзадач	Оптимизация системной производительности, максимизирующая пропускную способность	Иерархическая организация планировщика, статическое
DEISA	Европейский проект	Grid одного домена, не разделяемы во времени	Приложения в виде параллельного потока подзадач	Максимизация пользовательской производительности, минимизирующая время выполнения	Иерархическая организация планировщика, статическое

Анализ данной таблицы позволяет сделать следующие выводы:

- современные Grid системы используют иерархическую организацию планировщика;
- современные Grid системы чаще используют статическое планирование;
- с точки зрения политики планирования, все Grid системы ориентированы на реа-

лизацию одного из возможных вариантов критерия оптимизации;

- практические Grid системы чаще всего используются для однозадачных приложений либо приложений в виде параллельного потока подзадач.

Таким образом, повышение эффективности способов планирования для Grid систем связано с их совершенствованием на основе следующих свойств:

1. Использования множества различных критериев оптимизации.
2. Возможностью реализации для приложений в виде параллельно-последовательного потока подзадач.
3. Иерархическая организация планировщика.
4. Использование эффективных субоптимальных алгоритмов статического планирования.

Поэтому в данной работе предложен способ планирования для вычислительных Grid систем с перечисленными выше свойствами.

### Общая постановка задачи планирования

Рассмотрим прежде всего исходные данные для выполнения задачи планирования. Как было указано выше, работа посвящена повышению эффективности многопроцессорного статического планирования в вычислительных Grid системах для приложений в виде параллельно-последовательных подзадач. В этом случае приложение может быть представлено ориентированным ациклическим графом задачи (directed acyclic graph – DAG)  $G_i(T, E_i)$  [15]. В таком графе  $G_i$ , множество вершин  $T = \{t_1, \dots, t_m\}$  и множество дуг  $E_i$  представляют собой, соответственно, подзадачи и информационные зависимости в виде связей между ними. При этом вес вершины  $t_i$  в графе задачи определяется, как  $w_i$  и представляет собой вычислительную сложность  $i$ -й подзадачи. Информационная зависимость между  $i$ -й и  $j$ -й подзадачами задается с помощью дуги между ними в графе задачи с весом  $e_{ij}$ , и означает, что подзадача  $t_j$  может начать выполнение лишь тогда, когда завершила свое выполнение подзадача  $t_i$ . Кроме этого, перед выполнением, подзадача  $t_j$  должна получить необходимый объем данных  $e_{ij}$  от своей предшественницы (подзадачи  $t_i$ ).

Любой высокопроизводительный узел Grid системы, который является кластером либо параллельной системой, может быть

представлен неориентированным графом системы  $G_p(P, E_p)$ . В графе  $G_p$ , множество вершин  $P = \{p_1, \dots, p_n\}$  и множество ребер  $E_p$  представляют соответственно процессорные элементы (либо компьютеры кластера) и топологию каналов связи между ними. При этом вес вершины  $p_i$  в графе системы определяется как  $s_i$  и представляет собой производительность  $i$ -го процессора. А вес ребра, соединяющий две вершины в графе системы, например между  $i$ -ой и  $j$ -ой, определяется как  $u_{ij}$ , и представляет собой канал передачи данных (линк) между процессорами  $i$  и  $j$ , который имеет скорость (пропускную способность), равную  $u_{ij}$ . Для однородных узлов Grid системы значения весов вершин и весов ребер в графе имеют равные значения, а для неоднородных кластеров значения таких весов могут различаться.

Результат планирования может быть представлен с помощью диаграммы Ганта. Однако традиционная диаграмма Ганта содержит не всю информацию о пересылаемых данных с одного процессора на другой. Поэтому предлагается использовать модифицированную диаграмму Ганта [16], в которой расписание работы каждого процессора включает не только очередь исполняемых задач, но и порядок пересылаемых данных всех его каналов.

