

Речь

(с1) Уважаемые присутствующие, вашему вниманию представляется дипломная работа на тему: «Разработка горизонтально масштабируемой системы легковесных виртуальных машин».

(с2) Современный мир характеризуется высокой динамичностью, большими объёмами обрабатываемых данных, поэтому от IT инфраструктур компаний требуется возможность быстрого реагирования на изменившиеся требования бизнеса. За последние десятилетия наиболее распространённым подходом к решению этой задачи стало использование облачных вычислений, они предоставляют множество плюсов:

- 1) возможность быстро нарастить вычислительные ресурсы;
- 2) низкая вероятность технических сбоев;
- 3) сокращение капитальных и оперативных затрат бизнеса.

(с3) Для возможности эффективного использования облачных вычислений информационные системы должны обладать возможностью горизонтального масштабирования. Применяемые ранее монолитные архитектуры довольно сложно эффективно масштабировать, поэтому в настоящее время принято разрабатывать системы на основе множества микросервисов. Микросервис - это минимальная единица функциональности системы, слабо связанная с другими компонентами и обрабатывает однотипные запросы, обычно передаваемые по протоколу http. Микросервисы исполняются в виртуальной среде, как правило, для их запуска используются традиционные операционные системы, такие как: Windows, unix-подобные, FreeBSD и другие. Данные операционные системы, однако, не обладают архитектурой, ориентированной на такой режим исполнения и иногда могут накладывать дополнительные накладные расходы на вычислительные ресурсы, превосходящие затраты самого микросервиса. В противовес традиционным операционным системам существует новый подход под названием **Unikernel** идеально подходящий для запуска горизонтально масштабируемых микросервисов. **Unikernel** система ориентирована на выполнение единственной задачи, поэтому лишены чрезмерных накладных расходов.

(с4) В рамках данной работы были поставленные следующие цели:

- 1) Проектирование системы горизонтального масштабирования виртуальных машин и диспетчеризации запросов к исполняемым на них микросервисам;
- 2) Реализация компонентов системы в соответствии с разработанной моделью взаимодействия.

(с5) В данной работе была использована IncludeOS - это Unikernel система, представленная в 2016 году. Данная операционная система обладает следующими характеристиками:

- 1) наличием единого плоского образа операционной системы, скомпонованной вместе с прикладной задачей;
- 2) минимальным требованием к ресурсам хоста;
- 3) скорость старта вплоть до 200 миллисекунд;
- 4) написана на языке программирования C++ и исполняет прикладную задачу на нулевом кольце защиты;
- 5) ориентирована на работу с Web-сервисами.

(с6) Общая схема системы представлена на слайде №6. В неё входят сл. компоненты:

- 1) клиентский модуль;
- 2) сервер диспетчеризации запросов;
- 3) база данных с состоянием системы;
- 4) контроллер виртуальных машин;
- 5) агент, дополняющий виртуальную машину;
- 6) консоль администратора.

Рассмотрим теперь компоненты системы подробнее.

(с7) Бизнес-логика приложения взаимодействуют с двумя компонентами системы: с клиентским модулем и интерфейсом web-сервисов на уровне виртуальной машины. Клиентский модуль позволяет абстрагироваться от топологии инфраструктуры среды исполнения, его задача состоит в передаче формируемых на клиентской стороне запросов на транспортный уровень. Микросервис абстрагирован от данных клиентской стороны и маршрута прохождения запроса.

(с8) Сервер диспетчеризации реализуют диспетчеризацию запросов web-сервисов и горизонтальное масштабирование виртуальных машин. В зависимости от текущего состояния системы может быть принято решение о запуске дополнительных виртуальных машин и об остановке запущенных. Входящий запрос на обработку передаётся одной из виртуальных машин, где обрабатывается Web сервисом на уровне прикладной бизнес-логики. Текущие сведения о состоянии системы можно получить в консоли администратора.

(с9) На данном слайде приведена диаграмма процесса обработки входных запросов. При наличии простаивающих виртуальных машин, запрос на обработку отправляется одной из них. После получения ответа от выбранной виртуальной машины, данные пересылаются клиентскому модулю. Исходя из текущей загруженности системы может быть принято решение о запуске еще одного экземпляра виртуальной машины. При отсутствии свободных слотов виртуальной машины, выбирается для обработки запроса наименее загруженная.

(с10) Для предоставления обратной связи, каждая виртуальная машина с заданной частотой отправляет сведения о своём состоянии серверу диспетчеризации. При получении сведений о состоянии виртуальной машины, сервер либо регистрирует в базе данных новую виртуальную машину, либо обновляет состояние уже имеющейся. Исходя из тренда загруженности виртуальной машины, при падении нагрузки меньше порогового значения сервер диспетчеризации принимает решение об её остановки и отправляет соответствующую команду.

(с11) Для хранения и обработки состояний системы используется реляционная база данных, она включает в себя:

- 1) реестр ip-адресов, зарезервированных для использования виртуальными машинами;
- 2) пул исполняемых в текущий момент виртуальных машин;
- 3) набор параметров, позволяющих настраивать поведения системы.

(с12) Элементами научной новизны в данной работе является то, что:

- 1) Впервые создана модель горизонтально масштабируемой распределенной обработки запросов на основе легковесных виртуальных машин;
- 2) Создана работоспособная реализация предположенной системы на примере виртуальных машин с ядром IncludeOS.

(с13) Вся система тестировалась на демо-сервисе, который производил подсчёт количества простых чисел, вплоть до указанного числа. Устанавливался временной промежуток в 20 секунд и подсчитывалось количество обработанных запросов на различном количестве виртуальных машин. Все запросы по сложности вычисления были примерно одинаковые.

(с14) Работа всех компонентов системы при обработки запросов представлена на слайде №14.

(с15) Объём представленной системы приблизительно 3800 строк кода. При реализации данной системы использовалось три языка программирования, три операционные системы и более десятка различных библиотек и программных продуктов сторонних разработчиков.

(с16) В рамках данной работы были достигнуты следующие цели:

- 1) Спроектирована и реализована система горизонтального масштабирования виртуальных машин и диспетчеризации запросов к исполняемым на них микросервисам;
- 2) Реализованы компоненты системы в соответствии с разработанной моделью взаимодействия.
- 3) Результаты работы были продемонстрированы в 14-ой всероссийской научно-технической конференции «Наука и молодёжь — 2017» в секции «информационные технологии»

Доклад окончен, спасибо за внимание.