

Технические науки

УДК 004.75

РЕАЛИЗАЦИЯ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ ПЕРИФЕРИЙНОГО СЕРВИСА

Викторов Александр Сергеевич

аспирант кафедры информатики и вычислительной техники
Костромской государственной университет
Россия, г. Кострома

Аннотация: В статье рассматривается реализация распределенной вычислительной сети периферийного сервиса, дано описание архитектуры и механизмов, обеспечивающих основной функционал вычислительной сети и реализованные с использованием программных компонентов фреймворка Spring. Рассматриваемая вычислительная сеть представлена RESTful web-сервером, выполняющим функции шлюза приложений, и группой серверов приложений, автоматически организованных в иерархическую древовидную структуру.

Ключевые слова: беспилотный летательный аппарат БПЛА, периферийные вычисления, веб-сервер, Spring MVC, Internet of Drones.

DISTRIBUTED COMPUTING NETWORK OF THE EDGE SERVICE IMPLEMENTATION

Victorov Alexander Sergeevich

postgraduate, Department of Informatics and Computer Engineering of the
Kostroma State University
Russia, Kostroma

Abstract: The paper deals with distributed computing network of the edge service implementation, and also gives a description of the architecture and software components which has been implemented by using Spring framework for providing a based edge service functionality. Considered computing network implementation consist from web-server which providing application gateway functionality and application servers group which automatically form into hierarchical tree structure.

Keywords: unmanned aerial vehicle UAV, edge computing, web server, Spring MVC, Internet of Drones.

В данной статье рассматривается реализация распределенной вычислительной сети периферийного сервиса, предназначенной для распределенной обработки данных телеметрии, получаемых с группировки беспилотных летательных аппаратов БПЛА, и построенной с учетом концепции интернета беспилотных летательных аппаратов IoD (Internet of Drones) [1, 2]. Структура распределенной вычислительной сети представлена на рис. 1. В состав периферийного сервиса обработки данных телеметрии входят RESTful web-сервер, который предназначен для реализации механизма доступа к функционалу периферийного сервиса, и также несколько серверов распределенных приложений, из

которых сформирован вычислительный кластер периферийного сервиса. Каждый сервер распределенных приложений одновременно выполняет функции сервера баз данных, связанные с хранением данных, генерируемых во время процесса обработки.

