

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.62-021.131

СЕЛЕЗНЁВ
Александр Игоревич

**ТЕХНОЛОГИИ ВИРТУАЛИЗАЦИИ В СИСТЕМАХ ОБРАБОТКИ
ДАННЫХ**

Автореферат
на соискание степени магистра
по специальности 1–45 80 01 Системы и сети инфокоммуникаций

Научный руководитель
к.т.н., доцент
ШЕВЧУК Оксана Геннадьевна

Минск 2024

ВВЕДЕНИЕ

С каждым годом объемы передаваемой информации растут, что требует более производительных систем обработки данных. Для гибкой и эффективной обработки возрастающих массивов данных используется совокупность аппаратного и программного обеспечения, одним из элементов которой являются технологии виртуализации, позволяющие разделить ресурсы физического сервера (или серверных кластеров) между многими пользователями и приложениями. С помощью данных технологий становится возможным создание высокопроизводительных, масштабируемых и гибких систем обработки данных.

Целью магистерской диссертации является проектирование гибкой системы обработки данных на основе технологии виртуализации с использованием современных архитектурных средств.

В соответствии с поставленной целью были определены следующие задачи:

- изучение систем и основных этапов обработки данных;
- изучение и анализ основных типов виртуализации;
- исследование системных архитектур, необходимых для построения систем обработки информации на основе технологий виртуализации;
- проектирование гибкой системы обработки данных.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Связь работы с крупными научными программами

Тема диссертационной работы соответствует пункту 1 приоритетных направлений научной, научно-технической и инновационной деятельности Республики Беларусь на 2021–2025 гг., утвержденных Указом Президента Республики Беларусь №156 от 7 мая 2020 г. «Цифровые информационно-коммуникационные и междисциплинарные технологии, основанные на них производства». Работа выполнялась в учреждении образования «Белорусский государственный университет информатики и радиоэлектроники».

Цель и задачи исследования

Целью диссертационной работы является проектирование гибкой системы обработки данных на основе технологии виртуализации с использованием современных архитектурных средств.

Для достижения поставленной цели в диссертации решены следующие задачи:

- 1 Построение структурной схемы обобщенной системы обработки данных.
- 2 Выбор и обоснование необходимых технологий виртуализации.
- 3 Выбор и обоснование системной архитектуры.

4 Детализация характеристик элементов выбранной технологии виртуализации.

5 Выбор необходимого программного и аппаратного обеспечения.

6 Построение уточненной структуры обработки данных.

Личный вклад соискателя ученой степени

Содержание диссертации отображает личный вклад автора. Он заключается в научном обосновании используемых методов и средств технологии виртуализации в системах обработки данных, исследовании и формировании системных архитектур, оценке эффективности композиции выбранных инструментов и программных средств, обработке и анализе полученных результатов, формулировке выводов.

Определение целей и задач исследований, интерпретация и обобщение полученных результатов проводились совместно с научным руководителем кандидатом технических наук, доцентом О.Г. Шевчук.

Апробация диссертации и информация об использовании ее результатов

Базовые положения диссертационной работы были опубликованы в международном научном журнале «Молодой ученый».

Опубликование результатов диссертации

По результатам исследований, представленных в диссертации, опубликованы 2 печатные работы, в том числе: 2 статьи в научном журнале общим объемом 1,55 авторских листа.

Структура и объем диссертации

Диссертационная работа состоит из введения, общей характеристики работы, трех разделов с выводами по каждому разделу, заключения и библиографического списка.

Общий объем диссертационной работы составляет 80 страниц, из них 65 страниц текста, 26 рисунков на 12 страницах, список использованных библиографических источников (32 наименования на 3 страницах), список публикаций автора по теме диссертации (2 наименования на 1 странице), графический материал на 6 страницах.

