

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС РАСПРЕДЕЛЕННОГО ПЛАНИРОВЩИКА РЕСУРСОВ ИНФОКОММУНИКАЦИОННОЙ СИСТЕМЫ ОБЛАЧНОГО ЦЕНТРА ОБРАБОТКИ ДАННЫХ

Тутов А. В.¹,

*(Московский технический университет связи
и информатики, Москва, ФГБУН Институт проблем
управления им. В.А. Трапезникова РАН, Москва)*

Фархадов М. П.², Таратухин А. В.³, Керимов С. С.⁴

*(ФГБУН Институт проблем управления
им. В.А. Трапезникова РАН, Москва)*

В связи с возрастающим спросом на облачные сервисы разработка новых эффективных методов и алгоритмов планирования ресурсов облачных центров обработки данных является актуальной задачей, о чем свидетельствует непрекращающийся поток работ, посвященных данной теме. Критерии лучшего распределения ресурсов могут быть различными, такие как энергоэффективность, выполнение соглашений об уровне сервиса, надёжность и другие. На основе проанализированных работ были выбраны и разработаны модели, методы и алгоритмы распределения ресурсов, комплекс которых положен в основу предложенной в данной статье архитектуры распределенного планировщика ресурсов инфокоммуникационной системы облачного ЦОД на основе многокритериальной оптимизации ее характеристик и особенностей живой миграции виртуальных машин. Эффективность использованных моделей и методов подтверждена имитационным моделированием. Показано, что предложенный комплекс моделей и методов распределенного планировщика ресурсов позволяет сократить энергопотребление при выполнении показателей качества обслуживания.

Ключевые слова: инфокоммуникационная система, планировщик ресурсов, центр обработки данных, размещение виртуальных машины, энергоэффективность.

¹ Андрей Владимирович Тутов, ст. преподаватель (andrew_vidnoe@mail.ru).

² Маис Паиа оглы Фархадов, д.т.н., с.н.с., г.н.с. (mais.farhadov@gmail.com).

³ Арсений Викторович Таратухин, инженер (avt@ipu.ru).

⁴ Сервер Сейранович Керимов, инженер (serverdevel@ya.ru).

1. Введение

В настоящее время наблюдается стремительная цифровизация всех отраслей экономики. На смену программы «Цифровая экономика Российской Федерации» приходит национальный проект «Экономика данных», в котором особое внимание отводится цифровой инфраструктуре на базе отечественных облачных вычислений и сервисов для хранения и обработки данных. Центры обработки данных (ЦОД), в которых размещаются серверное и сетевое оборудование, а также системы хранения данных, являются ядром инфраструктуры для обработки и хранения данных. Большая часть облачных платформ в настоящее время зависит от иностранного программного обеспечения. Поэтому разработка отечественных технологий и программного обеспечения, включая планировщиков облачных ресурсов, является актуальной задачей.

В настоящее время облачные провайдеры, работающие по модели IaaS для обеспечения виртуальных машин (ВМ) ресурсами, используют либо статические, либо динамические политики. В первом случае выделяется фиксированный набор физических ресурсов, рассчитанный с использованием алгоритмов упаковки в контейнеры. Тогда как динамические политики учитывают изменения нагрузки и посредством миграции виртуальных машин в реальном времени справляются с ними. Эти политики могут быть реактивными или проактивными и обычно основаны на знании требований к ресурсам виртуальных машин, предоставленных пользователем или оцененных с использованием данных систем мониторинга и прогнозирования [13]. Статические и динамические политики распределения ресурсов могут применяться совместно и включаться в основной цикл работ по управлению ресурсами облачного ЦОД [6].

Исследованию и разработке таких систем посвящено множество работ, и их число возрастает с каждым годом, что подтверждает актуальность данной проблемы. Некоторые из недавних обзоров приведены в [23, 33].

В работах [3, 15, 20, 30, 40] рассматривалось статическое размещение виртуальных машин с использованием постановки

оптимизационной задачи о рюкзаке или размещении в контейнерах. Эти задачи относятся к классу NP-трудных комбинаторных задач. Поэтому на практике для выбора целевого сервера для размещения мигрирующих ВМ широко используются жадные эвристические алгоритмы, такие как «Первый подходящий по убыванию» (FFD – First Fit Decreasing), «Наилучший подходящий по убыванию» (BFD – Best Fit Decreasing) и их модификации [10, 11, 24, 31], которые не гарантируют получения оптимальных решений.

В последнее время метаэвристические алгоритмы муравьиных колоний (АСО) и генетические алгоритмы оптимизации стали использоваться для решения проблемы упаковки контейнеров и консолидации виртуальных машин [18, 20, 22, 29, 34, 40]. Размещение виртуальной машины как задача оптимизации химических реакций предложено в [26]. Однако такие подходы не гарантируют получения оптимальных решений.

Некоторые работы учитывают только одномерный ресурс или критерий оптимизации. Проблемы многокритериального размещения рассматривались в работах [1, 6, 8, 10, 20, 22, 26, 29, 40, 41].

Потребление энергии и потери ресурсов являются наиболее распространенными критериями, которые моделировались во многих работах [1, 9, 11, 20, 29, 31, 32, 34, 35, 40]. В [19] предложена стратегия размещения виртуальных машин с учетом глобального энергопотребления, которая оптимизирует энергопотребление облачных центров обработки данных с учетом затрат на серверы, сеть и охлаждение. Размещение виртуальных машин с учетом температуры рассматривалось в работах [1, 40].

Во многих работах учитываются метрики качества обслуживания, такие как нарушения SLA из-за снижения производительности [1, 7, 40], балансировка нагрузки [24], минимизация трафика [17, 21]. В работе [11] предложен комбинированный критерий, который учитывает как уровень нарушений SLA, так и потребление энергии.

Доступность и безопасность сервисов также могут учитываться при размещении виртуальных машин [29, 34].

