

На правах рукописи



СПИЦЫН Андрей Алексеевич

**УПРАВЛЕНИЕ ПРОЦЕССАМИ МИГРАЦИИ ВИРТУАЛЬНЫХ
МАШИН В ОБЛАЧНЫХ СРЕДАХ НА ОСНОВЕ РЕАЛИЗАЦИИ
ИЕРАРХИЧЕСКОЙ СТРАТЕГИИ БАЛАНСИРОВКИ НАГРУЗКИ**

Специальность: 2.3.5. Математическое и программное обеспечение
вычислительных систем, комплексов и
компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Воронеж – 2023

Работа выполнена в ФГБОУ ВО «Воронежский государственный технический университет».

Научный руководитель: **Мутин Денис Игоревич**, доктор технических наук

Официальные оппоненты: **Ломакина Любовь Сергеевна**, доктор технических наук, профессор, ФГБОУ ВО «Нижегородский государственный технический университет им. Р.Е. Алексеева», профессор кафедры «Вычислительные системы и технологии»

Перепелкин Дмитрий Александрович, доктор технических наук, профессор, ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина», профессор кафедры систем автоматизированного проектирования вычислительных средств

Ведущая организация: **АО «Научно-исследовательский институт вычислительных комплексов им. М. А. Карцева» (г. Москва)**

Защита состоится «26» июня 2023 года в 14³⁰ часов в конференц-зале на заседании диссертационного совета 24.2.286.04, созданного на базе ФГБОУ ВО «Воронежский государственный технический университет», по адресу: г. Воронеж, Московский просп., д. 14, ауд. 216.

С диссертацией можно ознакомиться в научно-технической библиотеке ФГБОУ ВО «Воронежский государственный технический университет» и на сайте www.cchgeu.ru.

Автореферат разослан «12» апреля 2023 г.

Ученый секретарь
диссертационного совета



Гусев Константин Юрьевич

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Стремительное развитие облачных сервисов породило множество методов управления ими. Большой вклад в развитие методов управления внесли Крупин А., Черняк Л., Li C., Wang Y., Mateusz G. В теоретическом плане ряд методов управления облачными сервисами сводится к оптимизации выбора физических исполнителей и порядка исполнения задач в распределенной системе внутри облака. Разумной идеей кажется модифицировать архитектуру системы управления облачными средами, расширив ее на случай нескольких уровней с учетом дополнительного миграционного слоя с использованием концепции контейнеризации.

Несмотря на глубоко развитую теорию назначений, существуют проблемы в облачных средах, связанные с алгоритмами распределения задач и виртуальных машин между физическими исполнительными устройствами в облаке. В данной области ведущие исследователи Романченко И., Самойленко А., Veegom A.A., Jennings N.R. развили множество подходов, ориентированных на классические задачи распределения процессов, обладающие как правило экспоненциальной сложностью. Ключевые показатели для этих подходов, к сожалению, не учитывают то, что задачи планирования в облаке происходят в реальном масштабе времени, и алгоритмы с экспоненциальной сложностью неприемлемы. Несмотря на значительные вычислительные мощности, процессы в облаке происходят быстрее, чем решаются задачи их распределения. Необходимы алгоритмы, основанные на декомпозиции больших задач на подзадачи по специальному критерию с полиномиальной по времени сложностью

Облачные среды часто проходят через периоды локальной перегрузки после неожиданного поступления пользовательских задач. В такие периоды задачи часто пропускают свои предельные сроки. Следовательно, необходимы иерархические или иные кластер-ориентированные стратегии балансировки нагрузки для облачных многокластерных центров обработки данных с приоритизацией локальной балансировки нагрузки сначала внутри кластера, а затем внутри центра обработки данных.

Таким образом, актуальность темы диссертационного исследования продиктована необходимостью дальнейшего развития средств математического и программного обеспечения управления миграцией виртуальных машин в облачных средах на основе иерархической стратегии балансировки нагрузки.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный технический университет» «Вычислительные комплексы и проблемно-ориентированные системы управления».

Целью работы является разработка математического и программного обеспечения управления процессами миграции виртуальных машин в облачных средах на основе иерархической стратегии балансировки нагрузки на виртуальные машины.

Задачи исследования. Для достижения поставленной цели необхо-

димо решить следующие задачи:

1. Модифицировать архитектуру системы управления облачными средами, расширив ее на случай нескольких уровней с учетом дополнительного миграционного слоя с использованием концепции контейнеризации.

2. Разработать эвристический алгоритм планирования задач в облаке, основанный на декомпозиции больших задач на подзадачи по специальному критерию с полиномиальной по времени сложностью.

3. Создать иерархическую стратегию балансировки нагрузки для облачных многокластерных центров обработки данных с приоритизацией локальной балансировки нагрузки сначала внутри кластера, а затем внутри центра обработки данных.

4. Разработать алгоритм решения многокритериальной задачи управления распределением задач в облачных средах на основе алгоритма роя частиц со сверточной фитнес-функцией.

5. Создать программный прототип системы управления распределением заданий в облачных средах.

Объект исследования: системы облачных вычислений.

Предмет исследования: математическое и программное обеспечение управления процессами миграции виртуальных машин в облачных средах на основе иерархической стратегии балансировки нагрузки.

Методы исследования. При решении поставленных в диссертации задач использовались теория вероятностей, теория принятия решений, а также методы объектно-ориентированного программирования.

Тематика работы соответствует следующим пунктам паспорта специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: п.3 «Модели, методы, архитектуры, алгоритмы, языки и программные инструменты организации взаимодействия программ и программных систем»; п.9 «Модели, методы, алгоритмы, облачные технологии и программная инфраструктура организации глобально распределенной обработки данных».

Научная новизна работы. В диссертации получены следующие результаты, характеризующиеся научной новизной:

1. Модифицированная многоуровневая архитектура системы управления облачными средами, отличающаяся наличием дополнительного миграционного слоя с использованием концепции контейнеризации, обеспечивающая сокращение времени задержки, вызванной осуществлением миграции.

2. Эвристический алгоритм планирования задач в облаке с полиномиальной по времени сложностью, отличающийся декомпозицией больших задач на подзадачи по специальному критерию и обеспечивающий соблюдение регламентных сроков исполнения задач.

3. Иерархическая стратегия балансировки нагрузки для облачных многокластерных центров обработки данных, отличающаяся приоритизацией локальной балансировки нагрузки сначала внутри кластера, а затем внутри центра обработки данных и обеспечивающая уменьшение среднего времени отклика и служебных издержек межкластерной коммуникации.

4. Алгоритм решения многокритериальной задачи управления распределением задач в облачных средах, отличающийся применением алгоритма роя частиц со сверточной фитнес-функцией и обеспечивающий оптимальное время выполнения и надежность компьютерных ресурсов и сетевых связей.

5. Структура программного прототипа системы управления распределением заданий в облачных средах, отличающаяся наличием подсистем обслуживания дополнительного миграционного слоя и декомпозицией больших задач на подзадачи, обеспечивающая выполнение требований к качеству обслуживания.

Теоретическая и практическая значимость исследования заключается в разработке математического и программного обеспечения управления миграцией виртуальных машин в облачных средах на основе иерархической стратегии балансировки нагрузки, а также информационного и программного обеспечения для экспериментальной оценки качества разработанных методов и алгоритмов.

Теоретические результаты работы могут быть использованы в проектных и научно-исследовательских организациях, занимающихся проектированием платформенно-инвариантных систем управления облачными средами в условиях штатной или нерегламентированной внешней нагрузки.

