СРАВНЕНИЕ ПОДХОДОВ К РЕСТРУКТУРИЗАЦИИ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ В УСЛОВИЯХ МАСШТАБИРОВАНИЯ В НЕСКОЛЬКИХ ДАТАЦЕНТРАХ

THE INFORMATION SYSTEM UNDER SCALING CONDITIONS IN SEVERAL DATA CENTERS MICROSERVICE ARCHITECTURE RESTRUCTURING APPROACHES COMPARISON

УДК 004.65+004.45

Васильев Никита Вячеславович, магистрант, Национальный исследовательский технологический университет «МИСиС», г. Москва, Россия, m1908870@edu.misis.ru

Vasilev N.V., master student, National Research Technological University «MISiS», Moscow, Russia, m1908870@edu.misis.ru

Аннотация

В статье рассматриваются подходы к построению микросервисной архитектуры в условиях масштабирования информационной системы в несколько датацентров с учетом особенностей: межсервисного взаимодействия и взаимодействия с реплицированными базами данных. Произведено сравнение различных подходов к построению микросервисной архитектуры с использованием технологии Kubernetes. Для сравнения использовалось нагрузочное тестирование построенных моделей информационной системы в определенной конфигурации, на основе чего были сделаны выводы о применимости различных подходов при реструктуризации архитектуры системы.

Annotation

The article discusses approaches to a microservice architecture development in terms of scaling an information system into several data centers, considering the peculiarities: interservice interaction and interaction with replicated databases. Different approaches to a microservice architecture development using Kubernetes technology were compared. For comparison, load testing of the constructed models of the information system in a certain configuration was used, based on which conclusions about the applicability of various approaches in restructuring the system architecture were drawn.

Ключевые слова: микросервисная архитектура, технология Kubernetes, информационные системы, микросервисы, масштабирование, датацентры, нагрузочное тестирование

Keywords: microservice architecture, Kubernetes technology, information systems, microservices, scaling, data centers, load testing

Сегодня в условиях ускоренной информатизации общества важной задачей при разработке и сопровождении информационных систем является повышение их отказоустойчивости путем масштабирования в несколько датацентров. Изначально большинство информационных систем проектируются и реализовываются в рамках применения исключительно одного центра обработки данных. Вследствие этого их масштабирование в несколько датацентров является серьезной проблемой, поскольку влечет за собой различного рода изменения в архитектуре приложения. Это, в свою очередь, требует проведения дополнительных исследований, т.к. выбор подходящего варианта реструктуризации архитектуры неочевиден.

Тема подходов к реструктуризации архитектуры информационной системы в условиях масштабирования в несколько датацентров в научных статьях раскрыта в недостаточной мере, потому что большинство подобных разработок принадлежат ко внутренним исследованиям и не подлежат огласке

согласно корпоративным политикам, эта статья направлена на восполнение данного пробела.

Масштабирование веб-приложения, имеющего микросервисную архитектуру, из одного в несколько датацентров (ДЦ) может быть осложнено такими факторами как увеличение задержки обработки запроса или увеличение трафика между несколькими датацентрами.

В данной работе рассматривается вопрос выбора подхода к построению архитектуры системы в нескольких датацентрах из числа альтернативных.

Одна из архитектурных особенностей – межсервисное взаимодействие. Рассматриваемые подходы к реализации:

1. Обмен данными между сервисами с использованием общего сетевого шлюза.
2. Обмен данными между сервисами с использованием сетевого шлюза, локального для каждого датацентра.

Другой особенностью является взаимодействие с реплицированными базами данных, для реализации существуют следующие подходы:

1. Ведущая репликация (master) в одном ДЦ и ведомая репликация (slave) в остальных.
2. Ведущая в каждом ДЦ с ведомыми подами для масштабирования.

При репликации Master-Slave существует только один экземпляр ведущей базы данных. Все остальные являются ведомыми – доступными только на чтение данных. При этом подходе возникают проблемы при увеличении числа ведомых экземпляров, так как всегда есть только один ведущий экземпляр [1, 2].