Для решения подобных многокритериальных задач чаще всего используют методы формирования обобщенного критерия [11, 20, 40, 41]. В работе [8] использовался метод анализа иерархии для выбора лучшего сервера, однако данный метод не подходит для получения решения в режиме реального времени. Время, необходимое для решения задачи оптимизации, является одним из основных факторов, влияющих на качество принятия решений в реальном времени. Один цикл работы контроллера длится несколько минут.

Задержки в принятии решений могут привести к значительным штрафам за нарушение соглашений SLA и дополнительным эксплуатационным расходам. Нерегулируемый рост задержек сделает невозможным внедрение инновационных высокодоходных услуг облачных дата-центров.

Несмотря на большое число исследований, ряд проблем и ограничений все еще остается. Например, множество противоречивых критериев, которые желательно учесть при управлении ресурсами, такие как энергопотребление, равномерность загрузки ресурсов, выполнение соглашений об уровне сервиса (SLA), минимизация трафика в условиях неравномерной и трудно прогнозируемой нагрузки.

Также существующие политики управления ресурсами, как правило, требуют точной оценки потребностей в ресурсах, поэтому необходимы исследования и поиск компромиссов между оптимальностью политики и ее устойчивостью к неточной информации о рабочей нагрузке [9].

Большинство обсуждаемых подходов не гарантируют точного решения и не полностью учитывают вопросы размерности и времени решения.

В данной работе предложена совокупность моделей, методов и алгоритмов, которая позволяет получить точное решение о размещении виртуальных машин независимо от масштаба ЦОД за период цикла работы контроллера, составляющего 2–5 минут. Разработанные методы и алгоритмы могут быть положены в основу программно-аппаратного комплекса распределенного планировщика ресурсов уровней инфраструктуры и платформы облачного ЦОД. В разделе 2 приведена типовая

структуры платформы облачных вычислений. В разделе 3 обсуждается архитектура распределенного планировщика ресурсов, в частности его структура, методы и алгоритмы блоков сбора данных, построения прогноза и принятия решений о размещении виртуальных машин. Также в данном разделе приведены результаты имитационного моделирования.

2. Типовая структура платформы облачных вычислений

Многоуровневая архитектура облачных вычислений в общем виде приведена на рис. 1 [13].

На нижнем уровне расположен **уровень инфраструктуры (IaaS)**, который состоит из двух подуровней: уровень аппаратуры – физические ресурсы ЦОД, такие как физические серверы, устройства хранения данных, компоненты сети. Второй подуровень – это уровень платформы виртуализации. На этом уровне создается пул виртуальных вычислительных ресурсов и хранилищ данных с использованием платформ виртуализации. На данном уровне необходимо эффективное распределение и использование виртуальных ресурсов в соответствии с потребностями пользователей облачных сервисов для минимизации нарушений соглашений об уровне сервиса (Service Level Agreement).

Уровень платформы (PaaS) расположен поверх уровня инфраструктуры, состоит из операционных систем и программных платформ, которые помогают автоматизировать процесс разработки приложений, их развертывания на виртуальных машинах и управления ими.

Уровень приложений (SaaS) состоит из самих облачных приложений, которые отличаются от традиционных приложений тем, что их можно автоматически или по требованию пользователя масштабировать для повышения эффективности работы, надежности, а также минимизации эксплуатационных затрат, поскольку оплата в облачных вычислениях взимается в зависимости от интенсивности использования ресурсов.

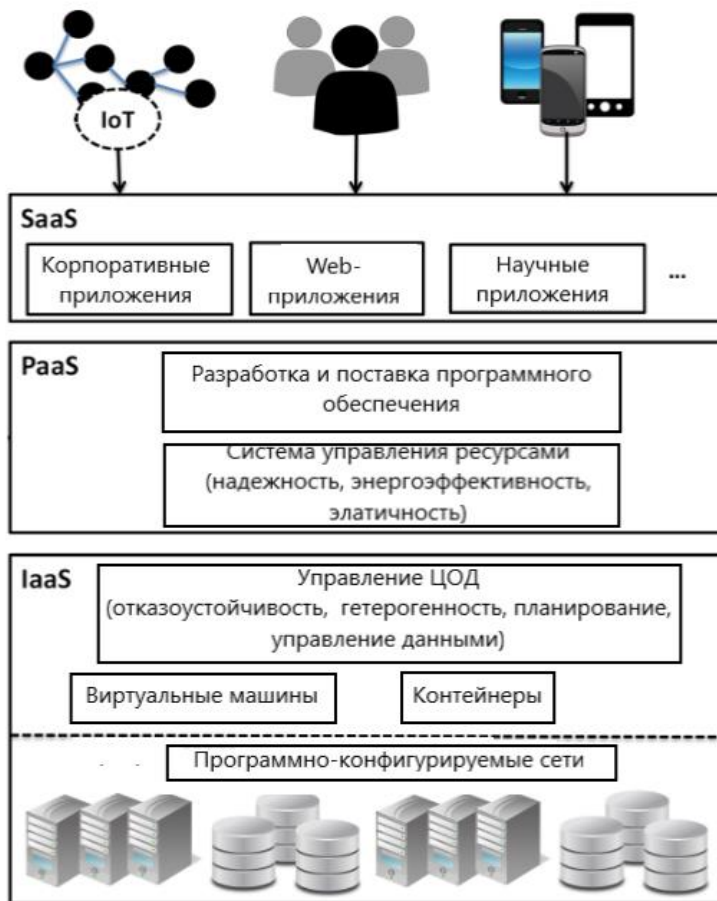


Рис. 1. Многоуровневая архитектура облачных вычислений

Масштабы современных ЦОД быстро растут, и на сегодняшний день они содержат от нескольких десятков до тысяч серверов, на которых размещаются сложные облачные приложения. Поэтому эффективное управление и планирование ресурсов является нетривиальной задачей, важной для повышения эффективности работы инфокоммуникационной системы ЦОД.