Положения, выносимые на защиту

1. Модифицированная многоуровневая архитектура системы управления облачными средами с дополнительным миграционным слоем обеспечивает сокращение времени задержки, вызванной осуществлением миграции.

2. Эвристический алгоритм планирования задач в облаке с полиномиальной по времени сложностью, отличающийся декомпозицией больших задач на подзадачи, обеспечивает соблюдение регламентных сроков исполнения задач.

3. Иерархическая стратегия балансировки нагрузки для облачных многокластерных центров обработки данных обеспечивает уменьшение среднего времени отклика и служебных издержек межкластерной коммуникации.

4. Алгоритм решения многокритериальной задачи управления распределением задач в облачных средах на основе алгоритма роя частиц обеспечивает оптимальное время выполнения надежность как компьютерных ресурсов, так и сетевых связей.

Результаты внедрения. Основные результаты внедрены в Военном учебно-научном центре Военно-воздушных сил «Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина» при проектировании распределенной корпоративной сети, в учебный процесс Воронежского государственного технического университета в рамках дисциплин: «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: XXVI-й и XXIX-й International Open Science Conference «Modern informatization prob-

lems in the technological and telecommunication systems analysis and synthesis» (Yelm, WA, USA, 2021 и 2023); Международной научно-практической конференции «Материалы и методы инновационных научно-практических исследований и разработок» (Пенза, 2021); 16-й Международной научно-технической конференции «Современные инструментальные системы, информационные технологии и инновации» (МТО-56, Курск, 2021); XXVII-th International Open Science Conference «Modern informatization problems in simulation and social technologies» (Yelm, WA, USA, 2022), а также на научных семинарах кафедры автоматизированных и вычислительных систем ВГТУ (2020-2023 гг.).

Достоверность результатов обусловлена корректным использованием теоретических методов исследования и подтверждена результатами сравнительного анализа данных вычислительных и натурных экспериментов.

Публикации. По результатам диссертационного исследования опубликовано 14 научных работ (7 – без соавторов), в том числе 6 – в изданиях, рекомендованных ВАК РФ (из них 2 – в изданиях Scopus и Wos и одно свидетельство о регистрации программы для ЭВМ). В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [4, 10] - модифицированная многоуровневая архитектура системы управления облачными средами с дополнительным миграционным слоем; [5] - эвристический алгоритм планирования задач в облаке с полиномиальной по времени сложностью; [7] - иерархическая стратегия балансировки нагрузки для облачных многокластерных центров обработки данных; [1, 14] - алгоритм решения многокритериальной задачи управления распределением задач в облачных средах на основе алгоритма роя частиц; [6] - информационное и программное обеспечение для экспериментальной оценки качества разработанных методов и алгоритмов.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 228 наименований. Работа изложена на 153 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность исследования, сформулированы его цель и задачи, научная новизна и практическая значимость полученных результатов, приведены сведения об апробации и внедрении работы.

В первой главе исследуются особенности разработки математического и программного обеспечения управления миграцией виртуальных машин в облачных средах на основе иерархической стратегии балансировки нагрузки, и анализируется современное состояние проблемы управления ими. Отмечено, что повысить эффективность управления облачными средами можно путем модификации архитектур системы управления, создания алгоритмов планирования размещением задач в виртуальных машинах и распределения виртуальных машин по физическим исполнительным устройствам. Результат анализа потребовал формализации данных задач, а также алгоритмизации их решения с учетом особенностей, отраженных на рис. 1.1. Сформулирована

цель и задачи исследования.

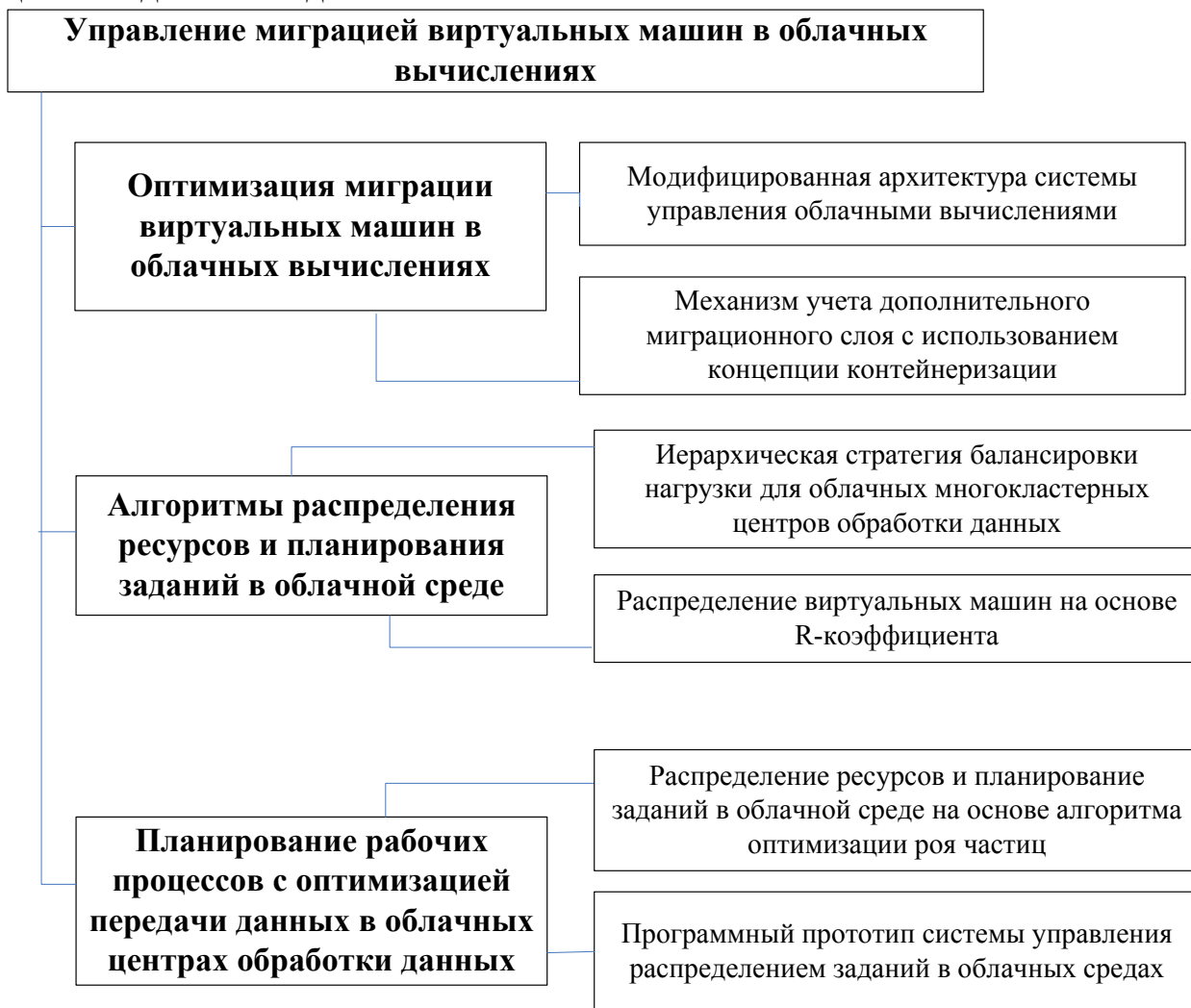


Рис. 1.1. Дизайн исследования

Вторая глава посвящена разработке механизмов оптимизации миграции виртуальных машин в облачных средах с помощью эффективного алгоритма размещения.

