

УДК 004.051

doi: 10.21685/2072-3059-2025-1-2

Подход к разработке архитектуры программного комплекса поддержки процесса перераспределения виртуальных машин виртуализированного центра обработки данных

Н. Ю. Бумажкина

Академия Федеральной службы охраны Российской Федерации, Орёл, Россия
orui@mvd.ru

Аннотация. *Актуальность и цели.* Распределенность разработанной системы поддержки обусловлена необходимостью получения параметров машин виртуальных и физических виртуализированного центра обработки данных (ЦОД) с учетом локальной или территориальной распределенности их серверных платформ. *Материалы и методы.* При разработке архитектуры программного комплекса использовалась парадигма «менеджер-агенты». При этом для поддержки кроссплатформенности сервисного программного обеспечения виртуализированного ЦОД рассмотрена возможность совместимости с интерфейсами прикладного программирования существующих распределенных систем мониторинга, таких как Zabbix, NetCrunch, OpenNMS и др. *Результаты.* Проведена экспериментальная оценка эффективности программного комплекса перераспределения виртуальных машин виртуализированного ЦОД. *Выводы.* Сформированная система поддержки отличается возможностью расширения и высокой адаптивностью к условиям современных виртуализированных ЦОД.

Ключевые слова: виртуализированный центр обработки данных, архитектура программного комплекса, виртуальная машина, имитационная модель

Для цитирования: Бумажкина Н. Ю. Подход к разработке архитектуры программного комплекса поддержки процесса перераспределения виртуальных машин виртуализированного центра обработки данных // Известия высших учебных заведений. Поволжский регион. Технические науки. 2025. № 1. С. 16–28. doi: 10.21685/2072-3059-2025-1-2

An approach to developing the architecture of a software package to support the process of redevelopment of virtual machines in a virtualized data center

N.Yu. Bumazhkina

Academy of the Federal Security Service of the Russian Federation, Orel, Russia
orui@mvd.ru

Abstract. *Background.* The distribution of the developed support system is due to the need to obtain the parameters of virtual and physical machines of a virtualized data center (DPC), taking into account the local or territorial distribution of their server platforms. *Materials and methods.* When developing the architecture of the software package, the “manager-agents” paradigm was used. At the same time, to support cross-platform service software of a virtualized data center, the possibility of compatibility with application programming interfaces of existing distributed monitoring systems such as Zabbix, NetCrunch, OpenNMS, etc. is considered. *Results.* An experimental evaluation of the effectiveness of the software package for relocating virtual machines in a virtualized data center has been

carried out. Conclusions. The formed support system is expandable and highly adaptable to the conditions of modern virtualized data centers.

Keywords: virtualized data center, software package architecture, virtual machine, simulation model

For citation: Bumazhkina N.Yu. An approach to developing the architecture of a software package to support the process of redevelopment of virtual machines in a virtualized data center. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences.* 2025;(1):16–28. (In Russ.). doi: 10.21685/2072-3059-2025-1-2

Введение

В настоящее время наблюдается увеличение объема обрабатываемых данных, что приводит к повышенному спросу на использование методов и технологий развития интеллектуального анализа данных [1, 2]. Одним из современных направлений в развитии этих методов является применение алгоритмов искусственного интеллекта, обусловленное необходимостью работы с большими данными и решения сложных вычислительных задач.

Технологической основой применения методов интеллектуального анализа данных выступает использование мощностей центра обработки данных (ЦОД). По мере роста объема данных, нуждающихся в обработке, увеличивается потребность в более мощных вычислительных и коммуникационных ресурсах [3, 4].

Важной задачей, решаемой системой администрирования ЦОД, является оптимизация использования их вычислительных и коммуникационных ресурсов. Одним из способов такой оптимизации является применение систем виртуализации этих ресурсов [5]. Центры обработки данных, использующие системы виртуализации, называются «виртуализированные центры обработки данных» [6].

Подход к предоставлению пользовательских услуг через виртуализацию в общем виде показан на рис. 1, он включает четыре взаимосвязанных уровня иерархии ЦОД [7].

Первый уровень включает вычислительные и коммуникационные ресурсы ЦОД, представленные множеством физических машин, чьи ресурсы распределяются виртуальным машинам в соответствии с их запросами.

Второй уровень – это виртуализация данных ресурсов, где многочисленные виртуальные машины используют выделенные гипервизорами ресурсы для установки необходимого пользователю программного обеспечения. Виртуализированные ЦОД, разработанные на основе одного вида гипервизоров, именуются гомогенными, а использующие разные виды гипервизоров – гетерогенными.

Третий уровень представляет программное обеспечение, с которым непосредственно взаимодействует пользователь.

Четвертый уровень – это уровень самих пользователей, которые получают в распоряжение услуги и сервисы с нужным качеством в рамках соглашения об уровне обслуживания. Программное обеспечение может изменять запросы на виртуализированные ресурсы, влияя на потребность в физических ресурсах. Виртуальные машины поддерживают гибкость запросов на вычислительные и коммуникационные ресурсы, но изначальная настройка на

уровне физических машин может не соответствовать изменяющимся потребностям. Увеличение ресурсов сверх выделенного объема невозможно, что снижает эффективность виртуализированного ЦОД.