При репликации Master-Master в каждом из ДЦ есть ведущий экземпляр базы, который синхронизируется с ведомыми экземплярами в том же ДЦ. При этом подходе любой из экземпляров может использоваться как для чтения, так и для записи [5].

При использовании общего сетевого шлюза запросы направляются как в первый, так и во второй датацентр без приоритезации запросов [4]. При этом довольно легко балансировать нагрузку между всеми экземплярами сервиса в

разных датацентрах, но используется большое количество траффика между датацентрами.

При использовании локальных сетевых шлюзов запросы из сервиса А в сервис Б приоритетно маршрутизируются в сервисы Б, находящиеся в том же датацентре. Такой подход делает сложным балансировку между экземплярами сервиса в разных датацентрах, но минимизирует количество траффика между датацентрами.

Сравнение различных подходов к взаимодействию с базами данных и между сервисами осуществлялось путем проведения нагрузочного тестирования каждой архитектуры [3, 6]. Результаты сравнения приведены в таблице 1.

Нагрузочное тестирование проводилось в следующей конфигурации:

1. Сервисы развертываются в двух датацентрах.
2. Сервисы на языке программирования Golang.
3. Все экземпляры сервисов и баз данных разворачивались в кластерах Kubernetes.
4. Нагрузочное тестирование проводилось средствами информационной системы «Яндекс.Танк».
5. Объем передаваемых в сервис данных и данных, записываемых в базу данных, должен составлять 1 Кб, без учета специфики протокола, используемого при передаче данных в другие сервисы или базы данных.
6. Нагрузка линейно увеличивалась до 400 запросов в секунду.

Таблица 1. – Результаты проведения нагрузочного тестирования различных моделей архитектуры информационной системы

Подход к построению архитектуры	**Количество ответов с кодом 200 – «ОК»**	**Средняя задержка ответа (сек)**	**Максимальная задержка ответа (сек)**
Общий сетевой шлюз	95 %	0.10	0.82
Локальный сетевой шлюз	99 %	0.02	0.65
Репликация Master-Slave	93 %	0.18	0.47

Репликация Master-Master	99 %	0.03	0.19

Результаты нагрузочного тестирования для моделей архитектуры информационной системы представлены на рисунках 1 – 4.

Для каждой из моделей представлено два графика: график со временем ответа сервисов в различных перцентилях (p100, p99, p98, p95, pp90, p85, p80, p75, p50 – точность времени обработки запросов) и график, отражающий количество успешно обработавших запросов (HTTP 500 – ошибка, HTTP 200 – запрос выполнился успешно).

