

УПРАВЛЕНИЕ РАСПРЕДЕЛЕНИЕМ ВИРТУАЛЬНЫХ МАШИН В ЦОД

Широкое внедрение парадигмы облачных вычислений привело к необходимости повышения эффективности управления ресурсами центров обработки данных. Учитывая частые изменения нагрузки на виртуальные машины и различные условия их функционирования, задачи размещения виртуальных машин и их миграции должны решаться в режиме онлайн. Для решения такого рода задач в статье предлагается подход к размещению новых виртуальных машин и миграции функционирующих. В работе оценена целесообразность использования адаптивного генетического алгоритма при управлении размещением виртуальных машин. Предложен адаптивный программно-определяемый подход к управлению виртуальными машинами центра обработки данных, который основывается на выборе политик управления ресурсами и виртуальными машинами с целью адаптации к влиянию возмущающих воздействий.

A widespread use of the cloud computing paradigm has increased the necessity and significance of improving the management efficiency of cloud data centers. Special attention is paid to solving cloud resource management problems. Due to the intensive changes of virtual machine workloads and different conditions of resource utilization the virtual machine placement and migration problems should be solved and optimized continuously in an online manner. To address such problems the authors present an approach to continuous new virtual machine allocation and virtual machine migration. The authors also evaluate a particular policy of the virtual machine allocation in a data center using an adaptive genetic algorithm. The proposed Adaptive Software Defined approach to the cloud data centers management is implemented by using the policy selector, that allows to select different algorithms or policies for resources and virtual machines management in order to adapt to the impact of disturbing influences.

Ключевые слова: облачные вычисления, программно-определяемая система, ИТ-инфраструктура, управление ресурсами, виртуализация, центр обработки данных.

Введение

Парадигма облачных вычислений кардинально меняет не только взгляд на предоставление ИТ-услуг, но и на понимание самих ИТ-услуг, ведь как ИТ-услуги в настоящее время рассматриваются ПО, платформы, ИТ-ресурсы и пр. Это, в свою очередь, потребовало изменений и принципов организации инфраструктуры предоставления ИТ-услуг. В современных центрах обработки данных (ЦОД), являющихся базой для предоставления облачных вычислений, используются технологии виртуализации, программно-определяемые сети, инновационные подходы в области обработки и хранения информации, а также другие решения, позволяющие предоставлять разнообразные ИТ-услуги пользователям и заказчикам. При этом существенно возрастает роль системы управления ИТ-инфраструктурой (СУИ) ЦОД, которая должна не только поддерживать качество ИТ-услуг на согласованном уровне при изменяющейся динамике запросов многочисленных

пользователей, постоянно растущей сложности информационных процессов, непрерывно увеличивающихся объемах обрабатываемой информации, но и эффективно использовать дорогостоящие вычислительные и коммуникационные ресурсы ЦОД.

При решении задачи повышения эффективности управления СУИ должна учитывать следующие особенности ИТ-инфраструктуры ЦОД:

- большое количество ресурсов каждого типа или вида;
- на серверах развертываются виртуальные машины (ВМ), на которых устанавливаются приложения, поддерживающие сервисы;
- ИТ-инфраструктура является более статичной и менее подвержена изменениям, чем приложения и программные средства;
- изменения в ИТ-инфраструктуре инициируются необходимостью изменений на уровне приложений и бизнес-процессов.
- каждый сервер используется одновременно несколькими приложениями;

– когда поверх ИТ-инфраструктуры ЦОД функционируют приложения заказчиков, то возможен конфликт интересов администраторов серверов и администраторов приложений.

Исследования [1, 2] показывают, что современные подходы и методы управления ЦОД должны учитывать использование программно-определяемых технологий, непредсказуемую интенсивность нагрузок и рост потребления ИТ-ресурсов при появлении новых сервисов. Операционные затраты на управление ресурсами в современных ЦОД являются большими и постоянно возрастают [3], а применение парадигмы программно-определяемых устройств на всех уровнях управления позволяет уменьшить расходы на администрирование, расширить возможности интеграции и обеспечить масштабирование СУИ.

При создании СУИ необходимо разрабатывать модели и решать задачи предоставления, распределения, сопоставления, масштабирования и оценки ИТ-ресурсов. Это необходимо делать с учетом того, что провайдеры облачных услуг, с одной стороны, несут ответственность за соответствие текущих значений показателей качества предоставляемых услуг согласованным с заказчиком значениям (например, времени отклика, задержки, вероятности потери запросов и др.), с другой стороны, заинтересованы в предоставлении высококачественных услуг с минимальным использованием полосы пропускания каналов связи, дискового пространства для хранения данных и других задействованных для предоставления услуг ИТ-ресурсов также с минимальным расходом энергоресурсов.

В то же время, поставщики услуг обязаны предоставлять ИТ-услуги с показателями качества, зафиксированными в соответствующем соглашении об уровне услуг (SLA). Для выполнения требований SLA провайдеры услуг часто вынуждены резервировать ресурсы, запускать дополнительные экземпляры приложений и проводить другие мероприятия, предусматривающие избыточное использование ИТ-ресурсов. Оплату за такую избыточность провайдеры перекладывают на своих клиентов, что в свою очередь приводит к повышению стоимости услуг и делает таких провайдеров неконкурентоспособными. В данном случае в выигрыше оказываются те провайдеры, которые задействуют в СУИ механизмы, отслеживающие тенденции увеличения запросов пользователей и динамически изменяющие объемы ресурсов по результатам прогнозирования. Таким образом, планирование

ресурсов и политика их распределения непосредственно влияют на стоимость облачных услуг.

