DOI 10.37882/2223–2966.2021.06.31

АНАЛИЗ ГИПЕРКОНВЕРГЕНТНОЙ ВЫЧИСЛИТЕЛЬНОЙ ИНФРАСТРУКТУРЫ ДЛЯ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ ВЫСОКОНАГРУЖЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

ANALYSIS OF HYPERCONVERGED COMPUTING INFRASTRUCTURE FOR DATA STORAGE AND PROCESSING OF HIGH-LOAD INFORMATION SYSTEMS

Yu. Sagalaev
O. Romashkova

Summary. The article is devoted to the consideration of the hyperconverged approach for building infrastructure as a storage platform and performing a virtualized workload. The advantages of a hyperconverged system in reducing the cost of supporting an information system relative to the traditional approach of infrastructure management are demonstrated. The issues of reliability and stability of such systems as a runtime environment for information systems for various purposes are considered. The experience of using the openstack open platform to build a ready-made solution is described.

Keywords: cloud computing, hypervisor, virtualization, IT infrastructure, data storage system, information system.

Сагалаев Юрий Романович
Аспирант, ГАОУ ВО «Московский городской педагогический Университет (МГПУ)» г. Москва
yrok472@mail.ru
Ромашкова Оксана Николаевна
Д.т.н., профессор, Российская академия народного хозяйства и государственной службы при Президенте РФ, г. Москва
ox-rom@yandex.ru

Аннотация. Статья посвящена рассмотрению гиперконвергентного подхода для построения инфраструктуры как платформы хранения и выполнения виртуализированной рабочей нагрузки. Продемонстрированы преимущества гиперконвергентной системы в сокращении издержек на поддержку информационной системы относительно традиционного подхода управлением инфраструктурой. Рассмотрены вопросы надежности и стабильности подобных систем в качестве среды выполнения информационных систем различного назначения. Описывается опыт применения открытой платформы openstack для построения готового решения.

Ключевые слова: облачные вычисления, гипервизор, виртуализация, ИТ-инфраструктура, система хранения данных, информационная система.

Введение

Существующая потребность в обеспечении качественного и быстрого отклика систем принятия решений диктует высокие требования к аппаратным и программным платформам, предоставляющих ресурсы для выполнения необходимого программного обеспечения. Одним из показателей качества системы является время отклика и получение конечного результата. Для достижения наилучшего результата недостаточно грамотно выстроить архитектуру самого приложения в связке с современным и производительным аппаратным обеспечением, поскольку важным критерием обеспечения высокой производительности зачастую оказывается программная платформа, реализующая управление системы в целом. Традиционный подход в построении систем хранения данных (СХД) претерпел значительные изменения за последние 10 лет. Необходимость в масштабировании инфраструктуры во время эксплуатации создало потребность в использовании программно-распределяемых хранилищ (software defined storage, SDS). SDS предоставляют автоматизированные и политико-ориентированные сервисы хранения, учитывающие особенности клиентов и приложений, использующие базовую инфраструктуру хранения и поддержку в целом программно-определяемой среды [1]. В рамках сохранения совместимости и возможности переноса все более распространенным становится виртуализация рабочей нагрузки, в том числе и модулей информационной системы (ИС). Гиперконвергентный подход в построении инфраструктуры (Hyper Convergent Infrastructure, HCI) сочетает в себе возможность совмещения SDS и гипервизора на одних и тех же физических узлах кластера, позволяет лучше рассчитывать стоимость и производительность всего комплекса на всех этапах эксплуатации комплекса (рисунок 1).

