УДК 681.3

**Русанова О.В.**, канд. техн. наук
**Шевело А.П.**

# ПЛАНИРОВАНИЕ ВЫЧИСЛЕНИЙ В КЛАСТЕРНЫХ СИСТЕМАХ

**Национальный технический университет Украины «КПИ»**

*Посвящено исследованию различных эвристических списочных алгоритмов планирования для кластерных систем с целью выбора наиболее эффективных из них. Рассмотрены известные и предлагаемые авторами подходы формирования очередей вычислений, а также их назначений на компьютеры кластерных систем. Разработана программная модель, с помощью которой выполнено сравнение различных алгоритмов планирования. Показано, что предлагаемые алгоритмы имеют более высокую эффективность по сравнению с известными подходами. Достоверность полученных результатов подтверждается при работе алгоритмов на реальной кластерной системе*

## *Введение*

В последние годы среди современных суперкомпьютеров удельный вес кластерных систем неуклонно растет. Так, за последние четыре года число кластерных систем в списке пятисот наиболее мощных компьютеров мира TOP500 увеличилось с 208 до 406. Поэтому подходы, связанные с повышением реальной производительности этих систем являются весьма актуальными.

Известно, что одним из основных методов повышения указанной производительности является эффективное планирование вычислительных ресурсов. Данная задача относится к классу *NP*-полных и в общем случае точного решения не имеет. Поэтому исследования в этой области сосредоточены на поиске эвристических подходов, с помощью которых могут быть получены квазиоптимальные результаты.

Решение данной задачи планирования усложняется бурным ростом количества компьютеров в кластерных системах. За последние четыре года максимальное число компьютеров в данных системах увеличилось с 8192 до 14240. Кроме того, кластеры могут включать как однородные так и неоднородные, с точки зрения производительности, компьютеры. Более того, компьютеры в кластерах могут быть объединены в различные топологии. По мнению авторов наиболее полно все перечисленные свойства могут быть учтены при разработке списочных алгоритмов планирования.

Работа посвящена исследованию различных списочных алгоритмов планирования для кластерных систем с целью выбора наиболее эффективных из них. Для этого разработана программная модель, с помощью которой может быть выполнено сравнение различных списочных алгоритмов для кластерных систем. Для подтверждения достоверности работы программной модели, проводится сравнение показателей эффективности алгоритмов планирования, полученных путём моделирования и при работе реальной кластерной системы.

## *Исследуемые эвристические списочные алгоритмы планирования*

Исходными данными списочных алгоритмов являются направленные ациклические графы задач, граф системы и критерий оптимизации планирования [1]. В качестве критериев оптимизации чаще всего используется минимальное время решения задачи при заданной стоимости системы.

Любые списочные алгоритмы можно разделить на два этапа [1]:

– определение приоритетов вершин (подзадач) графа задач на основании его характеристик и формирование очереди выполнения подзадач в порядке убывания их приоритетов;

– назначение подзадач на компьютеры системы.

Все списочные алгоритмы отличаются подходами, применяемыми при выполнении указанных этапов. В данной работе исследованы следующие шесть алгоритмов определения приоритетов подзадач при формировании очереди:

– приоритеты вершин определяются случайным образом, на основании которых и формируется очередь (случайная). Данный подход чаще других используется планировщиками кластерных систем;

– приоритеты вершин определяются на основании их весов, причём высший приоритет имеет вершина с минимальным весом;

– приоритеты вершин определяются на основании критических путей до конца графа задачи с учётом их весов. Данный подход применён в алгоритме *HEFT* [2], который является одним из наиболее известных в настоящее время;

– приоритеты вершин определяются подобно *HEFT* с тем отличием, что при равенстве критических путей больший приоритет имеют вершины с максимальной связностью;

– приоритеты *i*-ой вершины определяется по формуле:

$$P_{ri} = \frac{N_{кpi}}{N_{кр.гр}} + \frac{T_{кpi}}{T_{кр.гр}}, \qquad (1)$$

где $N_{кpi}$ – критический путь *i*-й вершины до конца графа задачи с учётом их колчества;

$N_{кр.гр}$ – критический путь графа по количеству вершин;

$T_{кpi}$ – критический путь *i*-й вершины до конца графа с учётом их весов;

$T_{кр.гр}$ – критический путь графа по сумме весов вершин. В случае равенства указанных значений, высший приоритет имеет вершина с максимальной связностью;

– приоритеты вершин определяются аналогично по формуле (1). В случае равенства этих значений высший приоритет имеет вершина с минимальным весом.