Задача планирования заключается в следующем. Пусть имеется Grid система, состоящая из  $K$  узлов. Каждый  $i$ -й узел состоит из  $P_i$  процессоров (компьютеров). Задано приложение из  $m$  подзадач с помощью DAG графа. Известны также топологии каждого из  $K$  узлов системы, описанные с помощью неориентированных графов. Требуется найти  $i$ -й узел Grid системы, который обеспечивает оптимизацию (максимизирует эффективность) решения заданного приложения.

В данной работе будем решать следующие задачи оптимизации планирования.

1. Необходимо найти такой  $i$ -й узел Grid системы, который обеспечивает минимальное общее время выполнения заданного приложения ( $T_i$ ). Математическая модель этой задачи может быть записана следующим образом. Найти

$$\min_{i=1, R} \{T_i\} \quad (1)$$

$$T_i = \sum_{j=1}^m \sum_{l=1}^{P_i} t_{jli} * X_{jli} + S_i + \max\{Tr, Tf_i\} \quad (2)$$

где  $t_{jli}$  – время выполнения  $j$ -й подзадачи в  $l$ -м процессоре  $i$ -го узла Grid системы;

$S_i$  – время доставки входных данных и результатов приложения в (из)  $i$ -го узла Grid системы;

$Tr$  – время готовности приложения к выполнению в узлах системы;

$Tf_i$  – время освобождения  $i$ -го узла Grid системы для выполнения заданного приложения в эксклюзивном режиме;

$X_{jli} = 1$ , если  $j$ -й подзадача выполняется  $l$ -м процессоре  $i$ -го узла,

$X_{jli} = 0$  в противном случае.

2. Необходимо найти  $i$ -й узел Grid системы с минимальной стоимостью, который обеспечит выполнение заданного приложения за ограниченное время ( $Tz$ ). Математическая модель этой задачи может быть записана следующим образом. Найти

$$\min_{i=1, R} \{C_i\} \quad (3)$$

при ограничении

$$T_i \leq Tz (i = 1, K) \quad (4)$$

В соотношении (3)  $C_i$  – стоимость  $i$ -го узла при выполнении заданного приложения. В данном случае вначале определяется подмножество узлов, время выполнения приложения в которых соответствует ограничению (4). Затем среди них необходимо найти узел, в котором приложение выполняется с минимальной стоимостью (соотношение (3)).

3. Необходимо найти  $i$ -й узел Grid системы с минимальной стоимостью, который обеспечит минимальное общее время выполнение заданного приложения. Математическая модель этой задачи может быть записана следующим образом. Найти узел Grid системы, удовлетворяющий соотношениям (1) и (2) при ограничении

$$C_i \leq C_{\min} \quad (5)$$

В этом случае вначале в соответствии с соотношениям (1) и (2) определяется подмножество узлов, обеспечивающих выполнение приложения за минимальное время. Затем среди них в соответствии с ограничением (5) выбирается узел с минимальной стоимостью.

### Модифицированный иерархический способ планирования

Для повышения эффективности планирования приложений в вычислительных Grid системах предлагается модификация иерархического способа за счет:

- использования двух различных типов алгоритмов планирования: для однородных и для неоднородных узлов;
- разработки новых эффективных алгоритмов для однородных и неоднородных узлов;
- разработки и применения различных способов трансформации графа задачи, которые целесообразно применять в качестве предварительного этапа при решении задачи планирования;
- использования одного из трех критериев оптимизации по выбору пользователя из рассмотренных выше.

Известно, что одной из важнейших характеристик Grid систем является неоднородность как вычислительных ресурсов, так и коммуникационных сетей. Поэтому алгоритмы планирования для традиционных однородных параллельных систем неприемлемы для Grid систем. В то же время анализ существующих вычислительных Grid систем показывает, что несмотря на то, что такие системы действительно состоят из неоднородных узлов, каждый из них состоит из однородных процессоров. Так, например, система DEISA включает десять разнородных суперкомпьютеров и суперкластеров, каждый из которых состоит из однородных процессоров. Поэтому имеет смысл использовать при планировании два типа алгоритмов – однородные и неоднородные.

