РАЗРАБОТКА И РАЗВЁРТЫВАНИЕ СЕРВЕРНОЙ ЧАСТИ МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ИГРЫ НА МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ

А. А. Михайлов

Белорусский государственный университет, г. Минск;
anthony.mikhaylov@gmail.com
науч. рук. – Н.А. Карпович, ст. преп.

В работе рассматривается проблема разработки серверной части игр, способных адаптироваться к резким изменениям в нагрузке. Для решения данной проблемы предлагается использовать микросервисную архитектуру в силу одного из её преимуществ – возможности динамического масштабирования отдельных компонент. Для этого каждая из компонент контейнеризируется и впоследствии оркестрируется платформами Kubernetes и Agones. Для создания динамически масштабируемого организатора игр используется платформа OpenMatch. В ходе работы была спроектирована архитектура серверной части проекта Medieval.IO. Все компоненты разработаны на языке программирования Golang. Для интеграции с клиентом компоненты были развёрнуты на облачной платформе Google Cloud Platform.

Ключевые слова: микросервис, Golang, Kubernetes, Docker, Agones, OpenMatch.

ВВЕДЕНИЕ

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 количества доступных серверов, или их мощностей и продолжительные обслуживания при обновлении версии игр или исправлении каких-то проблем и багов, при которых игроки не имеют доступа к игре. Особенно критичной данная проблема стала во время пандемии COVID-19, которая привела к увеличению игроков в многопользовательских играх [1]. В данной работе рассматривается один из возможных вариантов решения вышеуказанной проблемы.

АРХИТЕКТУРА СЕРВЕРНОЙ ЧАСТИ

Пример архитектуры серверной части игры на микросервисной архитектуре можно увидеть на рис.1. Компонента UsersService отвечает за хранение данных пользователей и предметов внутриигрового магазина. Компонента PaymentService занимается проведением транзакций обмена фиатных денег на внутриигровую валюту. Компонента Matchmaker отвечает за организацию сессий игр для клиентов, желающих принять участие

в игре. Компонента GameService занимается вычислениями состояний активных сессий игр на основании сообщений, поступающих от клиентов, и их последующей рассылкой клиентам.