Последние два алгоритма, а также модификация *HEFT* алгоритма предлагаются авторами данной работы.

В качестве алгоритмов назначения предлагается исследовать следующие четыре подхода:

– назначение подзадач на случайный компьютер кластерной системы. Этот алгоритм чаще других используется при работе с реальной кластерной системой;

– назначение на свободный компьютер, который имеет минимальное среднее расстояние до остальных в кластерной системе. В данном случае предполагается, что пропускная способность всех каналов системы одна и та же;

– назначение на наиболее приоритетный компьютер. В этом случае приоритет определяется как произведение времени пересылки данных для назначаемой подзадачи и производительности компьютера. Время пересылки данных вычисляется аналогично подходу, предложенному авторами в работе [3] и учитывает расстояние между компьютерами, пересылаемые объёмы данных и пропускные способности каналов. Таким образом, данный алгоритм является модифицированным подходом, изложенным в работе [3];

– назначение на наиболее производительный свободный компьютер кластерной системы.

## Программный комплекс для исследования алгоритмов планирования

В результате выполнения данной работы был разработан программный комплекс, который включает:

– программную модель для исследования описанных выше алгоритмов планирования вычислений в кластерных системах;

– программу для организации выполнения задач с учётом различных алгоритмов планирования в реальной кластерной системе.

В программной модели предусмотрено использование шести описанных выше алгоритмов формирования очередей и четырёх алгоритмов назначения. Пред-

лагаемая программная модель ориентирована на любую топологию кластерной системы, состоящую из однородных или неоднородных компьютеров.

Исходными данными для программной модели являются:

- граф задачи [1];
- граф кластерной системы [1];
- алгоритмы формирования очереди и назначения.

В модели предусмотрено два способа задания графа задачи:

- вручную пользователем с помощью редактора графа задачи;
- автоматически с помощью генератора графа задачи [4].

Граф кластерной системы и параметры компьютеров задаются с помощью специального редактора.

Результаты моделирования алгоритма планирования представляются в виде модифицированной диаграммы Ганта, в которой отображается как традиционное расписание вычислительных работ в кластерной системе, так и последовательность пересылаемых данных по каналам в процессе выполнения задачи. Пример модифицированной диаграммы Ганта приведен на рис. 1.

Для сравнения различных алгоритмов планирования предусмотрена система сбора статистических данных, входными данными которой являются:

- граф системы;
- начальная и конечная связность генерируемых графов задач и шаг изменения связности;
- начальное и конечное число вершин генерируемых графов задач;
- количество генерируемых графов задач для каждого значения связности.

Результатами работы системы сбора статистики являются следующие показатели эффективности алгоритмов планирования:

- коэффициент ускорения;
- коэффициент эффективности работы системы;
- коэффициент эффективности работы алгоритма планирования, определяемый по формуле

$$K_{эф.алг} = \frac{\max\left\{T_{кргр}, \left\lceil \sum_{i=1}^{m} W_i \Big/ n \right\rceil\right\}}{T_n},$$

где $T_n$ – время выполнения задачи на кластерной системе;

$n$ – число компьютеров в кластерной системе;

$m$ – количество вершин графа задачи;

$W_i$ – вес $i$-й вершины графа задачи.

Программа, предназначенная для выполнения задач в реальной кластерной системе, включает серверную и клиентскую части с удобным графическим интерфейсом пользователя. При выполнении задачи в реальной кластерной системе, вначале серверное приложение ожидает подключения клиентов, после чего рассылает им указания по работе. Клиенты получают исходные данные от сервера, обрабатывают свои подзадачи, обмениваются сообщениями с другими клиентами и формируют результаты вычислений.