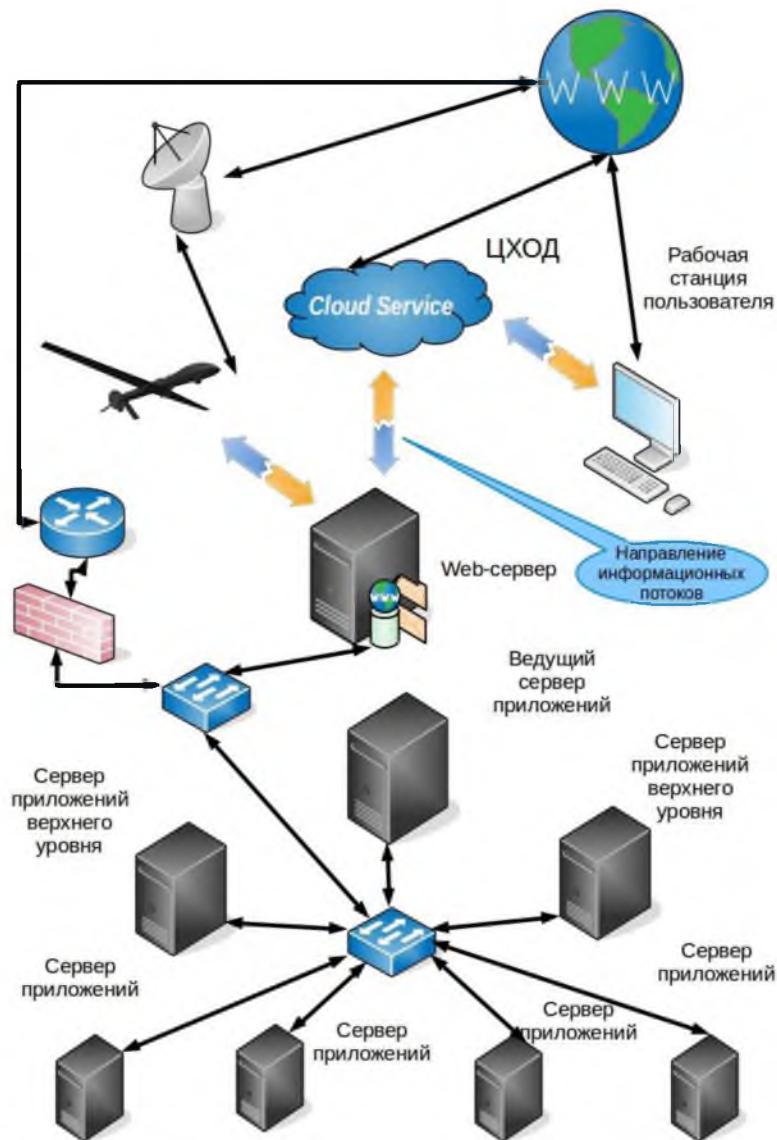


Рисунок 1 - Структура распределенной вычислительной сети

Для реализации web-сервера периферийного сервиса использовался фреймворк Spring MVC [3], а сама реализация выполнена с использованием архитектурного подхода REST (Representational State Transfer), в котором манипуляция ресурсами осуществляется через их представление. В основе фреймворка Spring MVC лежит реализация паттерна проектирования MVC (Model-view-controller). Все входящие HTTP запросы обрабатываются процессом DispatcherServlet. При получении HTTP запроса DispatcherServlet, используя интерфейс HandlerMapping [4], определяет контроллер ресурсов, на который перенаправляется запрос и который производит его обработку. Для объявления компонента контроллер ресурсов используется аннотация @RestController, а для сопоставления определенного HTTP запроса с обрабатывающим его методом, объявленным внутри контроллера, аннотация @RequestMapping. Основной функционал web-сервера связан с реализацией для периферийного сервиса механизма аутентификации

и авторизации [1] предварительно зарегистрированного заранее устройства, а также механизма инициализации и контроля WebSocket [5, 6] соединения, используемого для обмена данными телеметрии между сервером приложений, на котором будет производиться обработка данных телеметрии, и подключающимся устройством.

Каждое новое подключаемое устройство перед подключением к периферийному сервису проходит процедуру регистрации на сервере регистрации ЦХОД (центр хранения и обработки данных) для внесения информации об устройстве и владельце устройства в базу данных, после чего сервер регистрации отправляет пакет конфигурации для настройки параметров подключения к ЦХОД [1]. Дальнейшее подключение и доступ к сервисам вычислительной сети для БПЛА осуществляется посредством периферийного сервиса. Связь периферийного сервиса с ЦХОД осуществляется посредством web-сервера периферийного сервиса, который выполняет функции, связанные с синхронизацией потоков данных и процессов обработки между серверами приложений периферийного сервиса и ЦХОД. В частности после регистрации нового БПЛА при осуществлении первичного подключения БПЛА к вычислительной сети производит репликацию учетных данных нового устройства в локальную базу учетных данных периферийного сервиса. При подключении БПЛА к периферийному сервису производятся процедуры аутентификации и авторизации, после успешного завершения которых, генерируется маркер доступа (access token), который используется для получения доступа к сервису распределенной обработки данных телеметрии. При генерации маркера доступа учитывается тип запрашиваемой устройством обработки и права доступа, с которыми сопоставлена назначенная устройству роль. Информация о сгенерированном маркере сохраняется в базе данных web-сервера локально. Маркер доступа обновляется каждый раз, когда происходит смена базовой станции или истекает срок действия маркера [1].

