

МАТЕМАТИЧЕСКИЕ МОДЕЛИ РАЗМЕЩЕНИЯ И МИГРАЦИИ ВИРТУАЛЬНЫХ МАШИН НА ГЕТЕРОГЕННЫХ ПЛАТФОРМАХ ВИРТУАЛИЗАЦИИ

В. П. КОЧИН¹⁾, А. В. ШАНЦОВ¹⁾

¹⁾Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь

Аннотация. Определена необходимость применения гетерогенных платформ виртуализации, выделены основные преимущества их использования. Рассмотрена проблематика обеспечения эффективного размещения и миграции виртуальных машин на гетерогенной платформе. Обоснована потребность в формализации процессов размещения и миграции виртуальных машин. Предложена математическая модель размещения виртуальных машин, основанная на наборах требований виртуальных машин и наборе ресурсов, предоставляемых платформой виртуализации. Определено ограничение в размещении виртуальных машин на хостах платформы виртуализации. Предложена математическая модель миграции виртуальных машин, реализующая оптимизацию затрат на функционирование виртуальных машин. Разработаны алгоритм и математическая модель динамической балансировки виртуальных машин на гетерогенной платформе виртуализации. Представленная математическая модель учитывает требования к загруженности хостов платформы виртуализации, предоставляет оптимальный выбор наборов виртуальных машин и целевых хостов для осуществления балансировки.

Ключевые слова: информационные технологии; облачные вычисления; гетерогенная платформа виртуализации.

MATHEMATICAL MODELS OF VIRTUAL MACHINE PLACEMENT AND MIGRATION ON HETEROGENEOUS VIRTUALISATION PLATFORMS

V. P. KOCHYN^a, A. V. SHANTSOV^a

^aBelarusian State University, 4 Niezaliezhnasci Avenue, Minsk 220030, Belarus

Corresponding author: A. V. Shantsov (downseason@mail.ru)

Abstract. The necessity of using heterogeneous virtualisation platforms has been determined, and the main advantages of their use are highlighted. The problems of ensuring the efficient placement and migration of virtual machines on a heterogeneous platform are considered. The need to formalise the processes of virtual machine placement and migration is substantiated. The mathematical model for the placement of virtual machines is proposed, based on the sets of requirements of virtual machines and the set of resources provided by the virtualisation platform. A limitation has been defined in the placement of virtual machines on virtualisation platform hosts. The mathematical model of virtual machine migration is

Образец цитирования:

Кочин ВП, Шанцов АВ. Математические модели размещения и миграции виртуальных машин на гетерогенных платформах виртуализации. *Журнал Белорусского государственного университета. Математика. Информатика.* 2026;2:122–130. EDN: YHKHOJ

For citation:

Kochyn VP, Shantsov AV. Mathematical models of virtual machine placement and migration on heterogeneous virtualisation platforms. *Journal of the Belarusian State University. Mathematics and Informatics.* 2026;2:122–130. Russian. EDN: YHKHOJ

Авторы:

Виктор Павлович Кочин – кандидат технических наук, доцент; проректор по учебной работе и интернационализации образования.

Артём Владимирович Шанцов – аспирант кафедры технологий программирования факультета прикладной математики и информатики. Научный руководитель – В. П. Кочин.

Authors:

Victor P. Kochyn, PhD (engineering), docent; vice-rector for academic affairs and internationalisation of education.

kochyn@bsu.by

Artem V. Shantsov, postgraduate student at the department of programming technologies, faculty of applied mathematics and computer science.

downseason@mail.ru

proposed that implements cost optimisation for the operation of virtual machines. The algorithm and the mathematical model of dynamic balancing of virtual machines on a heterogeneous virtualisation platform have been developed. The presented mathematical model takes into account the requirements for the workload of virtualisation platform hosts, represents the optimal choice of virtual machine sets and target hosts for balancing.

Keywords: information technology; cloud computing; heterogeneous virtualisation platform.