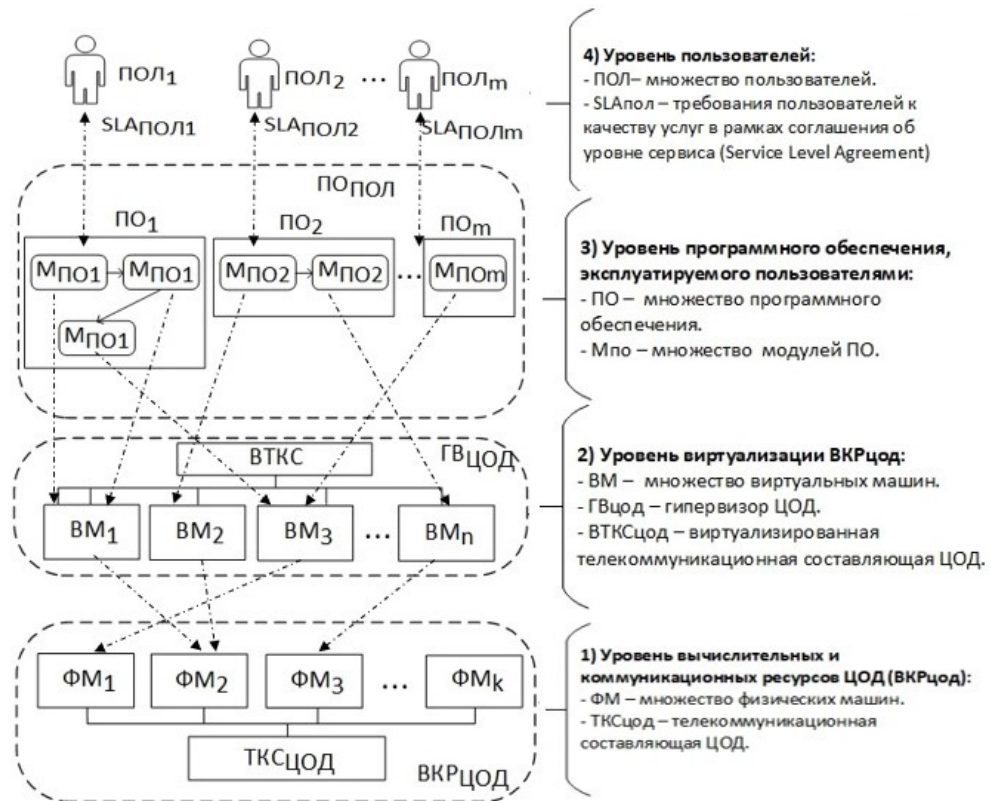


Рис. 1. Принцип динамического распределения вычислительных и коммуникационных ресурсов на основе технологии их виртуализации

Одним из способов решения задачи динамического распределения и перераспределения виртуализированных ресурсов является «переразмещение» виртуальных машин [8].

Основной способ переразмещения виртуальных машин – живая миграция, позволяющая переносить работающие виртуальные машины между физическими машинами без прерывания работы [9].

Алгоритмы живой миграции виртуальных машин (предварительное копирование (pre-cory) и последующее копирование (post-cory)) выполняются по плану, разработанному службой управления виртуализированного ЦОД. Результатом планирования является бинарная матрица миграции виртуальных машин, где единица означает размещение виртуальной машины на физической, а ноль – отсутствие размещения [10, 11]. В небольших ЦОД матрица создается вручную администратором, в крупных – автоматизируется. Формирование матрицы позволяет не только переразмещать виртуальные машины, но и оптимизировать процесс, освобождая ресурсы и снижая энергопотребление.

В данной статье предлагается подход к разработке архитектуры программного комплекса поддержки процесса перераспределения виртуальных машин виртуализированного ЦОД на основе полученных результатов проведенных ранее исследований [12].

Виртуализированный ЦОД предлагается рассматривать как распределенную систему, поскольку он соответствует следующим ее характеристикам:

- 1) совместное использование ресурсов;
- 2) открытость;
- 3) параллелизм;
- 4) масштабируемость;
- 5) отказоустойчивость;
- 6) прозрачность;
- 7) гетерогенность.

В свою очередь под распределенной системой понимается группа автономных компьютеров, физически разделенных, но объединенных сетью и управляемых программным обеспечением [13]. Компьютеры взаимодействуют, делятся ресурсами, файлами и выполняют задачи.

Система поддержки перераспределения виртуальных машин в виртуализированном ЦОД относится к распределенным системам, учитывающим локальную и/или территориальную распределенность серверных платформ. Для разработки архитектуры рекомендуется использовать парадигму «менеджер-агенты» и специализированную двухуровневую архитектуру, изображенную на рис. 2.