Гиперконвергентная инфраструктура

Гиперконвергентные вычислительные системы можно рассматривать применительно к любым вычислительным платформам (аппаратным, программным, облачным, нейроморфным, квантовым и др.), которые

Рис. 1. Сравнение подходов в построении инфраструктуры

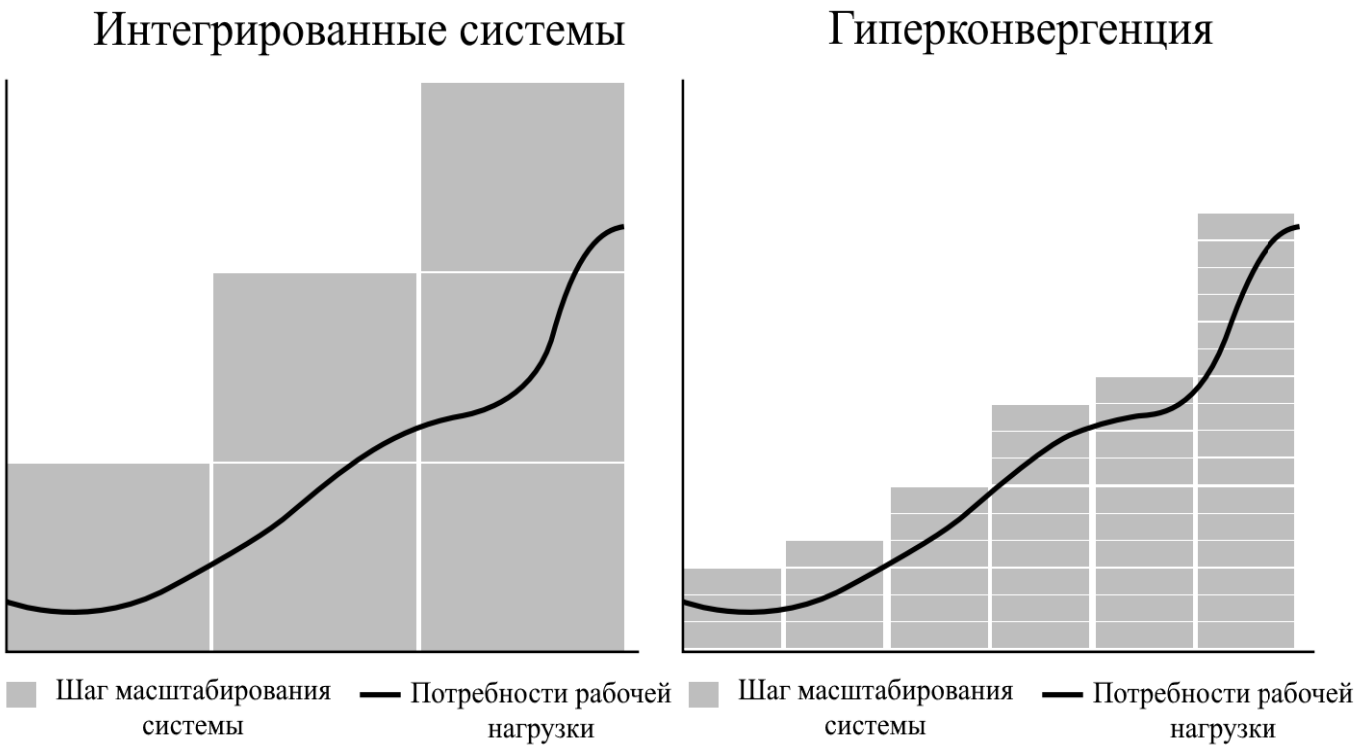

Рис. 2. Сравнение утилизации ресурсов при запуске нагрузки

могут обеспечивать пользователю доступ к различным сервисам [2]. HCI как подход упрощает архитектуру всей системы, ее обслуживания, поддержки и масштабирования. В отличии от традиционного подхода, где используются физически разрозненные аппаратные ресурсы и нецентрализованный программный слой, роль в управлении и обслуживании кластера и его нагрузки выполняет готовое программное решение, которое, в свою очередь, делится на несколько слоев:

1. Операционная система — основа для запуска всего программного обеспечения кластера;
2. Слой управления SDS, независящий от вышестоящих слоев;