Введение

Технологии виртуализации являются неотъемлемой частью современной вычислительной инфраструктуры, обеспечивая эффективное использование ресурсов, снижение затрат и повышение отказоустойчивости инфраструктуры. Однако по мере увеличения масштаба и сложности виртуализированных сред растет потребность в формализации процессов управления, особенно в части распределения нагрузки на облачной платформе. Без четко определенных процедур существенно возрастает риск нерационального использования ресурсов, снижения производительности и возникновения уязвимостей инфраструктуры [1–3].

Повысить гибкость и отказоустойчивость современных ИТ-инфраструктур, а также эффективность управления ими позволяет применение гетерогенных платформ виртуализации. Однако для реализации всех преимуществ гетерогенной платформы виртуализации необходима формализация процессов управления платформой, в частности процессов размещения, миграции и динамической балансировки виртуальных машин (ВМ).

Преимущества применения гетерогенных платформ виртуализации

Применение гетерогенных платформ виртуализации направлено на повышение гибкости и устойчивости ИТ-инфраструктуры и оказывает существенное влияние на защищенность обрабатываемых ресурсов. Гетерогенная платформа виртуализации включает в себя две и более подсистемы виртуализации, построенные на различных гипервизорах, объединенных в один вычислительный кластер [4]. Применение гетерогенных платформ виртуализации имеет ряд существенных преимуществ. Основными из них являются следующие преимущества.

1. Повышение гибкости и совместимости. Гетерогенные платформы виртуализации предполагают использование различных гипервизоров, архитектур и аппаратных решений в рамках одной ИТ-инфраструктуры, что, в свою очередь, позволяет:

- интегрировать унаследованные и современные информационные системы;
- адаптироваться под конкретные задачи и сервисы, используя наиболее подходящие технологии

в каждом случае.

2. Минимизация рисков зависимости от одного вендора (*vendor lock-in*). Применение нескольких платформ виртуализации (например, *VMware ESXi*, *Microsoft Hyper-V*, *Kernel-based Virtual Machine (KVM)*, *Xen*, *OpenStack* и др.) снижает зависимость от одного поставщика и обеспечивает:

- свободу выбора и конкуренции между вендорами, что влияет на конечную стоимость обслуживания платформы;

- гибкость при смене платформы, например, в случае изменения лицензирования или политик вендора;

- возможность миграции между платформами без полной реконфигурации инфраструктуры.

3. Оптимизация затрат и использование существующих ресурсов. Применение различных платформ виртуализации может быть оптимизировано по соотношению цены и производительности в зависимости от конкретных задач. Например, для тестовых сред и инфраструктуры платформы целесообразно использовать бесплатные решения с открытым исходным кодом (*KVM*, *Proxmox*), а для критичных бизнес-приложений – коммерческие решения, предоставляющие расширенные возможности управления и SLA (*VMware ESXi*, *Microsoft Hyper-V*).

4. Повышение устойчивости и отказоустойчивости. Гетерогенная структура платформы снижает риски при сбоях на уровне гипервизора или системного программного обеспечения. В случае если одна платформа виртуализации уязвима к определенной атаке или подвержена сбоям, другая платформа остается в работоспособном состоянии. Также критические подсистемы можно изолировать по разным гипервизорам, что повышает защищенность всей архитектуры.

5. Поддержка гибридных и мультиоблачных стратегий. Гетерогенные платформы виртуализации особенно актуальны при интеграции частных, публичных и периферийных облаков, что позволяет развертывать ВМ в разных средах с различными гипервизорами и сервисами (на публичных, например

Microsoft Azure, Amazon Web Services (AWS), Google Cloud, и частных облачных ресурсах) и осуществлять централизованное управление и оркестрацию с помощью кросс-платформенных инструментов (например, Kubernetes, Terraform, Ansible, OpenStack).