3. Двухуровневая архитектура планировщика ресурсов облачных центров обработки данных

3.1. СТРУКТУРА ПЛАНИРОВОЩИКА

Задачи планирования и управления облачными ресурсами возлагаются на планировщика ресурсов, в функции которого входят [1]:

- 1) первоначальное размещение виртуальных машин с заданными пользователями параметрами;
- 2) мониторинг загрузки ресурсов физических серверов и температуры;
- 3) выбор виртуальных машин для миграции в случае обнаружения недогруженных и перегруженных физических серверов;
- 4) выбор серверов для размещения мигрирующих машин;
- 5) запуск процесса миграции виртуальных машин.

Предлагаемая структура планировщика ресурсов приведена на рис. 2.

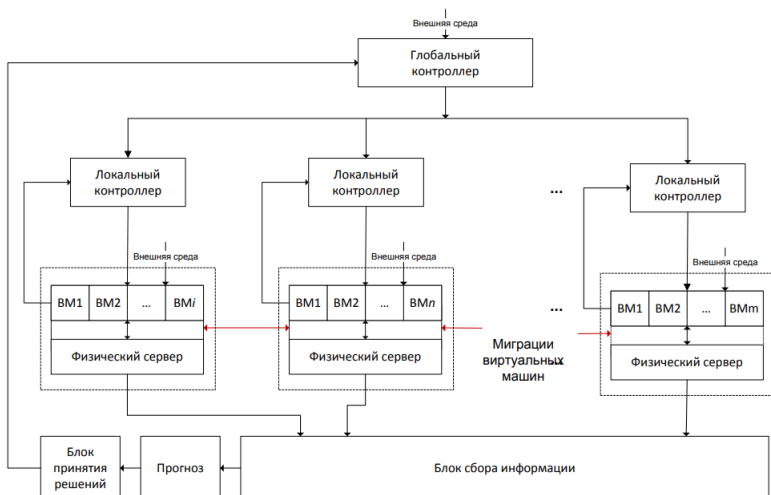


Рис. 2. Архитектура планировщика ресурсов

Планировщик ресурсов состоит из двух уровней контроллеров: локальных и глобального контроллеров. Каждый контроллер расположен на своем физическом хосте и интегрирован с платформой виртуализации. Локальные контроллеры расположены на серверах с виртуальными машинами. Они осуществляют мониторинг ресурсов серверов и виртуальных машин и выявляют потенциально опасные состояния физических серверов, связанных с перегрузкой или перегревом, а также состояние низкой энергоэффективности, когда физический сервер мало загружен. Данные о используемых ресурсах передаются в общее хранилище данных, доступ к которому имеет глобальный контроллер, у которого складывается все видение сети. На основании этой информации он принимает решение о миграции виртуальных машин с проблемного хоста и выбирает для них хосты назначения.

3.2. БЛОК СБОРА ДАННЫХ

В функции локального контроллера входит мониторинг серверов и выявление потенциально опасных состояний перегрузки и перегрева физических серверов, а также их недогрузки.

Для мониторинга используется метод скользящего окна. В соответствии с данным подходом, окно времени размером скользит по временному ряду, в котором измеряются l_w мгновенных значений загрузки процессора. Измеренные значения в одном окне усредняются. При достижении значения загрузки процессора порогового значения принимается решение о миграции виртуальных машин. Размер окна в широко используемом на практике планировщике ресурсов VMWare DRS для платформы виртуализации vSphere составляет 5 минут [25]. В [40] использовались двухминутное окно для наблюдения за загрузкой процессора и пятиминутное окно для наблюдения за температурой. Оптимальный размер скользящего окна для различных систем управления облачными ресурсами различается, но в работе [38] установлено, что он должен быть не менее чем в 3 раза больше средней длительности миграции виртуальных машин, чтобы процесс миграции не вносил помехи в наблюдение за серверами. Метод оценки общего времени миграции предложен

в [37]. Средние значения загрузки процессора, памяти, сети и температуры процессора каждого хоста сохраняются локальными контроллерами во внешнем хранилище данных, доступ к которому также имеет глобальный контроллер.

Метод «скользящего окна» совместно со статическими порогами может не уловить наличие тренда в изменении загрузки процессора. Поэтому необходимо прогнозировать значения температуры и загрузки процессора в ближайшем будущем, чтобы избежать ненужных миграций ВМ, когда превышение критического порога вызвано кратковременным всплеском нагрузки. В случае прогнозирования значений показателей за пределами пороговых значений локальный контроллер должен инициировать процесс миграции.

3.3. ПРОГНОЗ

Для выявления трендовой составляющей временного ряда во многих работах используются регрессионные методы. Так в работе [40] для прогнозирования перегрузок серверов использовался метод наименьших квадратов для поиска параметров линейной регрессии. Использование метода локальной регрессии обосновано в работе [11]. Кроме регрессионных моделей для прогнозирования нагрузки использовался метод ARIMA [43], авторегрессионная модель использовалась также в [39]. Для избавления от шумов и сглаживания данных в работах [28, 39] применялся фильтр Калмана. Для моделирования стационарных нагрузок разработана модель цепи Маркова [12]. Нейронные сети для прогнозирования перегрузок серверов использовались в работах [16, 42], модель иммунной сети построена в работе [8]. Однако для эффективного прогнозирования с использованием методов машинного обучения, как правило, необходим некоторый минимум наблюдений для обучения модели, так что для задачи прогнозирования перегрузки серверов с переменным числом виртуальных машин в режиме реального времени такое количество статистических данных может оказаться недоступным.

Поскольку облачная среда является сложной системой с обширным набором показателей, для прогноза его поведения

необходимы методы, учитывающие многообразие имеющихся параметров. По этой причине в данной работе предлагается использовать метод группового учета аргументов (МГУА) [4]. В отличие от нейронных сетей, где структура модели формируется на этапе обучения, в МГУА с каждой новой порции данных подбирается модель оптимальной структуры. МГУА обладает преимуществами перед нейронными сетями, когда отсутствует или почти отсутствует априорная информация о структуре модели и распределении ее параметров, а также когда число наблюдений крайне мало.