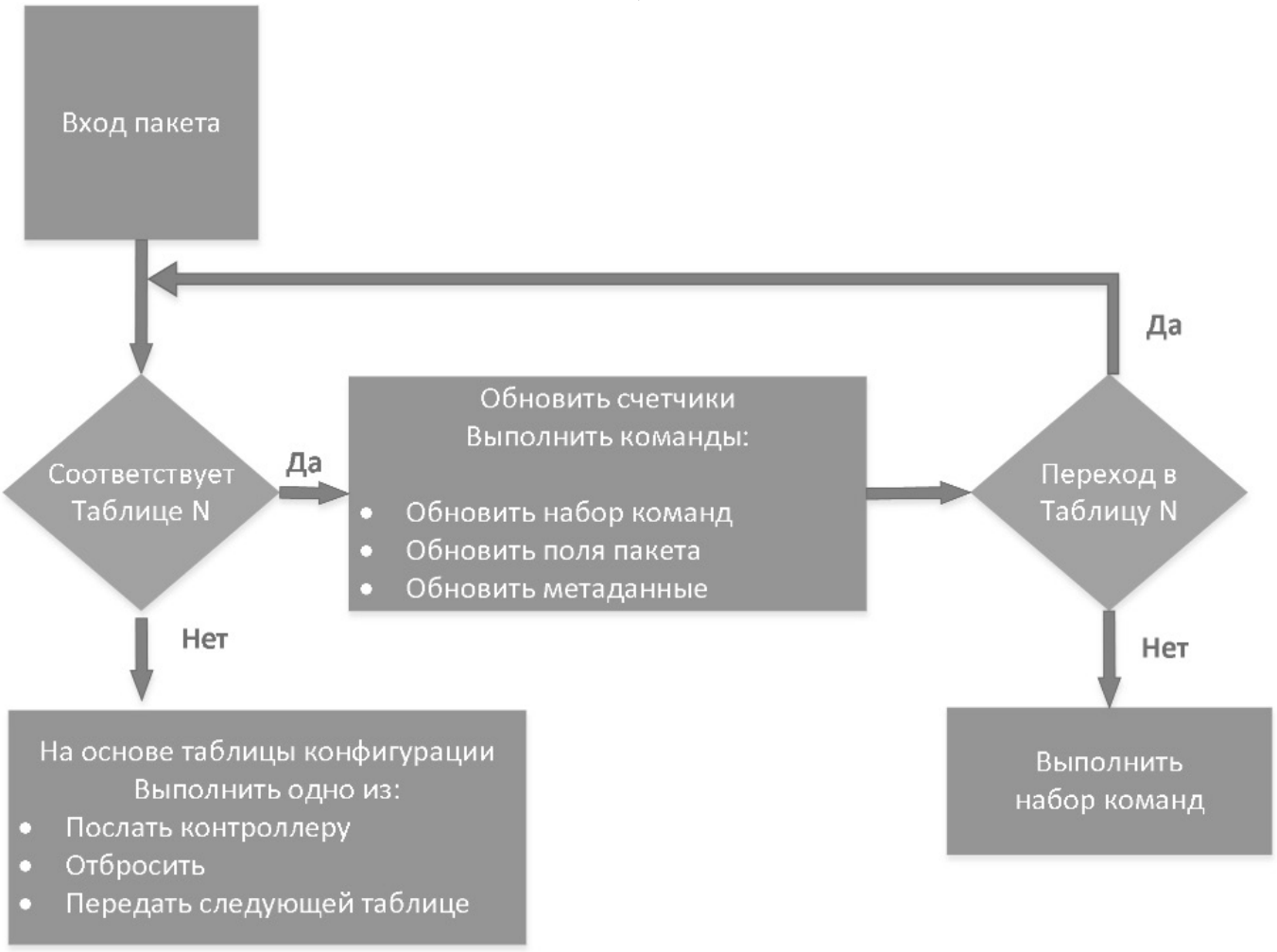

Рис. 3. Диаграмма прохождения пакета через переключатель OpenFlow

3. Гипервизор, выполняющий непосредственное исполнение виртуализированной полезной нагрузки;
4. Слой управления вычислительным кластером — система оркестрации виртуальными ресурсами, такими как виртуальные машины, виртуальные частные сети, маршрутизаторы, балансировщики нагрузки трафика.