Математические модели размещения, миграции и балансировки ВМ

Развитие облачных вычислений и мультиоблачных стратегий предприятий привело к неизбежной гетерогенности программных платформ виртуализации. Современные центры обработки данных все чаще включают в себя узлы, использующие различные гипервизоры (*VMware ESXi, KVM, Microsoft Hyper-V*) и контейнерные среды (*Docker, Firecracker, Kata*), управляемые через разнородные API и облачные оркестраторы. Такая среда обеспечивает гибкость, но одновременно порождает множество технических, экономических и организационных вызовов.

Математическое моделирование размещения и миграции ВМ в указанных условиях становится неотъемлемым элементом интеллектуального управления ресурсами центров обработки данных и обеспечивает:

- оптимальное использование вычислительных мощностей, снижение избыточности и простоя узлов;
- сокращение лицензионных расходов за счет балансировки между коммерческими платформами и бесплатными платформами с открытым исходным кодом (*open-source*);
- поддержание высокого уровня отказоустойчивости и SLA даже при динамическом перераспределении нагрузки;
- уменьшение времени и стоимости миграций, особенно при кросс-платформенных переходах;
- формализацию и автоматизацию процессов, что критично для масштабируемости ИТ-инфраструктуры и минимизации человеческого фактора.

Без систематического подхода к планированию размещения ВМ в условиях гетерогенной виртуализации организация оказывается перед трудноразрешимыми компромиссами между затратами, производительностью и надежностью аппаратных ресурсов.

Таким образом, исследования в данной области имеют не только теоретическую, но и высокую прикладную ценность – от создания энергоэффективных облаков до построения стратегий миграции на платформы с открытым исходным кодом с минимальными затратами и рисками.

Наличие двух или более типов гипервизоров гетерогенной платформы виртуализации налагает требования к формализации процессов размещения и миграции ВМ. Формализация процессов размещения ВМ предполагает разработку строгих алгоритмов и критериев, определяющих выбор физического хоста для каждой ВМ [5]. Этот подход обеспечивает предсказуемость, масштабируемость и автоматизируемость управления гетерогенной платформой. Алгоритмы формализованного размещения учитывают набор требований к размещению ВМ (количество центральных процессоров, объем оперативной памяти, объем дискового пространства, пропускную способность сети, энергопотребление и др.), позволяя тем самым достичь максимально эффективного размещения ВМ на хостах платформы.

Пусть $P = \{p_1, p_2\}$ – набор платформ виртуализации, $H = \{h_1^k, h_2^k, \dots, h_m^k\}$ – множество физических хостов, которые принадлежат платформе $k \in \{1, 2\}$, $V = \{v_1, v_2, \dots, v_n\}$ – множество ВМ, а $R = \{\text{CPU, RAM, Disk, ...}\}$ – ресурсы, потребляемые ВМ.

Каждая ВМ $v_i \in V$ характеризуется вектором требований к ресурсам

$$\vec{r}_i = (r_i^{\text{CPU}}, r_i^{\text{RAM}}, r_i^{\text{Disk}}, \dots).$$

Каждый хост $h_j^k \in H$ характеризуется вектором доступных ресурсов

$$\vec{f}_j = (f_j^{\text{CPU}}, f_j^{\text{RAM}}, f_j^{\text{Disk}}, \dots),$$

а также вектором свободных ресурсов

$$\vec{c}_j = (c_j^{\text{CPU}}, c_j^{\text{RAM}}, c_j^{\text{Disk}}, \dots).$$

Вектор свободных ресурсов $\vec{c}_j$ хоста h_j^k в момент времени t определяется как

$$\vec{c}_j(t) = \vec{f}_j - \sum_{i=1}^n \vec{r}_i,$$

где n – количество ВМ, размещенных на хосте h_j^k в момент времени t .