Проверка на уникальность

Проведена экспертиза диссертации Селезнёва Александра Игоревича «Технологии виртуализации в системах обработки данных» на корректность использования заимствованных материалов с применением сетевого ресурса «Антиплагиат» (адрес доступа: <https://antiplagiat.ru>) в on-line режиме 12.05.2024 г. В результате проверки установлена корректность использования заимствованных материалов (оригинальность диссертационной работы составляет 85 %).

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** рассмотрены проблемы необходимости гибкой и эффективной обработки возрастающих массивов данных, одним из решений которой является использование современных технологий виртуализации и оптимальной системной архитектуры при разработке гибкой системы обработки данных.

В **общей характеристике работы** показана связь работы с приоритетными направлениями научных исследований, цель и задачи исследования, личный вклад соискателя ученой степени, апробация результатов диссертации.

В **первой главе** магистерской диссертации рассматриваются системы обработки данных, технологии виртуализации и контейнеризации, системы оркестрации.

Во **второй главе** проведено исследование основных системных архитектур, с помощью которых возможно построение систем обработки данных. Особое внимание уделяется микросервисной архитектуре, так как в настоящее время она является одной из наиболее технологичных и перспективных.

В **третьей главе** диссертации сформулированы требования к проектированию гибкой системы обработки данных и поставлены задачи, необходимые для достижения этой цели. Каждая из задач представлена в виде обособленного этапа проектирования и успешно решена; совокупным результатом является разработанная модель гибкой современной системы обработки данных.

Проектируемая система обработки данных (СОД) должна удовлетворять следующим требованиям:

1 Производительность и надежность. СОД должна быть высокопроизводительной и надежной системой, способной выдерживать большие нагрузки и работать без потери данных.

2 Гибкость. Разрабатываемая система должна обладать высокой степенью интеграции и гибкостью, каждая из её частей должна быть отдельным самостоятельным блоком, допускающим, при необходимости, внесение изменений без остановки и перезагрузки всей системы.

3 Технологичность. При разработке СОД должны применяться современные технологии.

4 Безопасность. Все входные и выходные данные должны быть защищены от взлома и внешнего влияния.

Для разработки СОД, удовлетворяющей данным требованиям, необходимо решить следующие задачи:

1 Построение структурной схемы обобщенной системы обработки данных.

2 Выбор и обоснование необходимых технологий виртуализации.

3 Выбор и обоснование системной архитектуры.

4 Детализация характеристик элементов выбранной технологии виртуализации.

5 Выбор необходимого программного и аппаратного обеспечения.

6 Построение уточненной структуры обработки данных.

На начальном этапе была предложена структурная схема системы обобщенной обработки данных (см. рисунок 1).

