УДК 004.75
DOI 10.34822/1999-7604-2021-2-24-30

ПРИМЕНЕНИЕ ВИРТУАЛЬНЫХ КОНТЕЙНЕРОВ ПРИ СОЗДАНИИ МЕДИЦИНСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ

Н. Р. Урманцева ✉, Д. В. Хитрень
Сургутский государственный университет, Сургут, Россия
✉ *E-mail: nel-u@yandex.ru*

Рассмотрено применение виртуальных контейнеров для повышения эффективности обработки данных и быстрого развертывания инновационных решений для использования «облачных» приложений и их модернизации на основе микросервисов. Контейнеризация помогает ускорить выпуск новых версий программных продуктов, обеспечивает переносимость между гибридными и мультиоблачными средами, а также позволяет сократить инфраструктурные и эксплуатационные затраты. Описана концепция медицинской информационной системы врача – флеболога – сердечно-сосудистого хирурга с применением виртуальных контейнеров, а также ее реализованный в двух вариантах модуль по сбору флебологических данных.

Ключевые слова: облачные вычисления, виртуализация, виртуальный контейнер, Docker.

THE USE OF VIRTUAL CONTAINERS IN THE MEDICAL INFORMATION SYSTEMS DEVELOPMENT

N. R. Urmantseva ✉, D. V. Khitren
Surgut State University, Surgut, Russia
✉ *E-mail: nel-u@yandex.ru*

The article discusses the use of virtual containers, which can improve data processing efficiency and enable customers to quickly deploy innovative solutions to modernize applications and use cloud-based applications based on microservices. It states that containerization helps accelerate the release of new versions of software, provides portability between hybrid and multi-cloud environments, and also reduces infrastructure and operational costs. The concept of a medical information system of a cardiovascular surgeon-phlebologist with the use of virtual containers is described, as well as a system for collecting phlebology data in two versions.

Keywords: cloud computing, virtualization, virtual container, Docker.

Введение

Технологии виртуализации непрерывно развиваются с середины 1960-х гг. В настоящее время существуют два подхода к созданию независимых изолированных вычислительных пространств на одном физическом сервере: виртуальные машины (ВМ) – аппаратная виртуализация (рис. 1), которым нужен гипервизор, и виртуальные контейнеры (ВК) (рис. 2). В первом случае гипервизор позволяет выполнять одновременный запуск нескольких операционных систем (ОС) на одной ЭВМ, эмулируя физические характеристики аппаратного обеспечения конкретного компьютера и управляя компьютерами, выделяя и освобождая ресурсы для них. При этом «гипервизор осуществляет управление ресурсами и их разделение между различными операционными системами, выполняет изоляцию запущенных операционных систем друг от друга, а также обеспечивает их взаимодействие» [1, с. 7]. Во втором случае ВК – это виртуализация не на аппаратном уровне, а на уровне ОС: каждая гостевая ОС использует то же самое ядро, что и базовая.

Рис. 1. Виртуальные машины [2]

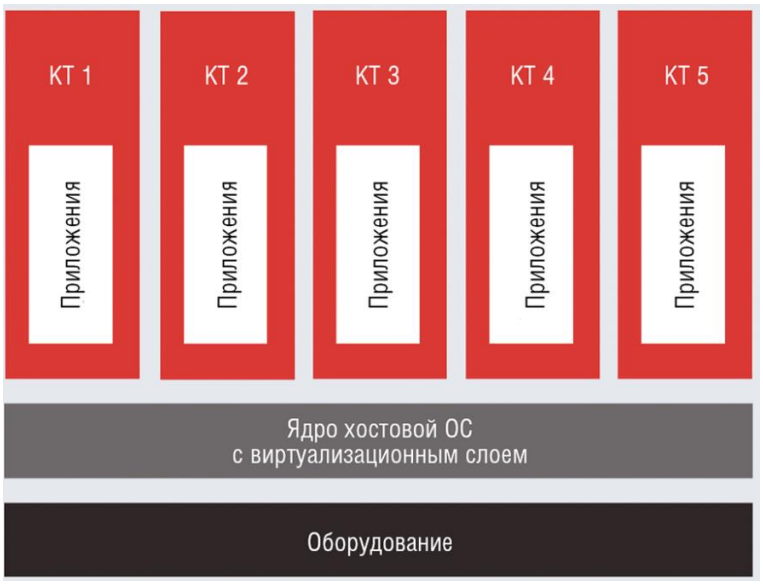

Рис. 2. Виртуальные контейнеры [2]