Общая структура модели задается обобщенным полиномом Колмогорова – Габора с ограниченной сложностью модели, которая определяется числом членов полинома, имеющего следующий вид [4]:

$$Y(x_1, \dots, x_M) = a_0 + \sum_{i=1}^M a_i x_i + \sum_{i=1}^M \sum_{j=1}^M a_{ij} x_i x_j + \sum_{i=1}^M \sum_{j=i}^M \sum_{k=j}^M a_{ijk} x_i x_j x_k + \dots,$$

где $X = (x_1, \dots, x_M)$ – вектор входных переменных (аргументов), а коэффициенты a неизвестны и их необходимо найти.

Параметры модели подбираются методом наименьших квадратов в соответствии с заданными значениями загрузки.

Для прогнозирования перегрузок серверов было разработано и зарегистрировано соответствующее программное обеспечение [2]. В общем виде алгоритм обнаружения перегрузок приведен в листинге 1. В предложенном алгоритме присутствует так называемый параметр безопасности, определяющий, насколько система чувствительна к перегрузкам. Значение параметра меньше единицы приводит к меньшей чувствительности к отклонениям загрузки процессора и соответственно к возможному увеличению нарушений SLA-соглашений.

2.5. БЛОК ПРИНЯТИЯ РЕШЕНИЙ

В случае выявления одного или нескольких недогруженных или перегруженных серверов глобальный контроллер должен составить новую карту размещения виртуальных машин в соответствии с определенными критериями.

На каждом физическом сервере в общем случае размещается несколько ВМ. Алгоритмы выбора ВМ для миграции исследовались в литературе: по минимуму оперативной памяти (ОП), случайный выбор, ВМ с максимальной корреляцией и др., среди них первый показал хорошие результаты, так как миграция такой ВМ с наименьшей ОП происходит быстрее всего [11].

Листинг 1.

```

Вход: порог, загрузка[], порядокМодели, параметрБезопасности,
        числоОкон
Выход: Является ли хост перегруженным
1 if  $len(загрузка) < числоОкон$  then
2   | return false;
3 end
4  $X \leftarrow 1:9$ ;
5 оценка  $\leftarrow$  MGUA(числоОкон, числоВходныхПеременных, X, загрузка[],
   числоТочекВОбучающейВыборке, числоТочекВПроверочнойВыборке,
   порядокМодели);
6  $x \leftarrow$  числоОкон;
7 switch порядокМодели do
8   | case 1 do
9     | прогноз  $\leftarrow$  оценка[0] + оценка[1]*x; break;
10  case 2 do
11   | прогноз  $\leftarrow$  оценка[0] + оценка[1]*x + оценка[2]*x2; break;
12  case 3 do
13   | прогноз  $\leftarrow$  оценка[0] + оценка[1]*x + оценка[2]*x2 + оценка[3]*x3;
   | break;
14 end
15 return параметрБезопасности * прогноз  $\geq$  порог

```

Требования к методу размещения виртуальных машин следующие [36]:

- для повышения энергоэффективности виртуальные машины следует размещать на меньшем числе физических серверов при соблюдении требований на качество обслуживания;

- следует учитывать как миграцию отдельных ВМ, так и групп виртуальных машин (кластеров);
- при миграции кластеров алгоритм должен попытаться разместить взаимосвязанные виртуальные машины на одном физическом сервере или близкорасположенных серверах, в лучшем случае в одной стойке;
- некоторым отдельным виртуальным машинам запрещено мигрировать;
- точное решение задачи оптимизации размещения ВМ должно быть получено за приемлемое время.

Выбор хоста назначения для мигрирующей ВМ в большинстве платформ осуществляется в соответствии с алгоритмом FFD: физические хосты упорядочиваются в соответствии с загрузкой ресурсов и выбирается первый незагруженный сервер.

Такой выбор позволяет сбалансировать нагрузку на серверы и обеспечить выполнение показателей качества, но не учитывает другие критерии, такие как энергопотребление и эффективность используемых ресурсов.

Для преодоления этих недостатков сформулируем следующую задачу оптимизации [6].

Пусть дано N серверов и N виртуальных машин для миграции. Для каждой ВМ задается производительность процессора VM_i^{CPU} , измеряемая в миллионах инструкций в секунду или, если все процессоры одинаковые, числом ядер. Объем оперативной памяти ВМ обозначим VM_i^{RAM} , $i = 1, \dots, N$.

Каждый физический сервер имеет свои характеристики: производительность процессора $PM_j^{CPU_0}$ и оперативную память ($PM_j^{RAM_0}$). Поскольку сервер может находиться в рабочем состоянии, часть ресурсов может быть занята другими ВМ. Занятую часть процессора обозначим как $PM_j^{CPU_1}$, а занятую часть памяти – $PM_j^{RAM_1}$. Предположим, что у любого физического сервера достаточно ресурсов для размещения любой виртуальной машины.

Глобальный контроллер должен определить серверы, на которые будут перемещены виртуальные машины. Поскольку размещение происходит динамически в течение одного цикла

контроллера, на каждом отдельном сервере можно разместить только одну виртуальную машину, так же как каждую виртуальную машину можно разместить только на одном сервере. Постановку задачи можно проиллюстрировать в виде полного двудольного графа (рис. 3).