Уровнями архитектуры разработанной распределенной системы поддержки перераспределения виртуальных машин на основе парадигмы «менеджер-агент» являются:

1. Уровень агентов – программных модулей, взаимодействующих с гипервизорами виртуализированного ЦОД и вышележащим уровнем. Функциями модуля-агента являются: получение данных от гипервизора о его типе, значениях наличия и остатка ресурсов, количестве активных виртуальных машин, потребности активных виртуальных машин в ресурсах.

2. Уровень менеджмента – программных модулей:

– формирование гомогенных подмножеств физических машин по типу используемых гипервизоров;

– формирование значений векторов ВНР (вектор наличия ресурсов), ВОР (вектор оставшихся ресурсов), ВВР (вектор выделения ресурсов) для подмножеств g_1 и g_2 [12];

– выполнение расчета значений множества векторов ВДР (вектор дисбаланса ресурсов) для подмножеств g_1 и g_2 [12];

– запуск копий модуля расчета матриц миграции ММГ для подмножеств g_1 и g_2 [12];

– формирование на основе рассчитанных матриц миграции планов миграции подмножеств g_1 и g_2 и передачи их уровню агентов.

Парадигма «менеджер-агент» распределяет задачи сбора и обработки данных между узлами системы, повышая гибкость и масштабируемость. Центральный менеджер координирует агентов, размещенных на серверных платформах. Агенты собирают данные о виртуальных и физических машинах

в локальных узлах и передают их менеджеру, который агрегирует и анализирует информацию, обеспечивая целостное представление о системе для оперативного выявления и устранения проблем.

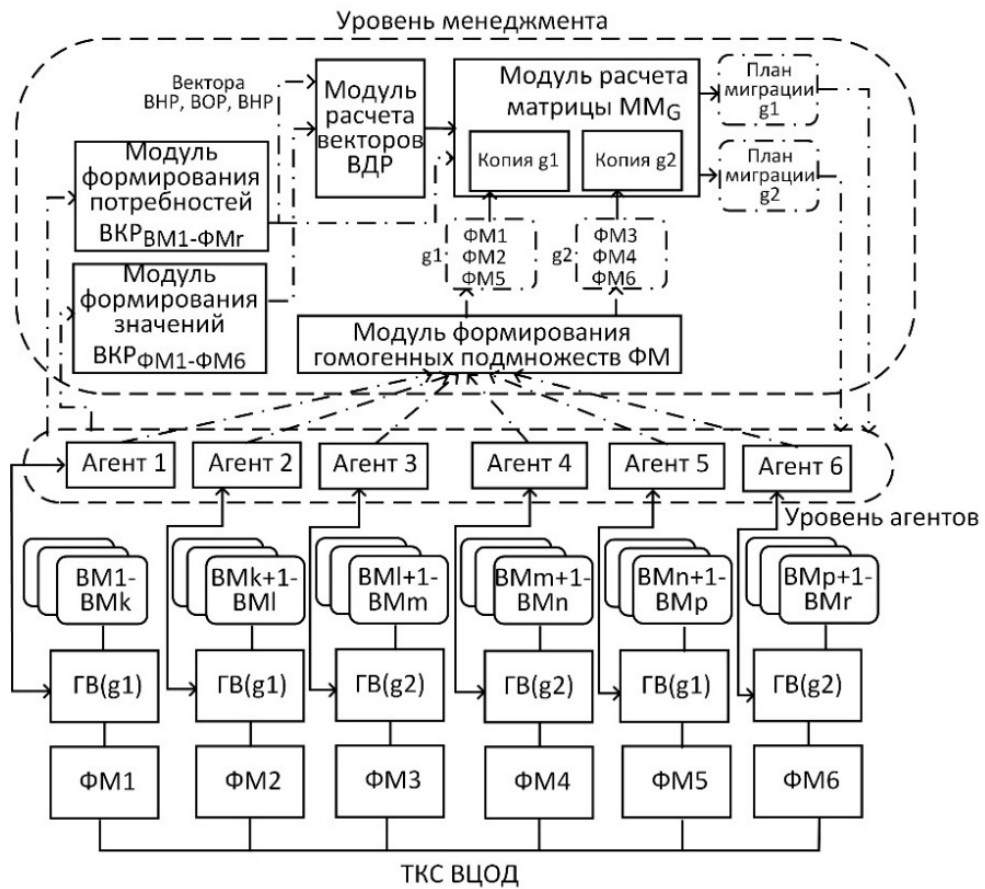


Рис. 2. Архитектура программного комплекса поддержки процесса переразмещения виртуальных машин на основе парадигмы «менеджер-агент»

Также данная парадигма обеспечивает высокую производительность, точность сбора данных и отказоустойчивость системы: при выходе агента из строя задачи перенаправляются другим агентам. Архитектура позволяет легко масштабировать систему за счет добавления новых серверов и агентов без значительных изменений в логике менеджера, что делает расширение быстрым и малозатратным.