Теоретические аспекты

Виртуальные контейнеры – это программная технология, инкапсулирующая компьютерные программы из базовой ОС, на которой она выполнена. ВК обеспечивают абстрагирование приложений от реальных сред, в которых они выполняются, поэтому приложения могут работать быстро и эффективно в различных вычислительных средах.

Еще один важный момент – это сравнение ВК с ВМ. Виртуальные машины – это гостевые ОС, работающие поверх ОС хоста с виртуальным доступом к системному оборудованию. Единственное сходство между ВК и ВМ – это возможность получения изолированных сред для запуска программных сервисов.

Следует отметить, что ВК не требуется отдельная ОС, они используют меньше ресурсов, чем ВМ, для их работы обычно достаточно лишь нескольких десятков мегабайт. Это, безусловно, преимущество по сравнению с ВМ, которые обеспечивают виртуализацию на аппаратном уровне. Более того, ВК запускаются с большей скоростью и совместно используют ядро ОС, задействуя таким образом лишь часть памяти, необходимой для загрузки всей ОС.

С помощью контейнерной виртуализации возможно изолировать работу пользовательского приложения от остальной системы. Например, пакет Docker [3] позволяет запускать процессы в изолированном окружении, в котором сам процесс сосуществует вместе со своими процессами-потомками. Хотя при этом процесс работает в той же ОС, что и остальные процессы, он их не идентифицирует. Таким образом, Docker может помочь разработчику разделить само приложение и систему, поместив его в Docker-контейнер и, если это необходимо, перенеся на другую подобную систему.

Оркестрирование представляет собой автоматизированный процесс управления связанными объектами – группами ВМ или ВК. Платформы оркестрирования ВК, например Kubernetes, идеально подходят для установки, масштабирования и управления рабочими нагрузками и службами. Kubernetes эффективно работает с ВК Docker и другими контейнерными системами, соответствующими спецификации Open Container Initiative (OCI). Преимущества платформ оркестрирования ВК также помогают упростить различные задачи управления, которые могут включать развертывание новых версий приложений и масштабирование контейнерных приложений.

Как отмечено в [4], крупные облачные провайдеры предоставляют своим клиентам сервисы для запуска ВК – готовые среды, снабженные удобным интерфейсом управления. Они не требуют крупных затрат, поскольку вся инфраструктура арендуется, а конфигурация создается в минимальном объеме, относящемся исключительно к запускаемому приложению. И по указанному пользователем небольшому объему параметров кластеры настраиваются провайдером.

Концепция МИС с применением виртуальных контейнеров

Виртуальные контейнеры – оптимальный вариант для стартап-компаний, например для частных профильных медицинских центров, занимающихся междисциплинарными исследованиями в области математического и нейросетевого моделирования физиологических процессов. В отличие от обычных медицинских информационных систем (МИС), в которых хранящаяся информация должна быть защищена законом о персональных данных № 152-ФЗ, медицинская информация для научных исследований может быть обезличена и только потом обработана с использованием функционала, заложенного в ВК. На рис. 3 изображена схема МИС с применением ВК в географически распределенной филиальной сети.