Модель размещения ВМ на платформе определяется рядом условий. Необходимым условием размещения ВМ является наличие свободных ресурсов на хосте. Условие размещения ВМ v_i на хосте h_j^k в момент времени t определяется бинарной переменной $x_{ij}(t)$, значение которой присваивается исходя из сравнения вектора потребления ресурсов ВМ $\vec{r}_i$ с вектором свободных ресурсов $\vec{c}_j$ хоста h_j^k :

$$x_{ij}(t) = \begin{cases} 1, & \text{если } \vec{r}_i \leq \vec{c}_j(t), \\ 0, & \text{если } \vec{r}_i > \vec{c}_j(t). \end{cases}$$

Дополнительными условиями размещения ВМ являются обеспечение заданного уровня обслуживания и обеспечение требуемого уровня защищенности ВМ. Для обеспечения требуемого уровня защищенности ВМ может назначаться определенная платформа p_k . В таком случае значение переменной $x_{ij}(t)$ принимает вид

$$x_{ij}(t) = \begin{cases} 1, & \text{если } \vec{r}_i \leq \vec{c}_j(t) \wedge v_i \in p_k, \\ 0, & \text{если } \vec{r}_i > \vec{c}_j(t) \vee v_i \notin p_k. \end{cases}$$

Обеспечение заданного уровня обслуживания часто требует одновременного размещения набора ВМ на одном физическом хосте в целях минимизации потерь в производительности и задержек в обработке данных. Необходимость совместного размещения подмножества ВМ $V_m = \{v_1, v_2, \dots, v_m\}$ из множества всех ВМ V на одном физическом хосте h_j^k платформы виртуализации p_k в момент времени t определяется бинарной переменной

$$x_m(t) = \begin{cases} 1, & \text{если } \sum_{i=1}^m \vec{r}_i \leq \vec{c}_j(t), \\ 0, & \text{если } \sum_{i=1}^m \vec{r}_i > \vec{c}_j(t). \end{cases}$$

Предложенная модель обеспечивает системный подход к решению задачи размещения ВМ за счет введения формализованных критериев, основанных на векторном представлении ресурсных требований ВМ и ресурсных характеристик физических хостов. Модель позволяет учитывать не только аппаратные ограничения, но и дополнительные требования – уровень обслуживания, аспекты информационной безопасности. Введение бинарных переменных, отражающих размещения ВМ, формализует логику принятия решений и создает основу для разработки алгоритмов автоматизированного управления ресурсами.

Формализованные подходы дают возможность реализовать автоматизированные системы миграции ВМ между гипервизорами. Миграция используется для балансировки нагрузки, повышения энергоэффективности, технического обслуживания хостов и обеспечения непрерывности предоставляемых сервисов при возникновении сбоев. Использование формальных правил позволяет предусматривать сценарии сбоев, обеспечивая автоматическое перераспределение нагрузки, резервирование и соблюдение заданного уровня обслуживания для ВМ [6].

Пусть t – дискретное время, $x_{ij}(t)$ – статус размещения ВМ v_i на хосте h_j^k в момент времени t , $M(t)$ – множество миграций в момент времени t .

Миграция ВМ определяется бинарной переменной

$$x_{ij}(t) = \begin{cases} 1, & \text{если } v_i \text{ перемещена на } h_j^k \text{ в момент времени } t+1, \\ 0 & \text{в противном случае.} \end{cases}$$

Ограничение миграции для каждой ВМ имеет вид

$$|x_{ij}(t+1) - x_{ij}(t)| \leq 1.$$

Тогда переменная миграции определяется как

$$m_{ij}(t) = |x_{ij}(t+1) - x_{ij}(t)|.$$

Общее количество миграций задается формулой

$$M(t) = \sum_{i=1}^n \sum_{j=1}^m m_{ij}(t).$$

Рассмотрим применение модели миграции ВМ на примере оптимизации энергопотребления гетерогенной платформы. Оптимизация процесса включает учет энергопотребления всех хостов платформы и затрат на миграцию ВМ. Тогда минимизация общих затрат принимает вид