Миграция виртуальных машин осуществляется планировщиком, установленным в вычислительных узлах с основной целью балансировки нагрузки на физические серверы. Миграция осуществляется на основе внутреннего дизайна облачной вычислительной системы. Время отклика, согласованное с пользователем и содержащееся в SLA, скомпрометировано из-за тяжелой обработки, необходимой для принятия решения о миграции и фактического осуществления миграции.

Алгоритм планирования вызывает программное обеспечение, расположенное на уровне OML, которое проверяет правильность запроса, и фактическая миграция осуществляется в бизнес-фазе, в которой реализуется эффективный алгоритм миграции/размещения. Здесь реализован алгоритм размещения Squirrel Whale Optimization Algorithm (S-WOA).

Перенос выполняется на основе пяти компонент, которые включают перенос виртуальной машины на физическую машину (PM); контейнер для

VM под таким же индексом; и контейнер для VM под разными РМ задач в виртуальных машин в контейнере; контейнер задачи VM. Фитнес-функция разработана и реализована для поиска наилучшего метода миграции. Фитнес-функция разработана с использованием нескольких параметров, которые включают пропускную способность, использование ресурсов и нагрузку.

На этапе трансформации осуществляется переход от виртуальной машины к контейнеру и наоборот. Производительность стратегии миграции в облаке на основе S-WOA оценивается с точки зрения количества экземпляров виртуальных машин, загрузки процессора, использования памяти, количества активированных РМ и времени. Взаимодействие между различными программами в слое OML показано на рис. 2.1.

В общем случае архитектура облачных вычислений может быть разделена на несколько уровней, таких как аппаратный уровень, инфраструктурный уровень, платформенный уровень и прикладной уровень.

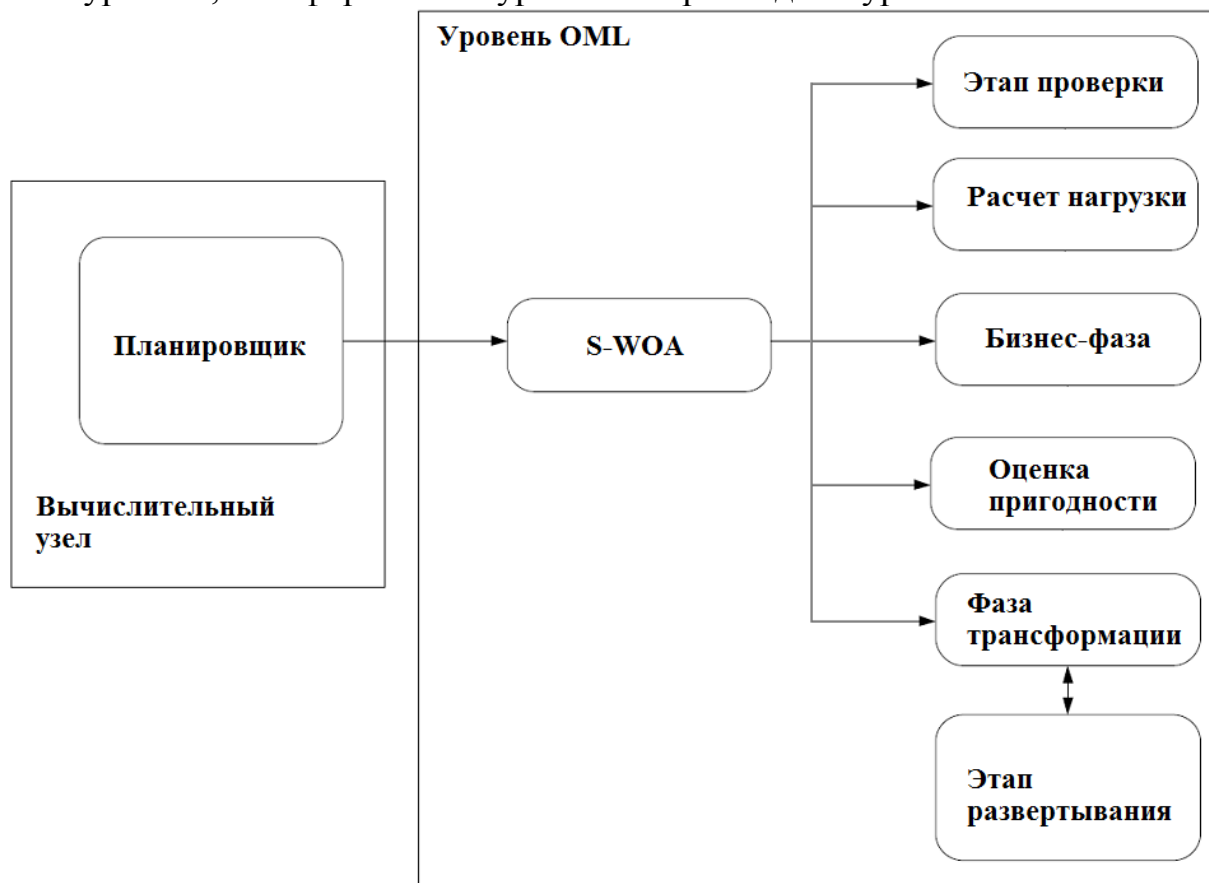


Рис. 2.1. Структура многоуровневой архитектуры OML

Предлагаемый алгоритм S-WOA разработан путем интеграции известных SSA и WOA. Кроме того, была разработана фитнес-функция, основанная на пропускной способности, использовании ресурсов и нагрузке. На этапе преобразования выполняется преобразование информации виртуальной машины в информацию контейнера или наоборот в нотации объекта миграции (MON). Наконец, на этапе развертывания алгоритм размещения, адаптированный на бизнес-этапе, используется в месте назначения для миграции, а на этапе преобразования доступен преобразованный код, который развертыва-

ется в соответствии с требуемой стратегией. После завершения развертывания исходный ресурс удаляется.

Определим облачную модель с различными виртуальными машинами, РМ и контейнерами. Облачная модель, содержащая n РМ, представлена в виде $J=\{J_1, ..., J_i, ..., J_n\}$, $1 \leq i \leq n$, и несколько виртуальных машин доступны в каждой РМ. Кроме того, V виртуальных машин определено как $V=\{V_1, ..., V_j, ..., V_m\}$, $1 \leq j \leq m$, где общее количество виртуальных машин обозначается как V_m . Кроме того, количество C доступных контейнеров определено как $C=\{C_1, ..., C_k, ..., C_p\}$, $1 \leq k \leq p$, где общий объем контейнеров обозначается как C_p . Таким образом, на этапе проверки всякий раз, когда возникает необходимость/запрос на миграцию, сначала проверяется истинность следующего: а) VM/Container существует, активен и работает, б) метрики VM/Container корректны. Однако каждый контейнер состоит из нескольких параметров, таких как загрузка процессора, память, производительность в MIPS, количество обрабатываемых элементов и частота, и уравнение выражается как

$$C_k = \{ D_k, H_k, B_k, A_k, I_k \} \quad (2.1)$$

где D_k - загрузка процессора в k -м контейнере, а H_k - количество памяти, используемой k -м контейнером, B_k - общее количество MIPS, используемых k -м контейнером, A_k - общее количество обрабатываемых элементов, используемых k -м контейнером, I_k - частота k -го контейнера. Для миграции следует учитывать 5 условий:

- условие-1 - от VM к РМ;
- условие-2 - контейнер для VM с той же РМ;
- условие-3 - контейнер для VM с разными РМ;
- условие-4 - миграция задач из VM в контейнер;
- условие-5 - миграция задач контейнера на виртуальную машину.

