

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.273:004.451.45

Саркисян
Эвелина Лерниковна

Алгоритм многопоточной обработки данных в микросервисной архитектуре

АВТОРЕФЕРАТ

на соискание степени магистра технических наук
по специальности 1-40 80 02 «Системный анализ, управление и обработка
информации (по отраслям)»

Научный руководитель

Навроцкий Анатолий
Александрович
кандидат физико-
математических наук, доцент

Минск 2023

ВВЕДЕНИЕ

В цифровой век технологии, построенные на надежной программной архитектуре, являются основой успеха в бизнесе. Наиболее важным фактором ценности архитектуры программного обеспечения является возможность экспериментировать с конечными пользователями и быстро воплощать новые бизнес-идеи в продукты. Технология предлагает различные подходы к решению проблем. Архитектура микросервисов — один из таких подходов в технологии проектирования сервисов. Она приобрела популярность благодаря успешному внедрению крупными компаниями. Однако при переходе от традиционной монолитной архитектуры к микросервисам общим узким местом является производительность. В данной работе рассматриваются алгоритмы повышения производительности с использованием многопоточной обработки.

Тема диссертации является актуальной в связи с популярностью использования микросервисов в веб-приложениях в настоящее время. С ростом сложности приложений и объемов данных возникает необходимость эффективной обработки с целью повышения производительности всей системы.

Цель диссертационной работы заключается в исследовании существующих алгоритмов и методов многопоточной обработки данных в микросервисной архитектуре, а также разработке алгоритма для увеличения производительности системы.

Объектом исследования является веб-приложение, состоящее из микросервисов. Предмет исследования представляет собой методы обработки данных в микросервисной архитектуре.

Задачи, решаемые в диссертации:

- анализ существующих архитектур построения приложений, оценка преимуществ и недостатков многопоточной обработки;
- сравнение современных подходов к обработке данных в микросервисной архитектуре;
- разработка алгоритма многопоточной обработки данных в микросервисной архитектуре.

В первом разделе диссертации производится обзор существующих моделей построения веб-приложений, способов взаимодействия, а также недостатки многопоточности. Во втором разделе проводится адаптация алгоритмов оптимизации многопоточной обработки данных. В третьем разделе проводится анализ производительности предложенных алгоритмов оптимизации. В четвертом разделе разрабатывается алгоритм динамического определения оптимального количества потоков при многопоточной обработке данных в микросервисной архитектуре и приводится тестирование его эффективности.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность исследования

Актуальность темы связана с растущей популярностью микросервисной архитектуры в настоящее время. С ростом сложности приложений и объемов данных, обрабатываемых ими, возникает необходимость эффективной обработки с целью повышения производительности всей системы.

Цель исследования

Целью диссертационной работы исследование существующих и разработка новых алгоритмов для оптимизации многопоточной обработки данных в микросервисной архитектуре.

Задачи исследования

1. Обзор существующих архитектур построения приложений, анализ актуальных способов взаимодействия, а также возможностей и недостатков многопоточной обработки;
2. Адаптация актуальных методов обработки данных в микросервисной архитектуре;
3. Разработка алгоритма оптимизации многопоточной обработки в микросервисной архитектуре.

Новизна полученных результатов

Новизна полученных результатов диссертационной работы заключается в адаптации существующих методов и разработке алгоритма оптимизации многопоточной обработки в микросервисной архитектуре.

Личный вклад соискателя

Соискателем выполнены все изложенные в работе разработки и исследования. Постановка задач и обсуждение результатов проводились совместно с научным руководителем. Обработка, интерпретация данных, а также выводы сделаны автором самостоятельно.

Апробация и опубликованность результатов диссертации

Основные положения диссертационной работы докладывались на следующих научных конференциях:

– The International Conference on Information Technologies and Systems ITS 2022 (Минск 2022).

По материалам диссертационной работы опубликована 1 печатная работа, Объем публикации составляет 2 печатные страницы.

Структура и объем диссертации

Диссертационная работа состоит из введения, четырех глав, заключения, списка использованных источников и приложения. Общий объем магистерской диссертации составляет 57 страниц, включая 26 иллюстраций, одну таблицу, библиографический список из 30 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В настоящей работе проводится исследование и разработке алгоритмов многопоточной обработки данных в микросервисной архитектуре.

Во введении определена область и указаны основные направления исследования, показана актуальность темы диссертационной работы.