$$\min \sum_{j=1}^m P_j u_j(t) + \lambda M(t),$$

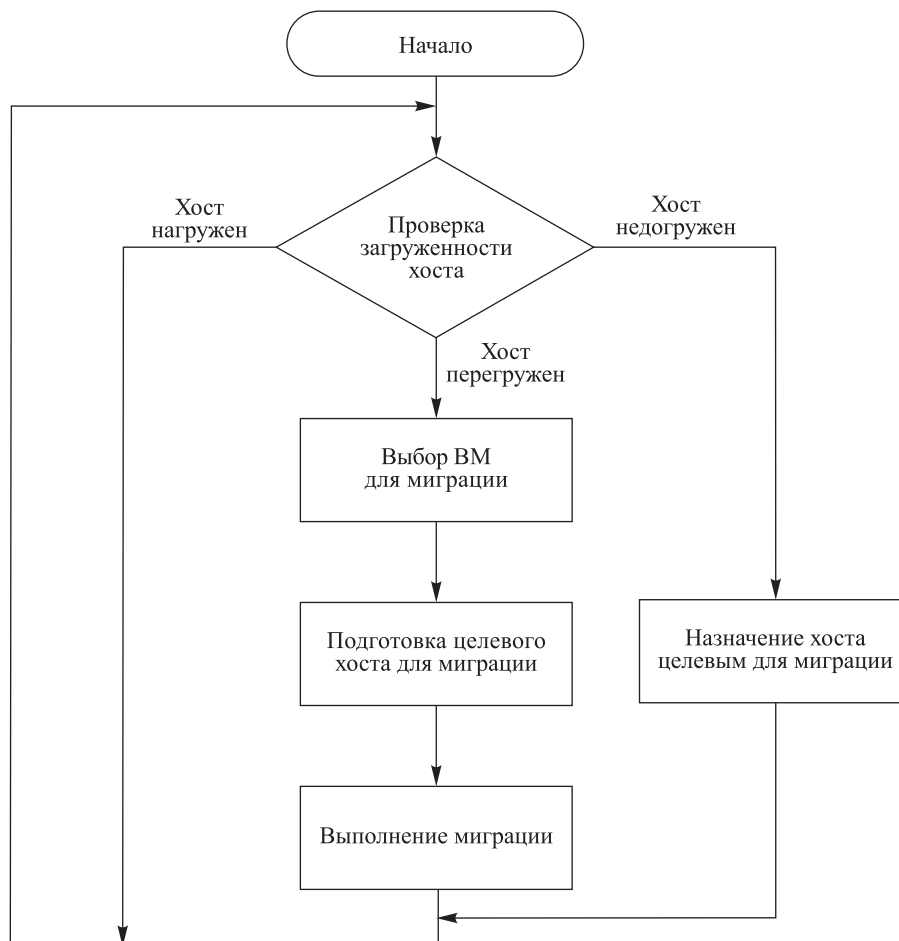
где P_j – энергопотребление хоста h_j^k ,

$$u_j(t) = \begin{cases} 1, & \text{если хотя бы одна ВМ перемещена на } h_j, \\ 0 & \text{в противном случае,} \end{cases}$$

а λ – весовой коэффициент миграции.

Далее на основе алгоритмов оптимизации [7] можно рассчитать оптимальное размещение ВМ при минимальном количестве миграций.

Формализация процесса балансировки нагрузки гетерогенной платформы виртуализации позволяет предусматривать сценарии сбоев, нарушений и кибератак [8], обеспечивая автоматическое перераспределение нагрузки и резервирование, в ходе которых осуществляется перенос ВМ с одного физического узла на другой без остановки предоставляемых сервисов. Алгоритм динамической балансировки представляет собой замкнутый цикл, состоящий из четырех этапов (см. рисунок).