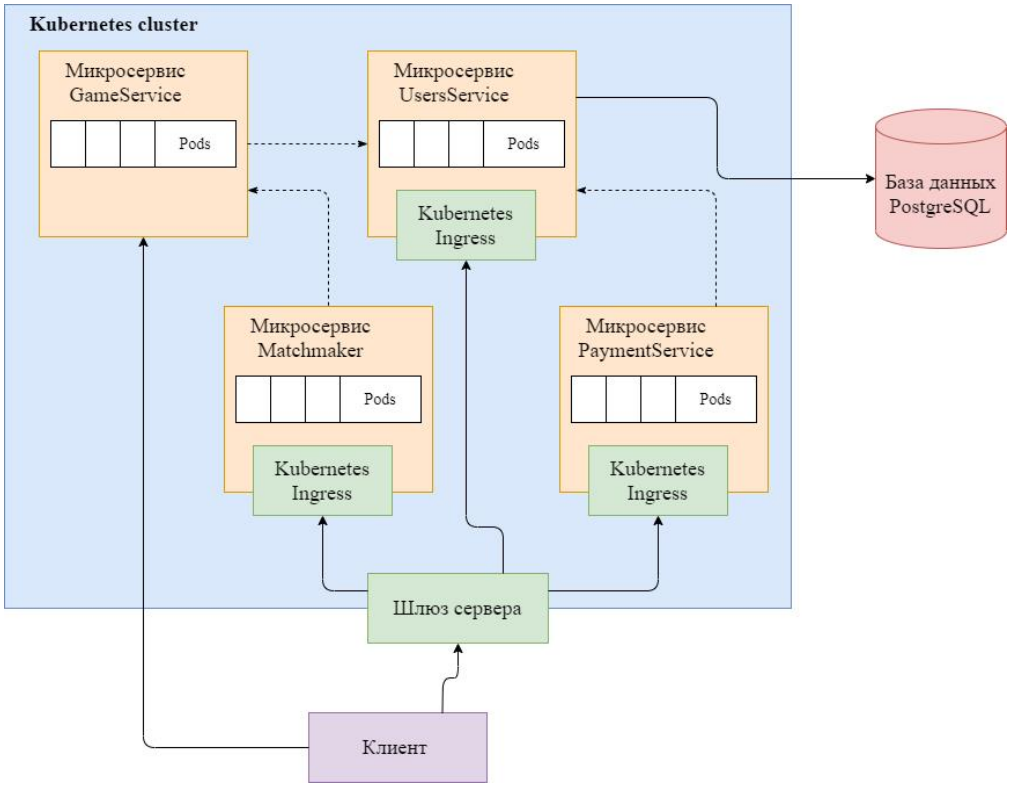

Рис. 1. Архитектура серверной части

Для динамического масштабирования компонент UsersService и PaymentService достаточно использовать ресурс платформы Kubernetes HorizontalPodAutoscaler [2], который будет выполнять масштабирование в зависимости от процента использования выделенного процессорного времени для экземпляров данных компонент. Однако для компонент GameService и Matchmaker данное решение не подойдёт в силу проблем с маршрутизацией запросов и сообщений и передачей информации между экземплярами компонент.

ИСПОЛЬЗОВАНИЕ ПЛАТФОРМЫ AGONES

Agones [3] (от греч. agōn – «соревнование», «состязание») – это платформа с открытым исходным кодом для развёртывания, хостинга, масштабирования и оркестрации игровых серверов для крупномасштабных многопользовательских игр, построенная на современном стандарте облачной индустрии – платформе Kubernetes.

Жизненный цикл экземпляра компоненты GameService можно увидеть на рис. 2.

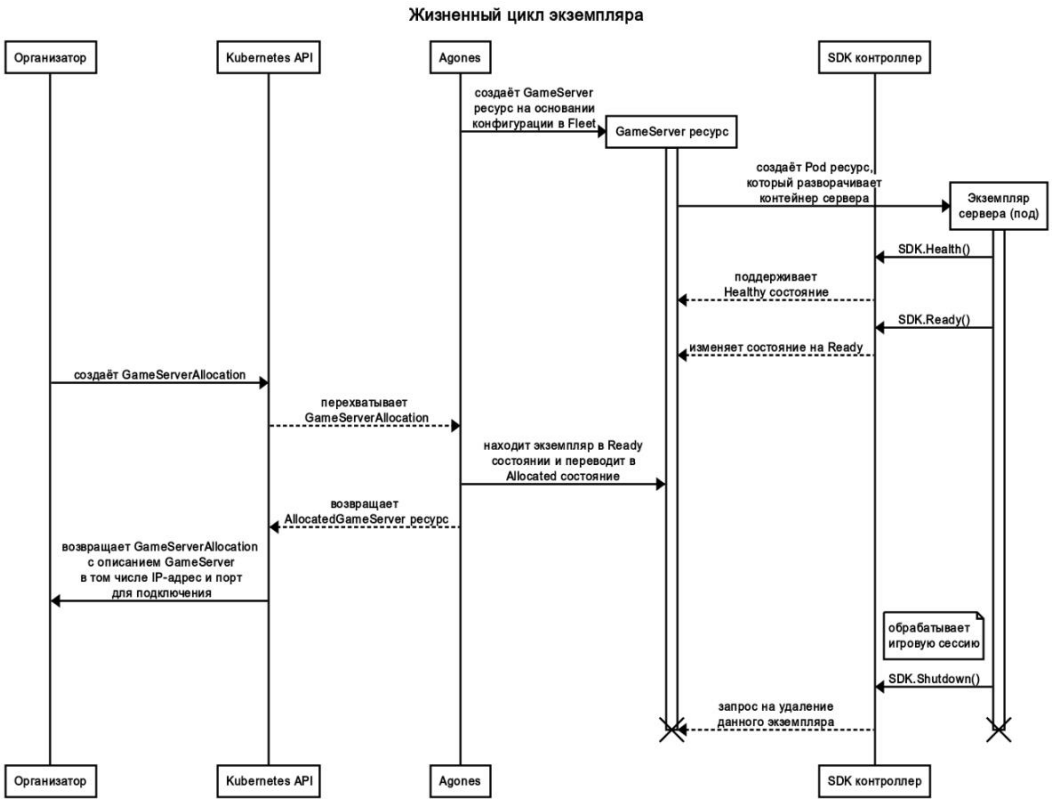

Рис. 2. Жизненный цикл экземпляра GameService

ИСПОЛЬЗОВАНИЕ ПЛАТФОРМЫ OPENMATCH

OpenMatch [4] – это платформа с открытым исходным кодом, предназначенная для упрощения создания масштабируемого организатора игр. Она предоставляет инфраструктуру, которая позволяет решить такие проблемы как обработка большого потока игроков и эффективный и быстрый поиск игроков в очереди на основе поставленных критериев. Данная платформа, как и Agones, использует Kubernetes как платформу для развёртывания и поддержки инфраструктуры организатора игр.