### *Результаты экспериментальных исследований*

Рассмотрим экспериментальные исследования, которые были проведены с помощью разработанного программного комплекса.

Для испытаний в качестве базовой топологии кластерной системы была использована «звезда», состоящая из восьми разнородных по производительности компьютеров. Такая топология является одной из наиболее распространёных в современных кластерных системах.

Для анализа различных алгоритмов формирования очередей и назначений использованы такие вышеупомянутые показатели эффективности, как коэффициент ускорения, коэффициент эффективности системы, а так же коэффициент эффективности алгоритмов планирования. Для каждого эксперимента генерировались графы задач с 8,16 и 32 вершинами. Графы задач генерировались со связностью в диапазоне от 90\10 до 10\90 с шагом 10. Количество генерируемых графа задач для каждого значения связности равно 100.

*Первый эксперимент* был посвящён анализу шести, описанных выше, алгоритмов формирования очередей задач. Полученные значения коэффициентов ускорения изображены на рис. 2

Для графов задач с числом вершин 8 в 67% случаев максимальный коэффициент ускорения имеет шестой алгоритм, в остальных 33% случаев максимальный коэффициент ускорения поровну имеют четвёртый и пятый алгоритмы. Для графов задач с числом вершин 16 в 67% случаев максимальный коэффициент ускорения имеет пятый алгоритм, в остальных же 33% случаев максимальный коэффициент ускорения поровну имеют четвёртый и шестой алгоритмы. Для графов задач с числом вершин 32 в 57,1% случаев максимальный коэффициент ускорения имеет пятый алгоритм, в 28,6% случаев – шестой, а в 14,3% случаев четвёртый алгоритм.

Таким образом, анализируя полученные результаты можно отметить, что в зависимости от размерности графа задач лидерами по коэффициенту ускорения являются пятый и шестой алгоритмы формирования очередей.

*Второй эксперимент* был посвящён анализу четырёх рассмотренных алгоритмов назначения. Полученные значения коэффициентов ускорения изображены на рис 3.

Для графов задач с числом вершин 8 в 60% случаев максимальный коэффициент ускорения имеет второй алгоритм, а в 40% случаев – третий алгоритм назначения. Для графов задач с числом вершин 16 в 50% случаев максимальный коэффициент ускорения имеет третий а в 33,3% случаев – четвёртый, а в 16,7% случаев второй. Для графов задач с числом вершин 32 максимальный коэффициент имеют поровну третий и четвёртый алгоритмы назначения.

Таким образом, второй алгоритм назначения имеет смысл использовать при равенстве количества вершин графа задачи числу компьютеров в кластерной системе. Имеет хорошие результаты третий алгоритм назначения при различных размерностях графа задачи и системы. Эффективности четвертого алгоритма увеличивается с ростом размерности графа задачи.

Для подтверждения достоверности полученных результатов при моделировании, было проведено выполненные вычислительных задач на реальной кластерной системе из восьми компьютеров с топологией «звезда». При этом использовались те же алгоритмы планирования, что и при моделировании. Полученные показатели эффективности алгоритмов во время моделирования вычислений и во время их реального выполнения совпадают в 91,3% случаев.

Это указывает на высокую степень соответствия модели вычислений реально работающей системе.

### Выводы

В работе рассмотрены различные списочные алгоритмы планирования для кластерных систем. Разработана программная модель, позволяющая выполнить сравнительный анализ наиболее известных и предполагаемых авторами списочных алгоритмов планирования. Показано, что предлагаемые алгоритмы имеют более высокие показатели эффективности по сравнению с известными подходами.