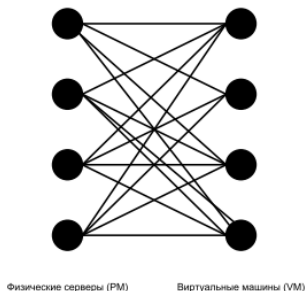


Рис. 3. Иллюстрация задачи размещения виртуальных машин по серверам в виде полного двудольного графа

Предположим, что виртуальная машина, размещенная на сервере, занимает всю выделенную ей память и время процессора. Обозначим u_{ij}^{CPU} загрузку ЦП сервера j после размещения VM i и u_{ij}^{RAM} – загрузку памяти сервера j после размещения VM i .

$$(1) \quad u_{ij}^{CPU} = \frac{PM_j^{CPU_1} + VM_i^{CPU}}{PM_j^{CPU_0}},$$

$$(2) \quad u_{ij}^{RAM} = \frac{PM_j^{RAM_1} + VM_i^{RAM}}{PM_j^{RAM_0}}.$$

Бинарные переменные задачи x_{ij} означают наличие или отсутствие размещения виртуальной машины i на сервере j .

Оптимизация производится по комбинированному критерию, являющемуся линейной комбинацией двух критериев неравномерности загрузки ресурсов и нарушения SLA-соглашений.

Критерий неравномерности загрузки ресурсов выглядит следующим образом:

$$(3) \quad f_{res}^j(u_{ij}^{CPU}, u_{ij}^{RAM}) = 1 - u_{ij}^{CPU} \cdot u_{ij}^{RAM}.$$

Этот критерий отражает, насколько полно используются ресурсы различных типов серверов. Значения этого критерия варьируются от 0 до 1. Чем ближе значение критерия к нулю, тем лучше используются ресурсы сервера.

Критерий нарушений SLA-соглашений выглядит следующим образом:

$$(4) \quad f_{SLA}^j(u_{ij}^{CPU}) = 1 - \frac{1}{1 + e^{-\left(u_{ij}^{CPU} - 0,8\right)}}.$$

Значение этой функции также принадлежит диапазону от 0 до 1. При достижении порога $u_{ij}^{CPU} = 0,8$ значение функции равно 0,5 и быстро увеличивается при превышении порогового значения. Этот критерий должен быть минимизирован.

Данные критерии перемножаются на неотрицательные веса α_1, α_2 , сумма которых равна 1. Веса отражают сравнительную важность критериев для лица, принимающего решения. Таким лицом может выступать системный администратор ЦОД. Математическая постановка задачи выглядит следующим образом:

$$(5) \quad \alpha_1 \sum_{i=1}^N \sum_{j=1}^N f_{res}^j(u_{ij}^{CPU}, u_{ij}^{RAM}) \cdot x_{ij} + \alpha_2 \sum_{i=1}^N \sum_{j=1}^N f_{SLA}^j(u_{ij}^{CPU}) \cdot x_{ij} \rightarrow \min_{x \in \Delta\beta}$$

где множество допустимых альтернатив $\Delta\beta$ формируется следующей системой ограничений:

$$(6) \quad \left\{ \begin{array}{l} \sum_{j=1}^N x_{ij} = 1 \quad \forall i \in \{1, 2, \dots, N\}, \\ \sum_{i=1}^N x_{ij} = 1 \quad \forall j \in \{1, 2, \dots, N\}, \\ x_{ij} \in \{0, 1\} \quad \forall i, j \in \{1, 2, \dots, N\}. \end{array} \right.$$

Сформулированная задача по структуре, необходимым условиям и характеру переменных эквивалентна известной

основной задаче о назначениях. Данная задача решается венгерским методом, который позволяет найти точное решение за время $O(N^3)$ [27]. Известно, что задача о назначениях может быть сведена к закрытой транспортной задаче путем замены последнего ограничения на $x_{ij} \geq 0$ [5].

В случае несимметричной задачи она может быть сведена к симметричной путем введения фиктивных серверов и виртуальных машин, по примеру транспортной задачи, к которой может быть сведена задача о назначении.

Предположение о том, что у каждого сервера достаточно ресурсов для размещения любой виртуальной машины, не всегда применимо. В этом случае граф назначений является неполным двудольным графом (рис. 4).

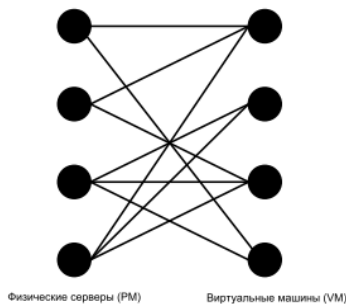


Рис.4. Иллюстрация задачи размещения виртуальных машин по серверам в виде неполного двудольного графа

В этом случае каждому коэффициенту целевой функции, соответствующему VM_i и PM_j , должно быть присвоено неэффективное большое число M , как это делается в симплекс-методе, чтобы данное назначение не было включено в решение задачи. Желательно выбрать число M равное N . Значение целевой функции при правильном размещении виртуальной машины всегда меньше N . Если значение целевой функции становится больше N , это означает, что ресурсов хоста недостаточно для обслуживания виртуальных машин и необходимо включить дополнительные физические серверы. Более того, можно быстро

определить, скольким виртуальным машинам не хватает места на физических серверах: для этого надо нацело разделить полученное значение целевой функции F на N .

Для оценки эффективности предложенного метода разработаны имитационные модели на платформе CloudSim [14], которая представляет собой набор основных классов, написанных на Java. Их можно использовать для построения модели среды облачных вычислений с желаемыми характеристиками. CloudSim поддерживает моделирование центров обработки данных, виртуализацию серверов с настраиваемыми политиками предоставления ресурсов виртуальным машинам, сетевые топологии, обмен сообщениями между приложениями. В режиме отладки и с помощью информационных сообщений можно контролировать ход моделирования.

Предлагаемый метод размещения ВМ встроен в модель энергоэффективного центра обработки данных, разработанную в [11]. Моделирование проводилось для центра обработки данных, состоящего из 100 физических серверов, половина из которых – HP ProLiant ML110G4, а другая половина – HP ProLiant ML110G5. Производительность ЦП измерялась в миллионах инструкций в секунду (MIPS) – 160 MIPS на каждое ядро для сервера HP ProLiant ML110G5 и 2660 MIPS для сервера HP ProLiant ML110C5. Каждый сервер имеет подключение к сети 1 Гбит/с. Задавалось энергопотребление серверов при разных уровнях нагрузки.