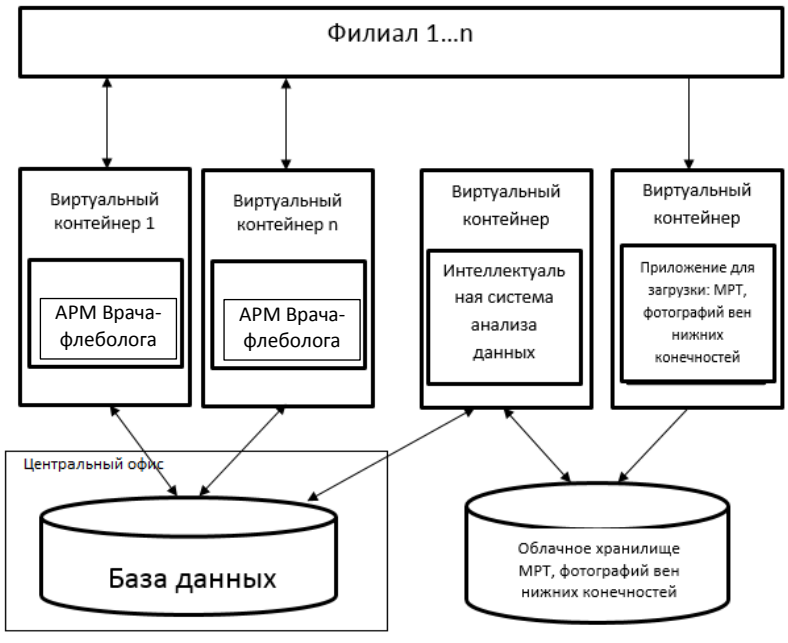

Рис. 3. Схема МИС с применением виртуальных контейнеров
Примечание: разработано авторами.

В предложенном для будущей реализации МИС флебологического центра автоматизированном рабочем месте (АРМ) врача – флеболога – сердечно-сосудистого хирурга приложение для загрузки МРТ-снимков и фотографий вен нижних конечностей, интеллектуальная система анализа данных, т. е. все нужные для работы компоненты, не разворачиваются в виртуальной среде, а находятся в специальных контейнерах. При таком подходе реализуется микросервисная архитектура, в которой компоненты информационной системы распределены. Пользователь системы подключается к приложению, запускает АРМ и работает напрямую с центральной базой данных. Сервер БД устанавливается в центральном или наиболее крупном подразделении, где возможно наибольшее число обращений к БД, либо на арендованном сервере у стороннего провайдера.

В облачном контейнере расположена интеллектуальная система анализа данных (а именно система поддержки принятия решений для постановки диагноза «хроническая венозная недостаточность»), построенная на базе сверточной нейронной сети, которая способна поставить в соответствие изображениям ног пациента код классификации хронических заболеваний вен CEAP (Clinical, Etiologic, Anatomic, Pathophysiologic). В качестве данных для обучения и тестирования нейронной сети используются фотографии нижних конечностей пациентов, которым были поставлены в соответствие различные классы патологий венозной системы и результаты МРТ-обследования в формате DICOM, необходимые для уточнения диагноза.

Для клиента на терминале при такой реализации разрабатываемой системы не меняется интерфейсная компонента, но увеличивается скорость доступа к БД результатов МРТ-исследований и фотографий вен нижних конечностей, для хранения которых при классической реализации МИС обычно бывает недостаточно емкости запоминающих устройств сервера клиники. Для администратора исчезает необходимость аппаратного обслуживания той части системы, которая отвечает за интеллектуальный анализ данных и хранение графических результатов обследования.

Существующие косвенные аналоги разрабатываемой системы (например, «Инфоклиника», Medesk и др.) обладают стандартной функциональностью, включают в себя модули для учета денежных средств, лекарственных препаратов, материалов и оборудования, ведения историй болезней пациентов, осуществления смс-рассылки и работы с сайтом медицинской организации, которые целесообразно использовать для более крупных медучреждений. Для флебологической клиники необходимо создание узкоспециализированной системы для научных исследований, прямых аналогов которой найдено не было. На данный момент в клинике используется информационная система, реализованная на базе конфигурации 1С «Управление небольшой фирмой».