Алгоритм динамической балансировки
Dynamic balancing algorithm

Этап 1: проверка загруженности хоста. Осуществляется проверка загруженности каждого хоста гетерогенной платформы виртуализации. Для этого вычисляется загрузка хоста по заданным критериям, таким как использование центрального процессора, оперативной памяти, дискового пространства и др.:

$$\text{Load}_j = \max \left(\frac{\text{CPU}_j^{\text{used}}}{\text{CPU}_j^{\text{total}}}, \frac{\text{RAM}_j^{\text{used}}}{\text{RAM}_j^{\text{total}}}, \frac{\text{Disk}_j^{\text{used}}}{\text{Disk}_j^{\text{total}}}, \dots \right).$$

Помимо вычисления метрик нагрузки Load_j , для каждого хоста определяются пороговые значения метрик θ_{low} и θ_{high} :

- если $\text{Load}_j > \theta_{\text{high}}$, то хост считается перегруженным и возникает необходимость в миграции ВМ;
- если $\theta_{\text{low}} < \text{Load}_j < \theta_{\text{high}}$, то хост находится в нагруженном состоянии и балансировка не требуется;
- если $\text{Load}_j < \theta_{\text{low}}$, то хост считается недогруженным и на него в случае необходимости переносятся ВМ.

Эман 2: выбор ВМ для миграции. Выполняется поиск ВМ с оптимальными критериями для миграции. Для этого на каждом перегруженном хосте осуществляется сортировка ВМ по ресурсоемкости:

$$v_i = w_1 \text{CPU}_i + w_2 \text{RAM}_i + w_3 \text{Disk} + \dots,$$

где w – весовые коэффициенты для центрального процессора, оперативной памяти, дискового пространства.

После выполнения сортировки осуществляется выбор минимальных подмножеств $V^* \subset V$ таких, что после их удаления $\theta_{\text{low}} < \text{Load}_j < \theta_{\text{high}}$.

В условиях гетерогенной платформы виртуализации необходимо учитывать возможность временной недоступности хоста, вызванной сетевой деградацией, отказом гипервизора, техническим обслуживанием, кибератакой или нарушением целостности среды выполнения.

Для формализации данного состояния введем бинарную переменную доступности хоста

$$a_j(t) = \begin{cases} 1, & \text{если хост } h_j \text{ доступен в момент времени } t, \\ 0, & \text{если хост } h_j \text{ недоступен.} \end{cases}$$

Тогда множество доступных хостов в момент времени t определяется как

$$H^{\text{av}}(t) = \{h_j \in H \mid a_j(t) = 1\}.$$

Эман 3: подготовка целевого хоста для миграции. Осуществляется проверка доступности необходимых ресурсов для переноса ВМ.

Выбор целевого хоста для миграции выполняется только среди множества $H^{\text{av}}(t)$. Условие допустимости размещения ВМ v_i на хосте h_j дополняется ограничением доступности

$$x_{ij}(t) \leq a_j(t).$$

Если для выбранной ВМ отсутствуют допустимые хосты во множестве $H^{\text{av}}(t)$, активируется процедура отложенной миграции:

- ВМ помещается в очередь ожидания Q_{mig} ;
- выполняется повторная попытка выбора через интервал Δt ;
- при превышении порогового времени ожидания инициируется аварийная миграция на резервную платформу.

Для каждой ВМ $v_i \in V^*$ выполняется оценка набора подходящих хостов и выбирается хост согласно заданным критериям платформы.

$$T_i = \{h_j \mid \bar{r}_i \leq \bar{c}_j - \bar{r}_j^{\text{used}}\}.$$

Эман 4: выполнение миграции.

Перед началом переноса проверяются:

- доступность исходного хоста $a_{\text{src}}(t)$;
- доступность целевого хоста $a_{\text{dst}}(t)$;
- сетевая связность между ними.

Если в процессе миграции обнаруживается потеря доступности целевого хоста, выполняется одна из следующих операций:

- откат (*rollback*) к исходному состоянию при использовании механизма предварительного копирования (*pre-copy*);
- переключение на альтернативный хост из множества допустимых хостов.

В случае полной недоступности исходного хоста инициируется процедура аварийного восстановления ВМ из последней консистентной точки (*checkpoint*), что может быть формализовано как

$$v_i(t+1) = \text{restore}(v_i, \text{backup}_k),$$

где backup_k – последняя корректная резервная копия ВМ.