Для подтверждения достоверности работы программной модели проведено сравнение показателей эффективности алгоритмов планирования, полученных путём моделирования и при работе реальной кластерной системы. Показатели

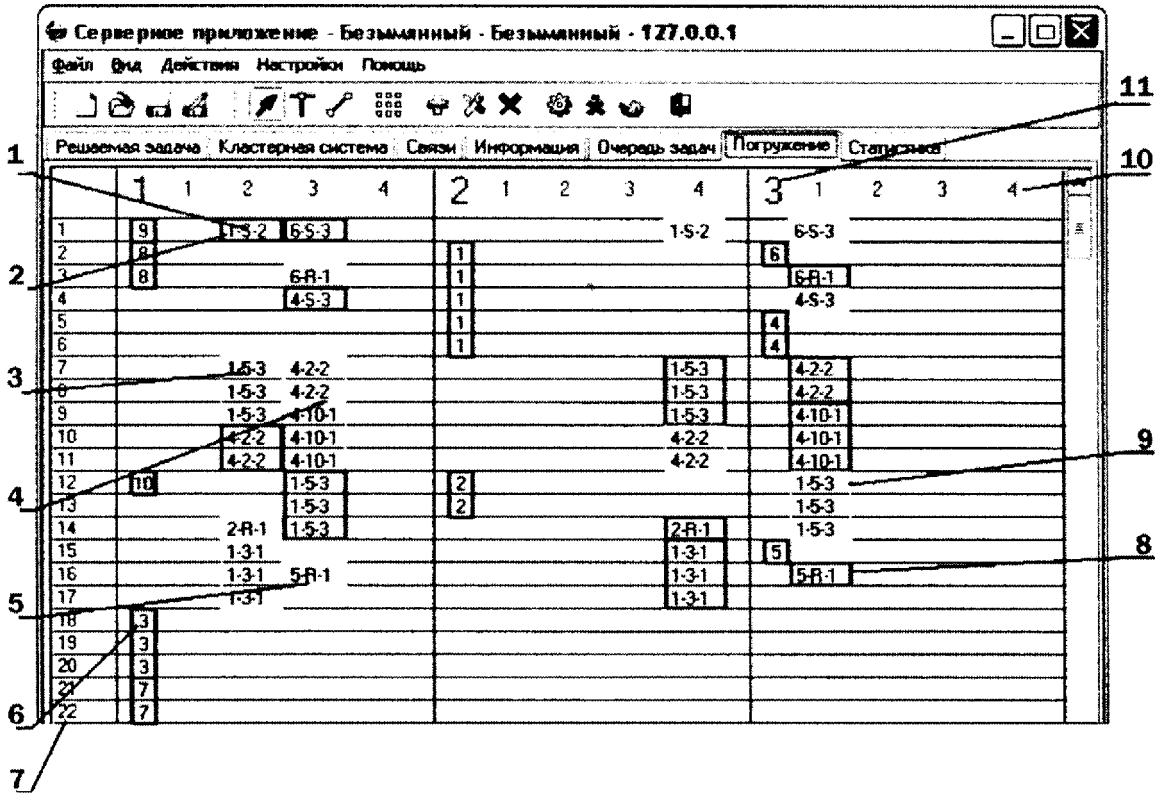
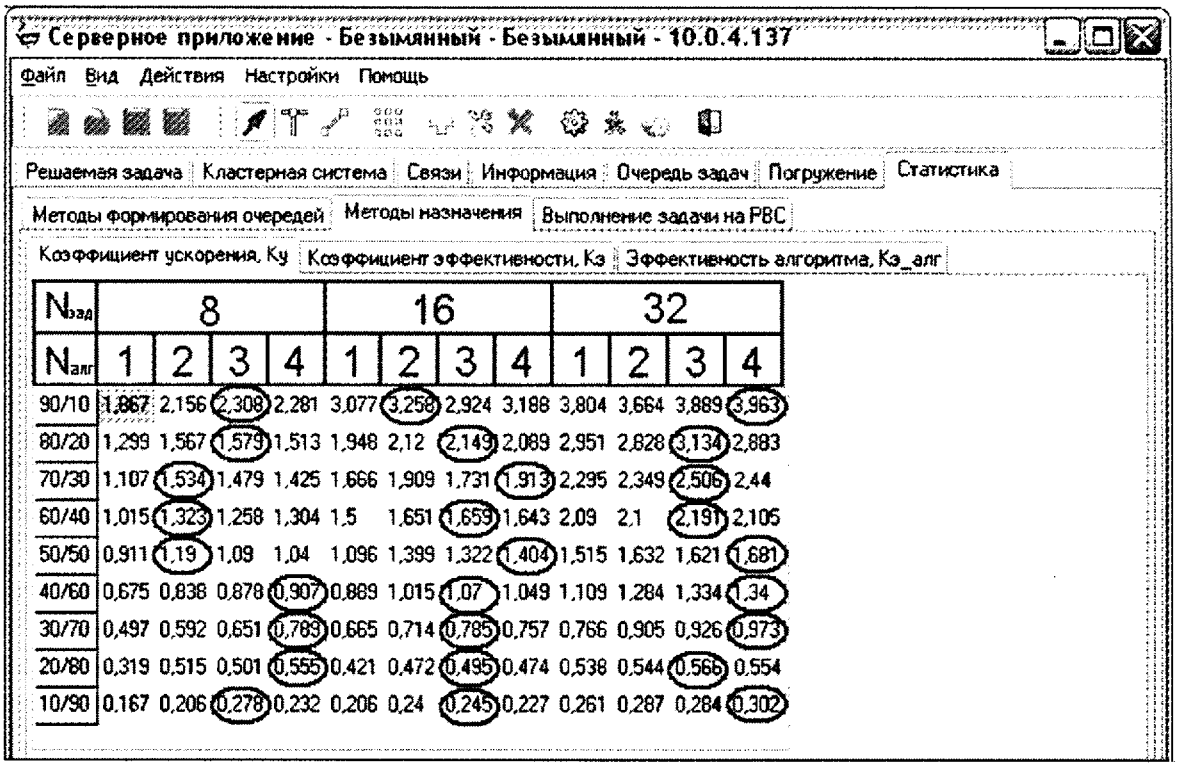



