

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

Аннотация к дипломной работе

**«Разработка и развёртывание серверной части
многопользовательской игры на микросервисной архитектуре»**

Михайлов Антон Артурович

Научный руководитель — ст. преподаватель кафедры
технологий программирования Карпович Н. А.

Минск, 2021

Реферат

Дипломная работа, 76 с., 52 рис., 5 формул, 30 источников, 15 приложений.

МИКРОСЕРВИСЫ, GOLANG, KUBERNETES, DOCKER, gRPC, МНОГОПОЛЬЗОВАТЕЛЬСКАЯ ИГРА, AGONES, OPENMATCH, GORM, ОБНАРУЖЕНИЕ СТОЛКНОВЕНИЙ.

Объект исследования — объектом исследования являются разработка и развёртывание серверной части многопользовательской игры, платформы Agones и OpenMatch, контейнеризация при помощи Docker и развёртывание микросервисных приложений на Kubernetes.

Цель работы — разработать серверную часть игры проекта Medieval.IO на микросервисной архитектуре, используя технологии, изученные в ходе прохождения производственной и преддипломной практик, и развёрнуть её на облачном сервисе для интеграции с мобильным и настольным клиентами.

В силу нынешней пандемии, вызванной COVID-19, люди стараются соблюдать социальную дистанцию и избегать больших скоплений. Однако необходимость в социализации и общении никуда не исчезла. На замену реальному взаимодействию пришло виртуальное. Ночные клубы заменили виртуальные комнаты в игре VR Chat, а игры в футбол с друзьями – виртуальные поединки в FIFA.

Однако у многих современных многопользовательских и онлайн игр достаточно часто встречаются одни и те же проблемы: проблемы с реагированием на возрастающий поток игроков, приводящий к нехватке или количеству доступных серверов, или их мощностей, и продолжительные обслуживания при обновлении версии игр или исправлении каких-то проблем и дефектов, при которых игроки не имеют доступа к игре.

Хоть данные проблемы и могут быть вызваны ошибками в логике сервера или в невозможности выделить новые сервера в силу отсутствия дополнительных аппаратных ресурсов, следует попробовать исключить

любые другие возможные проблемы, которые могут произойти. Одной из таких проблем является создание достаточного количества серверов в зависимости от нагрузки на игру и количества пользователей, находящихся в игре одновременно.

В данной работе был спроектирован и разработан на языке Golang динамически масштабируемый сервер игры для проекта Medieval.IO, используя такие технологии как Docker, Kubernetes, Agones и OpenMatch. Развёрнут данный сервер был на облачной платформе Google Cloud Platform.

Использование архитектуры, логики и ресурсов, разработанных для данного сервера, с замещением логики и механик самой игры даёт возможность создавать динамически масштабируемые сервера для сессионных игр.

Рэферат

Дыпломная праца, 76 ст., 52 мал., 5 формул, 30 крыніц, 15 дадаткаў.

МІКРАСЭРВІСЫ, GOLANG, KUBERNETES, DOCKER, gRPC, ШМАТКАРЫСТАЛЬНІЦКАЯ ГУЛЬНЯ, AGONES, OPENMATCH, GORM, ВЫЯЎЛЕННЕ СУТЫКНЕННЯЎ.

Аб'ект даследавання — аб'ектам даследавання з'яўляюцца распрацоўка і разгортванне сервернай часткі шматкарыстальніцкай гульні, платформы Agones і OpenMatch, кантэйнерызация пры дапамозе Docker і разгортванне мікрасэрвісных прыкладанняў на Kubernetes.

Мэта працы — распрацаваць серверную частку гульні праекта Medieval.IO на мікрасэрвіснай архітэктурцы, выкарыстоўваючы тэхналогіі, вывучаныя ў ходзе праходжання вытворчай і пераддыпломнай практык, і разгарнуць яе на воблачным сэрвісе для інтэграцыі з мабільным і настольным кліентамі.