Коэффициент качества задач (TQR) вычисляется с учетом количества задач с первоначальным крайним сроком выполнения задач и времени, затраченного на их выполнение:

$$TQR = \frac{1}{|t|} \sum_{r=1}^{|t|} \left(\frac{P_{ori} - P_{comp}}{G} \right) \quad (2.2)$$

где $|t|$ - число задач, P_{ori} - первоначальный срок выполнения задач, а P_{comp} - полное время выполнения задач. Далее:

$TQR > T$ - оценить загрузку облака, удовлетворяющую трем условиям;
 $TQR \leq T$, а нагрузка VM больше порогового значения, выполнить условие-4;

$TQR \leq T$, а нагрузка VM меньше или равна порогу, выполнить условие-5.

Значение нагрузки рассчитывается на основе используемых контейнером ресурсов для выполнения задачи, поставленной пользователем.

Память, процессор, количество обрабатываемых элементов, MIPS и частота - это параметры, учитываемые при вычислении нагрузки в облачной платформе. Поэтому нагрузка контейнера вычисляется с помощью:

$$L_k = \frac{1}{G_1} \sum_{q=1}^r \frac{(G^D + G^H + G^B + G^A + G^I) y}{(\max(G^D) + \max(G^H) + \max(G^B) + \max(G^A) + \max(G^I) + 1) e} \quad (2.3)$$

где G^D - общее количество процессоров, используемых в k -м контейнере, G^H - общее количество памяти, используемой в k -м контейнере, G^B MIPS в k -м контейнере, G^A - количество обрабатываемых элементов в k -м контейнере, G^I - количество частот в k -м контейнере, r - k число задач, а G_1 - нормализующий коэффициент. Если q -я задача выполняется k -м контейнером, то параметр $y=1$ или иначе y становится 0.

$$y = \begin{cases} 1, \text{если } q - \text{я задача выполняется } k - \text{м контейнером} \\ 0, \text{в противном случае} \end{cases} \quad (2.4)$$

Нагрузка VM вычисляется с учетом нагрузки контейнера и нормализующего коэффициента:

$$L_j = \frac{1}{G_2} \sum_{k=1}^p L_k \quad (2.5)$$

где L_k - нагрузка k -го контейнера, а $\frac{1}{G_2}$ - нормализующий коэффициент. Затем нагрузка PM вычисляется с учетом нагрузки VM с нормирующим коэффициентом, а выражение для нахождения нагрузки PM задается (2.6):

$$L_i = \frac{1}{G_3} \sum_{j=1}^m L_j \quad (2.6)$$

где L_j - нагрузка j -й VM, а $\frac{1}{G_3}$ - нормирующий коэффициент. Затем вычисление нагрузки в облаке следует трем ситуациям:

1. Если нагрузка PM больше порогового значения, то условие-1, в противном случае миграции не будет.
2. Если нагрузка VM превышает пороговое значение, то условие-2.
3. Если нагрузка PM и VM превышает пороговое значение, то условие-3. После этого, если $TQR \leq T$, то:
4. Если нагрузка VM превышает пороговое значение, то условие-4.
5. Если нагрузка VM является меньшей, чем пороговое значение, проанализировать условие-5.

Фитнес-функция оценивается для определения оптимального решения из множества решений. Пригодность S-WOA оценивается с использованием пропускной способности, использования ресурсов и нагрузки. Фитнес с минимальным значением - это оптимальное решение для миграции в облако. Здесь функция пригодности вычисляется для трех условий и приводится ниже. Фитнес-функция для первого условия, выражается следующим образом:

$$F_1 = \frac{1}{n} \sum_{i=1}^n S_i \quad (2.7)$$

где F_1 - фитнес-функция для условия-1, n - число PM;

$$S_i = \begin{cases} \frac{1}{2} \sum_{\substack{j=1 \\ j \in i}}^m (L_j + M_j), & \text{если } i \text{ не загружено} \\ 1, & \text{если перегружено} \end{cases} \quad (2.8)$$

где L_j - нагрузка j -й VM, а M_j - стоимость миграции:

$$M_j = \frac{1}{n} \sum_{i=1}^n \frac{u_i}{cm} \quad (2.9)$$

где u_i - число миграций, а c - постоянная. Тогда для условий-2 и 3 выражение фитнес-функции представляется в виде

$$F_2 = \frac{1}{n} \sum_{i=1}^n S_i \quad (2.10)$$

где $S_i = \begin{cases} \frac{1}{2} \sum_{\substack{j=1 \\ j \in i}}^m (L_{ij} + M_{ij}), & \text{если } i \text{ не загружено} \\ 1, & \text{если перегружено} \end{cases}$

и стоимость миграции выражается как

$$M_{ij} = \frac{1}{m} \sum_{j=1}^m \frac{u_j^*}{cp} \quad (2.11)$$

где u_j^* - количество миграций из контейнера в VM. Для условий-4 и 5 следует следовать тому же уравнению фитнес-функции условия-1.

Алгоритм оптимизации направлен на определение оптимальных весов для миграции в облаке. Псевдокод разработанного алгоритма S-WOA изображен на рис. 2.2, который демонстрирует пошаговое описание алгоритма.

Алгоритм 2.1: Псевдокод предложенного алгоритма S-WOA

Input: Популяция $X = \{X_1, X_2, \dots, X_u, \dots, X_v\}; 1 \leq u \leq v$

Output: X^* - лучшее решение

Procedure:

Begin

Инициализация популяции $X = \{X_1, X_2, \dots, X_u, \dots, X_v\}; 1 \leq u \leq v$

Вычисление фитнес-функции с использованием (2.5) и (2.8)

while ($h < h_{\max}$)

for каждого возможного решения

if $t > 0.5$

Обновить позицию S-WOA

End if

End for

Убедиться в наличии возможного решения вне области поиска

Обновить X^* при наличии лучшего решения

$h = h + 1$

End while

Return X^*

Рис. 2.2. Псевдокод предложенного алгоритма

Таким образом, создана модифицированная многоуровневая архи-

текстура системы управления облачными средами, отличающаяся наличием дополнительного миграционного слоя с использованием концепции контейнеризации, обеспечивающая сокращение времени задержки, вызванной осуществлением миграции.

Третья глава посвящена алгоритмизации распределения ресурсов и планирования задач в облачной среде.

При решении проблемы минимизации затрат, связанных с выполнением приложений в облачных ресурсах, создан новый алгоритм разделения задач под названием Оптимизированный по стоимости эвристический алгоритм (СОНА) для облачного планировщика для оптимизации стоимости выполнения задач рабочего процесса. В этом алгоритме большие задачи, которые, скорее всего, не будут соответствовать заданным пользователем срокам, разбиваются на подзадачи, чтобы сократить продолжительность планирования. Это делается для того, чтобы все задачи выполнялись в установленные сроки.

Псевдокод предложенного алгоритма представлен на рис. 3.1. Алгоритм начинается с определения прибывших задач рабочего процесса из очереди с набором задач рабочего процесса $Wt=(Wt_1, Wt_2, Wt_3, ..., Wt_n)$ с соответствующими длинами исполнения ($EL001, EL002, EL003, ..., EL00n$). Эти задачи должны быть выполнены на облачных ресурсах (виртуальные машины), которые образуют множество виртуальных машин ($VM_1, VM_2, ..., VM_K$). Вычисляется ожидаемое время завершения (ECT) каждой задачи в очереди задач рабочего процесса, которая должна быть запланирована на виртуальной машине. Для того чтобы задача соответствовала своему крайнему сроку, время не должно его превышать.

Затем создается очередь крайних сроков (DI_{queue}) для повторной очереди всех поступивших задач в очередь задач рабочего процесса (wt_{queue}) на основе их крайних сроков. После этого крайний срок для каждой задачи сравнивается с ее EL, и если EL Wt_i на VM_k больше крайнего срока (специальный критерий), то задача будет разбита на подзадачи и отправлена в готовую очередь для решения проблемы сопоставления. Алгоритм завершается, когда все несопоставленные задачи рабочего процесса в очереди крайнего срока рассмотрены.