Агенты для сбора данных, включая мониторинг оборудования и ресурсов, разработаны с учетом кроссплатформенности, поддерживают установку на различные операционные системы и аппаратные платформы. Это достигнуто за счет универсальных API, обеспечивающих интеграцию с существующими системами мониторинга.

Благодаря открытым интерфейсам и протоколам была обеспечена совместимость с такими системами, как Zabbix, NetCrunch, OpenNMS и др., что существенно упрощает интеграцию и позволяет снизить затраты на внедре-

ние системы поддержки [14]. Используя эти интерфейсы, менеджер может получить доступ к данным, уже собранным и обработанным текущими системами мониторинга, что исключает дублирование функций и значительно повышает эффективность работы.

Программные модули для распределенной системы поддержки перераспределения виртуальных машин предлагается реализовать на фреймворке PySyncObj, обеспечивающем интеграцию с системами мониторинга. Модули на уровне агентов собирают данные в реальном времени, взаимодействуя с уровнем менеджмента для непрерывного функционирования [15]. Уровень менеджмента отвечает за стратегическое планирование миграции виртуальных машин, используя векторы ВНР, ВОР и ВВР для адаптации к изменениям конфигурации ресурсов. Результаты расчетов матриц миграции передаются на уровень агентов для выполнения, обеспечивая гибкость, эффективность перераспределения виртуальных машин и оптимизацию использования ресурсов.

Разработка структуры имитационной модели гетерогенного виртуализированного центра обработки данных

Для оценки работоспособности и эффективности разработанного алгоритма необходимо решить задачу синтеза сложной системы с использованием средств компьютерного моделирования. Наилучшим выбором является применение инструментов имитационного моделирования как наиболее распространенного подхода [16, 17]. Цель моделирования – получение количественных (значения переменных) и качественных (свойства системы) результатов. Создание компьютерной модели требует учета всех внутренних связей объекта и точного отражения его факторов, что делает задачу сложной. Необходимо проанализировать доступные инструменты имитационного моделирования для решения поставленных задач.

В настоящее время на рынке программного обеспечения для имитационного моделирования представлено множество различных продуктов, разработчики которых предлагают широкий спектр систем для различных методов имитационного моделирования [17]. Обзор современных методов имитационного моделирования представлен в табл. 1.

Таблица 1

Программное обеспечение для имитационного моделирования

| Методы имитационного моделирования | Программные инструменты имитационного моделирования |
|------------------------------------|---|
| Дискретно-событийное моделирование | GPSS, AnyLogic, Arena, Extend, PowerSim Studio, ProModel, Pilgrim, SimScript, FlexSim и др. |
| Системная динамика | AnyLogic, Arena, SimBioSys, eMPlant, Plant Simulation, SimuLab, VenSim, Pilgrim, Dynamo, Stella и др. |
| Агентное моделирование | AnyLogic, SimAgent, SimBioSys, AgentSpeak, TeleScript, RePast, NetLogo, Mason и др. |

В связи с тем, что для архитектуры программного обеспечения, предназначенного для содействия перераспределению виртуальных машин, была выбрана парадигма «менеджер-агент», решение в пользу среды моделирова-

ния было принято в сторону продукта разработки российских специалистов – AnyLogic, разработка компании The AnyLogic Company. Эта платформа предоставляет возможность моделирования систем различных видов, включая дискретное событийное моделирование, системную динамику и агентное моделирование, что делает ее универсальным инструментом для решения широкого круга задач [18, 19].

Данная платформа для имитационного моделирования обладает множеством преимуществ: встроенные библиотеки, необходимые для научных исследований, поддержка программирования на Java для создания уникальных свойств объектов и классов, удобный и интуитивно понятный графический интерфейс, обширные примеры моделей и полноценная документация. Платформа также характеризуется гибкостью и наличием мощных аналитических инструментов.

Проектирование структуры имитационной модели гетерогенного виртуализированного центра обработки данных

Конструкция модели имитации для виртуализированного ЦОД с гетерогенной структурой включает следующие модули (генераторы компонентов структуры ЦОД):

1. Модуль генерации параметров ЦОД.
2. Модуль генерации параметров кластера.
3. Модуль генерации параметров виртуальных машин.
4. Модуль генерации параметров живой миграции.

В модуле генерации «Параметры ЦОД» создается количество кластеров физических машин, которые представлены как кластеры серверов, поскольку живая миграция виртуальных машин происходит только внутри одного кластера.

В модуле генерации «Параметры кластера» для физических машин указывается количество серверов и пропускная способность кластера. При этом в каждом сервере создаются ресурсы: С (процессор), М (RAM).

Тип ГВ (g_1 , g_2) представлен типом алгоритма живой миграции – g_1 – это Pre-copy; g_2 – это Post-copy.

В модуле генерации «Параметры виртуальных машин» задается количество виртуальных машин с запрашиваемыми/потребляемыми ресурсами: С (CPU), М (RAM). Для каждой виртуальной машины генерируются разные по объему ресурсы.