Как известно, повышение эффективности выполнения приложения в параллельных узлах Grid системы непосредственно связано с минимизацией числа пересылок. С этой целью предлагается использовать четыре способа трансформации графа задачи, которые целесообразно применять в качестве предварительного этапа при решении задачи планирования в метапланировщике.

При планировании вычислений в Grid системах чаще всего применяется один из следующих критериев оптимизации – максимизация пользовательской производительности, максимизация системной производительности или оптимизация экономических затрат. В предлагаемом подходе при плани-

ровании может быть применен один из следующих трех критериев по выбору пользователя. Первый из них относится к максимизации пользовательской производительности, второй к оптимизации экономических затрат, а третий к комбинации максимизации пользовательской производительности и оптимизации экономических затрат. Таким образом, предлагается более универсальный подход, который может быть применен для различных критериев оптимизации.

Рассмотрим основные этапы предлагаемого иерархического способа планирования для вычислительных Grid систем:

1. Пользователь посылает приложение в виде графа задачи и ресурсный запрос, учитывающий один из трех критериев оптимизации, описанных выше, метадиспетчеру. Метадиспетчер имеет следующую информацию об узлах: стоимость работы каждого узла за единицу времени, а также скорость доставки данных в каждый узел.

2. Метадиспетчер производит трансформацию графа задачи, затем посылает локальным менеджерам трансформированный граф задачи и получает в ответ возможное время запуска на каждом из них. Для определения этого времени на каждом из локальных ресурсов выполняется планирование заданного приложения с использованием одного из предлагаемых в данной работе алгоритмов: для однородных и для неоднородных узлов. Кроме того локальные менеджеры сообщают возможное время освобождения своих узлов для определения начального времени выполнения приложения.

3. В зависимости от заданного пользователем критерия оптимизации и имеющихся данных, метадиспетчер выбирает оптимальный узел Grid системы и посылает туда запрос на резервирование ресурсов, причем резервирование должно быть создано таким образом, чтобы воспользоваться зарезервированными ресурсами могло только это приложение.

4. Если резервирование выполнено, приложение посылается в очередь выбранного узла. В том случае если ресурсы, необходимые для запуска задания освободятся раньше времени начала резервирования, необходимо обеспечить разрешение запуска приложения.

### **Предлагаемые списочные алгоритмы планирования для однородных и неоднородных узлов Grid системы**

Рассмотрим эффективные алгоритмы планирования, применяемые на однородных и неоднородных локальных узлах Grid системы.

