

# ИТЕРАЦИОННЫЙ МЕТОД ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ РЕСУРСАМИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА ОСНОВЕ ПОРТФОЛИО ЭФФЕКТИВНЫХ МЕТАЭВРИСТИК

**А.Б. Клименко**

*Институт информационных наук и технологий безопасности Российского государственного  
гуманитарного университета*  
Россия, 117534, Москва, Кировоградская ул., 25, корпус 2,  
E-mail: anna\_klimenko@mail.ru

**Ключевые слова:** перепланирование, реконфигурация, распределение нагрузки, управление ресурсами.

**Аннотация:** В настоящее время продолжается интенсивное развитие вычислительных сетевых инфраструктур, что является основой для развития и эволюции распределенных вычислительных сред, которые, в силу расширения облачных вычислений к краю сети, а также появлению концепции т.н. «бессерверных вычислений», формируют широкий класс таких сред, которые обладают одновременно признаками гетерогенности, географической распределенности и динамики. Анализ представленных в открытой печати публикаций показал, что известные модели распределения нагрузки не учитывают эти особенности вычислительных сред, а именно: производится обобщение маршрута передачи данных до абстрактного канала, сами постановки задач распределения нагрузки рассматриваются вне контекста методов их решения, что важно в случаях комплексных многокритериальных задач. В рамках данного исследования представлена задача распределения вычислительной нагрузки, которая включает параметры, ограничения и целевые функции, характерные для распределенных гетерогенных динамических вычислительных сред, а также предложен итерационный метод решения этой задачи на основе предварительно подготовленного портфолио эффективных метаэвристических алгоритмов.

## 1. Введение

Широкий круг современных сред распределенных вычислений обладает признаками распределенности, динамики и гетерогенности.

В рамках данной работы будем рассматривать распределенность в географическом аспекте, то есть, вычислительная среда формально представляется графом, где при передаче данных из узла в узел могут принимать участие промежуточные узлы.

Динамика определяется изменениями как в топологии, так и в пользовательской нагрузке, и в критериях оценивания качества распределения нагрузки. Например, топология вычислительной среды динамична, когда вычисления организованы в сети автономных движущихся объектах, пользовательская нагрузка может изменяться при перемещении пользователей в транспорте, критерии оценивания качества могут также

меняться в зависимости от состояния внешней среды – например, в случае возникшей необходимости балансировки энергопотребления.

Гетерогенность является прямым следствием множественных и разнообразных устройств, используемых в качестве вычислителей, что заложено в концепции Интернета вещей и Интернета всего.

В то же время анализ представленных в открытом доступе моделей задач распределения нагрузки/вычислительных ресурсов показал, что, как правило, такие задачи формализуются следующим образом:

- классические задачи теории расписаний, несвязанные задачи и параллельные машины[1];
- задачи упаковки в контейнеры или полосы[2, 3];
- задачи структурно-параметрической оптимизации с обобщениями маршрутов передачи данных к выделенному устройству и определением задачи передачи данных как дополнительной вычислительной задачи[4, 5].

Перечисленные типы формализаций ориентированы на частные задачи распределения нагрузки, например, задача упаковки в контейнеры прежде всего моделирует ситуацию выделения ресурсов для информационно несвязанных задач разных пользователей в пределах кластера, и также частично захватывает проблематику структурной оптимизации – в случае, когда мы можем говорить о сокращении количества контейнеров (и это же применяется при консолидации виртуальных машин в кластере). Однако, параллельные машины и упаковка в полосы/контейнеры не дают ответа на вопрос о нагрузке на маршруты передачи данных, поскольку не отображают топологии сети. Следует отметить, что для задач в рамках классической теории расписаний разработано достаточное количество простых эвристик – эффективность которых также под вопросом в случае многокритериальности задачи и необходимости учета топологии.

Еще одной проблемой моделей распределения нагрузки является и то, что они не отражают такой аспект современных вычислительных сред, как динамика. Динамика требует периодической процедуры перепланирования и перемещения задач, и открытым остается вопрос, что лучше: использовать неэффективную эвристику и получить плохое по качеству распределение нагрузки, либо использовать сложный вычислительный метод, получить распределение высокого качества, но при этом в силу сложности процедуры планирования не уложиться в отведенное для решения задачи время.