Исследование алгоритма проведено в среде WorkflowSim на типовых рабочих процессах, представленных на рис. 3.2.

Для сравнения результатов реализован известный алгоритм HSLJF и проведено его сравнение и предложенного СОНА с использованием трех реальных приложений рабочего процесса Montage, CyberShake и Sipht. Можно сделать вывод, что сложность предложенного метода является полиномиальной. Экспериментальные результаты, полученные в результате моделирования, показывают, что СОНА может сократить продолжительность планирования всех задач по сравнению с JSLJF, тем самым снижая стоимость выполнения на 32,5% для SIPHT (рис. 3.3), 3,9% для Montage и 1,2% для CyberShake.

Алгоритм СОНА

Input: DAG: $G=(W,E)$; W - n рабочих потоков задач ($wt_1, wt_2, \dots, wt_n$); VM ($vm_1, vm_2, \dots, vm_n$)

Begin

```

     $wt_{queue} = (wt_1, wt_2, \dots, wt_n)$ 
     $VM_{list} = (vm_1, vm_2, \dots, vm_n)$ 
    While  $wt_{queue} \neq \emptyset$  do
        Вычислить ECT
        Создать  $Dl_{queue}$  - очередь дедлайнов задач
    *   Foreach ECT для  $wt_i$  из  $vm_k$  do
        Сравнить ECT для  $wt_i$  из  $vm_k$  с их дедлайнами
        If  $ECT_{wt_i} > Dl_{wt_i}$  Then разбить задачу на две
        Else
            If  $ECT_{wt_i} \leq Dl_{wt_i}$ 
                Сопоставить задачи с VM
            **   Обновить  $Dl_{queue}$ 
                While  $Dl_{queue} \neq \emptyset$  do
                    Repeat шаги с * по **
                End While
            If  $Dl_{queue} = \emptyset$ 
                End Else
            End Foreach
        End While
    End

```

Рис. 3.1. Псевдокод предложенного алгоритма

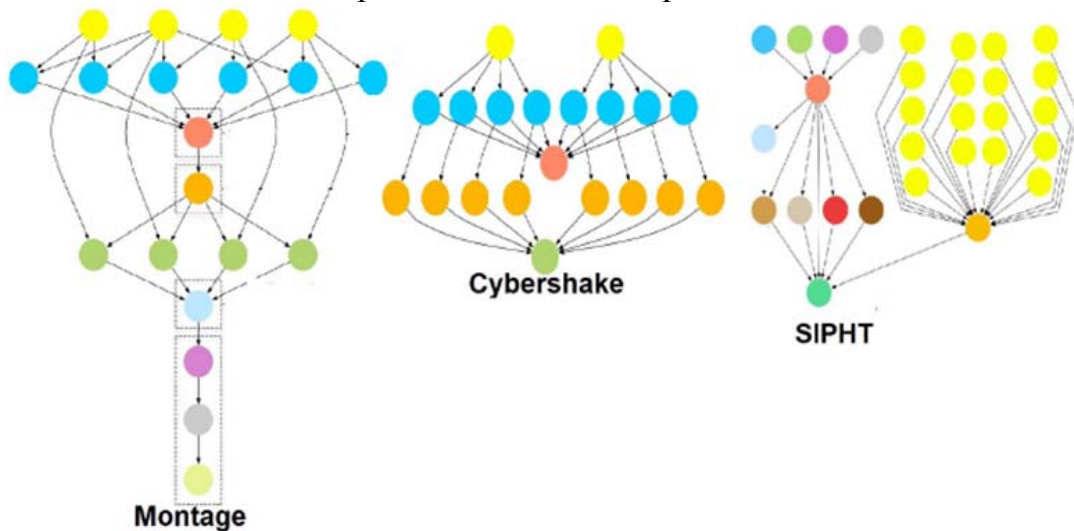


Рис. 3.2. Структура типовых рабочих процессов

Таким образом, представлен эвристический алгоритм планирования задач в облаке с полиномиальной по времени сложностью, отличающийся декомпозицией больших задач на подзадачи по специальному критерию и обеспечивающий соблюдение регламентных сроков исполнения задач.

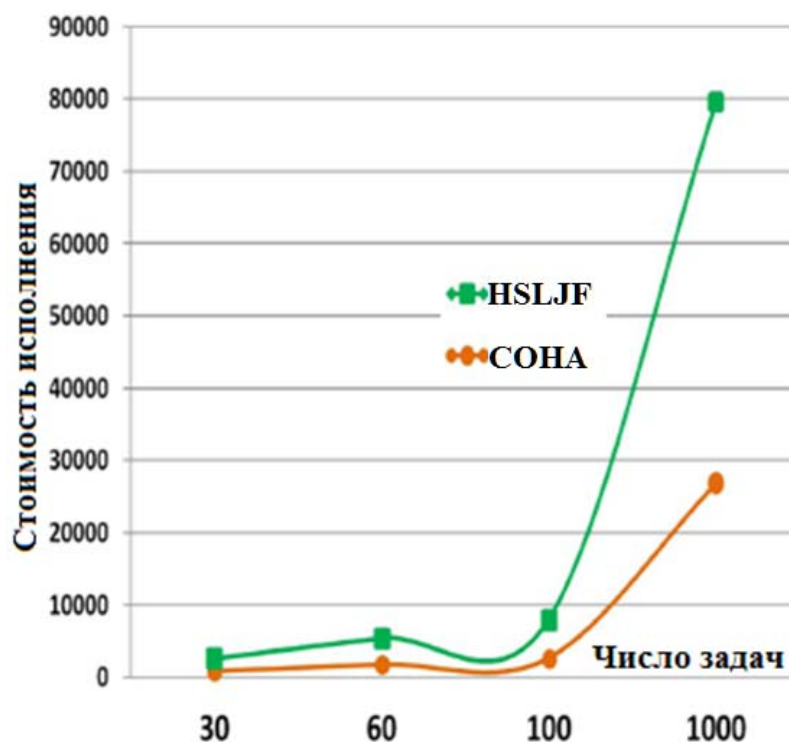


Рис. 3.3. Сравнение затрат на исполнение при различном количестве задач для рабочего процесса SIPHT

Как показано на рис. 3.4, облачная архитектура представляет собой набор центров обработки данных, и каждый центр обработки данных представляет собой конечное множество G кластеров S_k , связанных между собой, где каждый кластер содержит одну или несколько физических машин (PM), соединенных между собой коммутаторами, и в каждом PM работает несколько виртуальных машин (VM). Физические машины в кластерах неоднородны, поэтому нагрузка в одном кластере может быть очень высокой, в то время как в других кластерах может ничего не работать. Более высокая пропускная способность может быть достигнута, если добавить балансировку нагрузки между кластерами, а также балансировку нагрузки между физическими машинами внутри кластера.

Разработана иерархическая стратегия на двух уровнях: реализуется внутрикластерная и межкластерная балансировка: внутрикластерная балансировка нагрузки - запускается только тогда, когда некоторые менеджеры VM не могут локально сбалансировать перегрузку своих VM. Зная глобальное состояние каждого PM, менеджер может равномерно распределить глобальную перегрузку между своими физическими машинами.

Межкластерная балансировка нагрузки: на этом втором уровне выполняется глобальная балансировка нагрузки между всеми кластерами облачного центра обработки данных только в том случае, если на другом уровне не удастся достичь полного баланса нагрузки. Важно отметить, что на этом уровне алгоритм всегда успешно балансирует нагрузку на все эти кластеры.