Система сбора флебологических данных

На начальном этапе исследования был реализован сбор материалов для обучения и тестирования нейронной сети с помощью веб-приложения. Архитектура приложения состоит из двух функциональных компонент: frontend – разработан на Angular, и backend – на C#. Преимущество данного решения заключается в том, что оно уже развернуто на сервере в ВК. Данное приложение имеет систему регистрации пользователей: администратор выдает логин и пароль врачу-флебологу, с помощью которых он получает доступ к форме для загрузки данных. На форме для загрузки данных врачу необходимо выбрать тип загружаемых данных: фотографии или МРТ-снимки ног. Если врач выбирает «Фотографии ног», далее ему необходимо выбрать клинический раздел («С»), который описывает клинический статус пациента. После выбора клинического статуса пользователю предлагается переместить файлы в область загрузки, чтобы отправить их в БД на сервер. Конечный этап загрузки данных продемонстрирован на рис. 4.

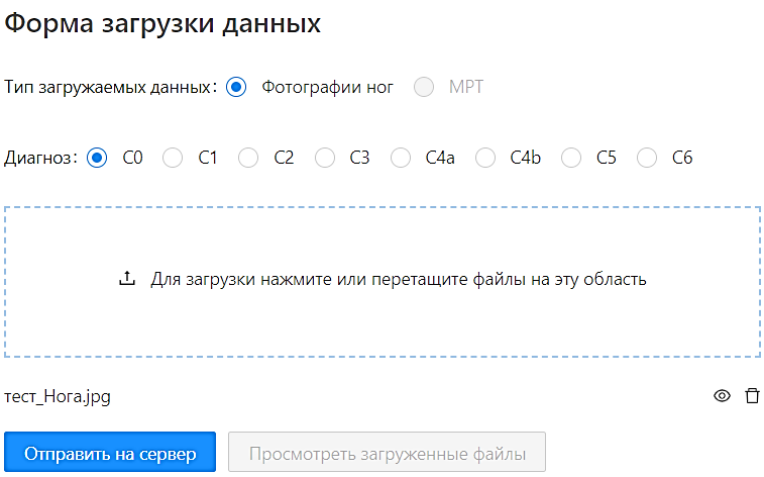

Рис. 4. Форма загрузки данных
Примечание: скриншот авторов.

Однако от данного решения пришлось отказаться по следующим причинам. Во-первых, разработка приложения с минимальным функционалом на Angular возможна при наличии множества готовых шаблонов, но написание программы с нуля, ее сопровождение и развитие проблематичны в связи со специфичностью фреймворка.

Во-вторых, в приложении нет возможности загрузить снимки МРТ – данная функция заблокирована.

В-третьих, в настоящий момент классификация CEAP изменилась (рис. 5), количество клинических подклассов «С» увеличилось, что требует внесения изменений в интерфейс системы.

C class	Description
C_0	No visible or palpable signs of venous disease
C_1	Telangiectasias or reticular veins
C_2	Varicose veins
C_{2r}	Recurrent varicose veins
C_3	Edema
C_4	Changes in skin and subcutaneous tissue secondary to CVD
C_{4a}	Pigmentation or eczema
C_{4b}	Lipodermatosclerosis or atrophie blanche
C_{4c}	Corona phlebectatica
C_5	Healed
C_6	Active venous ulcer
C_{6r}	Recurrent active venous ulcer

CVD, Chronic venous disease.
Each clinical class subcharacterized by a subscript indicating the presence (symptomatic, *s*) or absence (asymptomatic, *a*) of symptoms attributable to venous disease.

Рис. 5. Модифицированная классификация CEAP 2020 г. [5]

Инструментом для разработки приложения по сбору данных, реализованного в виде сайта, был выбран Django [6].

Неоспоримое преимущество этого фреймворка заключается в том, что регистрация пользователя предусмотрена самим Django, в отличие от Angular, в котором приходилось создавать функционал регистрации нового пользователя с нуля. Здесь frontend реализован на HTML с подключенным bootstrap, a backend – на Python.