В результате деления на уровни, каждый новый слой не влияет на отказ предыдущего. Важной особенностью HCI кластера является возможность точечного масштабирования под обновляющиеся требования, предъявляемые к системе. Типовой рекомендуемой конфигурацией каждого из узлов является наличие современного серверного многоядерного процессора с поддержкой набора инструкций, поддерживающих виртуализацию [3], а также наличие твердотельных носителей для выполнения задачи кэширования данных в SDS. Зависимость потребления мощностей кластера полезной нагрузкой (реляционные СУБД) классическим подходом и HCI в виде диаграмм представлен на рисунке 2. «Большой шаг» при масштабировании системы в традиционной схеме ведет к неоптимальной схеме всей сети, поскольку не происходит полноценная равномерная утилизация ресурсов кластера. Все доступные ресурсы выше линии потребности рабочей нагрузки оказываются не использованными, в тоже время невозможно перераспределить свободные ресурсы под новую задачу, требуется вмешательство и ручной перенос физических ресурсов между серверами кластера [4]. Неиспользованные ресурсы продолжают потреблять электроэнергию в простое, что вызывает расходы на охлаждение [5].

HCI подход позволяет выстраивать кластер с «меньшим» шагом, тем самым не допуская потери «излишков» ресурсов. Применение единого ПО на всех узлах позволяет быстро добавлять новые ресурсы уже к существующему кластеру. По этой причине вместо команды IT