Основное преимущество этой стратегии заключается в приоритизации локальной балансировки нагрузки сначала внутри кластера, а затем внутри центра обработки данных. Цель этой стратегии состоит в том, чтобы умень-

шить количество сообщений между физическими машинами. Как следствие этой цели, накладные расходы, вызванные новой стратегией, уменьшаются.

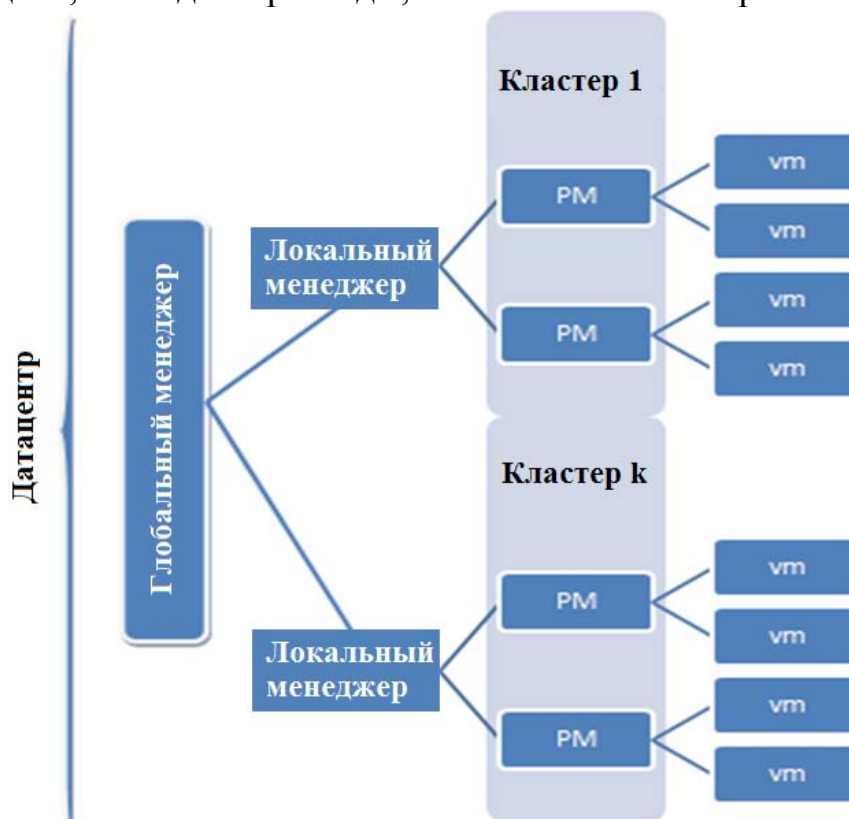


Рис. 3.4. Архитектура предлагаемой стратегии

В диссертации подробно рассмотрены оценка нагрузки и политика обмена информацией; определение состояния кластера; политика передачи процессов; классификация нагрузки физической машины и кластера.

Итак, представлена иерархическая стратегия балансировки нагрузки для облачных многокластерных центров обработки данных, отличающаяся приоритизацией локальной балансировки нагрузки сначала внутри кластера, а затем внутри центра обработки данных и обеспечивающая уменьшение среднего времени отклика и служебных издержек межкластерной коммуникации.

В главе 4 представлена структура программного прототипа системы управления распределением заданий в облачных средах, алгоритмической и программной реализации механизмов управления исполнением рабочих процессов в облачной инфраструктуре, основанные на полученных результатах.

Цель - оптимизировать время передачи данных при минимизации времени выполнения и повышении надежности. Подход предлагается в контексте облака IaaS с несколькими центрами обработки данных, где приложения рабочего процесса должны быть запланированы для выполнения в разных центрах обработки данных.

Приложение рабочего процесса (w) моделируется как ориентированный ациклический граф (DAG) $G = (N; E)$, где N - множество n узлов, обозначающих задачи t_i ($1 \leq i \leq n$), а E - набор направленных ребер. Ребро $e(i,j) \in E$

соответствует зависимости между задачей t_i и задачей t_j , в которой t_i является непосредственной родительской задачей t_j , а t_j - непосредственной дочерней задачей t_i . Дочерняя задача не может быть выполнена до тех пор, пока не будут выполнены все ее родительские задачи. Матрица данных размерности $n \times n$ представляет объем данных, которыми обмениваются задачи.

В облаке IaaS компьютерные ресурсы создаются в виде виртуальных машин (VM). Как правило, облачные провайдеры предлагают несколько типов виртуальных машин. Каждый тип vm_i определяется с точки зрения его вычислительной мощности CC_{vm_i} , а следовательно, линейно связанного с ней времени выполнения. Определим время вычислений $CT(t_i, vm_p)$ как расчетное время выполнения задачи t_i в виртуальной машине типа vm_p в соответствии с ее мощностью. I_{ti} аналогично определяется в операциях с плавающей запятой (FLOP) и вычисляется из уравнения (4.1):

$$CT(t_i, vm_p) = \frac{I_{t_i}}{CC_{vm_p}} \quad (4.1)$$

Рассмотрим U центров обработки данных в облачной инфраструктуре. Необходимо учитывать времена передачи $TT_{(i,j)}$ между двумя задачами, соответствующее весу ребра $(i,j) \in E$ в DAG, которое соответствует времени передачи данных из задачи t_i (выполняемой на vm_p , размещенной в дата-центре U_a) в задачу t_j на vm_k , в дата-центре U_b):

$$TT_{(i,j)} = \frac{data_{ij}}{Transfer_{rate_{(p,k)}}} \quad (4.2)$$

где $data_{ij}$ - размер выходных данных, полученных в t_i и переданных в t_j ; $Transfer_{rate_{(p,k)}}$ - скорости передачи между дата-центрами.

При выполнении двух взаимодействующих задач на одной виртуальной машине время передачи равно нулю, а при запуске на разных виртуальных машинах, размещенных в одном центре обработки данных, время передачи равно нулю. Таким образом, предлагается новая модель, описывающая несколько платформ дата-центров обработки данных. В модели общее время вычислений $TCT(t_i, vm_p)$ задачи в виртуальной машине вычисляется по (4.3):

$$TCT(t_i, vm_p) = CT(t_i, vm_p) + \sum_{i=1}^m s_m r_m TT_{(i,j)} \quad (4.3)$$

где m - число ребер, в которых t_i является родительской задачей; $s_m=0$, если t_i и t_j работают на одной и той же виртуальной машине или 1 в противном случае; $r_m=0$ vm_p и vm_k размещаются в одном центре обработки данных или 1 в противном случае.

Общее время вычислений $TCT(w)$ всех задач в рабочем процессе определяется из (4.4):

$$TCT(w) = \max \{TCT(t_i; vm_p)\} \quad (4.4)$$

В исследовании под надежностью будем понимать частоту отказов. Считаем, что отказы виртуальных машин и каналов связи в системе имеют пуассоновское распределение. Пусть λ_p обозначает интенсивность отказов

vm_p , а $\lambda_{L(k;p)}$ - интенсивность отказов линии связи $L(k;p)$ между vm_k и vm_p .