При разработке сайта необходимо было реализовать функцию классификации фото по категориям. Врач, загружая фото, выбирает клинический класс, и в зависимости от выбранного им класса в папку проекта с наименованием, соответствующим выбранному классу,

загружается фото. Следует уточнить, что класс патологии может быть не определен, в таком случае пользователю следует сохранить фото в папку с неклассифицированными фотографиями. Если врач загружает архив с МРТ-снимками, то архив загружается в папку проекта с наименованием «МРТ».

Пользовательский сценарий выглядит так: врач-флеболог переходит на сайт, регистрируется, заходит в систему под своей учетной записью, открывает форму загрузки данных, выбирает, что ему необходимо загрузить: фото ног или МРТ-снимки. Если пользователь выбирает «Загрузить МРТ», перед ним появляется кнопка, с помощью которой он выбирает необходимый для загрузки архив и нажимает кнопку «Загрузить». Если пользователь выбирает «Загрузить фото ног», ему необходимо выбрать наименование клинического класса, необходимое фото и загрузить его. Фото будет загружено в папку с нужным названием. После каждой загрузки происходит перенаправление на страницу с формой загрузки данных, что позволяет производить те же самые действия заново. Реализация загрузки фотографий нижних конечностей пациента и МРТ-снимков в БД на сервере через сайт продемонстрирована на рис. 6–7.

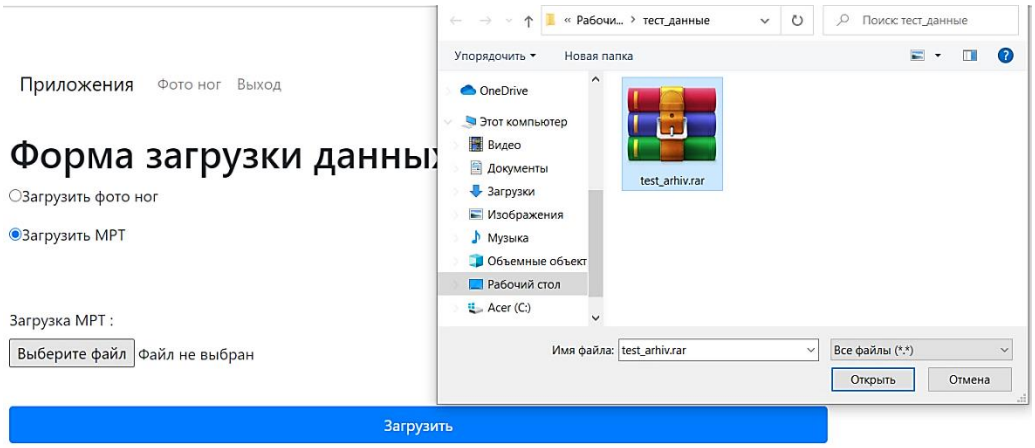

Рис. 6. Форма загрузки МРТ-снимков в БД
Примечание: скриншот авторов.

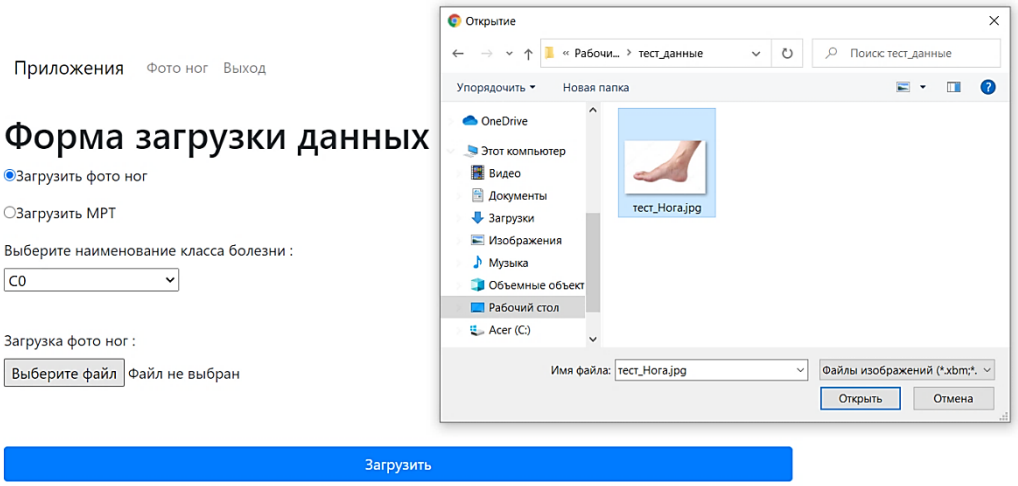

Рис. 7. Форма загрузки фото ног произвольного класса патологии «С» в БД
Примечание: скриншот авторов.