На основе данной платформы был разработан компонент Matchmaker, которую можно увидеть на рис. 3. Компоненты FrontendService, BackendService, QueryService, Redis предоставляет OpenMatch. Компоненты Matchmaker и MatchFunction предоставляются разработчиком.

ТЕСТИРОВАНИЕ ДИНАМИЧЕСКОГО МАСШТАБИРОВАНИЯ

Разработанный сервер был развёрнут на облачной платформе Google Cloud Platform. Для проверки динамического масштабирования сервера были созданы 500 тестовых клиентов, которые посылали запрос на участие в игре, подключались к выделенному серверу игры и выполняли движения вперёд и назад. Изначально был развёрнут всего 1 экземпляр компонента

Matchmaker и 10 экземпляров компонента GameService с размером сессии игры в 20 участников. После запуска тестовых клиентов сервер прошёл через 2 этапа масштабирования и создал 25 дополнительных экземпляров – 15 для обработки созданных сессий игр и 10 в качестве буфера. Среднее время ожидания составило 30 секунд, несмотря на то что полученная нагрузка была в 2.5 раза больше ожидаемой.

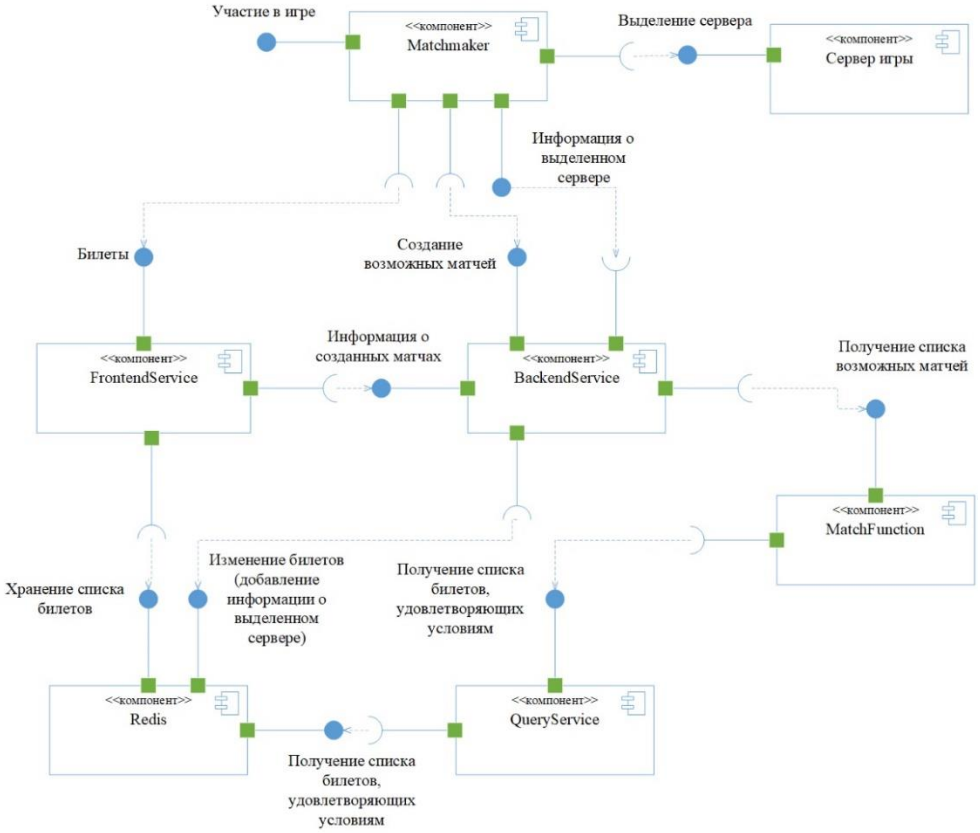

Рис. 3. Компонент Matchmaker

Библиографические ссылки

1. Global Gaming Study: More Gamers Spending More Money in COVID lockdowns – Which Publishers Will Benefit? L. Jaeger, N. Jarb, A. David. [Электронный ресурс] – Режим доступа: https://www.simon-kucher.com/nl/blog/new-global-gaming-industry-study-gamers-spend-more-money-and-time-increase-social-contact – Дата доступа: 01.06.2021.
2. Horizontal Pod Autoscaler [Электронный ресурс] – Режим доступа: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/ – Дата доступа: 01.06.2021.
3. Agones [Электронный ресурс] – Режим доступа: https://agones.dev/site/ – Дата доступа: 01.06.2021.
4. OpenMatch [Электронный ресурс] – Режим доступа: https://open-match.dev/site/ – Дата доступа: 01.06.2021.