В процессе моделирования использовались четыре типа виртуальных машин, соответствующие экземплярам Amazon:

- 2500 MIPS, ОЗУ 0,85 ГБ;
- 2000 MIPS, 3,75 ГБ ОЗУ;
- 1000 MIPS, 1,7 ГБ ОЗУ;
- 500 MIPS, 613 ГБ ОЗУ.

Использовались реальные данные рабочей нагрузки, доступные в CloudSim [44]. Эти фрагменты рабочей нагрузки содержат данные об использовании процессора, собранные с более чем 1000 виртуальных машин проекта PlanetLab в течение случайно выбранных 10 дней весной 2010 года. Для проведения повторяемых экспериментов в разумные сроки в этой работе

были взяты только первые 150 фрагментов нагрузок виртуальных машин. Интервал измерения нагрузки – 5 минут.

Моделирование проводилось со всеми 10 рабочими нагрузками, включенными в CloudSim. Изменялись значения весовых коэффициентов критериев от 0,1 до 0,9 с шагом 0,1. Результаты сравнивались с алгоритмом FFD. Параметры нагрузки приведены в таблице 1 [11].

Таблица 1. Числовые характеристики нагрузки (загрузка процессора (%))

Дата	Число VM	Среднее	Станд. отклон.	Кварт. 1	Медиана	Кварт. 3
03/03/2011	1052	12,31%	17,09%	2%	6%	15%
06/03/2011	898	11,44%	16,83%	2%	5%	13%
09/03/2011	1061	10,70%	15,57%	2%	4%	13%
22/03/2011	1516	9,26%	12,78%	2%	5%	12%
25/03/2011	1078	10,56%	14,14%	2%	6%	14%
03/04/2011	1463	12,39%	16,55%	2%	6%	17%
09/04/2011	1358	11,12%	15,09%	2%	6%	15%
11/04/2011	1233	11,56%	15,07%	2%	6%	16%
12/04/2011	1054	11,54%	15,15%	2%	6%	16%
20/04/2011	1033	10,43%	15,21%	2%	4%	12%

Эффективность алгоритмов оценивалась по таким показателям, как общее энергопотребление центра обработки данных (E) и нарушение SLA-соглашений (SLAV), которые были объединены в комбинированную метрику $ESV = E \cdot SLAV$.

Всего было проведено 100 экспериментов. Результаты одного из них приведены на рис. 4, где показаны значения комбинированной метрики ESV для нагрузки на дату на 03/06/2011.

Во всех экспериментах предлагаемый алгоритм размещения виртуальных машин превосходит алгоритм FFD как по сравнению с частными критериями, так и по показателю ESV. На рис. 5 приведены ящичковые диаграммы, отражающие во сколько раз метрика ESV по предлагаемому алгоритму

в различных экспериментах меньше чем по алгоритму FFD в имитационных моделях с различными нагрузками.

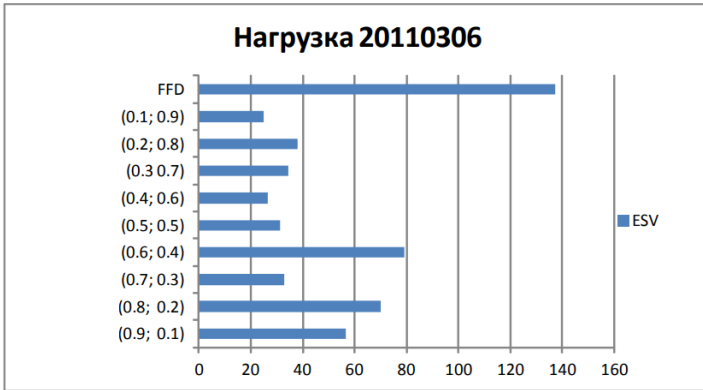


Рис. 4. Сравнение значений метрики *ESV* для нагрузки 06/03/2011 для алгоритма *FFD* и предлагаемого алгоритма с различными значения весовых коэффициентов

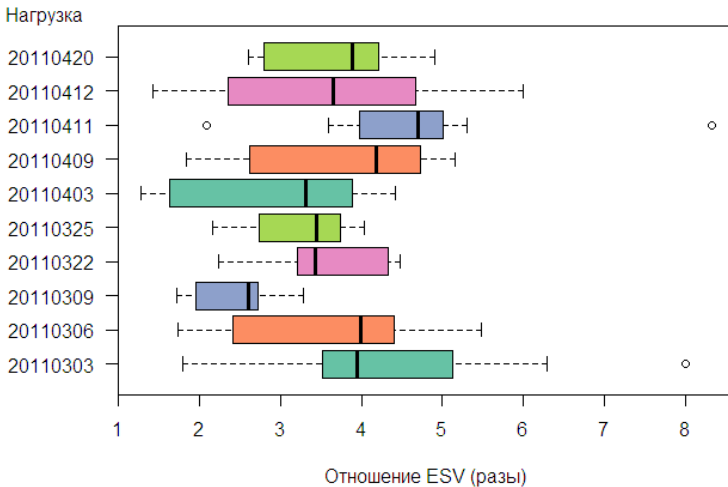


Рис. 5. Ящичковые диаграммы распределения отношения метрики *ESV* для алгоритма *FFD* к предлагаемому алгоритму для различных нагрузок

В целом моделирование показало, что предложенный метод размещения виртуальных машин позволяет получить в среднем в 3,5 раза лучшие результаты по комбинированной метрике ESV (энергопотребление и нарушения SLA-соглашений), чем используя распространенный на практике эвристический алгоритм FFD.