Таким образом, алгоритм динамической балансировки расширяется до отказоустойчивой модели, учитывающей частичные и временные сбои элементов гетерогенной платформы.

Далее для каждой найденной пары (v_i, h_j) осуществляется проверка сетевой доступности и выполняется миграция, которая включает:

- фоновую синхронизацию памяти (копирование содержимого оперативной памяти на новый хост с минимальным воздействием на реализуемые процессы);
- кратковременную приостановку ВМ, перенос остаточной памяти и запуск на новом сервере;
- обновление сетевых маршрутов (сохранение сетевой идентичности ВМ при смене физического местоположения).

После выполнения миграции выполняется возврат к этапу 1.

Экспериментальные исследования

Целью экспериментальных исследований являлась оценка эффективности предложенных математических моделей размещения и динамической балансировки ВМ на гетерогенной платформе виртуализации по следующим критериям:

- уровню энергопотребления хостов;
- средней загрузке ресурсов;
- количеству миграций;
- устойчивости к временной недоступности узлов.

Конфигурация тестовой платформы. Моделирование проводилось для гетерогенного кластера, включающего 6 физических хостов, 2 типа гипервизоров (тип А и тип В) и 48 ВМ с различными профилями нагрузки. Параметры хостов приведены в табл. 1.

Таблица 1

Параметры хостов

Table 1

Host parameters

Параметры	Значение
CPU	32 vCPU
RAM	128 Гб
Disk	4 Тб
Среднее энергопотребление	420 Вт

ВМ были разделены на три класса.

1. *Compute-intensive* (высокая нагрузка на центральный процессор).
2. *Memory-intensive* (высокое потребление оперативной памяти).
3. *Mixed workload* (смешанная нагрузка).

Нагрузка моделировалась как дискретный временной процесс с шагом $\Delta t = 5$ мин в течение 24 ч.

Сравниваемые алгоритмы. Для оценки эффективности предложенной модели сравнивались три подхода.

1. Статическое размещение (*baseline*) – первоначальное размещение без последующей балансировки.
2. Классическая балансировка без учета доступности хостов – перераспределение по пороговым значениям нагрузки.
3. Предложенная модель – формализованная балансировка с учетом доступности хостов и стоимости миграции, а также с учетом механизма отложенной и аварийной миграции.

Результаты эксперимента. В ходе экспериментальных исследований были получены следующие результаты.

1. Энергопотребление. Данные о среднесуточном энергопотреблении кластера приведены в табл. 2.

Таблица 2

Среднесуточное энергопотребление кластера

Table 2

Average daily energy consumption of the cluster

Подход	Среднее энергопотребление в сутки, кВт · ч	Изменение, %
Статическое размещение	60,5	–
Классическая балансировка	54,2	–10,4
Предложенная модель	49,8	–17,7

Снижение энергопотребления достигается:

- за счет консолидации нагрузки;
- отключения недогруженных хостов;
- минимизации избыточных миграций.

2. Количество миграций. Как следует из табл. 3, снижение количества миграций составило 36,8 %, что подтверждает эффективность введенного штрафного коэффициента λ в целевой функции.

Таблица 3

Среднесуточное количество миграций

Table 3

Average daily number of migrations

Подход	Среднее число миграций в сутки
Классическая балансировка	38
Предложенная модель	24

3. Балансировка нагрузки. Данные о среднеквадратичном отклонении загрузки центрального процессора по хостам приведены в табл. 4.

Таблица 4

Среднеквадратичное отклонение загрузки центрального процессора по хостам

Table 4

Root mean square deviation of CPU load across hosts

Подход	$\sigma(\text{CPU})$, %
Статическое размещение	18,4
Классическая балансировка	9,7
Предложенная модель	7,2

Из табл. 4 видно, что предложенная модель обеспечивает более равномерное распределение нагрузки.