В модуле генерации «Параметры живой миграции» реализованы два алгоритма живой миграции виртуальных машин (Pre-copy и Post-copy) на основе результатов анализа [7] и три алгоритма перераспределения ресурсов ЦОД:

- 1) на основе метода XEN SandPiper – XEN SandPiper;
- 2) на основе метода вычисления нагрузки виртуализированного сервера (МНВС);
- 3) разработанный алгоритм перераспределения виртуальных машин ЦОД на основе метаэвристики ACS (МП ВМ ЦОД).

Принципы перераспределения ресурсов, реализованные в алгоритмах XEN SandPiper и МНВС рассмотрены в [12].

Вид интерфейса настройки структуры виртуализированного ЦОД в разработанной имитационной модели представлен на рис. 3.

The interface is divided into four main sections:

- Параметры ЦОД (DC Parameters):** Includes 'Количество кластеров' (2), 'Режим генерации' (manual/auto), and a 'generate' button.
- Параметры кластера (Cluster Parameters):** Includes '№ кластера' (кластер1), 'Имя кластера' (кластер1), 'Количество серверов' (5), 'Пропускная способность' (1000 Mb/s), '№ сервера' (сервер1), 'CPU' (5), 'RAM' (64 | 256), and buttons for 'generate', 'save', 'DataBase', and 'GEN_SERVER'.
- Параметры ВМ (VM Parameters):** Includes 'Режим генерации' (single_mode/multi_mode), 'Количество ВМ' (10), 'CPU' (4 | 8), 'RAM' (4 | 32), and buttons for 'generate', 'save', 'DataBase_VM', 'id_server', 'id_VM', and 'install_VM'.
- Параметры ЖМ (Hypervisor Parameters):** Includes 'Алгоритм миграции' (Pre-copy/Post-copy), 'Режим миграции' (Consistent/Parallel), 'Server_Source' (0), 'Server_Destination' (0), 'Name_VM', 'Count_hope', 'preinstall', 'migrate', 'Алгоритм распределения ВМ' (XEN SandPiper/MHBC/МП ВМ ЦОД), and a 'Distribute' button.

Additional buttons at the bottom include 'Clean_db' and 'main menu'.

Рис. 3. Интерфейс настройки структуры виртуализированного центра обработки данных

Результаты проведения имитационного эксперимента

Для оценивания результата работы алгоритма перераспределения виртуальных машин в виртуализированном ЦОД с различной структурой и степенью гетерогенности в рамках разработанной имитационной модели был спланирован сравнительный имитационный эксперимент. Эффективность работы разработанного алгоритма перераспределения виртуальных машин сравнивается с алгоритмом XSP и алгоритмом VSL.

При каждом прогоне имитационного эксперимента входные параметры создавались случайным образом генераторами нагрузки имитационной модели: количество физических машин, тип гипервизора каждой физической машины, начальные значения ресурсов каждой физической машины, количество виртуальных машин, запущенных на каждой физической машине, начальное значение ресурсов, используемых каждой виртуальной машиной.

Вид интерфейса сгенерированных параметров прогона имитационной модели представлен на рис. 4. Исследуемые алгоритмы перераспределения виртуальных машин подключались к модели через AnyLogic Python API (библиотека AnyLogic Cloud).

Степень гетерогенности виртуализированного ЦОД определяется фиксированным коэффициентом гетерогенности $K_{\text{гет}} = \{0; 0,5\}$ – соотношения физических машин с гипервизорами: Xen – g_1 и KVM – g_2 . Если $K_{\text{гет}} = 0$, то виртуализированный ЦОД полностью гомогенный (с гипервизором Xen или KVM), а если $K_{\text{гет}} = 0,5$, то виртуализированный ЦОД максимально гетерогенный (50 физических машин с гипервизором Xen и 50 физических машин с гипервизором KVM).

Имитационный эксперимент имитирует нагрузку на виртуализированный ЦОД с использованием случайно определяемого числа виртуальных машин и их параметров. Цель – оценить предельные возможности и слабые места алгоритма перераспределения виртуальных машин при масштабировании инфраструктуры. С ростом числа физических машин нагрузка на алгоритм возрастает, что может снизить его производительность и эффективность.

