

УДК 004.75

ИССЛЕДОВАНИЕ KUBERNETES ОПЕРАТОРОВ В ЗАДАЧАХ АВТОМАТИЧЕСКОГО РАЗВЕРТЫВАНИЯ MEC ПРИЛОЖЕНИЙ

Борисов Н.А. (ИТМО)

Научный руководитель – инженер Филянин И.В.

(ИТМО)

Введение. С учетом роста потребностей в обработке данных в реальном времени, а также с увеличением числа устройств Internet of Things (IoT), существует необходимость в эффективных решениях, которые бы обеспечивали высокую производительность, низкую задержку и надежную передачу данных.

Для решения этой проблемы ETSI (European Telecommunications Standards Institute) разработал архитектурную модель, которая позволяет размещать вычислительные ресурсы и приложения на краю сети, ближе к конечным пользователям или устройствам. MEC представляет собой перспективное направление для оптимизации процессов общения между клиентской и серверной частью приложения.

Основная часть. MEC (Multi-Access Edge Computing) — это архитектурная концепция в области сетей связи, которая представляет собой вычислительную платформу, размещенную на краю сети, ближе к конечным пользователям. Основная идея MEC заключается в том, что обработка данных и вычисления могут выполняться непосредственно на уровне края сети, что позволяет уменьшить задержки и увеличить производительность для приложений и сервисов, работающих в мобильных сетях [1]. В рамках MEC архитектуры вычислительные ресурсы и сервисы должны быть размещены вблизи к конечным пользователям, в ближайших узлах сети, таких как базовые станции или вычислительные устройства на краю сети [2].

Существует 3 уровня инфраструктуры и ресурсов, которые используются для предоставления услуг и приложений MEC:

User Equipment Level (Уровень пользовательского оборудования): это самые близко расположенные к конечным пользователям устройства. На этом уровне находятся мобильные устройства, такие как смартфоны, IoT устройства, датчики и другие источники запросов, которые могут взаимодействовать с MEC сервисами и приложениями.

Edge Level (Уровень края сети): здесь расположены MEC серверы или устройства, которые предоставляют вычислительные и сетевые ресурсы для обработки и предоставления MEC приложений и сервисов на краю сети.

Remote Level (Удаленный уровень): на этом уровне могут располагаться централизованные службы управления, хранения данных или другие общие ресурсы, которые поддерживают работу MEC-сервисов.

Структура MEC:

MEC host — это объект, содержащий платформу MEC и инфраструктуру виртуализации, которая предоставляет вычислительные ресурсы, хранилище и сетевые ресурсы для запуска приложений MEC.

MEC platform представляет собой набор основных функциональных возможностей, необходимых для запуска приложений MEC на конкретной платформе виртуализации и предоставления услуг MEC.

MEC app создаются в инфраструктуре виртуализации узла MEC на основе конфигурации или запросов к приложению.

Operation support system — OSS отвечает за предоставление доступа к пользовательским запросам, пересылаемым с пользовательского оборудования (UE) через прокси-сервер (UALCMP).

User Application LifeCycle Management Proxy (UALCMP) — Прокси-сервер управления жизненным циклом пользовательского приложения предоставляет приложению устройства доступ к API MEC.

MEC Orchestrator — это платформа, которая обеспечивает развертывание, управление и оркестровку приложений на границе сети.

MEC Platform Manager (MEPM) отвечает за управление и управляемость платформы MEC. Это компонент, который управляет развертыванием, масштабированием и управлением жизненным циклом MEC-приложений на MEC-платформе.

Kubernetes операторы — это программные контроллеры, расширяющие функциональность Kubernetes, позволяя автоматизировать задачи развертывания и управления приложениями. Они представляют собой код, который понимает особенности вашего приложения и способен автоматически управлять жизненным циклом приложения. Оператор представляет собой pod-демон, который отслеживает API Kubernetes на предмет изменений, связанных с конкретным ресурсом (CR). При обнаружении изменения оператор принимает меры для обеспечения соответствия желаемому состоянию ресурса.

Custom Resource Definition (CRD) — это расширение API Kubernetes, позволяющее определять собственные типы ресурсов. Определение объекта CRD создает новый пользовательский ресурс с указанным вами именем.

Custom Resource (CR) — Кастомные ресурсы предоставляют возможность определить собственные типы ресурсов, которые не входят в стандартные объекты Kubernetes.

Использование Kubernetes операторов в контексте развертывания MEC приложений открывает новые возможности для эффективного управления инфраструктурой. Они обеспечивают единый и унифицированный способ управления MEC приложениями, а также позволяют более гибко масштабировать и обновлять MEC приложения в распределенной среде. Если в экосистеме Kubernetes нет оператора, который реализует желаемое поведение, можно написать свой собственный кастомный контроллер. Оператор можно реализовать, используя любой язык программирования, который может выступать в качестве клиента для Kubernetes API.

Выводы. Проведённый анализ позволил изучить основные концепции и принципы MEC, а также разобраться с методами применения Kubernetes операторов для развёртывания MEC платформы. Перед разработкой своего собственного оператора мне необходимо определиться со списком разворачиваемых компонентов будущей платформы. Набор приложений я получу от научного руководителя и других студентов, которые тоже принимают участие в разработке платформы.

Список использованных источников:

1. Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher and Valerie Young Mobile Edge Computing A key technology towards 5G: white paper. Valbonne: ETSI, 2015. 16 с.
2. Dario Sabella, Vadim Sukhomlinov, Linh Trang, Sami Kekki, Pietro Paglierani, Ralf Rossbach, Xinhui Li, Yonggang Fang, Dan Druta, Fabio Giust, Luca Cominardi, Walter Featherstone, Bob Pike, Shlomi Hadad Developing Software for Multi-Access Edge Computing: white paper. Valbonne: ETSI, 2019. 38 с.
3. Michael Dame The Kubernetes Operator Framework Book: . Birmingham: Packt Publishing Ltd., 2022. 306 с.