Снижение качества ИТ-услуги, произошедшее, например, в результате увеличения количества запросов пользователей, может быть скомпенсировано увеличением объемов ресурсов, выделенных этой услуге. Поэтому при создании СУИ особое внимание уделяется решению задач управления ресурсами ЦОД с учетом частых изменений нагрузки на ВМ и различных условий функционирования ВМ. Причем задачи размещения ВМ и их миграции должны решаться в режиме онлайн.

Традиционные подходы к управлению ресурсами ЦОД и, в частности, к управлению ВМ не учитывают в полной мере динамику происходящих процессов, и обычно вырабатывают управляющие воздействия, когда инциденты и проблемы уже возникли. В работе предложен подход к управлению размещением и миграцией ВМ, который основан на учете динамики процессов, происходящих в ЦОД, при развертывании новых ВМ одновременно с миграцией уже работающих ВМ.

Значительное влияние на эффективность процесса управления ресурсами ЦОД оказывает использование исторических данных и предсказание потребности в ресурсах на краткосрочную или среднесрочную перспективу. В предложенном подходе используется среднесрочное предсказание в потребности ресурсов физического сервера и учитывается интенсивность потребления ресурсов каждой ВМ, что позволяет повысить качество принимаемых решений, связанных с миграцией ВМ и размещением новых ВМ.

Таким образом, целью статьи является разработка подхода к управлению виртуальными машинами с учетом прогноза потребностей в ресурсах.

Анализ публикаций

В работах [4, 5, 6] отмечается, что в настоящее время учеными и технологическими компаниями прилагаются значительные усилия для разработки методов, алгоритмов и систем управления ресурсами ЦОД, предоставляющих облачные услуги. Причем исследования сосредоточены в основном в направлениях предоставления, распределения, планирования емкости, сопоставления и масштабирования ресурсов облачных и виртуализированных ЦОД.

При этом предложено большое количество систем с использованием программно-определяемых технологий и программных платформ для предоставления и мониторинга ресурсов, включая точные методы, эвристики, механизмы предсказания нагрузки, методы учета разбалансированности нагрузки физического сервера, взаимовлияния ВМ, расположенных на одном физическом сервере и пр. Большинство из предложенных подходов для управления ресурсами в облачных и виртуализированных ЦОД основаны на использовании принципа консолидации ресурсов и ВМ, а также на использовании минимального количества физических серверов при обслуживании запросов клиентов. Причем физические серверы, на которых не осталось работающих ВМ, переводятся в режим минимального энергопотребления или отключаются.

Для управления ресурсами и ВМ в режиме онлайн с учетом динамики изменений среды в современных облачных решениях широко применяется механизм миграции ВМ [7]. В [8] предложена система для автоматизации мониторинга, выявления перегруженных элементов ИТ-инфраструктуры, определения нового плана размещения ВМ на физических серверах и выполнения миграции ВМ согласно этому плану. Однако применяемые в этой системе фиксированные пороги для определения степени утилизации ресурсов не позволяют эффективно управлять ресурсами в условиях смешанных нагрузок и нестационарных моделей использования ресурсов.

В работе [9] предложен алгоритм управления миграцией ВМ, минимизирующий число физических серверов, необходимых для обслуживания нагрузок на указанном в SLA уровне. Авторы используют оценочную модель для обычного предсказания необходимости будущего потребления ресурсов, поэтому их алгоритм не учитывает число миграций ВМ.

В работе [10] предложена автоматизированная система управления емкостью и нагрузкой, которая интегрирует несколько контроллеров ресурсов в трех разных областях и временных масштабах.

В [11] предложена система, позволяющая адаптивно подстраивать выделенные для различных ВМ ресурсы в условиях облачных вычислений. Система использует онлайн модель предсказания потребности в ресурсах для определения краткосрочных потребностей в ресурсах. Предложенный подход использует специ-

альную программу внутри каждой ВМ, что недопустимо для многих систем и платформ облачных вычислений.

Анализ проблемы эффективной динамической консолидации ВМ проведен в работе [12]. Рассмотрены проблемы онлайн, офлайн, детерминированной и динамической консолидации ВМ и приведены адаптивные эвристики для реализации динамической консолидации ВМ. Показано, что для решения проблемы обеспечения энергоэффективности и достижения заданной производительности в процессе консолидации ВМ адаптивные эвристики работают лучше, чем оптимальный онлайн-детерминированный алгоритм. Предложенная модель управления включает в себя глобальный и локальный менеджеры. Разработанный алгоритм вырабатывает новый план размещения ВМ по физическим серверам с учетом недогруженных и перегруженных физических серверов. План миграции генерируется алгоритмом в дискретном режиме через заданные интервалы времени.