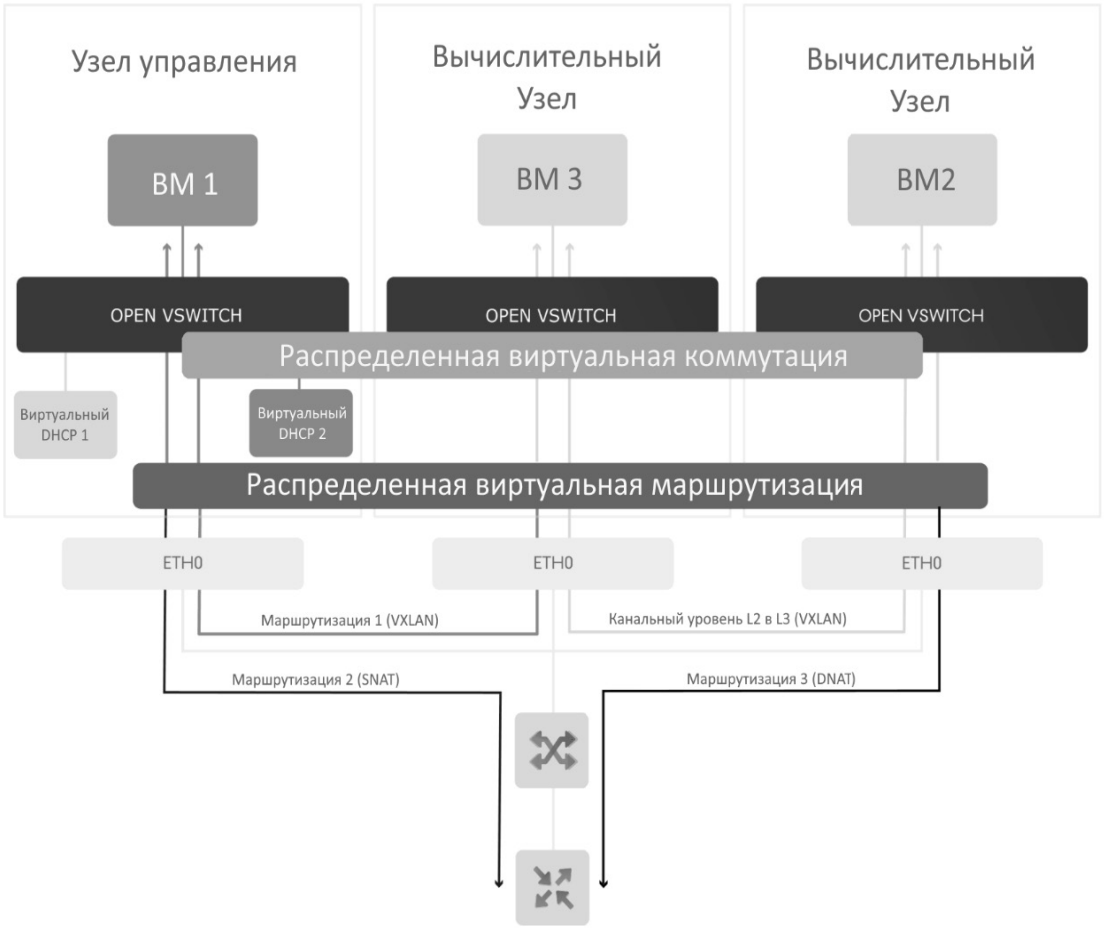

Рис. 4. Сравнение утилизации ресурсов при запуске нагрузки

специалистов для управления хранилищами данных и серверным оборудованием порой достаточно одного системного администратора [6].

Использование гиперконвергентного подхода в построении информационных систем

Выполнение соглашения об уровне обслуживания (Service Level Agreement — SLA) является важным требованием при планировании инфраструктуры под задачи информационной системы (ИС). Чтобы обеспечить коммерческий успех системы, необходимо полностью учитывать целевые ограничения качества обслуживания (Quality of Service — QoS) при планировании заданий пользователя, чтобы максимально удовлетворить потребности пользователя в QoS [7]. SLA подразумевает не только как быстро будет работать система, но и как стабильно доступна система во времени.

Для достижения требуемого уровня обслуживания ИС применяются подходы приоритезации и ограничения производительности виртуальных сетей и виртуальных носителей, обслуживающих виртуализированные модули ИС.

В качестве механизма приоритезации виртуального трафика в сетях, объединяющие модули ИС используется открытый протокол OpenFlow для управления передачи данных сети, реализованный в виртуальных маршрутизаторах кластера. В процессе переадресации определенные поля заголовка пакета могут модифицироваться оговоренным образом (рисунок 3).

Подобный способ позволяет выделять весь трафик, проходящий через маршрутизаторы виртуальных сетей, реализуя приоритезацию в виде организации очередей, разрыв соединения в случае отсутствия настроенного TLS или наличия неопределенного трафика в сети. В отличие от физического маршрутизатора, виртуальный переключатель применяет новые правила на ходу организации нового виртуального маршрутизатора или сети, не требует добавления новой физической сети кластера. Таким образом, можно выстроить

Таблица 1. Описание характеристик тестовых узлов

Компонент	Характеристика
Процессор	Intel Xeon Silver 4208 CPU3.20 ГГЦ
SSD	INTEL D3-S4510 960 Гбайт
Оперативная память	DIMM DDR4 2933 МГц
Сеть	Ethernet 1 Гбит/с

Таблица 2. Результаты тестирования I/O-тестов в HCI с использованием различных профилей производительности

Название профиля	Размер блока	Режим	Результат
Базовый	1 Мбайт	запись	257 Мбит/с
Случайная запись	4 кбайт	запись	65,9 Мбит/с
Случайное чтение	4 кбайт	чтение	71,8 Мбит/с
Последовательная запись	1 Мбайт	запись	247 Мбит/с
Последовательное чтение	1 Мбайт	чтение	137 Мбит/с
Случайное чтение/запись	4 кбайт	Запись/чтение	19,4 Мбит/с; 45,1 Мбит/с

сложную внутреннюю сеть из десятков виртуальных сетей и маршрутизаторов, подчиняющихся определенным правилам, не прибегая к приобретению нового физического оборудования. На рисунке 4 отображена схема физического представления подключения виртуальной сети в кластере. Технология VXLAN, используемая для виртуальных сетей, позволяет создавать логические сети L2 в сетях L3 путем инкапсуляции (туннелирования) кадров Ethernet через пакеты UDP.