4. Заключение

В работе предложена совокупность методов и алгоритмов программно-аппаратного комплекса распределенного планировщика ресурсов в инфокоммуникационной системе облачного ЦОД с учётом оптимизации основных её характеристик на основе живой миграции виртуальных машин. В работе планировщика положен алгоритм прогнозирования перегрузки серверов на основе метода группового учета аргументов, что позволяет минимизировать число ненужных миграций виртуальных машин и тем самым повысить стабильность облачных сервисов. Разработанный метод многокритериального динамического размещения виртуальных машин по критериям нарушения SLA-соглашений и эффективности использования вычислительных ресурсов позволяет получить точное решение в режиме реального времени. Внедрение этих алгоритмов в планировщик ресурсов облачных ЦОД позволит уменьшить значение комбинированного критерия ESV (произведение энергопотребления и нарушения SLA-соглашений) в среднем в 3,5 раза по сравнению с широко распространённым на практике эвристическим алгоритмом FFD.

Литература

1. ВОРОЖЦОВ А.С., ТУТОВА Н.В., ТУТОВ А.В. *Динамическое распределение вычислительных ресурсов центров обработки данных // Т-Comm – Телекоммуникации и транспорт. – 2016. – Т. 10, №.7.*

2. ВОРОЖЦОВ А.С., ТУТОВА Н.В., ТУТОВ А.В. *Программа для прогнозирования перегрузки серверов с использованием комбинаторного метода группового учета аргументов на языке программирования Java*. – Свидетельство о регистрации программы для ЭВМ RUS 2018666780 07.12.2018.
3. ВОРОЖЦОВ А.С., ТУТОВА Н.В., ТУТОВ А.В. *Оптимизация размещения облачных серверов в центрах обработки данных* // Т-Comm – Телекоммуникации и транспорт. – 2015. – Т. 9, №6. – С. 4–8.
4. ИВАХНЕНКО А.Г., СТЕПАШКО В.С. *Помехоустойчивость моделирования*. – Киев: Наукова Думка, 1985. – 216 с.
5. КРОТОВ В.Ф., ЛАГОША Б.А., ЛОБАНОВ С.М. и др. *Основы теории оптимального управления*. – М.: Высшая школа, 1990. – 430 с.
6. ТУТОВ А.В. и др. *Многокритериальная оптимизация размещения виртуальных машин по физическим серверам в облачных центрах обработки данных* // Т-Comm – Телекоммуникации и транспорт. – 2021. – Т. 15, №1. – С. 28–34.
7. ТУТОВ А.В. *Модели и методы распределения ресурсов инфокоммуникационной системы облачных центров обработки данных* // Научные технологии в космических исследованиях Земли. – 2018. – Т. 10, №6. – С. 100–107.
8. ХАНТИМИРОВ Р.И. *Прогнозирование нагрузки в облачной вычислительной среде с использованием нейросетей Элмана, обучаемых системой искусственного иммунитета* // Нейрокомпьютеры: разработка, применение. – 2015. – №3. – С. 59–64.
9. ALHARBI F. et al. *An ant colony system for energy-efficient dynamic virtual machine placement in data centers* // Expert Systems with Applications. – 2019. – Vol. 120. – P. 228–238.
10. ALHAMMADI A.S.A., VASANTHI V. *Multi-objective algorithms for virtual machine selection and placement in cloud data center* // Int. Congress of Advanced Technology and Engineering (ICOTEN–2021). – IEEE, 2021. – P. 1–7.

11. BELOGLAZOV A., BUYYA R. *Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers* // Concurrency and Computation: Practice and Experience. – 2012. – Vol. 24, No. 13. – P. 1397–1420.
12. BELOGLAZOV A., BUYYA R. *Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints* // IEEE Trans. on Parallel and Distributed Systems. – 2012. – Vol. 24, No. 7. – P. 1366–1379.
13. BUYYA R. et al. *A manifesto for future generation cloud computing: Research directions for the next decade* // ACM computing surveys (CSUR). – 2018. – Vol. 51, No. 5. – P. 1–38.
14. CALHEIROS R.N. et al. *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms* // Software: Practice and experience. – 2011. – Vol. 41, No. 1. – P. 23–50.
15. CAMATI R.S., CALSAVARA A., LIMA JR L. *Solving the virtual machine placement problem as a multiple multidimensional knapsack problem* // ICN-2014. – 2014. – Vol. 264.
16. DINESH KUMAR K., UMAMAHESWARI E. *An efficient proactive VM consolidation technique with improved LSTM network in a cloud environment* // Computing. – 2024. – Vol. 106, No. 1. – P. 1–28.
17. FARZAI S., SHIRVANI M.H., RABBANI M. *Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters* // Sustainable Computing: Informatics and Systems. – 2020. – Vol. 28. – P. 100374.
18. FELLER E., RILLING L., MORIN C. *Energy-aware ant colony based workload placement in clouds* // IEEE/ACM 12th Int. Conf. on Grid Computing – 2011. – IEEE, 2011. – P. 26–33.
19. FENG H., DENG Y., LI J. *A global-energy-aware virtual machine placement strategy for cloud data centers* // Journal of Systems Architecture. – 2021. – Vol. 116. – P. 102048.