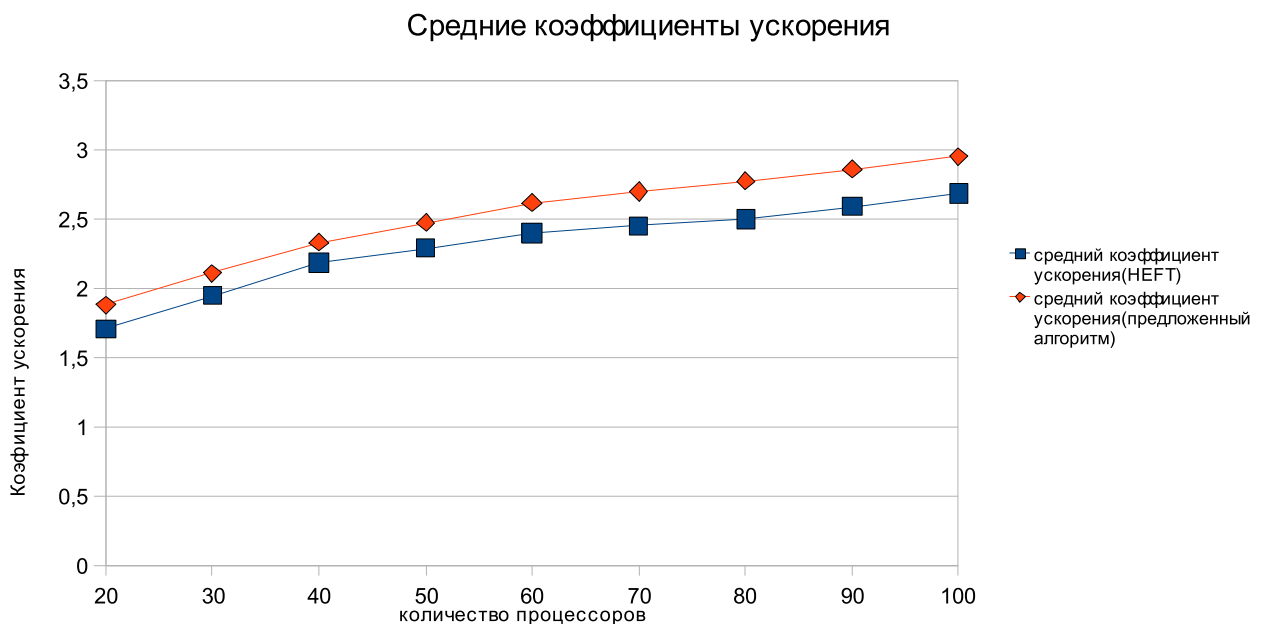
Для однородных узлов Grid системы предлагается использовать, разработанный авторами универсальный двухпроходный списочный эвристический алгоритм (TLHSA) [17], который имеет лучшие показатели эффективности в среднем на 12-25% по сравнению с широкоизвестными аналогичными алгоритмами, такими, как – "Mapping Algorithm" и "Mapping with Duplication Algorithm", разработанными El-Rewini и T. Lewis [18]. TLHSA отличается от известных тем, что при определении величин вычислительной "срочности" подзадач (вершин) принимаются во внимание временные задержки на передачу данных. При определении величин вычислительной "срочности" учитываются следующие характеристики вершин графа задачи: ранний срок выполнения; поздний срок выполнения; резервное время выполнения; вес вершины; сумма весов входящих дуг. Значение вычислительной "срочности" каждой вершины в графе задачи определяется как разность между суммой весов входящих в нее дуг и резервным временем выполнения данной подзадачи. Очередь готовых к выполнению подзадач формируется в порядке убывания значений их вычислительной "срочности". Назначение вершины выполняется на тот процессор, для которого время начала выполнения данной подзадачи на процессоре является минимальным. При этом не имеет значения, занят ли этот процессор в данный момент времени или свободен.

Для неоднородных узлов Grid системы предлагается использовать алгоритм, который является сочетанием TLHSA и известного гетерогенного алгоритма HEFT [19]. Для формирования очереди подзадач предлагается использовать подход подобный тому, который существует в TLHSA. Отличие заключается в способе определения позднего срока выполнения вершин. В алгоритме для однородных узлов время позднего срока выполнения определяется как разность между длиной критического пути графа задачи и длиной самого длинного пути, начиная от данной вершины и заканчивая конечной с

учетом весов вершин, но без учета весов дуг. В алгоритме же для неоднородных узлов целесообразно вместо весов вершин использовать среднее время выполнения вершин на процессорах с различной производительностью неоднородного узла Grid системы.