В модели надежность задачи t_i равна вероятности ее успешного выполнения на виртуальной машине, которой она назначена, и вероятности успешной передачи данных этой задаче от ее непосредственных предшественников. $R(t_i; vm_p)$ для t_i , выполненного или отображенного на vm_p , вычисляется как

$$R(t_i, vm_p) = e^{-\lambda_p CT(t_i, vm_p)} \quad (4.5)$$

Элемент $R_{(j,i)}$, соответствующий надежности передачи между t_j , выполняемой на vm_k , и t_i , выполняемой на vm_p , может быть вычислен так:

$$R_{(i,j)} = e^{-\lambda_{L(k,p)} TT_{(j,i)}} \quad (4.6)$$

С учетом (4.5) и (4.6) надежность $R(t_i)$ задачи t_i , может быть определена из (4.7), где $pred_{t_i}$ - множество задач-предшественников задачи t_i :

$$R(t_i) = \prod_{t_j \in pred_{t_i}} R(t_j, vm_p) R_{(i,j)} \quad (4.7)$$

Надежность рабочего процесса w :

$$R(w) = \prod_{t_i \in N} R(t_i) \quad (4.8)$$

Чтобы максимизировать качество планирования рабочего процесса, необходимо минимизировать:

$$R_{index}(\omega) = \sum_{vm_p \in VM} \sum_{t_i \in N} (\lambda_p CT(t_i, vm_p) + \lambda_{L(k,p)} TT_{(j,i)}) \quad (4.9)$$

Задача может быть формально определена следующим образом: найти расписание S для рабочего процесса w с минимальным $TCT(w)$ и максимальным $R(w)$ (таким образом, с минимальным $R_{index}(w)$).

Чтобы решить эту проблему, предлагается стратегия, основанную на популяционной метаэвристике PSO (алгоритм роя частиц). PSO имеет меньше параметров, чем другие метаэвристики, и быстро сходится. В работе используется дискретный вариант PSO под названием discrete particle swarm Optimization (DPSO). Каждое положение частицы - это возможное решение. В нашей модели частица имеет n измерений, соответствующих n задачам рабочего процесса. N задач должны быть назначены m виртуальным машинам, где $m < n$.

Подход, основанный на DPSO (RDPSO – рис. 4.1), стремится оптимизировать время выполнения и надежность одновременно. Для оценки решения с RDPSO, сверточная фитнес-функция выражается через обе метрики – нормализованные коэффициентами α_1 , α_2 время выполнения и надежность RDPSO:

$$Fitness = \alpha_1 \frac{TCT}{T_{max}} + \alpha_2 \frac{R_{index}}{R_{index_{max}}} \quad (4.10)$$

Для оценки эффективности предложенного подхода разработана имитационная модель, основанная на инструменте CloudSim для моделирования облака с несколькими платформами центров обработки данных. Решение оценено в различных конфигурациях платформ (3dcx3vm, 3dcx5vm, 5dcx3vm

и 5dсх5vm) с использованием четырех известных приложений. Время выполнения, переданные данные и общая надежность были сопоставлены с результатами, полученными алгоритмом RHEFT.

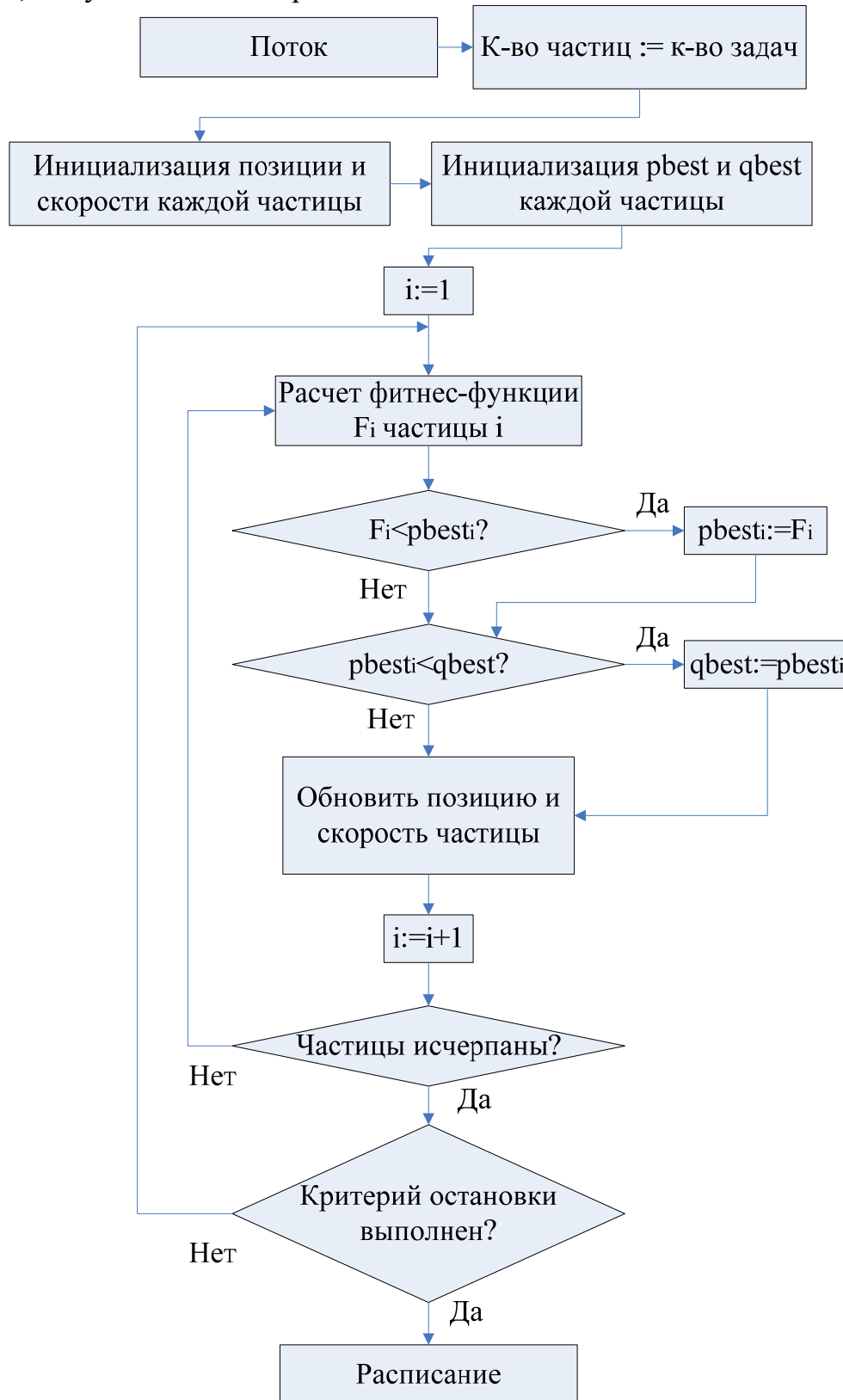


Рис. 4.1. Структурная схема алгоритма RDPSO

Экспериментальные результаты, представленные на рис. 4.2, представляют собой время выполнения для разного количества задач. Использовано

четыре приложения рабочего процесса с 30 (25 для Montage), 50 (60 для SIPHT) и 100 задачами. Улучшение времени для RDPSO составляет 28-30%, уменьшение вероятности отказов – от 20% и более.