20. FERDAUS M.H. et al. *Virtual machine consolidation in cloud data centers using ACO metaheuristic* // Proc. of the 20th Int. Conf. Euro-Par–2014, Parallel Processing:, Porto, Portugal, August 25–29, 2014. 20. – Springer Int. Publishing, 2014. – P. 306–317.
21. FERDAUS M.H. et al. *An algorithm for network and data-aware placement of multi-tier applications in cloud data centers* // Journal of Network and Computer Applications. – 2017. – Vol. 98. – P. 65–83.
22. GAO Y. et al. *A multi-objective ant colony system algorithm for virtual machine placement in cloud computing* // Journal of computer and system sciences. – 2013. – Vol. 79, No. 8. – P. 1230–1242.
23. GILL S.S., BUYYA R. *Sustainable cloud computing realization for different applications: a manifesto* // Digital Business: Business Algorithms, Cloud Computing and Data Engineering. – 2019. – P. 95–117.
24. GULATI A. et al. *Vmware distributed resource management: Design, implementation, and lessons learned* // VMware Technical Journal. – 2012. – Vol. 1, No. 1. – P. 45–64.
25. HUEBSCHER M.C., MCCANN J.A. *A survey of autonomic computing—degrees, models, and applications* // ACM Computing Surveys (CSUR). – 2008. – Vol. 40, No. 3. – P. 1–28.
26. KIANI M., KHAYYAMBASHI M.R. *A network-aware and power-efficient virtual machine placement scheme in cloud datacenters based on chemical reaction optimization* // Computer Networks. – 2021. – Vol. 196. – P. 108270.
27. KUHN H.W. *The Hungarian method for the assignment problem* // Naval research logistics quarterly. – 1955. – Vol. 2, No. 1–2. – P. 83–97.
28. KUSIC D. et al. *Power and performance management of virtualized computing environments via lookahead control* // Cluster computing. – 2009. – Vol. 12. – P. 1–15.
29. LU J. et al. *Optimal machine placement based on improved genetic algorithm in cloud computing* // The Journal of Supercomputing. – 2022. – P. 1–29.

30. LUO J.Y. et al. *A cut-and-solve algorithm for virtual machine consolidation problem* // Future Generation Computer Systems. – 2024. – Vol. 154. – P. 359–372.
31. MOGES F.F., ABEBE S.L. *Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework* // Journal of Cloud Computing. – 2019. – Vol. 8, No. 1. – P. 2.
32. MURTAZAEV A., OH S. *Sercon: Server consolidation algorithm using live migration of virtual machines for green computing* // IETE Technical Review. – 2011. – Vol. 28, No. 3. – P. 212–231.
33. RANI K., SANGWAN O.P., GARG R. *A critical review on energy efficient Rani scheduling techniques in cloud computing* // AIP Conference Proc. – AIP Publishing, 2023. – Vol. 2938, No. 1.
34. SAXENA D. et al. *A secure and multiobjective virtual machine placement framework for cloud data center* // IEEE Systems Journal. – 2021. – Vol. 16, No. 2. – P. 3163–3174.
35. SHAW R., HOWLEY E., BARRETT E. *An energy efficient anti-correlated virtual machine placement algorithm using resource usage predictions* // Simulation Modelling Practice and Theory. – 2019. – Vol. 93. – P. 322–342.
36. TOUTOV A.V. et al. *Resource Allocation Algorithms for Single, Cluster and Tired Virtual Machines* // Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED–2023). – IEEE, 2023. – P. 1–4.
37. TOUTOV A., VOROZHTSOV A., TOUTOVA N. *Analytical approach to estimating total migration time of virtual machines with various applications* // Int. Journal of Embedded and Real-Time Communication Systems (IJERTCS). – 2020. – Vol. 11, No. 2. – P. 58–75.
38. VOROZHTSOV A. S., TOUTOVA N.V., TOUTOV A.V. *Resource control system stability of mobile data centers* // Systems of Signals Generating and Processing in the Field of on Board Communications. – IEEE, 2018. – P. 1–4.
39. WU Y. et al. *Load prediction using hybrid model for computational grid* // 8th IEEE/ACM Int. Conf. on Grid Computing. – IEEE, 2007. – P. 235–242.

40. XU J., FORTES J. *A multi-objective approach to virtual machine management in datacenters* // Proc. of the 8th ACM Int. Conf. on Autonomic Computing. – 2011. – P. 225–234.
41. XU J., FORTES J. *Multi-objective virtual machine placement in virtualized data center environments* // IEEE/ACM Int. Conf. on Green Computing and Communications & Int. Conf. on Cyber, Physical and Social Computing. – IEEE, 2010. – P. 179–188.
42. YANG Q. et al. *A new method based on PSR and EA-GMDH for host load prediction in cloud computing system* // The Journal of Supercomputing. – 2014. – Vol. 68. – P. 1402–1417.
43. ZHANG Q. et al. *Dynamic energy-aware capacity provisioning for cloud computing environments* // Proc. of the 9th Int. Conf. on Autonomic computing. – 2012. – P. 145–154.
44. URL: <https://github.com/Cloudslab/cloudsim/tree/master/modules/cloudsim-examples/src/main/resources/workload/planetlab> (дата обращения: 05.05.2024).

ARCHITECTURE OF A DISTRIBUTED SOFTWARE AND HARDWARE COMPLEX RESOURCE SCHEDULER OF INFOCOMMUNICATION SYSTEM OF CLOUD DATA CENTER

Andrew Toutov, Moscow Technical University of Communications and Informatics, Moscow, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, postgraduate (andrew_vidnoe@mail.ru).

Mais Farhadov, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Doctor of Science, professor (mais.farhadov@gmail.com).

Arsenij Taratukhin, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, engineer (avt@ipu.ru).

Server Kerimov, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, engineer (serverdevel@ya.ru)

Abstract: With the development of cloud technologies, the development of methods and algorithms for a resource scheduler for cloud data centers is an urgent task, as evidenced by the continuous flow of works devoted to this topic. The criteria for the best allocation of resources can be different, such as energy efficiency, fulfillment of service level agreements, reliability and others. Based on the analyzed works, mod-

els, methods and algorithms for resource distribution were selected and developed, the complex of which forms the basis of the distributed resource scheduler architecture proposed in this article for an infocommunication system of a cloud data center based on multi-criteria optimization of its characteristics and features of live migration of virtual machines. The effectiveness of the models and methods used has been confirmed by simulation modeling and makes it possible to reduce energy consumption when meeting quality of service indicators.

Keywords: infocommunication system, resource scheduler, data processing center, virtual machine placement, energy efficiency.

УДК 004.41

ББК 32.972

DOI: 10.25728/ubs.2024.109.12

*Статья представлена к публикации
членом редакционной коллегии А.А. Печниковым.*

*Поступила в редакцию 22.02.2024.
Опубликована 31.05.2024.*