В исследовании [13] представлена мультиагентная система DRPM (Dynamic Resources Provisioning and Monitoring) которая предназначена для управления облачными ресурсами провайдера с соблюдением требований SLA. Система включает в себя алгоритм для определения ВМ, которая является претендентом на миграцию. При этом задействуются глобальный агент и множество локальных агентов. Модель системы DRPM разработана в среде моделирования CloudSim, которая широко используется для оценки работы алгоритмов и методов управления ресурсами в условиях виртуализации и облачных вычислений. Результаты моделирования показали, что система DRPM позволяет увеличить утилизацию ресурсов провайдера, снизить потребление электроэнергии и избежать нарушений требований SLA. Однако система работает в дискретном режиме, составляя план миграции через фиксированные промежутки времени путем реализации трех фаз: мониторинг, анализ, выполнение. Кроме того, использование локальных агентов внутри ВМ часто недопустимо в условиях облачных вычислений.

В предыдущих исследованиях авторов [14-17] разработаны теоретические и практические основы создания систем управления ИТ-инфраструктурой, включая декомпозиционно-компенсационный подход, двухуровневую модель системы управления ресурсами с координатором, адаптивный генетический алгоритм и

концепцию управления корпоративной ИТ-инфраструктурой. Статья продолжает теоретические исследования в области управления ресурсами ЦОД с акцентом на разработку алгоритмов и политик управления ресурсами и ВМ, обладающих свойством адаптации к возмущающим воздействиям, с учетом динамики развертывания новых ВМ и процессов миграции ВМ, которые происходят одновременно, а также учетом тенденций потребности в ресурсах для достижения энергетической эффективности.

Модель системы управления ЦОД

Провайдер облачных услуг предоставляет различные по содержанию и назначению сервисы для клиентов. Клиенты провайдера могут предоставлять некоторые сервисы для других клиентов путем реализации многопользовательского многозвенного ЦОД, который представлен сложной сетевой и вычислительной ИТ-инфраструктурой. Между провайдерами облачных услуг заключаются соглашения по предоставлению услуг на базе нескольких ЦОД. Для этого вырабатываются соответствующие политики, которые согласовывают предоставление ресурсов по требованию.

Для предоставления ИТ-сервисов провайдеры услуг обычно используют одну или несколько ВМ, которые размещаются на физических серверах. При ухудшении качества услуг вследствие увеличения нагрузки на ВМ используются методы горизонтального или вертикального масштабирования. Для такой схемы предоставления ИТ-услуг предлагается модель системы управления, приведенная на рис. 1.

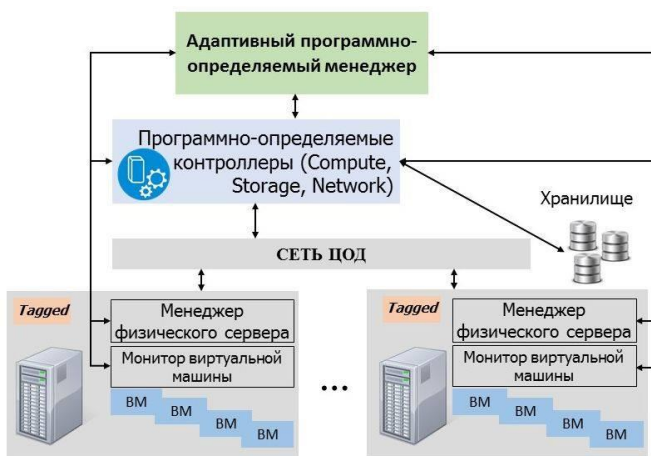


Рис. 1. Модель системы управления ЦОД

Объект управления представлен множеством физических серверов, каждый из которых ха-

рактеризуется соответствующим аппаратным обеспечением с установленной операционной платформой и имеет фиксированную емкость ресурсов, которая характеризуется производительностью процессоров, объемом ОЗУ и ПЗУ, пропускной способностью сетевого подключения, производительностью работы дисковой подсистемы и др. На каждом сервере размещается одна или несколько ВМ. Серверы соединяются в группы и, в конечном итоге, образуют вычислительную базу ЦОД. Коммуникационная сеть ЦОД обеспечивает взаимодействие серверов и в настоящее время создается с использованием программно-определяемых технологий. Сервера физически группируются в виде стоек и подключаются к коммутатору стойки (ToR Switch).

ЦОД с поддержкой облачных услуг ориентирован на обработку пакетной и транзакционной нагрузки. Проблемы обеспечения гетерогенной нагрузки решаются путем использования моделей IaaS, PaaS and SaaS [18]. Модель IaaS является основой для предоставления базовых сервисов виртуализации и функционирования моделей PaaS и SaaS. Для хранения данных о сервисах и ВМ предлагается использовать централизованное хранилище наряду с распределенным хранилищем [19, 20, 21].

Предоставление дополнительных ресурсов осуществляется исходя из наличия свободных ресурсов. Запросы клиентов на увеличение объемов предоставляемых ресурсов должны удовлетворяться без ущерба для остальных клиентов. При этом владелец ЦОД всегда стремится избежать чрезмерного выделения ресурсов, которые в текущий момент времени не используются.