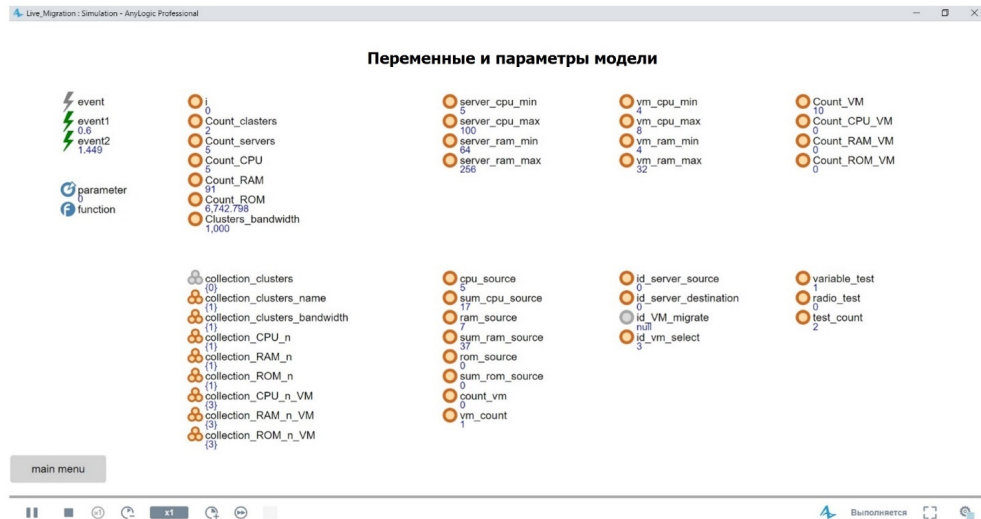


Рис. 4. Интерфейс генерации параметров и переменных имитационного прогона

Эффективность работы алгоритма перемещения виртуальных машин в виртуализированном ЦОД определяется коэффициентом использования физических машин. Этот коэффициент демонстрирует соотношение количества неиспользуемых физических машин к их общему числу в кластере. Для каждого уровня гетерогенности будет выполняться 10 прогонов перед применением алгоритмов перераспределения ресурсов и 10 прогонов после.

Усредненные результаты проведенных экспериментов имитационной модели представлены на рис. 5.

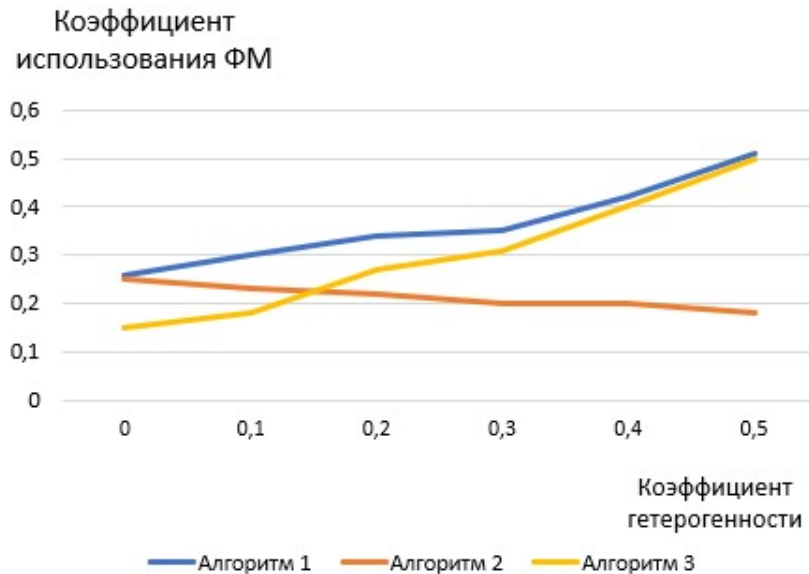


Рис. 5. Усредненные результаты имитационного эксперимента по оцениванию значения коэффициента использования физических машин после реализации исследуемых алгоритмов перераспределения виртуальных машин для виртуализированного ЦОД с коэффициентом гетерогенности

Заключение

Сформированная система поддержки отличается возможностью расширения и высокой адаптивностью к условиям современных виртуализированных ЦОД. Она позволяет централизованно управлять и следить за процессами, одновременно отвечая требованиям кроссплатформенной совместимости и интеграции с имеющимися средствами мониторинга, что превращает ее в ключевой инструмент для обеспечения надежности и продуктивности работы ЦОД.

Эффективное применение фреймворка PySyncObj и умелое распределение функциональных зон на два уровня архитектуры обеспечивает высокую производительность и надежность в управлении виртуальными машинами. Механизм поддержки перераспределения виртуальных машин не только облегчает работу администраторов, но и способствует более эффективному использованию вычислительных ресурсов, что особенно актуально в условиях быстрого развития современной ИТ-инфраструктуры.

Разработанная архитектура программного обеспечения для поддержки процесса перемещения виртуальных машин в гетерогенных ЦОД позволяет динамически перераспределять виртуальные машины, основываясь на дисбалансе виртуализированных вычислительных ресурсов, которые организованы в подгруппы физических машин с гомогенной архитектурой. Имитационное моделирование подтверждает успешное достижение поставленных задач исследования.