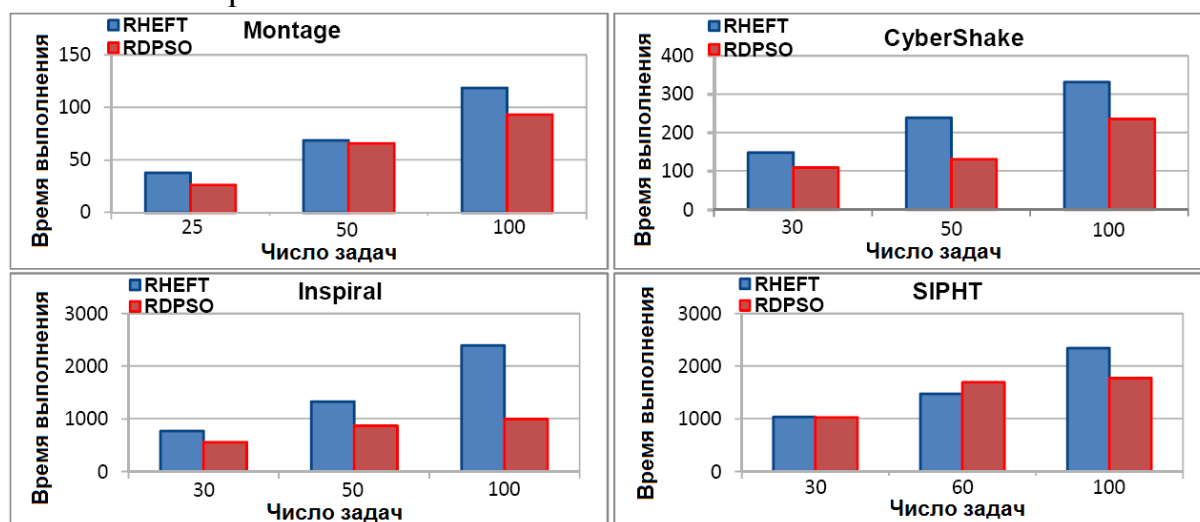


Рис. 4.2. Гистограммы времени выполнения для Montage, CyberShake, Inspiral и SIPHT, с 30, 50 и 100 задачами

Разработана структурная схема прототипа системы диспетчеризации заданий в облачных средах (4.3). Программная реализация прошла государственную регистрацию в ФИПС.



Рис. 4.3. Структура программного прототипа системы управления распределением заданий в облачных средах

Таким образом, создан алгоритм решения многокритериальной задачи управления распределением задач в облачных средах, отличающийся применением алгоритма роя частиц со сверточной фитнес-функцией и обеспечивающий оптимальное время выполнения и надежность как компьютерных ресурсов, так и сетевых связей.

Разработана структура программного прототипа системы управления распределением заданий в облачных средах, отличающаяся наличием подсистем обслуживания дополнительного миграционного слоя и декомпозицией больших задач на подзадачи, обеспечивающая выполнение требований к качеству обслуживания.

Заключение

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Создана модифицированная многоуровневая архитектура системы управления облачными средами с дополнительным миграционным слоем, обеспечивающая сокращение времени задержки, вызванной осуществлением миграции.

2. Разработан эвристический алгоритм планирования задач в облаке с полиномиальной по времени сложностью, обеспечивающий соблюдение регламентных сроков исполнения задач.

3. Создана иерархическая стратегия балансировки нагрузки для облачных многокластерных центров обработки данных, обеспечивающая уменьшение среднего времени отклика и служебных издержек межкластерной коммуникации.

4. Разработан алгоритм решения многокритериальной задачи управления распределением задач в облачных средах на основе алгоритма роя частиц, обеспечивающий оптимальное время выполнения и надежность как компьютерных ресурсов, так и сетевых связей.

5. Создана структура программного прототипа системы управления распределением заданий в облачных средах, отличающаяся наличием подсистем обслуживания дополнительного миграционного слоя и декомпозицией больших задач на подзадачи, обеспечивающая выполнение требований к качеству обслуживания. Элементы программного обеспечения зарегистрированы в ФИПС.

Основные результаты диссертации опубликованы в следующих работах:

Публикации в изданиях списка ВАК

1. Спицын А.А., Мутин Д.И. Распределение ресурсов и планирование заданий в облачной среде на основе алгоритма оптимизации роя частиц и R-фактора// Моделирование, оптимизация и информационные технологии. 2020; 8(4). <https://moitvvt.ru/ru/journal/pdf?id=869>. DOI: 10.26102/2310-6018/2020.31.4.023.

2. Спицын А.А. Оптимизированный по стоимости эвристический алгоритм для планирования рабочих процессов в облачной среде сервисов

IaaS// Системы управления и информационные технологии, №1(83), 2021. – С. 30-37.

3. Спицын А.А. Алгоритм настройки параметров расписания облачных вычислений на основе оптимизации роя частиц// Вестник Рязанского государственного радиотехнического университета, № 75, 2021. С. 44-52.

Публикации в изданиях, индексируемых в WoS и Scopus

4. Spitsyn A.A., Mutin D.I., Bikkulov I.M., Frantsisko O.Yu., Atlasov I.V. Multi-objective selection of structure variants for a corporate heterogeneous integrated system of information management// International Journal on Information Technologies & Security, № 1 (vol. 13), 2021, p. 27-38. **(WoS)**

5. Abdalov A.V., Grishakov V.G., Loginov I.V., Spitsyn A.A. Resource planning algorithm for reconfigurable information system development in the case of several sources of resources// AIP Conference Proceedings, 2402, 050012, <https://doi.org/10.1063/5.0071507>. **(Scopus)**

Свидетельство о государственной регистрации программы для ЭВМ

6. Спицын А.А., Львович И.Я., Зеленина А.Н. Диспетчерское управление распределением процессов в облачных вычислениях. – Свидетельство о регистрации программы для ЭВМ № 2021611994 от 10.02.2021. М.: ФИПС, 2021.

Статьи и материалы конференций

7. Spitsyn A.A., Mutin D.I. Approach to creating a hierarchical load balancing strategy in cloud structures// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2021'AS): Proceedings of the XXVI-th International Open Science Conference (Yelm, WA, USA, January 2021). - Yelm, WA, USA: Science Book Publishing House, 2021. – p. 167-175.

8. Спицын А.А. Результаты вычислительного эксперимента по планированию облачных вычислений на основе оптимизации роя частиц// Информационные технологии моделирования и управления, №1(123), 2021. – С. 68-79.

9. Спицын А.А. Исследование расширенной архитектурной модели облачных вычислений // Сб. ст. Междунар. НПК «Материалы и методы инновационных научно-практических исследований и разработок» (Пенза, 17.03.2021 г.). – Уфа: OMEGA SCIENCE, 2021. – с. 36-44.

10. Спицын А.А., Мутин Д.И. Оптимизация миграции виртуальных машин в облачных вычислениях с помощью эффективного алгоритма размещения// Матер. 16-й Международной научно-технической конференции «Современные инструментальные системы, информационные технологии и инновации» (МТО-56). – Курск, 2021. С. 251-256.

11. Спицын А.А. Оптимизация планирования в облачных вычислениях// IV Всероссийская научная конференция с международным участием «Информационные технологии в моделировании и управлении: подходы, методы, решения». – Тольятти, 2021. С. 315-323.

12. Spitsyn A.A. About optimization approaches to cloud load balancing//

Modern informatization problems in simulation and social technologies (MIP-2022'SCT): Proceedings of the XXVII-th International Open Science Conference (Yelm, WA, USA, January 2022). - Yelm, WA, USA: Science Book Publishing House, 2022. – p. 174-183.

13. Спицын А.А. Управление расписанием облачных вычислений на основе оптимизации роя частиц// Решение: матер. 11-й Всеросс. НПК. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2022. – С. 255-259.

14. Spitsyn A.A., Mutin D.I. Architecture of the prototype of the task distribution management system in cloud environments// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2023'AS): Proceedings of the XXIX-th International Open Science Conference (Yelm, WA, USA, January 2023). - Yelm, WA, USA: Science Book Publishing House, 2023.

Подписано в печать 24.03.2023.

Формат 60x84/16. Бумага для множительных аппаратов.

Усл. печ. л. 1,0. Тираж 80 экз. Заказ №

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14