Рис.1. Пример диаграммы Ганта, где: 1 - передача исходных данных; 2 - номер задачи, от которой передаются данные; 3 - номер задачи, для которой передаются данные; 4 - номер компьютера-приемника данных; 5 - возврат результатов на сервер; 6 - номер выполняемой задачи; 7 - время выполнения задачи; 8 - передача данных; 9 - прием данных; 10 - номер канала; 11 - номер компьютера


| $N_{зад}$ | 8 | | | | | | 16 | | | | | | 32 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N_{алг}$ | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 6 |
| 90/10 | 1,895 | 1,874 | 2,071 | 2,13 | 2,23 | 2,382 | 2,483 | 2,638 | 3,056 | 3,053 | 3,162 | 3,337 | 3,397 | 3,555 | 3,737 | 3,951 | 3,903 | 3,946 |
| 80/20 | 1,589 | 1,441 | 1,579 | 1,683 | 1,731 | 1,766 | 2,006 | 2,089 | 2,025 | 2,123 | 2,308 | 2,076 | 2,731 | 2,782 | 3,049 | 3,204 | 3,372 | 3,329 |
| 70/30 | 1,585 | 1,244 | 1,418 | 1,511 | 1,725 | 1,61 | 1,864 | 1,796 | 1,839 | 2,168 | 2,111 | 2,116 | 2,576 | 2,546 | 2,4 | 2,69 | 2,963 | 2,863 |
| 60/40 | 1,349 | 1,181 | 1,331 | 1,409 | 1,453 | 1,459 | 1,465 | 1,386 | 1,605 | 1,702 | 1,888 | 1,75 | 2,083 | 2,186 | 2,14 | 2,237 | 2,466 | 2,344 |
| 50/50 | 1,219 | 1,069 | 1,144 | 1,215 | 1,209 | 1,183 | 1,381 | 1,32 | 1,41 | 1,409 | 1,547 | 1,525 | 1,833 | 1,566 | 1,768 | 1,77 | 1,922 | 1,979 |
| 40/60 | 0,895 | 0,87 | 0,819 | 1,022 | 1,06 | 1,112 | 1,032 | 1,096 | 1,012 | 1,173 | 1,178 | 1,11 | 1,368 | 1,244 | 1,302 | 1,445 | 1,464 | 1,364 |
| 30/70 | 0,768 | 0,618 | 0,799 | 0,754 | 0,838 | 0,722 | 0,772 | 0,722 | 0,723 | 0,828 | 0,852 | 0,958 | 0,963 | 0,905 | 0,91 | 1,037 | 0,987 | 1,058 |
| 20/80 | 0,636 | 0,487 | 0,697 | 0,665 | 0,487 | 0,729 | 0,537 | 0,44 | 0,44 | 0,553 | 0,557 | 0,529 | 0,582 | 0,514 | 0,559 | 0,613 | 0,618 | 0,595 |
| 10/90 | 0,378 | 0,217 | 0,226 | 0,406 | 0,291 | 0,437 | 0,246 | 0,22 | 0,244 | 0,299 | 0,268 | 0,311 | 0,299 | 0,287 | 0,28 | 0,304 | 0,337 | 0,326 |