Существующая практика резервирования ресурсов с запасом для того, чтобы упредить нежелательные последствия, вызванные нехваткой ресурсов при растущих потребностях клиентов и росте рабочей нагрузки, приводит к увеличению эксплуатационных затрат и расходов на электроэнергию. Поэтому в СУИ следует использовать методы и алгоритмы, позволяющие сократить объемы зарезервированных ресурсов без снижения качества предоставляемых ИТ-сервисов и исключения случаев нарушения SLA.

Подход к адаптивному размещению виртуальных машин. Основной задачей при реализации управления облачным ЦОД является эффективное перераспределение ресурсов между пользователями услуг в условиях аварий,

неисправностей, непредсказуемых изменений нагрузки, увеличения численности клиентов и количества их запросов. Перераспределение ресурсов должно происходить так, чтобы не было случаев нарушения SLA при минимизации потребления электроэнергии

Для решения задачи управления ресурсами и ВМ предлагается использовать программно-определяемый подход с комбинацией централизованного и децентрализованного управления. Использование такого подхода к управлению ЦОД позволяет выбирать алгоритмы или политики управления ресурсами и ВМ таким образом, чтобы осуществлялась адаптация к воздействию таких факторов, как изменение нагрузки, установка дополнительных обновлений, развертывание нового программного обеспечения и аппаратных платформ, внесение изменений в структуру и производительность предоставляемых услуг.

Адаптивный программно-определяемый менеджер. В предлагаемой СУИ адаптивный программно-определяемый менеджер (АПОМ) осуществляет централизованное управление облачным ЦОД путем управления физическими и виртуальными ресурсами и выбором политик управления для адаптации к внешним факторам.

В АПОМ реализован адаптивный программно-определяемый подход, который основан на выборе политики управления в зависимости от текущего и предсказанного состояния ВМ, серверов и ЦОД в целом с учетом трендов использования ресурсов. АПОМ может взаимодействовать с аналогичными по функциональности модулями управления других ЦОД, обмениваясь управляющей информацией по глобальной сети или выделенным каналам связи. АПОМ также реализует политику допуска клиентов к сервисам. АПОМ реализуется на отказоустойчивом кластере с высокой доступностью.

АПОМ осуществляет управление ресурсами физических серверов. Для этого каждый сервер соединен с АПОМ с помощью виртуального канала, созданного на основе политик контроллера программно-определяемой сети (SDN). АПОМ периодически получает данные о текущем и предсказанном состоянии ресурсов пу-

тем получения сообщений от каждого сервера. Сообщения содержат метку, время отправки, общую емкость ресурсов сервера Res (для процессора, памяти, сетевого подключения и дискового пространства) и предсказанную утилизацию этих ресурсов, вычисленную на стороне сервера. Функции управления ресурсами отдельного сервера выполняет менеджер физического сервера (МФС). Причем Res записывается следующим образом:

$$Res = \{CPU, RAM, NET, STIO\} \quad (1)$$

Для обозначения серверов, которые задействованы в процессах управления, используются метки. Меткой "CONSOLIDATION" отмечаются серверы, которые в текущее время находятся в процессе консолидации. Меткой "PLACEMENT" отмечены серверы, используемые в данное время для размещения ВМ. Меткой "SCALING" отмечены серверы, находящиеся в процессе настройки внутренних ресурсов, предоставляемых локальным ВМ на этом сервере. Меткой "AVAILABLE" помечаются сервера, доступные для размещения и консолидации ВМ.

Подход к размещению новых и мигрирующих ВМ. Для управления виртуальными машинами АПОМ запускает отдельный экземпляр процесса управления для каждого запроса от сервера или для каждого запроса на создание ВМ. Первым выполняются запросы от серверов, на которых наблюдается (или прогнозируется) превышение использования имеющихся ресурсов, когда для ВМ требуются или требуются дополнительные ресурсы, которые сервер принципиально не сможет выделить. Далее обслуживаются запросы клиентов на создание новых ВМ. После чего обслуживаются запросы серверов, которые, согласно прогнозу, будут недозагружены.

Разработанный алгоритм работы АПОМ размещения новых, а также управления функционирующими ВМ приведен на рис. 2. В результате работы алгоритма формируется множество ВМ G^{VM} , которые необходимо разместить на серверах ЦОД.