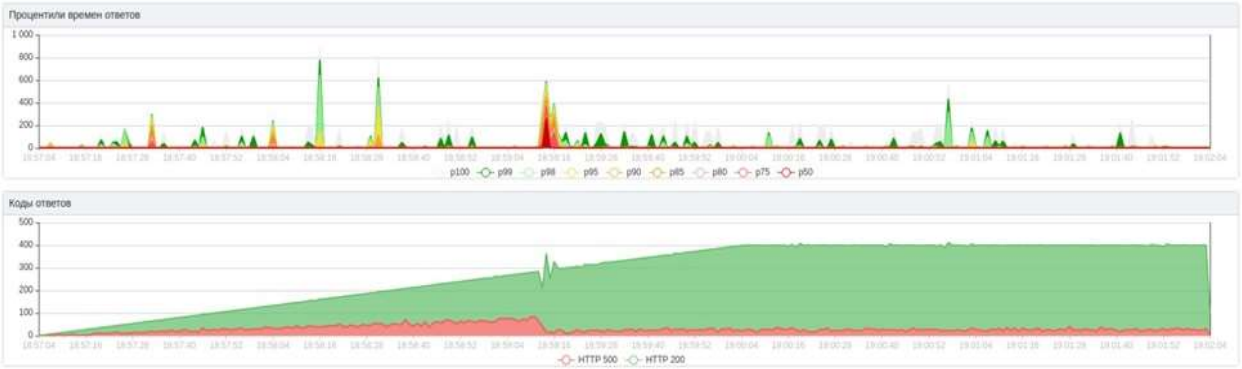

Рисунок 1. – Общий сетевой шлюз

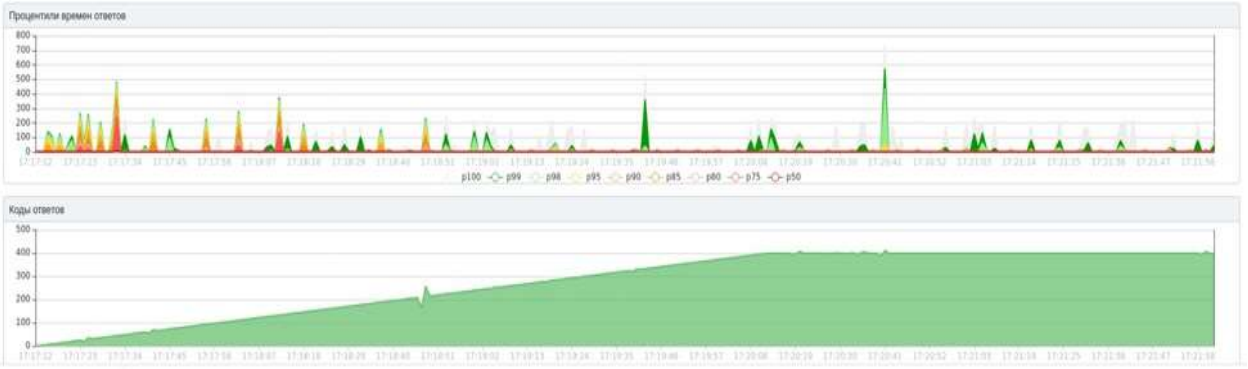

Рисунок 2. – Локальный сетевой шлюз

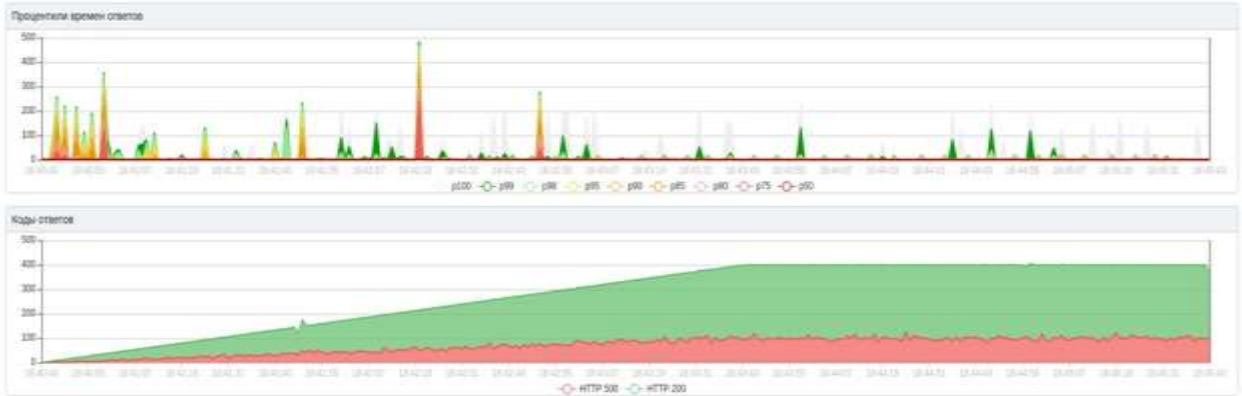

Рисунок 3. – Репликация Master-Slave

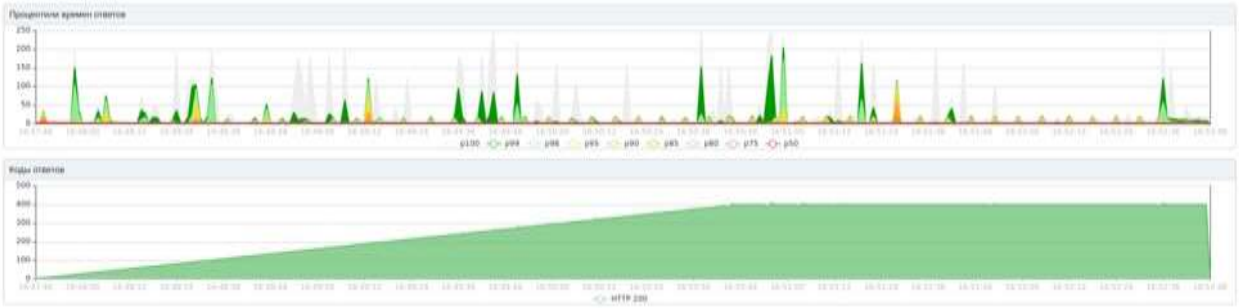

Рисунок 4. – Репликация Master-Master

По результатам реализации выбранной архитектуры можно сделать выводы о полученных положительных эффектах: повышение среднего времени наработки системы на отказ, уменьшение количества сетевых запросов между датацентрами и уменьшение среднего времени обработки запроса при использовании подходов к построению архитектуры с использованием репликации Master-Master и локальных сетевых шлюзов.

Литература

1. Balalaie A., Heydarnoori A., Jamshidi P. «Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture» // IEEE Software. – 2016. – №33. – С. 42-52.
2. Song M., Shin H. Replica Striping for Multi-resolution Video Servers // Boavida F., Monteiro E., Orvalho J. (eds), Protocols and Systems for Interactive Distributed Multimedia. IDMS 2002. Lecture Notes in Computer Science, vol 2515. – Heidelberg: Springer-Verlag. – 2002.
3. Артамонова А.А., Сравнительный анализ двух систем управления контейнерами: Docker Swarm И Kubernetes // Синергия наук. – 2018. – №23. – С. 1173-1182.
4. Дьяченко Р.А., Ковшун А.Н., Борисов С.Н. Обзор возможностей балансировки сетевой нагрузки на основе Haproxy // Сборник научных статей VII Международной научно-практической конференции. – 2017. – С. 167-170.
