

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПРОГРАММНОГО МОДУЛЯ ОБРАБОТКИ ВЫСОКОНАГРУЖЕННОЙ ОЧЕРЕДИ СООБЩЕНИЙ НА ОСНОВЕ NOSQL БАЗЫ ДАННЫХ ДЛЯ СИСТЕМЫ С МИКРОСЕРВИСНОЙ АРХИТЕКТУРОЙ

Гагарин В. Ю.

(Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»),

Вагнер А. В.

(ООО «Торроу Технолоджис»)

Научный руководитель – кандидат технических наук, доцент Тропченко А. А.

(Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»)

Сегодня программным инженерам доступен целый спектр технологий оптимизации нагрузки в приложениях с микросервисной архитектурой. Уже определён ряд базовых подходов для обеспечения отказоустойчивости программных комплексов. Один из таких подходов – внедрение подсистемы управления очередями сообщений. Часто для поставленных задач достаточно внедрения готового решения от Kafka, RabbitMQ или NATS. Иногда бывают случаи, когда, по ряду причин, появляется необходимость в собственной разработке подобной подсистемы. Определим, какие факторы здесь обязательно стоит учесть, и какие существуют особенности реализации таких программных модулей.

Введение. Логика некоторых бизнес-операций промышленного приложения требует асинхронного выполнения в фоновом режиме через очередь сообщений. Использование готовых решений не всегда обосновано и выгодно. Во-первых, это усложняет задачу поддержки инфраструктуры системы в целом. Во-вторых, так как, как правило, используются различные системы хранения данных для брокера сообщений и остальной части приложения, требуется поддерживать согласованность/консистентность данных между ними. В таких случаях целесообразно иметь собственный программный модуль обработки высоконагруженной очереди сообщений, чтобы быстро, качественно, эффективно и удобно настраивать его под свои нужды.

Основная часть. Собственная разработка проводится на базе высоконагруженной платформы Togrow. Платформа имеет микросервисную архитектуру. В качестве основного средства разработки используется фреймворк .NET Core, а в качестве основы системы хранилища данных – нереляционная СУБД MongoDB. Микросервисы платформы разделены на логические слои по своим функциональным возможностям и уровню доступа к данным. В качестве протокола межсервисного взаимодействия используется протокол GRPC – система удалённого вызова процедур от компании Google. При выполнении той или иной бизнес-операции часть логики приложения может выполняться асинхронно, например, рассылка уведомлений пользователям. Для обеспечения такого подхода через очередь сообщений необходимо определить объект – сообщение очереди (QueueMessage), которое имеет отношение к определённому пакету (QueueBatch). В таком случае первый микросервис через издателя очереди кладёт это сообщение в очередь и завершает бизнес-операцию, а затем подписчик очереди получает это сообщение из очереди и вызывает метод второго микросервиса для непосредственной отправки уведомлений. Основные компоненты подсистемы:

- Queue Controller – микросервис, с помощью которого можно положить сообщение в очередь (Издатель);
- Queue Processor – микросервис, который подписан на изменения коллекции в базе данных. Извлекает новые сообщения из очереди и выполняет их (Подписчик);
- Torgow Message Queue – общая библиотека для согласования работы компонентов подсистемы.

Выводы. Разработан программный модуль управления очередями сообщений на базе существующих брокеров. Дана оценка эффективности разработанного решения. Подсистема внедрена в промышленную версию разрабатываемой высоконагруженной платформы Torgow. Планируется дальнейшее апробирование результатов в научно-образовательном, междисциплинарном журнале (входит в SCOPUS).

Гагарин В. Ю. (автор)

Подпись

Тропченко А. А. (научный руководитель)

Подпись