В первой главе диссертационной работы проведен анализ предметной области. Подробно проанализированы существующие архитектуры построения веб-приложений: монолитная, микросервисная и сервис-ориентированная. Каждый из подходов обладает как некоторыми преимуществами, так и недостатками. Для дальнейшего проведения исследований выбрана микросервисная архитектура.

Приведены актуальные способы взаимодействия в микросервисной архитектуре – в данной диссертации исследование проводится на приложении с взаимодействием через асинхронный брокер сообщений.

Рассмотрена возможность использования многопоточной обработки, ее возможности и недостатки. В связи с неравномерностью сообщений, поступающих на обработку, необходимо учитывать ситуации с большой нагрузкой или отсутствием сообщений вовсе. Следовательно, возникает проблема определения оптимального количества потоков при многопоточной обработке в микросервисной архитектуре.

Во второй главе адаптированы методы оптимизации многопоточной обработки данных такие как, оптимизация за счет реализации пула потоков, пакетной обработки, барьерная синхронизация и Fork/Join алгоритм.

Проанализированные алгоритмы позволяют сократить общие накладные расходы на инициализацию и управление потоков, а также позволяют более эффективно использовать ресурсы системы и предотвратить некоторые проблемы синхронизации. Однако каждый из них имеет ограничения в использовании, выбор зависит от задач и требований программы, а также от аппаратных характеристик системы, таких как количество доступных процессоров и ядер, объем доступной памяти, типы и скорости доступа к разделяемым ресурсам.

В третьей главе проводится анализ производительности предложенных методов оптимизаций многопоточной обработки данных таких как, оптимизация за счет реализации пула потоков, пакетной обработки, барьерной синхронизации и Fork/Join алгоритма.

Анализ результатов показал, что внедрение рассмотренных алгоритмов позволяет улучшить производительность при выполнении большого количества задач путем уменьшения накладных расходов на вызов одной задачи. Благодаря адаптированным алгоритмам оптимизации общее среднее время на получение и

обработку данных, поступающих из асинхронного брокера сообщений, сократилось на 11.3%.

Однако данные алгоритмы не позволяют динамически регулировать количество потоков в зависимости от нагрузки. В следующей четвертой проводится разработка нового алгоритма для решения этой проблемы.

В четвертом разделе разработан алгоритм динамического определения количества потоков в микросервисной архитектуре. Подробно описаны характеристики предложенного алгоритма и проанализировано влияние размера пула потоков на производительность системы.

Результаты проведенного тестирования подтверждают способность алгоритма автоматически адаптироваться к изменениям в рабочей нагрузке, включая как увеличение, так и снижение количества потоков, чтобы максимально эффективно использовать ресурсы и поддерживать производительность системы. Это позволяет оптимизировать выполнение задач без необходимости ручной настройки количества потоков.

ЗАКЛЮЧЕНИЕ

В процессе выполнения диссертационной работы проведен анализ проблемы многопоточной обработки сообщений в микросервисной архитектуре.

Рассмотрены архитектуры построения современных веб-приложений, способов взаимодействия и предложены алгоритмы повышения эффективности многопоточной обработки за счет реализации пула потоков и пакетной обработки. Благодаря уменьшению накладных расходов на вызов одной задачи обеспечивается улучшение производительности при выполнении большого количества задач. Внедрение предложенных оптимизаций привело к заметному снижению среднего времени выполнения на 11.3%, что свидетельствует об улучшении производительности системы.

Для решения проблемы выбора оптимального количества потоков разработан алгоритм динамического определения оптимального количества потоков при обработке сообщений в микросервисной архитектуре. Предложенный алгоритм позволяет в режиме реального времени регулировать размер пула потоков в зависимости от нагрузки, что позволяет оптимизировать выполнение задач, распределяя нагрузку между доступными ресурсами более эффективно, избегать их избыточного использования или недостатка. В качестве стартового значения для размера потоков определено количество логических процессоров на текущем оборудовании. Тестирование показало, что предложенный алгоритм позволил уменьшить среднее время обработки на 7.61%.

Предложенные в диссертационной работе алгоритмы позволяют существенно увеличить скорость многопоточной обработки данных в микросервисной архитектуре, а также максимально эффективно использовать ресурсы и поддерживать производительность системы

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1. Саркисян, Э.Л. Алгоритмы и методы повышения производительности веб-приложений / Э. Л. Саркисян // Информационные технологии и системы 2022 (ИТС 2022) = Information Technologies and Systems 2022 (ITS 2022) : материалы международной научной конференции, Минск, 23 ноября 2022 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2022. – с. 161–162