В конечном счете существуют две системы сбора флебологических данных для системы поддержки принятия решений в постановке диагноза «хроническая венозная недостаточ-

ность», построенной на базе сверточной нейронной сети. В ходе дальнейшего исследования необходимо использовать сайт, разработанный на Django, в связи с явными недостатками первой системы, а именно: устаревшаей классификацией CEAP, отсутствием функционала для загрузки МРТ-снимков и отсутствием возможности сопровождения проекта.

Проект на Django был развернут в Docker-контейнере [7], который позволяет избавиться от необходимости установки на компьютер различных сервисов. К их числу можно отнести веб-сервер, БД (в данном случае Sqlite 3) и прочие компоненты инфраструктуры приложения. Вся инфраструктура прописана в конфигурационном файле и запускается одной командой вместе с самим приложением. Более того, имеющиеся web-приложения/сайты, если их несколько, могут оказаться привязаны к разным версиям одних и тех же программ, использование же Docker-контейнеров позволит изолировать их окружения, упростить тестирование и развертывание.

Заключение

Мировой рынок коммерческого контейнерного программного обеспечения будет расти вплоть до 2023 г. в среднем на 30 % ежегодно, превысив к концу периода $1,6 млрд – такой прогноз аналитиков IHS Markit опубликован в отчете Technology Multi-Tenant Server Software Market Tracker [8].

Безусловно, виртуальные контейнеры зарекомендовали себя как актуальное для IT-индустрии решение, сокращающее время вывода продукта на рынок, а также снижающее стоимость его разработки и эксплуатации. Они обеспечивают удобство разработки, позволяя не тратить время на настройку окружения (при использовании оркестрации). «Облачный подход» в целом дает возможность сократить расходы на физическое оборудование, что особенно критично, когда речь идет об интеллектуальном анализе больших объемов графических данных в медицинских информационных системах.

Литература

1. Белоножко П. П., Белоус В. В., Куцевич Н. А., Храмов Д. А. Свободные облачные аппаратно-программные платформы. Аналитический обзор // Науковедение : интернет-журнал. 2016. Т. 8, № 6. URL: http://naukovedenie.ru/PDF/61TVN616.pdf (дата обращения: 30.03.2021).

2. Серверная виртуализация: гипервизоры против контейнеров // LAN : журнал сетевых решений. 2017. № 01–02. URL: https://www.osp.ru/lan/2017/01-02/13051363 (дата обращения: 30.03.2021).

3. Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. URL: https://kubernetes.io/ (дата обращения: 30.03.2021).

4. Зачем и как использовать контейнеры: разбираемся с Docker, Kubernetes и другими инструментами. URL: https://tproger.ru/articles/containers-explained/ (дата обращения: 30.03.2021).

5. Стадии развития варикозного расширения вен. URL: https://www.angiolife.com.ua/vse-o-varikoze/ (дата обращения: 30.03.2021).

6. Дронов А. В. Django 2.1. Практика создания веб-сайтов на Python. СПб. : БХВ-Петербург, 2019. 665 с.

7. Моуэт Э. Использование Docker. Разработка и внедрение программного обеспечения при помощи технологии контейнеров / пер. с англ ; науч. ред. А. А. Маркелов. М. : ДМК Пресс, 2017. 354 с.

8. Почему растет популярность контейнеризации. URL: https://www.cnews.ru/articles/2020-07-20/ (дата обращения: 30.03.2021).