Список литературы

1. Жукабаева Т. К., Кусаинова А. Т. Технология Больших данных (Big Data). Основные характеристики и перспективы применения // ResearchGate. URL: <https://www.researchgate.net>
2. Hemn B. A. A brief survey on big data: technologies, terminologies and data-intensive applications // *Journal of Big Data*. 2022. Vol. 9 (1). P. 36. doi: 10.1186/s40537-022-00659-3
3. Bakhtiyar S., Oleg S. Building the Software Defined Data Center // *Proceedings of the Institute for System Programming of RAS*. 2018. Vol. 30 (6). P. 7–24. doi: 10.15514/ISPRAS-2018-30(6)-1.
4. Yaseein S. H., Maen A., Ahmed S. A., Saleh A. Data Centre Infrastructure: Design and Performance // *Latest Advances and New Visions of Ontology in Information Science*. 2023. doi: 10.5772/intechopen.109998
5. Бумажкина Н. Ю. К вопросу о виртуализации вычислительных и телекоммуникационных ресурсов центра обработки данных // *Научная сессия ТУСУР – 2024: XXIX Международная научно-техническая конференция студентов, аспирантов и молодых ученых*. Томск, 2024. С. 107–110.
6. Shabanov B. M., Samovarov O. I. Building the Software Defined Data Center // *Proceedings of the Institute for System Programming of RAS*. 2018. doi: 10.15514/ISPRAS-2018-30(6)-1
7. Bumazhkina N. Yu., Zaharova I. N., Kochkurov A. E. On the use of live migration technologies for virtual machines in the task of optimizing data center resources // *Modern informatization problems in simulation and social technologies (MIP-2024'SCT) Proceedings of the XXIX-th International Open Science Conference*. Yelm, WA, USA, 2024. Vol. 2. P. 133–137.
8. Mollamotalebi M. Shahnaz H. Multi-objective dynamic management of virtual machines in cloud environments // *Journal of Cloud Computing*. 2017. Vol. 6 (1). doi: 10.1186/s13677-017-0086-z

9. Choudhary A., Govil M., Singh G., Awasthi L., Pilli E., Kapil D. A critical survey of live virtual machine migration techniques // *Journal of Cloud Computing: Advances, Systems and Applications*. 2017. Vol. 6 (23). P. 2–41. doi: 10.1186/s13677-017-0092-1
10. Shakya A., Garg D., Nayak P. Hybrid Live VM Migration: An Efficient Live VM Migration Approach in Cloud Computing // *Advanced Informatics for Computing Research*. 2018. P. 600–611. doi: 10.1007/978-981-13-3140-4_54
11. Mishra M., Sahoo A. On theory of VM placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach // *IEEE International Conference on Cloud Computing (CLOUD)*. 2011. P. 275–282. doi: 10.1109/CLOUD.2011.38
12. Карпова О. В., Филонова Ю. Б., Кузина В. В. Критерии оценки качества и рисков технологического процесса разработки программного продукта // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2024. № 1. С. 56–69.
13. Bae S. JavaScript Data Structures and Algorithms: An Introduction to Understanding and Implementing Core Data Structure and Algorithm // *Library of Congress Control Number: 2019930417*. 2019. doi: 10.1007/978-1-4842-3988-9
14. Федорова Л. М. Системы мониторинга. Обзор и сравнение // *Вестник науки и образования*. 2020. Т. 4, № 10 (88). С. 16–18.
15. Ramasubramani V., Adorf C. S., Dice B. D. signac: A Python framework for data and workflow management // *17th Python in Science Conference At: Austin, TX*. 2018. doi: 10.25080/Majora-4af1f417-016
16. Майер Р. В. Компьютерное моделирование : учеб.-метод. пособие для студентов педагогических вузов [Электронное учебное издание на компакт-диске]. Глазов : Глазов. гос. пед. ин-т, 2015. 24,3 Мб.
17. Аверина Т. А. Построение алгоритмов статистического моделирования систем со случайной структурой : учеб. пособие. Новосибирск : РИЦ НГУ, 2015. 155 с.
18. Стенников В. А., Барахтенко Е. А., Майоров Г. С. Разработка мультиагентной модели интегрированной энергоснабжающей системы в программной среде AnyLogic // *Вестник Иркутского государственного технического университета*. 2020. Т. 24 (5). С. 1080–1092.
19. Егоров С. Выбор языка имитационного моделирования, или не заколачивайте гвозди микроскопом. URL: <https://www.anylogic.ru/blog/vybor-yazyka-imitatsionnogo-modelirovaniya-ili-ne-zakolachivayte-gvozd-i-mikroskopom/> (дата обращения: 09.01.2024).

References

1. Zhukabaeva T.K., Kusainova A.T. Big Data Technology (Big Data). Main characteristics and application prospects. *ResearchGate*. (In Russ.). Available at: <https://www.researchgate.net>
2. Hemn B.A. A brief survey on big data: technologies, terminologies and data-intensive applications. *Journal of Big Data*. 2022;9(1):36. doi: 10.1186/s40537-022-00659-3
3. Bakhtiyar S., Oleg S. Building the Software Defined Data Center. *Proceedings of the Institute for System Programming of RAS*. 2018;30(6):7–24. doi: 10.15514/ISPRAS-2018-30(6)-1
4. Yaseein S.H., Maen A., Ahmed S.A., Saleh A. Data Centre Infrastructure: Design and Performance. *Latest Advances and New Visions of Ontology in Information Science*. 2023. doi: 10.5772/intechopen.109998
5. Bumazhkina N.Yu. On the issue of virtualization of computing and telecommunication resources of the data processing center. *Nauchnaya sessiya TUSUR – 2024: XXIX Mezhdunarodnaya nauchno-tehnicheskaya konferentsiya studentov, aspirantov i molodykh uchenykh = Scientific session TUSUR – 2024: the 29th International scientific*