При принятии решения о назначении подзадач приложения на процессор узла Grid системы целесообразно применить подход, аналогичный тому, который используется в алгоритме HEFT. Такой подход позволяет получить достаточно точный результат. Однако предлагается усовершенствовать модель пересылки данных между процессорами в узле Grid системы при выполнении назначения по сравнению с алгоритмом HEFT. Предлагается при пересылке данных между процессорами использовать модель коммуникационной стоимости, в соответствии с которой в каждом процессоре имеется автономный контроллер ввода-вывода. Это означает, что процессор может одновремен-

но вычислять задачу и обмениваться данными с другим процессором. Коммуникационные задержки между подзадачами, назначенными на один и тот же процессор приравниваются к нулю. Значение же коммуникационной задержки между двумя связанными подзадачами, выполняемыми на различных процессорах является функцией, зависящей от размера сообщения (веса дуги в графе задачи), маршрута и пропускной способности канала между процессорами. Кроме того, предполагается, что пересылка любого сообщения не прерывается и любой процессор узла Grid системы в текущий момент времени может обмениваться данными только с одним процессором. В процессе моделирования было проведено сравнение показателей эффективности предлагаемого алгоритма с известным алгоритмом HEFT. Результаты сравнения средних коэффициентов ускорения представлены на рис. 1.



**Рис. 1. Сравнение средних значений коэффициентов ускорения при использовании предлагаемого алгоритма и HEFT**

Согласно полученным данным видно, что использование предлагаемого алгоритма планирования для неоднородных узлов Grid системы приводит к увеличению коэффициента ускорения в среднем на 10% по сравнению с классическим алгоритмом HEFT.

#### Способы трансформации графов задач

Как было указано выше, повышение эффективности выполнения приложения в параллельных узлах Grid системы непосред-

ственно связано с разработкой и применением различных способов трансформации графа задачи, которые целесообразно применять в качестве предварительного этапа при решении задачи планирования. В работе [20] авторами данной статьи предложено следующие четыре способа трансформации графа задачи: дублирование вершины; объединение предшествующей и последующей вершин; объединение двух параллельных ветвей; объединение двух предшествующих ве-

ршин. Нами также была разработана программная модель для исследования различных алгоритмов статического планирования вычислений для Grid систем. С помощью данной модели был выполнен сравнительный анализ алгоритмов планирования без трансформации и алгоритмов с учетом пятнадцати различных сочетаний рассмотренных трансформаций графов задач. Наилучшие результаты по показателям эффективности были получены для алгоритмов с восьмым (сочетание первого и третьего способов), тринадцатым (сочетание первого, третьего и четвертого способов) и пятнадцатым (сочетание всех четырех способов) способами трансформации. Показатели эффективности этих алгоритмов в среднем на 25% выше, чем алгоритмов без трансформации графа задачи.

### Заключение

1. Проанализированы основные способы планирования в Grid системах. Определен иерархический способ планирования, как наиболее эффективный подход для вычислительных интер Grid систем.

2. Рассмотрена общая постановка задачи планирования, используемая в данной работе. Для этого в соответствии с теорией расписаний определены исходные данные, результаты планирования, а также описаны решаемые задачи оптимизации.

3. Предложен усовершенствованный иерархический способ планирования вычислений для Grid систем, ориентированный на один из трех критериев оптимизации по выбору пользователя. Повышение эффективности предлагаемого подхода обеспечивается за счет использования двух новых эффективных алгоритмов планирования для однородных и неоднородных узлов, а также за счет разработанных способов трансформации графа задачи.

Предложенные алгоритмы доведены до практической реализации в виде программного продукта, который может быть использован при разработке иерархических планировщиков современных вычислительных Grid систем.