5. Котиков П., Нечай А. Репликация данных между серверами баз данных в среде геоинформационных систем // Вестник Российского Нового

Университета. Серия: сложные системы: модели, анализ и управление. – 2015. – №1. – С. 88-91.

6. Позин Б.А., Галахов И.В. Модели в нагрузочном тестировании // Программирование. – 2011. – №1. – С. 20-35.

Literature

1. Balalaie A., Heydarnoori A., Jamshidi P. «Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture» // IEEE Software. – 2016. – No. 33. – P. 42-52.
2. Song M., Shin H. Replica Striping for Multi-resolution Video Servers // Boavida F., Monteiro E., Orvalho J. (eds), Protocols and Systems for Interactive Distributed Multimedia. IDMS 2002. Lecture Notes in Computer Science, vol 2515. – Heidelberg: Springer-Verlag. – 2002.
3. Artamonova A.A., Comparative analysis of two container management systems: Docker Swarm and Kubernetes // Synergy of Sciences. – 2018. – No. 23. – P. 1173-1182.
4. Dyachenko R.A., Kovshun A.N., Borisov S.N. Review of Haproxy-based Network Load Balancing Capabilities // Collection of Scientific Articles of the VII International Scientific and Practical Conference. – 2017. – P. 167-170.
5. Kotikov P., Nechai A. Data replication between database servers in the environment of geoinformation systems // Bulletin of the Russian New University. Series: complex systems: models, analysis and control. – 2015. – No. 1. – P. 88-91.
6. Pozin B.A., Galakhov I.V. Models in Load Testing // Programming. – 2011. – No. 1. – P. 20-35.