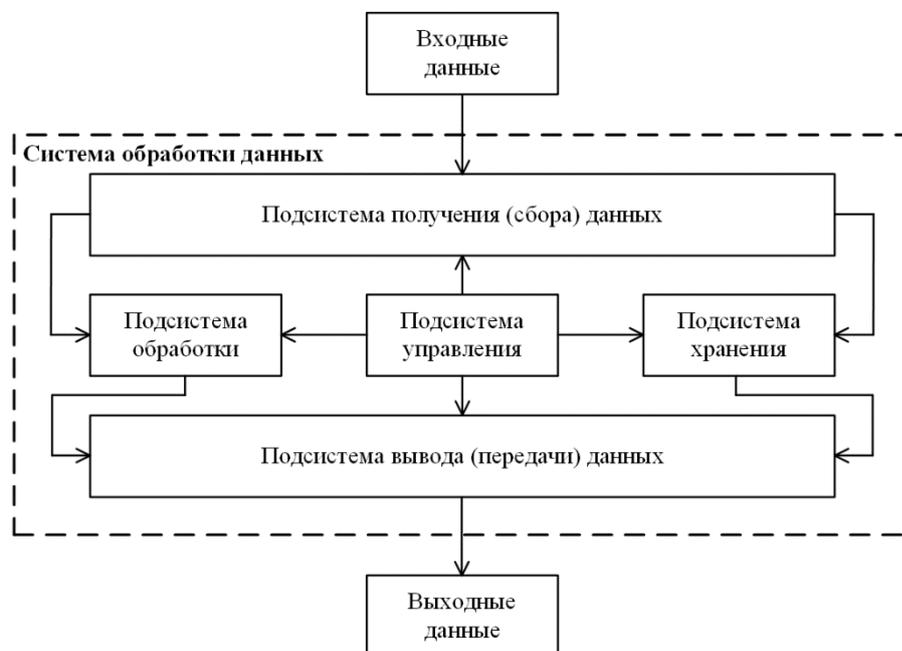


Рисунок 1 – Обобщенная структура системы обработки данных

Входные данные могут поступать от пользователя напрямую или из отдельных массивов данных. Поступающие данные могут быть различны, поэтому в блоке «Подсистема получения (сбора) данных» происходит их фильтрация и исключение неподходящих данных. «Подсистема управления» предназначена для управления всем процессом обработки в ручном/автоматическом режимах и имеет прямой доступ ко всем другим блокам. В блоке «Подсистема хранения» обеспечивается хранение необходимых данных как для последующей обработки, так и для передачи обработанных данных в блок «Подсистема вывода (передачи) данных», из которого данные передаются конечному потребителю.

Для проектирования СОД были рассмотрены две популярные технологии виртуализации – виртуальные машины и контейнеры. В результате анализа и сравнения данных технологий по требуемым критериям в качестве основной технологии виртуализации выбрана технология контейнеризации, обладающая большей гибкостью, масштабируемостью и отказоустойчивостью.

Исходя из требований к разрабатываемой системе был произведен анализ и выбор наиболее подходящей системной архитектуры. В качестве критериев сравнения архитектур используются следующие:

- структура;
- масштабируемость;
- разработка;
- развертывание.

Для проектируемой СОД выбирается микросервисная архитектура (МА), так как она отвечает поставленным требованиям в большей мере, чем монолитная архитектура.

Технология контейнеризации предполагает собой использование контейнеров. Несмотря на то, что в настоящее время одним из популярных видов контейнеров является Docker, перед выбором в качестве основного элемента технологии контейнеризации для проектирования СОД важно произвести сравнение и с другим распространённым контейнером – Linux Containers (LXC). Анализ и сравнение данных видов контейнеров был произведен с учетом ранее сформулированных требований, в результате которого были выбраны контейнеры Docker.

Для эффективного управления контейнерами требуется выбрать подходящую технологию оркестрации. С этой целью рассматриваются архитектуры популярных технологий оркестрации Kubernetes и Docker Swarm.

Архитектура Kubernetes построена на принципах распределенной системы. Основными структурными компонентами архитектуры Kubernetes являются мастер-узел, рабочие узлы и поды. Архитектура системы оркестрации Docker Swarm представляет собой систему взаимодействия swarm менеджеров (swarm node) и swarm рабочих узлов (swarm worker node).

Для проектируемой СОД выбирается технология оркестрации Kubernetes благодаря её гибкости, повышенной безопасности и ориентированности под микросервисы.

Для выбора подходящего программного и аппаратного обеспечения необходимо определиться с тем, где будет размещаться проектируемая СОД, поэтому требуется рассмотреть варианты размещения СОД, детализировать основные особенности каждого из них и произвести их сравнение. После анализа достоинств и недостатков локального и облачного размещения инфраструктуры был выбран облачный тип размещения.

В соответствии с выбранным типом размещения был произведен анализ видов облачного размещения и сравнение наиболее распространённых провайдеров для выбора оптимального варианта. Рассмотрены достоинства и недостатки облачных провайдеров AWS и Microsoft Azure. Обе рассмотренные облачные платформы схожи по многим критериям, но AWS обладает большей гибкостью и поэтому выбирается в качестве облачной платформы для проектируемой СОД.

Для проектирования оптимальной СОД на системах выбранного облачного провайдера предложена уточненная модель обработки данных, представленная на рисунке 2.