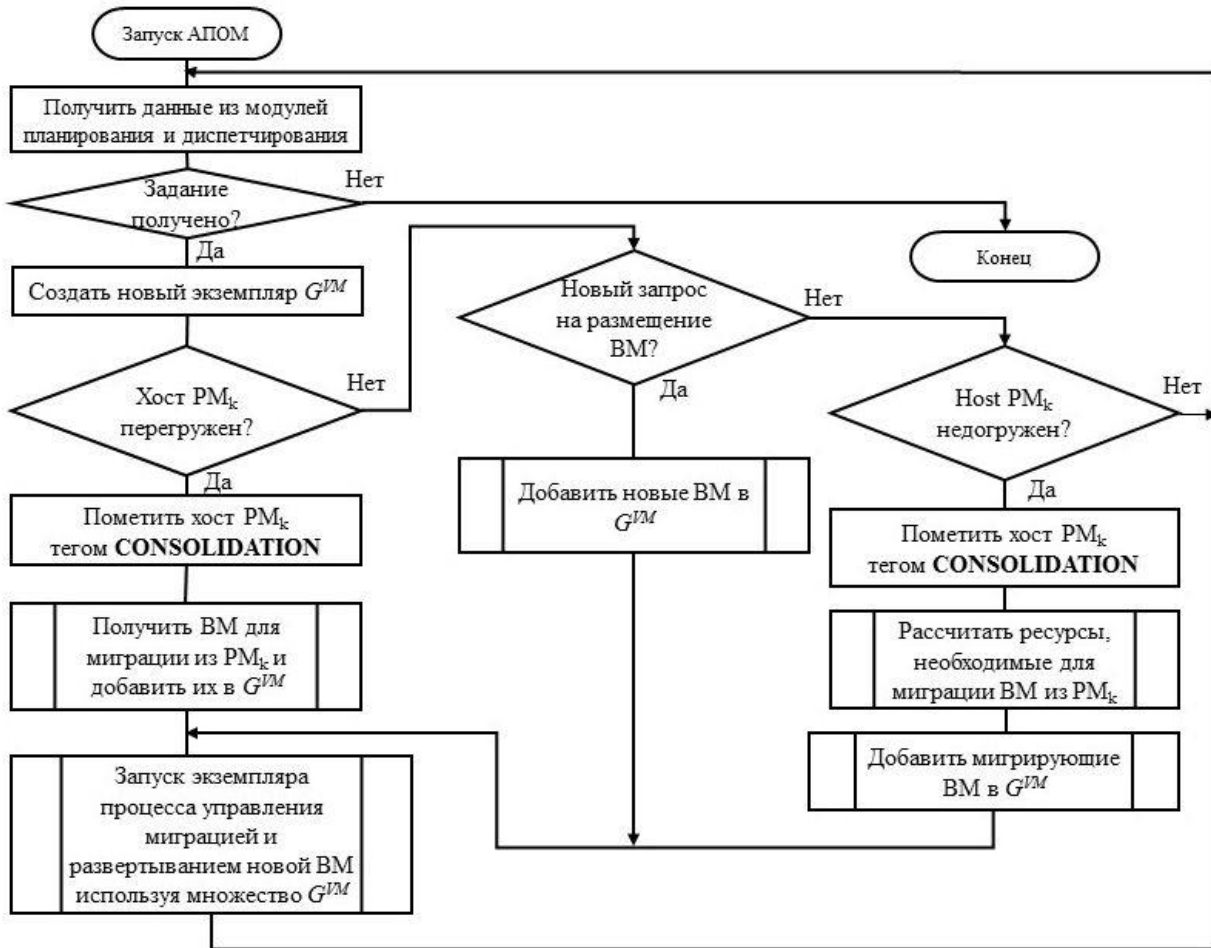


Рис. 2. Алгоритм управления миграцией и размещением VM

При обработке полученного списка G^{VM} АПОМ выбирает подходящие серверы из всех работающих в данный момент в ЦОД путем обработки локальных данных о прогнозируемом состоянии серверов. В результате АПОМ получает множество G^{PM} серверов, претендующих на размещение VM. Затем АПОМ рассылает сообщения, содержащие требования $CPU_K^{VM}, RAM_K^{VM}, NET_K^{VM}, STIO_K^{VM}$ к новой / мигрирующей VM, серверам из множества G^{PM} . Сервер, имеющий, согласно прогнозу, достаточно ресурсов для размещения этой VM, помечает себя меткой "PLACEMENT" и отправляет сообщение в АПОМ, в котором указывает прогнозируемый объем доступных ресурсов. После обработки ответов, полученных от серверов, АПОМ создает целевой список серверов GT^{PM} и реализует политику, базирующуюся на адаптивном программно-определяемом подходе. При этом из списка GT^{PM} выбирается сервер, наиболее подходящий для размещения заданной VM согласно выбранной в данный момент политики, например: выбор сервера с максимальной загрузкой или выбор сервера с такой загрузкой ресур-

сов, при которой при размещении новой VM будет минимальная разбалансировка ресурсов (отсутствие существенных перекосов в сторону использования одного из ресурсов сервера). Кроме того, на сообщения АПОМ могут отвечать серверы с достаточным прогнозируемым количеством ресурсов и помеченные меткой "AVAILABLE".

Алгоритм работы экземпляра процесса управления показан на рис. 3.