Основное предназначение периферийного сервера заключается в оптимизации и ускорении процессов обработки данных телеметрии при помощи частичной обработки собираемых данных на вычислительных мощностях, расположенных в непосредственной близости к источникам данных [2]. Сервера распределенных приложений сгруппированы в иерархическую древовидную структуру, причем уровень иерархии, который занимает сервер, определяется индексом производительности сервера, который вычисляется автоматически при инициализации программного обеспечения периферийного сервиса путем проведения ряда заранее сформированных тестов. Конфигурация вычислительной сети, формируемая автоматически во время инициализации вычислительных узлов, храниться на web-сервере. При инициализации новый вычислительный узел регистрируется на web-сервере, на основании вычисленного индекса производительности производится динамическое изменение конфигурации вычислительной сети в режиме реального времени. При изменении конфигурации вычислительной сети уже зарегистрированные узлы оповещаются об изменении конфигурации посредством механизма, основанного на технологии XMLRPC. Для реализации механизма взаимодействия между серверами, находящимися на разных уровнях иерархии используется модель взаимодействия ведущий – ведомый, при этом сервер, находящийся на верхнем уровне иерархии, отвечает за балансировку вычислительной нагрузки на узлах, находящихся на нижнем уровне иерархии. После генерации маркера доступа web-сервер отправляет запрос на выбор сервера приложений, который будет производить обработку, к ведущему серверу приложений, который запускает процедуру балансировки вычислительной нагрузки. Данная процедура заключается в поиске вычислительного узла, который будет производить обработку данных, путем обхода иерархической древовидной структуры, сформированной узлами вычислительной сети. При получении вычислительного задания сервер, находящийся на верхнем уровне иерархии, на основании значения коэффициента резервирования и собственной остаточной доступной вычислительной мощности, вычисляемой на основе коэффициента загрузки ЦП, принимает

решение выполнить вычислительную задачу самостоятельно или распределить вычислительную задачу на сервера, находящиеся на нижнем уровне иерархии. При распределении вычислительной задачи на нижний уровень иерархии используется алгоритм балансировки нагрузки, описанный в [2]. Во время процесса обработки данных телеметрии также возникает необходимость хранения данных используемых в вычислительном цикле, при этом используется техника репликации, особенностью которой является то, что распространение изменения состояния реплик данных производится от серверов находящихся на верхнем уровне иерархии к серверам на нижнем уровне иерархии.

После выбора сервера приложений, который будет производить обработку, выбранный сервер отправляет запрос к web-серверу на инициализацию WebSocket соединения. После инициализации WebSocket соединения через него запускается передача данных телеметрии на обработку от БПЛА к выбранному серверу. Протокол WebSocket является протоколом полнодуплексной связи, реализованный поверх протокола TCP и предназначен для двухстороннего обмена сообщениями между клиентом и сервером в режиме реального времени. Поверх протокола WebSocket [5] используется протокол обмена текстовыми сообщениями STOMP, который служит для представления информации на прикладном уровне и позволяет взаимодействовать различным брокерам сообщений, поддерживающим STOMP формат. В протоколе STOMP информация передается кадрами, каждый из которых содержит команду, необязательные заголовки в формате ключ-значение и тело сообщения. Входящие сообщения обрабатываются на периферийном сервере приложением, которое производит проверку прав доступа к вычислительным ресурсам перед запуском обработки входящего сообщения и идентификацию, обрабатывающего приложения. Проверка прав доступа на обработку на вычислительном кластере периферийного сервиса осуществляется при помощи Json токена.

Список литературы:

1. Викторов А. С. Система аутентификации для интернета беспилотных летательных аппаратов // июльские научные чтения Сборник научных трудов по материалам III и II Международных научно-практических конференций. 2017. Смоленск: Общество с ограниченной ответственностью "НОВАЛЕНКО", 2017. С. 65-71.
2. Викторов А.С. Алгоритм балансировки вычислительной нагрузки для периферийной вычислительной системы распределенной обработки данных телеметрии // Достижения естественных и технических наук в XXI веке. Белгород: ООО Агентство перспективных научных исследований (АПНИ), 2017. С. 144-150.
3. Walls C. Spring in Action, Fourth Edition. 4 изд. New York: Manning Publications Company, 2014. 624 с.
4. Deinum M., Rubio D., Long J. Spring 5 Recipes: A Problem-Solution Approach. 4 изд. New York: Apress, 2017. 831 с.
5. Gutierrez F. Spring Boot Messaging: Messaging APIs for Enterprise and Integration Solutions. 4 изд. New York: Apress, 2017. 196 с.
6. Хабаров С.П., Красовский И.А., Киев А.Х. Удаленное управление на базе технологии Websocket // Информационные системы и технологии: теория и практика. Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет им. С.М. Кирова, 2017. С. 120-126.