Рис. 2. Значения коэффициентов ускорения для различных алгоритмов формирования очередей задач




| $N_{зад}$ | 8 | | | | 16 | | | | 32 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $N_{алг}$ | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 90/10 | 1,867 | 2,156 | 2,308 | 2,281 | 3,077 | 3,258 | 2,924 | 3,188 | 3,804 | 3,664 | 3,889 | 3,963 |
| 80/20 | 1,299 | 1,567 | 1,579 | 1,513 | 1,948 | 2,12 | 2,149 | 2,089 | 2,951 | 2,828 | 3,134 | 2,883 |
| 70/30 | 1,107 | 1,534 | 1,479 | 1,425 | 1,666 | 1,909 | 1,731 | 1,913 | 2,295 | 2,349 | 2,506 | 2,44 |
| 60/40 | 1,015 | 1,323 | 1,258 | 1,304 | 1,5 | 1,651 | 1,659 | 1,643 | 2,09 | 2,1 | 2,191 | 2,105 |
| 50/50 | 0,911 | 1,19 | 1,09 | 1,04 | 1,096 | 1,399 | 1,322 | 1,404 | 1,515 | 1,632 | 1,621 | 1,681 |
| 40/60 | 0,675 | 0,838 | 0,878 | 0,907 | 0,889 | 1,015 | 1,07 | 1,049 | 1,109 | 1,284 | 1,334 | 1,34 |
| 30/70 | 0,497 | 0,592 | 0,651 | 0,789 | 0,665 | 0,714 | 0,785 | 0,757 | 0,766 | 0,905 | 0,926 | 0,973 |
| 20/80 | 0,319 | 0,515 | 0,501 | 0,555 | 0,421 | 0,472 | 0,495 | 0,474 | 0,538 | 0,544 | 0,566 | 0,554 |
| 10/90 | 0,167 | 0,206 | 0,278 | 0,232 | 0,206 | 0,24 | 0,245 | 0,227 | 0,261 | 0,287 | 0,284 | 0,302 |

Рис. 3. Значения коэффициентов ускорения для различных алгоритмов назначения

эффективности алгоритмов планирования совпали в 91,3% случаев, что указывает на высокую степень соответствия модели реально работающей системе.

### Список литературы

1. *G. Loutskii, O. Rusanova.* Heuristic Mapping and Scheduling algorithm for Distributed Memory systems. Computer Systems and Networks: designing, application, utilization, – Poland, Rzeszow, 1997, No.1, – P. 253 – 263.

2. *H. Topcouglu, 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, 2002, – P. 260 – 274.

3. *G. Loutskii, O. Rusanova.* List scheduling Algorithm for Parallel and Distributed Systems. Computer Systems and Networks: designing, application, utilization,- Poland, Rzeszow, 2000, T.1, – P. 95 – 100.

4. *G. Loutskii, O. Rusanova.* Direct Acyclic Graphs Scheduling in the Parallel System. Computer Systems and Networks: designing, application, utilization,– Poland, Rzeszow, 1998, part 2, – P. 163 – 170.