Algorithm 2. Непрерывная миграция и размещение новых ВМ

Input: G^{VM} , N , P

Output: Решение о миграции или распределении ВМ на PM

```

1   $K \leftarrow 0$ 
2   $VM \leftarrow G_K^{VM}$ 
3  while  $VM$  is not NULL do
4      Создать новый экземпляр  $G^{PM}$ 
5      Используя политику  $P$ , получить  $N$  подходящих PMs
      претендентов на размещение  $VM$  и добавить их в  $G^{PM}$ 
6       $GT^{PM} \leftarrow \text{NULL}$ 
7       $Res^{VM} \leftarrow G_K^{VM} \{CPU_K^{VM}, RAM_K^{VM}, NET_K^{VM}, STIO_K^{VM}\}$ 
8      for  $i=1$  to  $N$  do
9          Отправить требования  $Res^{VM}$  машины  $VM$  в  $PM_i$ 
10         Получить ответ от  $PM_i$ 
11         if ( $PM_i$  имеет ресурсы для  $VM$ ) and ( $PM_i$  помечен
            меткой AVAILABLE) then
12             Пометить  $PM_i$  меткой PLACEMENT
13             Получить значения  $Res^{PM}$  от  $PM_i$ 
14              $GT^{PM} \leftarrow PM_i$ 
15         end if
16     end for
17     if  $GT^{PM}$  is NULL then
18         if idle PMs exists then
19             Используя политику  $P$ , включить подходящий PM
            и пометить его меткой AVAILABLE
20             Разместить  $VM$  на  $PM$ 
21         else
22             Отправить требования  $Res^{VM}$  машины  $VM$  в
            АПОМ другого ЦОД  $DC$ 
23             Получить ответ от АПОМ другого ЦОД  $DC$ 
24             if response is negative then
25                 return невозможно распределить  $VM$ 
26             else
27                 Распределить все ВМ из  $G^{VM}$  в другом ЦОД
                 $DC$ 
28                 return
29             end if
30         end if
31     else
32         Используя политику  $P$ , получить  $PM$  из  $GT^{PM}$ ,
            подходящую для распределения  $VM$ 
33         Распределить  $VM$  на  $PM$ 
34         Пометить все PM в  $GT^{PM}$ , кроме  $PM$ , тегом
            AVAILABLE
35     end if
36      $K \leftarrow K + 1$ 
37      $VM \leftarrow G_K^{VM}$ 
38 end while

```

Рис. 3. Алгоритм работы экземпляра процесса управления непрерывной миграцией и размещением новых ВМ

Сервер с загрузкой процессора меньше установленного порога не отвечает на сообщения от

АПОМ, так как, согласно прогнозу, работающие на нем ВМ следует переместить на другие серверы для перевода этого сервера в спящий режим или его выключения.

Если ответы на сообщения АПОМ не получены и подходящий для размещения ВМ из множества G^{VM} сервер не найден, то АПОМ принимает решение перевести из спящего режима или включить сервер в стойке, где находится максимальное количество работающих серверов. Затем АПОМ отправляет включенному серверу сообщение о создании новой ВМ или размещения у себя мигрирующей ВМ и помечает сервер меткой "AVAILABLE".

АПОМ принимает решение, исходя из анализа таких данных: значениях индикаторов и метрик, указанных в SLA; загрузки сетевого интерфейса сервера и коммутатора стойки; чувствительности к перегрузкам, создаваемым процессом миграции; взаимного влияния ВМ друг на друга в пределах одного сервера; минимизации размещения ВМ с пульсирующими нагрузками на одном сервере. При этом АПОМ может учитывать информацию о прогнозе нагрузки на новую ВМ или не учитывать ее. Учет такой информации позволяет уменьшить число лишних миграций ВМ в будущем.

АПОМ анализирует выполнение условия

$$Res_{AVAIL}^{PM} \geq Res_K^{VM} \quad (2)$$

где Res_{AVAIL}^{PM} – доступное количество ресурсов сервера, Res_K^{VM} – ресурсы, которые требуются для ВМ из списка G^{VM} .

Если условие (2) для каждого ресурса из множества (1) удовлетворены для некоторого множества серверов, то АПОМ обрабатывает это множество.

Когда экземпляр процесса управления по выбору целевого сервера закончен, АПОМ отправляет сообщение целевому серверу создать новую ВМ и помечает этот сервер меткой "AVAILABLE". Затем отправляет сообщения серверам из списка GT^{PM} для пометки их как "AVAILABLE". На стороне сервера локальным МФС проводятся несколько проверок перед тем, как сервер примет предложение о размещении ВМ.

Производится вычисление

$$Res_{AVAIL}^{PM} = Res_{Th}^{PM} - Res_P^{PM}, \quad (3)$$

где Res_{Th}^{PM} – максимальный порог использования ресурса, Res_P^{PM} – прогнозируемое использование ресурса.

Выражение (3) для каждого ресурса из множества (1) вычисляется локально менеджером сервера. Если выполняется одно из условий $CPU_{AVAIL}^{PM} < 0$, $RAM_{AVAIL}^{PM} < 0$, $NET_{AVAIL}^{PM} < 0$, $STIO_{AVAIL}^{PM} < 0$, то сервер начинает процесс локальной консолидации.

Причем процесс консолидации ВМ среди серверов с загрузкой выше допустимой запускается с наивысшим приоритетом. Сервер, некоторое количество ВМ которого будут мигрировать, помечается меткой "CONSOLIDATION". Таким образом, для размещения ВМ из G^{VM} , множество серверов GT^{PM} будет содержать серверы с меткой "AVAILABLE". При размещении новой ВМ определяется, входит ли она в состав сервиса с масштабированием и балансированием нагрузки или в состав группы сервисов.

АПОМ периодически получает сообщения о прогнозе доступных ресурсов Res от всех серверов в ЦОД путем обмена сообщениями с локальными МФС. Для выбора соответствующей политики управления используется следующая метрика для каждого ресурса из (1)

$$CAP_{Res}^{DC} = \frac{\sum_{i=1}^M Res_i^{VM}}{\sum_{j=1}^N Res_j^{PM}}, \quad (4)$$

где M – количество ВМ, работающих в определенный промежуток времени, N – число серверов, Res_i^{VM} – запрошенное количество ресурсов из (1) i -й ВМ, Res_j^{PM} – текущее использование ресурсов (1) j -м сервером.