Гетерогенность вычислительной среды также накладывает определенные требования на постановку задачи планирования

Предлагаемая в данном исследовании модель задачи распределения нагрузки в динамичной гетерогенной вычислительной среде обобщает формализации задач, ставших на фоне развития данной области частными, при этом отличается от ранее предложенных следующим:

- наличием параметра трудоемкости процедуры смены конфигурации;
- наличием параметра транзитной нагрузки на узлы;
- наличием множеств общих и индивидуальных целевых функций и ограничений.

Также предложен итерационный метод решения поставленной задачи с использованием эффективной метаэвристики из сформированного портфолио эффективных алгоритмов.

## 2. Постановка задачи распределения вычислительных ресурсов в распределенной динамичной гетерогенной среде

Для распределения нагрузки по фрагменту географически распределенной сети используются следующие входные данные.

1. Граф задач, предназначенных к решению:  $G_1 = \{ \langle g_i, r_i \rangle, R \}$ , где  $g_i$  – вычислительная сложность задачи,  $r_i$  – требования к ресурсам устройства, на котором будет происходить размещение, включая: требования к объему памяти, требования к пропускной способности канала, к производительности и т.д. Данный граф – ациклический и направленный, где ребра  $R$  взвешены объемом передаваемых данных между задачами.
2. Граф сети представляется произвольным направленным мультиграфом, где вершины взвешены характеристиками узлов сети (производительность, объем памяти, энергопотребление и т.д.), ребра взвешены скоростями передачи данных по каналам связи. То есть:  $G_2 = \{ M, C \}$ ,  $M = \{ m_{ij} \}$  – ресурсы, которыми располагает узел,  $C = \{ c_{ij} \}$  – каналы связи.
3. Имеются общие критерии оценивания качества распределения задач  $S_0 = \{ s_{kj} \}$ .
4. Гетерогенность сети и специфика используемых устройств продуцирует индивидуальные критерии качества распределения, специфичные отдельным узлам, и составляет множество:  $P_0 = \{ p_{ij} \}$ .
5. Имеются общие ограничения  $constr = \{ constr_k \}$ .
6. Процедура управления характеризуется параметрами  $\langle g_r, r_r, t_r \rangle$  где  $g_r$  – вычислительная сложность задачи перепланирования,  $r_r$  – требования к ресурсам узла, где выполняется расчет нового закрепления,  $t_r$  – время выполнения перепланирования (перемещение данных).
7. Остатки ресурсов транзитных узлов определяется как:  $\forall i, k, L m_{i\_ост} = m_i - \sum_{k=1}^L r_k$ , где  $L$  – количество задач, для которых узел  $i$  включен в маршрут передачи данных.

Таким образом, управление системой РВ будет заключаться в решении следующей задачи: необходимо для графов  $G_1$  и  $G_2$  найти такие закрепление задач за устройствами и  $T$ , чтобы при имеющихся ограничениях:  $r_i \leq m_j$ ,  $constr_j$ ,  $r_r \leq m_j$  обеспечить  $S_0 \rightarrow \max, P_0 \rightarrow \max, m_{i\_ост} \rightarrow \max, \forall i \in \{ Ro_\alpha \}$ , где  $\{ Ro_\alpha \}$  – множество маршрутов.

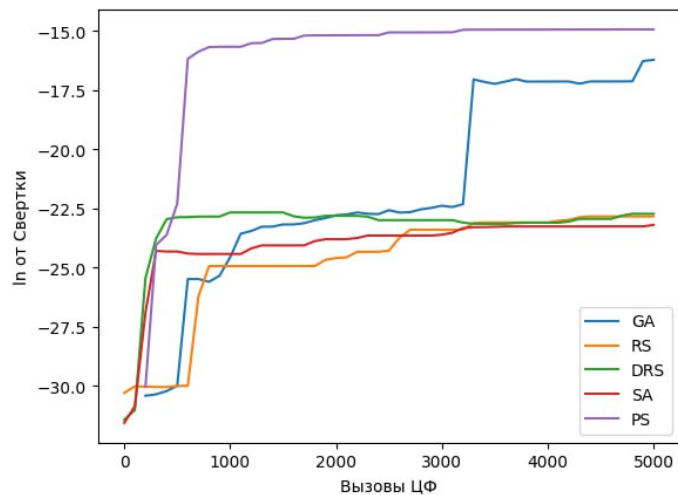
## 3. Итерационный метод повышения эффективности управления ресурсами распределенных вычислительных систем на основе портфолио эффективных метаэвристик