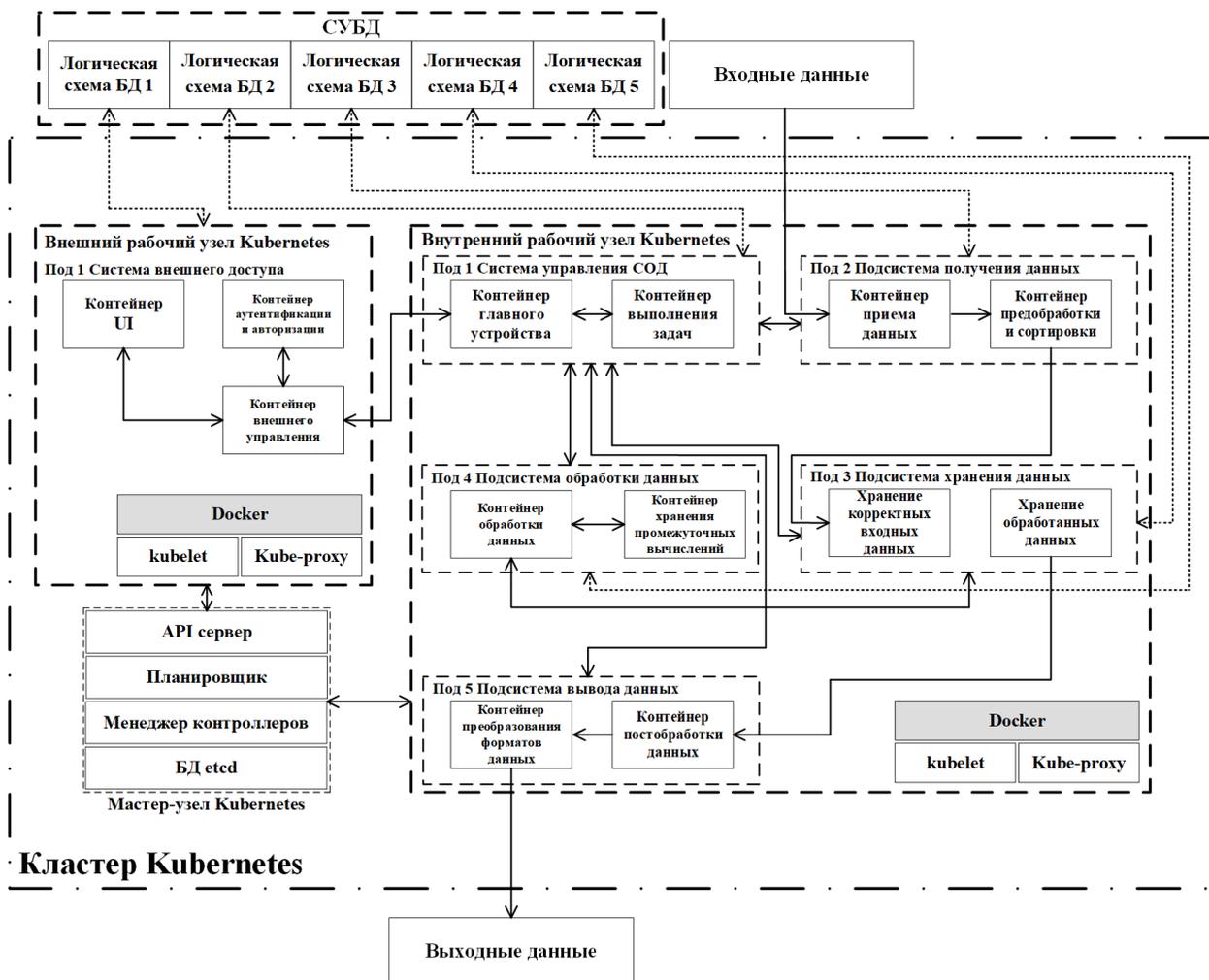


Рисунок 2 – Структурная схема уточненной модели СОД

Из рисунка 2 видно, что уточненная модель СОД представляет собой микросервисную структуру взаимосвязанных между собой контейнеров Docker. Контейнеры сгруппированы между собой в отдельные поды (в зависимости от назначения) и представлены в виде двух обособленных рабочих узлов, используя возможности оркестрации Kubernetes. Каждый из контейнеров в данном случае представляет собой отдельный микросервис – так как выполняет одну конкретную задачу и имеет простые каналы обмена данными.