Выражение (4) определяет текущую емкость ресурсов ЦОД с целью определения тренда использования каждого ресурса.

Менеджер физического сервера. Каждый сервер характеризуется использованием ресурсов процессора, памяти, сетевого подключения, загрузкой интерфейса устройства хранения и др. На каждом сервере выполняется мониторинг виртуальной машины (МВМ), например Xen [22], ES Xi [23] или Hurer-V [24], а также специальная ВМ с МФС для управления виртуальными машинами средствами МВМ. МФС также может быть модулем в МВМ.

МФС накапливает исторические данные об использовании локальных ресурсов сервера. На основе исторических данных МФС вычисляет прогноз использования ресурсов на среднесрочный период, сопоставимый со временем миграции работающей ВМ. Данные о прогнозе использования

ресурсов вместе с меткой времени сохраняются локально и отправляются в АПОМ. МФС генерирует команды запуска, останова и паузы для ВМ при управлении МВМ или гипервизором. В результате мониторинга МФС выявляет одну или несколько ВМ, для которых не выполняются требования SLA. На основании политик, полученных от АПОМ, МФС принимает решение о выделении большего объема локальных ресурсов этим ВМ.

Экспериментальные исследования

Рассматривается случай применения политики размещения ВМ в ЦОД с гомогенной конфигурацией серверов. Предлагается решение задачи размещения ВМ на основе адаптивного генетического алгоритма (АГА), предложенного в [16]. Экспериментальные исследования проводились для решения задачи распределения ВМ с помощью трех алгоритмов: управляемого генетического алгоритма (УГА) [25], АГА [16] и классического генетического алгоритма (КГА). Исследования проведены для различного соотношения количества ресурсов сервера и количества ВМ.

Оценка качества работы КГА, УГА и АГА выполнена относительно количества высвобожденных серверов, которые могут быть выключены или переведены в спящий режим. Предполагается, что алгоритмы запускаются администратором, когда возникает необходимость оптимизации размещения ВМ.

На рис. 4 показана зависимость числа высвобожденных серверов от размерности задачи (числа работающих ВМ) для случая, когда запрошенные ресурсы процессоров и памяти для всех ВМ распределены случайным образом в диапазоне [0.05; 0.6].

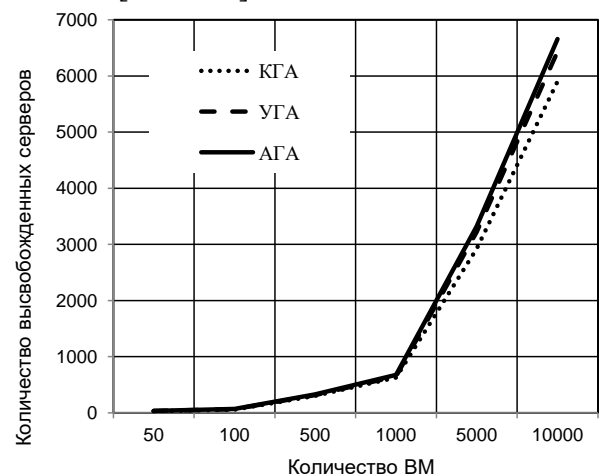


Рис. 4. Зависимость количества высвобожденных серверов от размерности задачи

Анализ экспериментальных исследований в широком диапазоне изменения запрашиваемых ресурсов позволил сделать следующие выводы: 1) использование УГА и АГА является более эффективным, чем использование КГА; 2) АГА всегда позволяет получить лучшие результаты по распределению ВМ на серверах, независимо от условий эксперимента; 3) в случае дисперсии в широком диапазоне требований [0,05; 0,6], использование АГА является предпочтительным.

Выводы

Для решения задачи размещения и миграции ВМ ЦОД предложен адаптивный программно-определяемый подход, основанный на использовании алгоритма распределения новых и миграции работающих ВМ. Адаптивный программно-определяемый менеджер использует комбинации эвристик, при этом учитываются следующие эвристики: индикаторы SLA, загрузка сетевого интерфейса сервера и коммутатора стойки, чувствительность к накладным расходам при миграции, взаимное влияние ВМ друг на друга при работе на одном сервере, а также минимизация размещения ВМ с пульсирующими нагрузками на одном сервере. АПОМ

позволяет выбирать различные алгоритмы или политики управления ресурсами и ВМ для адаптации к воздействию внешних факторов, таких как: изменение нагрузки, установка обновлений программ и оборудования, использование новых программных и аппаратных платформ, изменение в структуре и производительности работы поставщика услуг.

Проведены экспериментальные исследования для оценки конкретной политики распределения ВМ с использованием адаптивного генетического алгоритма. Сравнение результатов работы алгоритмов АГА и УГА показало, что использование АГА эффективнее распределяет ВМ с различными начальными требованиями к ресурсам.

Далее авторы продолжают работу по таким направлениям как: разработка системы приоритетов для политик управления ресурсами ЦОД с целью адаптации к влиянию внешних и внутренних факторов; разработка методики и приоритетов применения эвристик при выборе целевого сервера для миграции или развертывания новых ВМ; посредством имитационного моделирования исследование непрерывного распределения новых и мигрирующих ВМ.