QoS подход в искусственном ограничении пропускной способности сети и производительности дисков применяется для достижения прогнозируемой нагрузки на кластер. Зная необходимую пиковую нагрузку отдельно взятого модуля, можно ограничить ресурсы так, чтобы не допустить ситуаций, связанных с отказом всей системы из-за утечек памяти.

Еще одним подходом в достижении необходимого SLA значения является избыточная репликация данных между узлами, позволяя использовать автоматические сценарии восстановления целостности нагрузки, без необходимости вмешательства персонала в работу кластера.

Применение Openstack решения в разработке производительной высоконагруженной ИС

В качестве практического примера для определения производительности системы был развернут демонстрационный HCI кластер с использованием трех идентичных узлов, приведенных в таблице 1. Программным слоем всей системы служит дистрибутив Centos 7 на базе ядра Linux, объектное хранилище на базе SDS CEPH [8] и система openstack для управления вычислительным кластером.

Наиболее популярным гипервизором с открытым исходным кодом, поддерживающий управление openstack сервисом nova является KVM (Kernel-based Virtual Machine) [9]. Данные виртуальной машины, запущенной на базе KVM, хранятся на SDS с настроенным уровнем репликации равным трем, соответствующим массиву RAID-1 из трех носителей. Используемыми инструментами объективного измерения производительности на вычислительном кластере были выбраны утилиты командной строки fio и stress. Fio — это системная утилита для тестирования подсистемы ввода/вывода, которая используется в тестах производительности и нагрузочного тестирования аппаратного обеспечения [10].

Используя описанное ПО, получим усредненное значение производительности дисковой подсистемы и SDS целиком, в сценариях близким к высоконагруженным ИС. В ходе выполнения утилиты с использованием набора профилей, повторяющих поведение ИС в случайном обращении и записи на SDS, сформировалась оценка производительности для каждого из набора, представленная в таблице 2.

Каждый тест выполняется 10 раз подряд отдельно от остальных в течении одной минуты. Для каждого теста указывается разный объем блока данных, использующиеся при обращении к SDS. Так, случайная запись и чтение с блоком размера 4 кбайт хорошо повторяет сценарии обращения к СУБД и другим службам ИС в ходе ее повседневного использования, а последова-

тельное чтение с блоком размера 1 Мбайт демонстрирует выгрузку большого объема данных для последующего анализа.

Полученные результаты производительности SDS ожидаемо отличаются от результатов, полученных аналогичными тестами на системах без использования SDS. Сниженная производительность напрямую связана с необходимостью синхронизации всего SDS, что влечет за собой вычислительные издержки, а также необходимость передачи данных между узлами во время операций при обращении к виртуальной файловой системе кластера. Средние значения с использованием SDS в среднем меньше на 10–15% относительно номинальной производительности носителей. В качестве решения проблемы уменьшения производительности SDS можно применить ряд мер:

1. Увеличение сетевой пропускной способности между узлами кластера. Применение более скоростных интерфейсов со скоростями 10 Гбит/с или их агрегация, использование Infiniband сети вместо классической ethernet подхода позволяет утилизировать ресурсы носителей более полно и с меньшими задержками. Настройка длины кадра в сети передачи также играет роль в производительности кластера в целом, уменьшается нагрузка на ЦПУ, следовательно, это сказывается на времени отклика.
2. Правильный выбор требуемого уровня кодирования и отказоустойчивости. Задачи высоконагруженной ИС предполагают активный и не всегда предсказуемый уровень обращения к данным. Следует использовать действительно требуемый уровень отказоустойчивости и не использовать виды кодирования, не предназначенные для активных операций по записи и чтению небольших фрагментов, таких как избыточное кодирование с использованием кодов Рида-Соломона. Использование репликации выше значения уровня три повышает нагрузку кластера, при этом, не давая практической пользы в виду маловероятности одновременного выхода из строя более двух узлов, при условии энергонезависимости друг от друга.
3. Распределение данных между разными слоями SDS. Хранение больших архивных данных на отдельных носителях SDS, использование «быстрого» слоя из твердотельных носителей под СУБД и других модулей ИС.
4. Использование QoS политик для каждого виртуализированного модуля ИС. Искусственное ограничение производительности отдельно взятых виртуальных машин позволяет освободить ресурсы для других более критически важных модулей ИС.