Мастер-узел Kubernetes служит для администрирования и управления внешним и внутренним рабочими узлами. По умолчанию он настроен для автоматической работы и доступ к нему предоставляется с помощью «Контейнера главного устройства», расположенного во внутреннем рабочем узле.

Внешний рабочий узел представляет собой 3 взаимодействующих контейнера Docker, расположенных в поде 1, для авторизации и настройки СОД: «Контейнер UI», «Контейнер аутентификации и авторизации» и «Контейнер внешнее управление». «Контейнер UI» является микросервисом, служащим для отображения пользовательского интерфейса авторизации и настроек обработки СОД. «Контейнер аутентификации и авторизации» позволяет авторизоваться в системе для управления и настройки системы. «Контейнер внешнего управления» предназначен для корректного взаимодействия пользователя и ядра СОД. Все данные, используемые в этом узле, хранятся во внешней логической схеме базы данных (БД) 1, которая представляет собой одну из схем единой внешней системы управления базами данных (СУБД), что отвечает особенностям построения системы с использованием микросервисов.

В данной уточненной структурной схеме все отдельные логические схемы БД взаимодействуют с подами, так как иная реализация (взаимодействие непосредственно контейнеров с БД) значительно сложнее (в этом случае нужно разрабатывать отдельное ПО, в отличие от поддержки взаимодействия подов с контейнерами и БД в Kubernetes по умолчанию) и выходит за рамки уточненной модели СОД. При построении системы используется внешняя СУБД (все логические схемы БД расположены вне кластера Kubernetes) ввиду многочисленных недостатков для варианта внутри кластера Kubernetes, основным из которых является значительное увеличение сложности реализации комплексной системы оркестрации.

Внутренний рабочий узел предназначен для обработки поступающих данных и состоит из следующих подов:

1 Система управления СОД. В этом поде располагаются «Контейнер главное устройство» и «Контейнер выполнение задач». «Контейнер главное устройство» позволяет производить администрирование и управление элементами СОД, а «Контейнер выполнения задач» играет роль планировщика и менеджера задач.

2 Подсистема получения данных служит для первоначальной обработки входящих данных и состоит из «Контейнера приема данных» и «Контейнера предобработки и сортировки». «Контейнер приема данных» отвечает за корректную передачу данных в «Контейнер предобработки и сортировки», где происходит верификация входящих данных. В процессе этой операции данные временно записываются и хранятся в логической схеме БД 3; после завершения этапа предобработки и сортировки – неверные данные удаляются.

3 Подсистема хранения данных. Этот под представляет собой совокупность контейнеров «Хранение корректных исходных данных» и «Хранение обработанных данных». В контейнер «Хранение корректных исходных данных» данные поступают из «Контейнера предобработки и сортировки», которые дополнительно проверяются и записываются в постоянное хранилище логической схемы БД 4. Контейнер «Хранение

обработанных данных» служит для хранения данных после их обработки из подсистемы обработки данных.

4 Подсистема обработки данных состоит из «Контейнера обработки данных» и «Контейнера хранения промежуточных вычислений», взаимодействующих с логической схемой БД 5. В «Контейнере обработки данных» происходит операция преобразования в требуемые выходные данные, по завершению которой обработанные данные поступают в подсистему хранения данных.

5 Подсистема вывода данных содержит «Контейнер постобработки данных» и «Контейнер преобразования форматов данных». «Контейнер постобработки данных» служит для финального преобразования данных в унифицированный вид. «Контейнер преобразование форматов данных» приводит выходные данные к требуемым пользователю/принимающей системе формату и передает их на выход.