- and engineering conference of students, postgraduate students and young scientists. Tomsk, 2024:107–110. (In Russ.)
6. Shabanov B.M., Samovarov O.I. Building the Software Defined Data Center. *Proceedings of the Institute for System Programming of RAS*. 2018. doi: 10.15514/ISPRAS-2018-30(6)-1
 7. Bumazhkina N.Yu., Zaharova I.N., Kochkurov A.E. On the use of live migration technologies for virtual machines in the task of optimizing data center resources. *Modern informatization problems in simulation and social technologies (MIP-2024'SCT) Proceedings of the XXIX-th International Open Science Conference*. Yelm, WA, USA, 2024;2:133–137.
 8. Mollamotalebi M. Shahnaz H. Multi-objective dynamic management of virtual machines in cloud environments. *Journal of Cloud Computing*. 2017;6(1). doi: 10.1186/s13677-017-0086-z
 9. Choudhary A., Govil M., Singh G., Awasthi L., Pilli E., Kapil D. A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing: Advances, Systems and Applications*. 2017;6(23):2–41. doi: 10.1186/s13677-017-0092-1
 10. Shakya A., Garg D., Nayak P. Hybrid Live VM Migration: An Efficient Live VM Migration Approach in Cloud Computing. *Advanced Informatics for Computing Research*. 2018:600–611. doi: 10.1007/978-981-13-3140-4_54
 11. Mishra M., Sahoo A. On theory of VM placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach. *IEEE International Conference on Cloud Computing (CLOUD)*. 2011:275–282. doi: 10.1109/CLOUD.2011.38
 12. Karpova O.V., Filonova Yu.B., Kuzina V.V. Criteria for assessing the quality and risks of the technological process of software product development. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences*. 2024;(1):56–69. (In Russ.)
 13. Bae S. JavaScript Data Structures and Algorithms: An Introduction to Understanding and Implementing Core Data Structure and Algorithm. *Library of Congress Control Number: 2019930417*. 2019. doi: 10.1007/978-1-4842-3988-9
 14. Fedorova L.M. Monitoring systems. Review and comparison. *Vestnik nauki i obrazovaniya = Bulletin of science and education*. 2020;4(10):16–18. (In Russ.)
 15. Ramasubramani V., Adorf C.S., Dice B.D. signac: A Python framework for data and workflow management. *17th Python in Science Conference At: Austin, TX*. 2018. doi: 10.25080/Majora-4af1f417-016
 16. Mayer R.V. *Komp'yuternoe modelirovanie: ucheb.-metod. posobie dlya studentov pedagogicheskikh vuzov = Computer modeling: teaching guide for students of pedagogical universities*. Glazov: Glazov. gos. ped. in-t, 2015;24,3 Mb. (In Russ.)
 17. Averina T.A. *Postroenie algoritmov statisticheskogo modelirovaniya sistem so sluchaynoy strukturoy: ucheb. posobie = Construction of algorithms for statistical modeling of systems with random structure: teaching guide*. Novosibirsk: RITs NGU, 2015:155. (In Russ.)
 18. Stennikov V.A., Barakhtenko E.A., Mayorov G.S. Development of a multi-agent model of an integrated energy supply system in the AnoLogic software environment. *Vestnik Irkutskogo gosudarstvennogo tekhnicheskogo universiteta = Bulletin of Irkutsk State Technical University*. 2020;24(5):1080–1092. (In Russ.)
 19. Egorov S. *Vybor yazyka imitatsionnogo modelirovaniya, ili ne zakolachivayte gvozdi mikroskopom = Choosing a simulation language, or don't hammer nails with a microscope*. (In Russ.). Available at: <https://www.anylogic.ru/blog/vybor-yazyka-imitatsionnogo-modelirovaniya-ili-ne-zakolachivayte-gvozdi-mikroskopom/> (accessed 09.01.2024).

Информация об авторах / Information about the authors

Наталья Юрьевна Бумажкина

адъюнкт, Академия Федеральной службы
охраны Российской Федерации (Россия,
г. Орёл, ул. Приборостроительная, 35)

E-mail: orui@mvd.ru

Natalya Yu. Bumazhkina

Adjunct, Academy of the Federal Security
Service of the Russian Federation
(35 Priborostroitelnaya street, Orel, Russia)

Автор заявляет об отсутствии конфликта интересов / The author declares no conflicts of interests.

Поступила в редакцию / Received 15.01.2025

Поступила после рецензирования и доработки / Revised 25.01.2025

Принята к публикации / Accepted 14.02.2025