Добавление в модель задачи параметра, определяющего трудоемкость операции перепланирования, ставит вопрос о выборе методов, которые бы позволяли оценить их трудоемкость при решении пр-сложных задач. В этом смысле метаэвристические алгоритмы достаточно легко оцениваемы – трудоемкость пропорциональна числу вызовов целевых функций при допущении что вычислительная сложность целевой функции превышает сложность вспомогательных вычислений (генерацию решений и организацию циклов). Поскольку процедура изменения конфигурации является дополнительной задачей, следовательно, чем меньше будет ее трудоемкость, тем меньше ресурсов будет использовано. Следует отметить, что здесь имеется противоречие, поскольку с увеличением трудоемкости применяемой метаэвристики увеличивается соответственно площадь обследованного поискового пространства и, следовательно, качество получаемого результата. Кроме того, общеизвестным является

и тот факт, что метаэвристики требуют перенастройки параметров при изменении входных данных и не гарантировано, что один и тот же экземпляр алгоритма будет одинаково успешно решать задачи с различными исходными данными.

Учитывая, что представленная в предыдущем разделе задача является новой, поведение метаэвристик на формируемом поисковом пространстве неясно и, следовательно, целесообразно провести ряд тестовых испытаний для выявления классов алгоритмов, использование которых будет эффективно. К рассмотрению были приняты следующие алгоритмы: имитация отжига, направленный случайный поиск, случайный поиск, оптимизация роением частиц, генетический алгоритм.

Исходные данные решаемой задачи следующие: 12 узлов случайной производительности, 10 задач, связанных передачей данных, случайной трудоемкости, каждому узлу соответствует индивидуальная ЦФ – значение вероятности безотказной работы при  $t=100$ ч. Данная векторная целевая функция была скаляризована путем применения мультипликативной свертки. Пример результатов работы различных метаэвристик представлен на рис. 1.



**Рис.1.** Эффективность метаэвристик в зависимости от количества вызовов целевой функции.

Опираясь на результаты тестовых реализаций алгоритмов, получаем таблицу зависимостей вида «наиболее эффективный алгоритм – количество вызовов ЦФ». И, далее, предлагается использовать следующий алгоритм с последовательным улучшением результата, при этом к входным данным добавляются объемы задач при пересылке по сети, а к стоимостной ЦФ добавляется критерий перемещения пользовательских задач на целевые узлы.

#### Описание алгоритма:

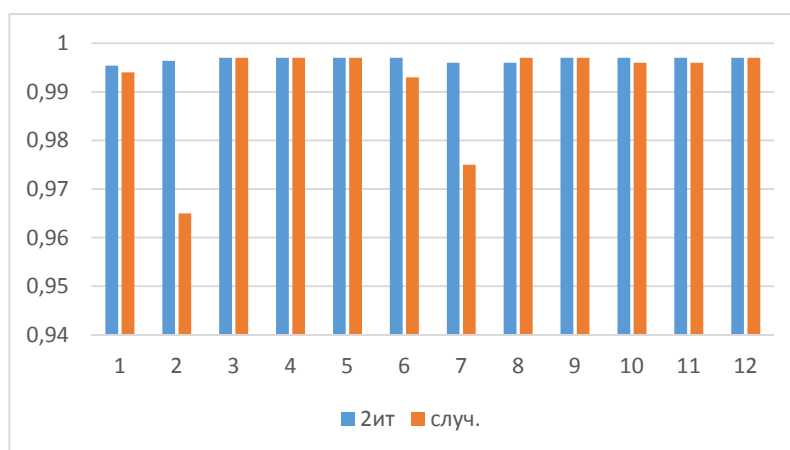
- 1) Все задачи располагаются на узле, ассоциированном с планировщиком.
- 2) Получить пессимистичную оценку времени, необходимого на передачу задач на целевые узлы  $t_{transit}$ .
- 3) Получить пессимистичную оценку времени решения пользовательских задач на целевых узлах. Получим значение  $t_{transit}+t_{app}$
- 4)  $T_{opt} = T_0 - t_{transit} - t_{app}$  – минимальное время, которое может быть гарантированно затрачено на процесс планирования вычислений.
- 5) Если  $T_{opt} > 0$  то:
  - Для  $T_{opt}$  выбираем такой алгоритм, который к концу этого отрезка показывает максимальную эффективность.  $T_{opt}$  пропорционально числу вызываемых ЦФ. Для