Одним из основных преимуществ использования контейнеров Docker и средства оркестрации Kubernetes в микросервисной архитектуре разрабатываемой СОД является то, что Kubernetes в данном случае полностью отвечает за балансировку нагрузки – в любой момент времени при увеличении нагрузки на какой-либо из контейнеров возможно автоматическое создание его реплик, что позволяет в полной мере раскрыть достоинства МА.

Предложенная уточненная модель СОД может быть полностью имплементирована в среде AWS, так как все ее основные технологии доступны в этой среде. Спроектированная СОД может являться базисом для построения определенной системы обработки данных с заданными параметрами и, в этом случае, существует возможность простой адаптации посредством организации отдельных БД для каждого из микросервисов, что значительно улучшает масштабируемость всей системы в целом. В случае выбора сложного и комплексного варианта обработки данных возможно выделение подсистемы обработки данных в отдельный рабочий узел.

ЗАКЛЮЧЕНИЕ

В магистерской диссертации спроектирована структура гибкой системы обработки данных на основе технологии контейнеризации с использованием микросервисной архитектуры. Для этого проведено исследование основных этапов обработки данных, технологий виртуализации и основных системных архитектур, необходимых для построения системы обработки данных. Особое внимание уделяется микросервисной архитектуре, так как в настоящее время она является наиболее технологичной и перспективной, что детально рассмотрено на примере перехода от классической монолитной архитектуры к микросервисной. Для построения гибкой и современной системы обработки данных сформулированы требования, которым должна удовлетворять проектируемая система. Разработка этой системы представляет собой совокупность задач проектирования в виде обособленных этапов.

На начальном этапе проектирования предложена и рассмотрена обобщенная структура системы обработки данных. В результате анализа технологий виртуализации выбрана технология контейнеризации. Выбор микросервисной архитектуры в качестве системной для проектирования гибкой системы обработки данных был во многом обусловлен особенностями технологии контейнеризации.

На этапе детализации элементов выбранной технологии виртуализации было проведено сравнение основных видов контейнеров – LXC и Docker. В процессе сравнения выбраны контейнеры Docker, ввиду их ориентированности на микросервисную архитектуру и простоту использования в разработке. Для управления контейнерами произведен подробный анализ основных систем оркестрации – Kubernetes и Docker Swarm. Обе технологии обладают достаточным функционалом для использования в качестве системы управления СОД и соответствуют выдвинутым при проектировании требованиям, но технология Kubernetes обладает большей гибкостью, повышенной безопасностью и изначально ориентирована на микросервисы, что обуславливает именно её выбор.

На этапе выбора необходимого программного и аппаратного обеспечения детально рассмотрены различные виды размещения инфраструктуры и их особенности для использования в качестве основного варианта проектируемой системы обработки данных. Было выбрано облачное размещение, так как именно этот вариант позволяет построить гибкую, современную и высокоэффективную систему. Но следует учитывать, что этот тип размещения характеризуется недостатком в виде возможных проблем с безопасностью и конфиденциальностью данных, хотя, с другой стороны, материальные вложения в него кратно меньше других вариантов. На основании данного выбора произведен анализ и сравнение двух основных мировых облачных провайдеров – AWS и Microsoft Azure. Выбрана платформа AWS за счёт большей гибкости, хотя при значительном расширении проектируемой системы обработки данных можно использовать и Microsoft Azure, ввиду её более удобной адаптируемости для крупных корпораций.

Особенностью спроектированной системы обработки данных является использование современных технологий виртуализации на производительной и технологичной платформе на основе микросервисов, что обеспечивает высокую надежность и безопасность.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1–А Селезнёв, А. И. Контейнеризация в системах обработки данных / А. И. Селезнёв, И. Л. Селезнёв // Молодой учёный. – 2023. – № 43 (490). – С. 7–11.

2–А Селезнёв, А. И. Актуальность применения микросервисной архитектуры в системах обработки данных / А. И. Селезнёв, И. Л. Селезнёв // Молодой ученый. – 2023. – № 48 (495). – С. 22-32.