

Имитационная модель реактивной микросервисной корпоративной информационной системы

Введение. Для достижения высокой отказоустойчивости высоконагруженных приложений с клиент-серверной архитектурой применяют горизонтальное масштабирование. Это влечёт за собой расходы на приобретение дополнительного оборудования. Поэтому все более актуальной становится возможность повышения надежности корпоративных информационных систем (КИС) путем более рационального использования вычислительных ресурсов.

Анализ публикаций. Вопросы оптимизации работы информационных системы рассмотрены в работах [1,3]. В них предлагается использовать принципы неблокирующих технологий обработки запросов и реактивного программирования.

Цель работы заключается в разработке имитационной модели функционирования корпоративной информационной системы, построенной на основе микросервисной архитектуры.

Основная часть. Серверная часть информационной системы можно рассматривать как систему массового обслуживания (СМО). В такую систему в случайные моменты времени поступают заявки на обслуживание. Эти заявки обслуживаются с помощью имеющихся в распоряжении системы сервисных каналов. Между источником заявок и каналами их обслуживания находится накопитель заявок – очередь.

Особенность рассматриваемой системы массового обслуживания состоит в том, что для обработки заявок используется неблокирующий подход[2]. Это значит, что каждый принимающий слой выполняя свои операционные обязанности формирует асинхронный запрос к следующему слою и не дожидаясь ответа приступает к обработке следующего запроса.

Имитационная модель позволяет симулировать работу дублирующих микросервисов с возможностью выхода из строя каждого из сервисов в случайные моменты времени.

Модель содержит источник заявок, формируемых пользователями, шлюз, модуль сервисов с сервером баз данных и генератор сигнала остановки одного из микросервисов на обслуживание.

Одним из основных элементов модели является шлюз. Этот элемент решает задачу маршрутизации запросов в соответствии с входными данными к соответствующим каналам соответствующих обработчиков. Он создает новые неблокирующие асинхронные запросы к микросервисам и обрабатывает пришедшие от них готовые ответы. Также в его обязанности входит балансировка нагрузки и переключение потока запросов на работоспособный сервис в случае аварийной или плановой остановки одного из них.

Таким образом можно классифицировать данную СМО как многоканальную, многофазную с неограниченной очередью и нетерпеливыми заявками.

В результате моделирования получаем время жизни заявки в СМО (рисунок 1). В среднем оно составляет 0,156 с. Заявками с наименьшим временем пребывания в системе – GET-запросы на загрузку статичных ресурсов.

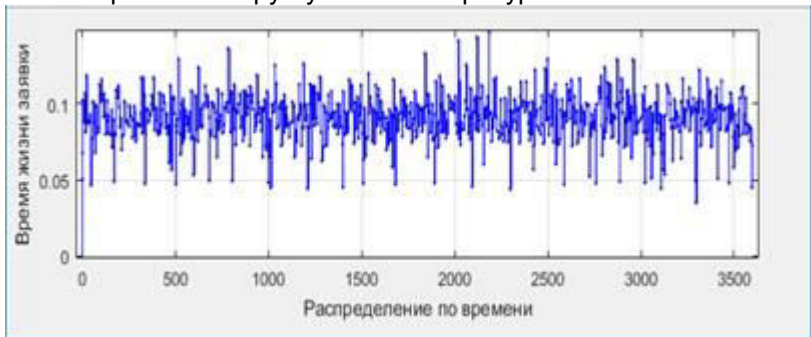


Рисунок 1 – Время пребывания заявки в системе

В результате проведенных имитационных экспериментов были определены загруженности шлюза и сервисов в разных режимах работы, количество потерянных запросов в случае аварийной остановки одного из сервисов.

Список использованных источников

1. Reactor 3 Reference Guide // Project Reactor – [Электронный ресурс].
URL: <https://projectreactor.io/docs/core/release/reference/index.html#getting-started-introducing-reactor>. (Дата обращения: 15.03.2022).
2. Томаш Нуркевич, Бен Кристенсен. Реактивное программирование с применением Rxjava / пер. с англ. — М.: ДМК Пресс, 2017. - 358 с.
3. Файзутдинов Р.Н. – Математическое моделирование сложных систем. – Казань: ВАС, 2013 – 69 с.