этого предварительно подготовлены данные по результатам предыдущего эксперимента.

- Решаем задачу на протяжении времени  $T_{opt}$ , применяя выбранный алгоритм.
  - По полученному решению (реальному распределению задач) пересчитываем  $d = T_o - t_{app}$ , где  $d$  – величина, на которую увеличилось время, которое может быть потрачено на планирование.
  - Если при приращении времени эффективность применяемых алгоритмов изменяется менее чем на 10%, считаем решение окончательным и переходим к п.6.
  - Если приращение эффективности более чем на 10% то  $T_{opt} = d/2$ , переходим к п.5.
- В противном случае:
- Выбираем алгоритм, дающий самую высокую эффективность на самом коротком промежутке времени и первое полученное решение, удовлетворяющее ограничениям, считаем окончательным при этом решая задачу CSP (constraint satisfaction problem).

6. Конец.

Экспериментальные исследования применения описанного алгоритма относительно случайного распределения продемонстрировали следующий эффект (рис. 1)



**Рис. 2.** Сравнение значений индивидуальных значений ВБР узлов после выполнения второй итерации алгоритма и при случайном распределении задач.

Показано улучшение значений индивидуальных ЦФ узлов – значений функции вероятности безотказной работы на 100 часов моделирования. Время пересылки задач на узлы: с применением разработанного алгоритма – 7,49 ед.вр., при случайном распределении задач – 8,24 ед.вр. В процессе получения распределения на первой итерации был выбран алгоритм оптимизации роением частиц (Particle Swarm Optimization), на второй итерации наиболее эффективным был выбран направленный случайный поиск (Directed random search). Таким образом, получено распределение задач по фрагменту сети, выполненное с минимальными трудозатратами (и, соответственно, временем), при котором мы также получаем уменьшение времени реконфигурации (переноса задач на целевые машины).

## 4. Заключение

В данной статье рассмотрен вопрос смены конфигурации в распределенной гетерогенной и динамичной вычислительной среде. Предложен метод повышения эффективности управления ресурсами распределенных вычислительных систем на

основе портфолио эффективных метаэвристик, который базируется на предложенной новой модели задачи распределения вычислительных ресурсов с включением таких параметров, как время реконфигурации и ресурсы узлов, задействованных в передаче данных. Получение решения является итерационным процессом и заключается в последовательном выборе метаэвристики, наиболее эффективной на доступном временном интервале. Применение алгоритма демонстрирует наличие эффекта – на приведенном примере до 4% в значениях индивидуальных ЦФ и до 10% в сокращении времени перемещения задач на целевые узлы.

## Список литературы

1. Pinedo M. Planning and Scheduling in Manufacturing and Services. New York: Springer, 2009.
2. Coffman E., Csirik J., Johnson D., Woeginger G. An introduction to bin packing, 2004.
3. Han Xin, Iwama Kazuo, Ye Deshi, Zhang, Guochuan. Strip Packing vs. Bin Packing, 2006. DOI: 10.1007/978-3-540-72870-2\_34.
4. Топорков В.В. Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004. 320 с.
5. Барский А.Б. Параллельные информационные технологии: учебное пособие / М.: Интернет-Университет Информационных Технологий, 2007; М.: БИНОМ. Лаборатория знаний, 2007. 502 с.