Заключение

Проведено исследование возможности применения HCI подхода, а также его преимущества для обеспечения отказоустойчивости и приемлемой скорости в работе высоконагруженных ИС, требующих низкий уровень задержек при обращении к данным. Рассмотрен пример конкретного HCI кластера и его производительность в различных сценариях обращения к SDS. Опираясь на полученные результаты производительности кластера в разных режимах работы, можно сделать выводы о том, что не смотря на ожидаемый уровень падения производительности, HCI подход дает преимущества отказоустойчивости и быстрого восстановления ИС. Использование механизмов репликации и применение виртуальных сетей позволяет минимизировать издержки на приобретение и обслуживание нового физического оборудования, позволяя максимально утилизировать доступные ресурсы. Исходя из полученной оценки естественной и гибридной схемы взаимодействия с носителями, для соблюдения требуемого QoS был сформирован комплекс мер при планировании и организации высоконагруженной системы.

ЛИТЕРАТУРА

1. Мазур Э.М. Распределенные системы хранения данных: анализ, классификация и выбор // Перспективы развития информационных технологий. 2015. № 26. URL: https://cyberleninka.ru/article/n/raspredelennye-sistemy-hraneniya-dannyh-analiz-klassifikatsiya-i-vybor (дата обращения: 30.04.2021).
2. Соловьев В.М. Конвергентные и гиперконвергентные вычислительные системы // Изв. Сарат. ун-та Нов. сер. Сер. Математика. Механика. Информатика; Izv. Saratov Univ. (N.S.), Ser. Math. Mech. Inform. 2018. № 1. URL: https://cyberleninka.ru/article/n/konvergentnye-i-giperkonvergentnye-vychislitelnye-sistemy (дата обращения: 02.05.2021).
3. Системные требования платформы openstack. URL: https://docs.openstack.org/murano/rocky/admin/deploy_murano/prerequisites.html (дата обращения: 02.05.2021).
4. Tang Q., Gupta S.K. S., Varsamopoulos G., Energy-effcient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach, IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 11, pp. 1458–1472, 2008.
5. Кудрявцев А.О., Кошелев В.К., Избышев А.О., Дудина И.А., Курмангалеев Ш.Ф., Аветисян А.И., Иванников В.П., Велихов В.Е., Рябинкин Е.А. Разработка и реализация облачной системы для решения высокопроизводительных задач // Труды ИСП РАН. 2013. № . URL: https://cyberleninka.ru/article/n/razrabotka-i-realizatsiya-oblachnoy-sistemy-dlya-resheniya-vysokoproizvoditelnyh-zadach (дата обращения: 02.05.2021).

6. Носкова А.И., Токранова М.В. Преимущество гиперконвергентных систем над облачными технологиями // Интеллектуальные технологии на транспорте. 2017. № 2. URL: https://cyberleninka.ru/article/n/preimuschestvo-giperkonvergentnyh-sistem-nad-oblachnymi-tehnologiyami (дата обращения: 05.05.2021).
7. Sun H.-B., Ding Y.-S. QoS scheduling of fuzzy strategy grid workflow based on the bio-network // Int. J. Comput. Sci. Eng. 2011. Vol. 6, № 1–2. P. 114–121.
8. Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. Long, Carlos Maltzahn. Ceph: A scalable, high-performance distributed file system. In Proceedings of the 7th symposium on Operating systems design and implementation, pp. 307–320. USENIX Association, 2006.
9. Газуль С.М. Формирование динамически конфигурируемой информационной инфраструктуры организации: 08.00.05: — Санкт-Петербург., 2018.
10. Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, Anthony Liguori. kvm: the linux virtual machine monitor. In Proceedings of the Linux Symposium, volume 1, pp. 225–230, 2007.

Московский городской педагогический университет