### Список литературы

1. Y. Zhu, A Survey on Grid Scheduling Systems, Department of Computer Science, Hong Kong University of science and Technology, 2003.
2. Technical Report No. 2006-504 Scheduling Algorithms for Grid Computing: State of the Art and Open Problems. Fangpeng Dong and Selim G. Akl. School of Computing, Queen's University Kingston, Ontario January 2006.
3. F. Berman, *High-Performance Schedulers*, chapter in *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann Publishers, 1998.
4. V. Hamscher, U. Schwiegelshohn, A. Streit, R. Yahyapour, *Evaluation of Job-Scheduling Strategies for Grid Computing*, in Proc. of GRID 2000 GRID 2000, First IEEE/ACM International Workshop, pp. 191-202, Bangalore, India, December 2000.
5. F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su and D. Zagorodnov, *Adaptive Computing on the Grid Using AppLeS*, in IEEE Trans. On Parallel and Distributed Systems (TPDS), Vol.14, No.4, pp.369-382, 2003.
6. K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, *Grid Information Services for Distributed Resource Sharing*, in Proc. the 10th IEEE International Symposium on High- Performance Distributed Computing (HPDC-10), pp. 181-194, San Francisco, California, USA, August 2001.
7. A. Khokhar, V. K. Prasanna, M. E. Shaaban, and C. L. Wang, *Heterogeneous Computing: Challenges and Opportunities*, IEEE Computer, Vol. 26, No. 6, pp. 18-27, June 1993.
8. H. J. Siegel, H. G. Dietz, and J. K. Antonio, *Software Support for Heterogeneous Computing*, in ACM Computing Surveys, Vol.28, No.1, pp. 237 – 239, March 1996.
9. K. Cooper, A. Dasgupta, K. Kennedy, C. Koelbel, A. Mandal, G. Marin, M. Mazina, J. Mellor-Crummey, F. Berman, H. Casanova, A. Chien, H. Dail, X. Liu, A. Olugbile, O. Sievert, H. Xia, L. Johnsson, B. Liu, M. Patel, D. Reed, W. Deng, C. Mendes, Z. Shi, A. YarKhan and J. Dongarra, *New Grid Scheduling and Rescheduling Methods in the GrADS Project*, in Proc. of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), pp.199--206, Santa Fe, New Mexico USA, April 2004.
10. K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, *A Resource Management Architecture for Metacomputing Systems*, In D.G. Feitelson and L. Rudolph, editors, in Proc of the 4th Workshop on Job Scheduling Strategies for Parallel Processing, LNCS Vol. 1459 pp. 62–82, Orlando, Florida USA, March 1998.
11. <http://www.openpbs.org>
12. <http://www.cs.wisc.edu/condor>

13. R. Wolski, N. T. Spring and J. Hayes, *The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing*, in the J. of Future Generation Computing Systems, Vol. 15, No. 5-6, pp. 757-768, January 1999.
14. F.D. Sacerdoti, M.J. Katz, M.L. Massie and D.E. Culler, *Wide area cluster monitoring with Ganglia*, in Proc. of IEEE International Conference on Cluster Computing, pp.289 – 298, Hong Kong, December 2003.
15. Луцкий Г.М., Русанова О.В., “Проблемы отображения и планирования транспьютерных систем”, (англ.) 1-ая Международная Конференция по Параллельной Обработке и Прикладной Математики - PRAM'94 R. Wyrzykowski and H. Piech eds., Czestochowa. 1994, pp.77-83.
16. G. Loutsky, O. Rusanova Scheduling Problems on the Parallel and Distributed Systems - an Overview, “Computer Systems and Networks: designing, application, utilization”. - Poland, Rzeszow, 2000, tom 1, pp.101-105(Engl).
17. A.Korochkin, O.Rusanova Scheduling Problems for Parallel and Distributed Systems, ACM, SIGADA, Vol.XIX, Issue 3, New York, USA, 1999, P.195-201.
18. Блинова Т.А., Русанова О.В. Составление расписания работ для конвейерных вычислительных систем.Тез. докл. регион. семинара "Распределенная обработка информации" /Новосибирск, 1989. - С.24.
19. H. Torcuoglu, S. Hariri, M.Y. Wu, Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing, IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 3, pp. 260 - 274, 2002.
20. Rusanova O.V., Shevelo A.P. List scheduling algorithm modification for MPP systems // Вісник НТУУ «КПІ».– Інформатика, управління та ОТ.-2006.-№45.-С.101-111.

Поступила в редакцію 18.12.2009