4. Обработка временной недоступности хоста. В эксперименте моделировалось временное отключение одного хоста на интервале 2 ч. Получены следующие результаты:

- в классической модели наблюдалась потеря 3 ВМ до восстановления узла;
- в предложенной модели 2 ВМ были автоматически перенесены, а 1 ВМ восстановлена из последней консистентной точки, при этом время деградации сервиса сократилось на 41 %.

Таким образом, расширение алгоритма динамической балансировки механизмами контроля доступности существенно повышает устойчивость платформы.

Проведенные эксперименты показали, что предложенная формализованная модель обеспечивает:

- снижение энергопотребления на 17–18 %;
- уменьшение количества миграций более чем на треть;
- более равномерную загрузку ресурсов;
- повышение устойчивости к временной недоступности узлов.

Наибольший эффект достигается в сценариях переменной нагрузки и при наличии разнородных профилей ВМ.

Заключение

В условиях стремительной цифровизации, роста количества рабочих нагрузок и усложнения ИТ-архитектур применение гетерогенных платформ позволяет гибко адаптироваться к современным требованиям, снижать операционные риски, оптимизировать затраты и повышать уровень защищенности активов. Формализация процессов размещения ВМ, внедрение эффективных механизмов миграции и балансировки нагрузки, комплексное обеспечение безопасности являются критически важными аспектами построения надежных и управляемых виртуализированных инфраструктур. В условиях постоянного роста числа ВМ и усложнения атакующих методов только системный и формализованный подход может обеспечить устойчивость и эффективность информационных систем, размещенных на гетерогенных платформах.

Библиографические ссылки

1. Кочин ВП, Шанцов АВ. Комплексная система защиты информации облачных ресурсов. В: Научно-исследовательский институт технической защиты информации. *Комплексная защита информации. Материалы XXVI научно-практической конференции*; 25–27 мая 2021 г.; Минск, Беларусь. Минск: Владимир Сивчиков; 2021. с. 332–334.
2. Кочин ВП. Модель сложных интегрированных систем. *Журнал Белорусского государственного университета. Математика. Информатика*. 2024;1:71–78. EDN: NANGXU.
3. Kochyn VP, Zherelo AV. Designing a secure fail-safe cloud repository of paperworks of students and employees of educational institutions. *Journal of the Belarusian State University. Mathematics and Informatics*. 2021;3:104–108. DOI: 10.33581/2520-6508-2021-3-104-108.
4. Кочин ВП, Шанцов АВ. Гибридный режим функционирования облачных платформ. В: Романов ОА, редактор. *Информационно-коммуникационные технологии: достижения, проблемы, инновации (ИКТ-2022). Сборник статей II Международной научно-практической конференции*; 30–31 марта 2022 г.; Полоцк, Беларусь. Новополоцк: Полоцкий государственный университет имени Евфросинии Полоцкой; 2022. с. 189–193.
5. Соловьев ВП, Удовиченко АО. Метод планирования размещения группы виртуальных машин с перераспределением ресурсов. *Программные продукты и системы*. 2012;1:134–138.
6. Алексанков СМ. Модель динамической миграции виртуальных машин с гибридным подходом. *Научно-технический вестник информационных технологий, механики и оптики*. 2017;17(4):725–732. DOI: 10.17586/2226-1494-2017-17-4-725-732.
7. Васильева ЮО, Костенко ВА, Чупахин АА. Влияние политик размещения виртуальных машин на эффективность использования ресурсов центров обработки данных. *Известия Российской академии наук. Теория и системы управления*. 2020;3:95–100. DOI: 10.31857/S0002338820030166.
8. Ворожцов АС, Тутова НВ, Тутов АВ. Динамическое распределение вычислительных ресурсов центров обработки данных. *Т-Сотт. Телекоммуникации и транспорт*. 2016;10(7):47–51. EDN: WINJQL.

Получена 26.05.2025 / исправлена 08.03.2026 / принята 09.06.2026.
Received 26.05.2025 / revised 08.03.2026 / accepted 09.06.2026.