Список литературы

1. Buyya R., Calheiros R. N., Son J., Dastjerdi A. V., Yoon Y. Software-defined cloud computing: Architectural elements and open challenges/ *Advances in Computing, Communications and Informatics (ICACCI) International Conference*. – 2014. – pp. 1–12.
2. Jararweh Y., Al-Ayyoub M., Benkhelifa E., Vouk M., Rindos A. Software defined cloud: Survey, system and evaluation/ *Future Generation Computer Systems*. – 2015. – vol. 58, May 2016. – pp. 56–74.
3. Hariri S., Khargharia B., Chen H., Yang J., Zhang Y., Parashar M., Liu H. The autonomic computing paradigm/ *Cluster Comput.* – 2006. – vol. 9 (1). – pp. 5–17.
4. Hameed A., Khoshkbarforousha A., Ranjan R., Jayaraman P. P., Kolodziej J., Bala-ji P., Zeadally S., Malluhi Q. M., Tziritas N., Vishnu A. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems/ *Computing*. – 2014. – pp. 1–24.
5. Madni S. H. H., Latiff M. S. A., Coulibaly Y. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities/ *Journal of Network and Computer Applications*. – 2016. – vol. 68. – pp. 173–200.
6. Singh S., Chana I. A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges/ *Journal of Grid Computing*. – 2016. – pp. 1–48.
7. Clark C., Fraser K., Hand S., Hansen J. G., Jul E., Limpach C., Pratt I., Warfield A. Live migration of virtual machines/ *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*. – 2005. – vol. 2. – pp. 273–286.
8. Wood T., Shenoy P. J., Venkataramani A., Yousif M. S. Black-box and Gray-box Strategies for Virtual Machine Migration/ *NSDI*. – 2007. – vol. 7. – pp. 17–17.
9. Bobroff N., Kochut A., Beaty K. Dynamic placement of virtual machines for managing SLA violations/ *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium*. – 2007. – pp. 119–128.
10. Zhu X., Young D., Watson B. J., Wang Z., Rolia J., Singhal S., McKee B., Hyser C., Gmach D., Gardner R. 1000 islands: Integrated capacity and workload management for the next generation data center/ *Autonomic Computing (ICAC'08). International Conference*. – 2008. – pp. 172–181.
11. Shen Z., Subbiah S., Gu X., Wilkes J. Cloudscale: elastic resource scaling for multi-tenant cloud systems/ *Proceedings of the 2nd ACM Symposium on Cloud Computing*. – 2011. – p. 5.

12. 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. – pp. 1397–1420.
13. Al-Ayyoub M., Jararweh Y., Daraghme M., Althebyan Q. Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure/ *Cluster Computing*. – 2015. – vol. 18, no. 2. – pp. 919–932.
14. Ролик А.И. Декомпозиционно-компенсационный подход к управлению уровнем услуг в корпоративных ИТ-инфраструктурах / А.И. Ролик // *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр.* – К.: Век+, – 2013. – № 58. – С. 78–88.
15. Ролик А.И. Управление уровнем услуг корпоративной ИТ-инфраструктуры на основе координатора / А.И. Ролик // *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр.* – К.: Век+. – 2013. – № 59. – С. 98–105.
16. Теленик С.Ф. Адаптивный генетический алгоритм для решения класса задач распределения ресурсов ЦОД / С.Ф. Теленик, А.И. Ролик, П.С. Савченко // *Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка.* – К.: «ВЕК+», 2011. – № 54. – С. 164–174.
17. Ролик А.И. Концепция управления корпоративной ИТ-инфраструктурой / А.И. Ролик // *Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка.* – К.: «ВЕК+», 2012. – № 56. – С. 31–55.
18. Mell P., Grance T. The NIST definition of cloud computing (draft),” NIST special publication. – 2011. – vol. 800, no. 145. – p. 7.
19. Apache.org. HDFS Architecture. http://hadoop.apache.org/core/docs/current/hdfs_design.html
20. Calder B., Wang J., Ogus A., Nilakantan N., Skjolsvold A., McKelvie S., Xu Y., Srivastav S., Wu J., Simitci H. Windows Azure Storage: a highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. ACM. – 2011. – pp. 143–157.
21. Ghemawat S., Gobiuff H., & Leung, S.-T.. The Google File System. *SOSP '03: Proceedings of the nineteenth ACM Symposium on Operating Systems Principles* Bolton Landing, NY: ACM. – 2003. – pp. 29–43.
22. XenSource, Inc. [2008]. “TheXen™ virtual machine monitor”, <http://www.cl.cam.ac.uk/research/srg/netos/projects/archive/xen/>
23. VMware, Inc. [2016]. “vSphere Hypervisor,” <http://www.vmware.com/products/vsphere-hypervisor/>
24. Microsoft Corporation. [2016] “Hyper-V overview,” <https://technet.microsoft.com/library/hh831531.aspx>
25. Теленик С.Ф. Генетичні алгоритми вирішення задач управління ресурсами і навантаженням центрів оброблення даних / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, С.А. Андросов // *Автоматика. Автоматизація. Електротехнічні комплекси та системи.* – 2010. – №1 (25). – С. 106–120.