У сілу цяперашняй пандэміі, выкліканай COVID-19, людзі імкнуцца выконваць сацыяльную дыстанцыю і пазбягаць вялікіх навал. Аднак неабходнасць у сацыялізацыі і зносінах нікуды не знікла. На замену рэальнаму ўзаемадзеянню прыйшло віртуальнае. Начныя клубы замянілі віртуальныя пакоі ў гульні VR Chat, а гульні ў футбол з сябрамі – віртуальныя паядынкі ў FIFA.

Аднак у многіх сучасных шматкарыстальніцкіх і анлайн гульнях досыць часта сустракаюцца адны і тыя ж праблемы: праблемы з рэагаваннем на нарастальны струмень гульцоў, які прыводзіць да недахопу або колькасці даступных сервераў, або іх магутнасцяў, і працяглая абслугоўвання пры абнаўленні версіі гульняў ці выпраўленні нейкіх праблем і дэфектаў, пры якіх гульцы не маюць доступу да гульні.

Хоць гэтыя праблемы і могуць быць выкліканыя памылкамі ў логіцы сервера або ў немагчымасці вылучыць новыя сервера ў сілу адсутнасці дадатковых апаратных рэсурсаў, варта паспрабаваць

выключыць любыя іншыя магчымыя праблемы, якія могуць адбыцца. Адной з такіх праблем з'яўляецца стварэнне дастатковай колькасці сервераў у залежнасці ад нагрузкі на гульні і колькасці карыстальнікаў, якія знаходзяцца ў гульні адначасова.

У дадзенай працы быў спраектаваны і распрацаваны на мове Golang дынамічна маштабуемы сервер гульні для праекта Medieval.IO, выкарыстоўваючы такія тэхналогіі як Docker, Kubernetes, Agones і OpenMatch. Разгорнуты дадзены сервер быў на воблачнай платформе Google Cloud Platform.

Выкарыстанне архітэктury, логіцы і рэсурсаў, распрацаваных для дадзенага сервера, з замяшчэннем логікі і механік самой гульні дае магчымасць ствараць дынамічна маштабуемыя серверы для сесійных гульняў.

Abstract

Diploma, 76 p., 52 illustrations, 5 formulas, 30 sources, 15 appendixes.

MICROSERVICES, GOLANG, KUBERNETES, DOCKER, gRPC, MULTIPLAYER, AGONES, OPENMATCH, GORM, COLLISION DETECTION.

The object of study — the development and deployment of the server side of a multiplayer game, the Agones and OpenMatch platforms, containerization using Docker and the deployment of microservice applications on Kubernetes.

The purpose of the work — to develop the server part of the game of the Medieval.IO project on a microservice architecture, using the technologies studied during the production and pre-graduation practices, and deploy it on a cloud service for integration with mobile and desktop clients.

Due to the current pandemic caused by COVID-19, people are trying to keep a social distance and avoid large gatherings. However, the need for socialisation and communication has not disappeared. Real interaction has been replaced by virtual interaction. Nightclubs have been replaced by virtual rooms in VR Chat, and football games with friends have been replaced by virtual duels in FIFA.

However, many of today's multiplayer and online games share many of the same problems: problems responding to an increase in player traffic, resulting in a lack of either the number or capacity of available servers, and lengthy maintenance issues with game updates or bug fixes that prevent players from accessing the game.

While these problems can be caused by faulty server logic or the inability to allocate new servers due to a lack of extra hardware resources, an attempt to rule out any other possible problems that might occur should be undertaken. One such problem is creating enough servers depending on the load of the game and the number of users in the game at the same time.

In this paper, a dynamically scalable game server for Medieval.IO project was designed and developed in Golang language using such technologies as Docker, Kubernetes, Agones and OpenMatch. This server was deployed on Google Cloud Platform.

Using architecture, logic and resources developed for this server and replacing logic and mechanics of the game itself makes it possible to create dynamically scalable game servers for session games.