

## РАСПРЕДЕЛЕННЫЙ БРОКЕР СООБЩЕНИЙ КАФКА ДЛЯ ВЫСОКОСКОРОСТНОЙ ПЕРЕДАЧИ И АГРЕГАЦИИ ДАННЫХ

В. Л. Радищевский, А. Д. Кульневич  
Научный руководитель: Е. И. Губин

Национальный исследовательский Томский политехнический университет  
E-mail: vladrad95@mail.ru

### Введение

Цель настоящей работы рассмотреть архитектуру и особенности распределенного брокера сообщений Kafka, а так же оценить такие параметры производительности как пропускная способность и задержка.

Система очереди сообщений является одной из наиболее важных базовых технологий, благодаря которым можно построить крупномасштабную распределенную систему, объединяя слабосвязанные и автономные вычислительные единицы. В крупномасштабной системе с присутствием множества вычислительных узлов система очереди сообщений становится потенциально узким местом производительности, которое может ограничить потенциал использования вычислительных ресурсов. Поэтому немаловажной особенностью является выбор и настройка системы очереди сообщений для широкомасштабной распределенной системы. Системы очереди сообщений, такие как ZeroMQ, ActiveMQ и RabbitMQ, широко применяются во многих различных промежуточных средах. Разработанная в корпорации LinkedIn служба обмена сообщениями Apache Kafka обладает такими качествами как масштабируемость, высокая производительность и отказоустойчивость. Она нашла широкое применение во многих крупных компаниях и корпорациях, таких как Yahoo, Netflix, PayPal и многие другие [\*]. С развитием Apache Hadoop, который стал де-факто системой для большой

инфраструктурой обработки данных за счет использования распределенной файловой системы (HDFS), структуры параллельной обработки (MapReduce), системой планирования заданий и управления кластером YARN, он стал полноценной многопользовательской платформой, поддерживающей такие режимы обработки данных, как пакетная (Batch) обработка, вычисления в реальном времени (Real-time computing), интерактивная (Interactive) обработка, и т. д.

### Apache Hadoop

Hadoop – это распределенная вычислительная среда, предназначенная для создания кластера для реализации распределенного хранения и вычислений. Распределенная файловая система HDFS предназначена для хранения файлов, поблочно распределенных между узлами вычислительного кластера. Более того применяются методы для хранения данных в HDFS, реализующие высокую надежность и доступность, такие как репликация данных, синхронизация и автоматическая обработка отказов. Помимо HDFS, текущая версия Hadoop включает в себя фреймворк MapReduce для программирования распределенных вычислений, планировщик заданий и ресурсов YARN, а так же распределенный сервис конфигурирования и синхронизации ZooKeeper, используемый Kafka для координации между потребителями сообщений (consumers) в условиях распределенной передачи.

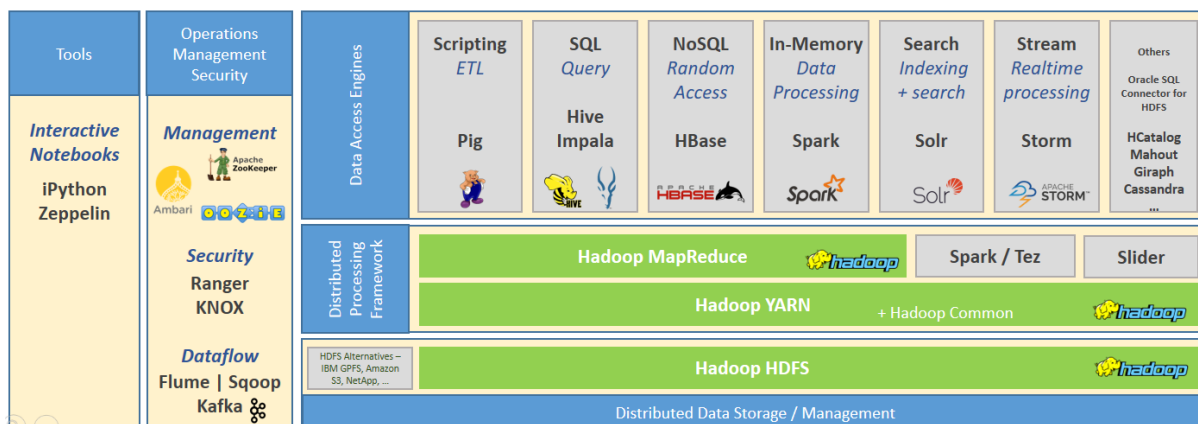


Рис. 1. Экосистема Hadoop [2]

## Apache Kafka

Apache Kafka – распределенная система обмена сообщениями. Ее архитектура делает систему высокопроизводительной, надежной, и масштабируемой. Система Kafka представляет собой брокер (Broker), экземпляры которого установлены на узлах распределенной системы и отвечают за хранение сообщений. Брокер хранит сообщения в различных темах (Topics), которые в свою очередь, разделены на разделы (Partition). Продюсеры (Producers) записывают данные в темы, а потребители (Consumers) читают данные. Разделы так же распределены по различным брокерам кластера. Kafka управляет сообщениями в каждом разделе устанавливая смещения (Offsets) для определения текущей позиции чтения данных. Сообщения в каждом разделе могут реплицироваться и использоваться для восстановления в случае сбоя. Kafka предоставляет API-интерфейсы продюсеров и потребителей, используемые для передачи и получения сообщений из тем. Потребители контролируют операцию чтения за счет смещения offset. Поэтому для того, чтобы повторно извлекать сообщения нужно сбросить значение offset на предыдущие позиции. Поскольку операции по контролю смещений выполняются потребителями, брокеры Kafka не знают о статусе чтения сообщений по темам. Брокеры хранят сообщения в течение заданного времени перед удалением из памяти.

### Оценка производительности

Оценка производительности работы производилась по схеме, представленной на рисунке 2. Приложение Kafka Streams для обработки и агрегации сообщений расположено вместе с Kafka брокером на одном компьютере. Приложения с Producer и Consumer были расположены на другом компьютере в одной локальной сети.

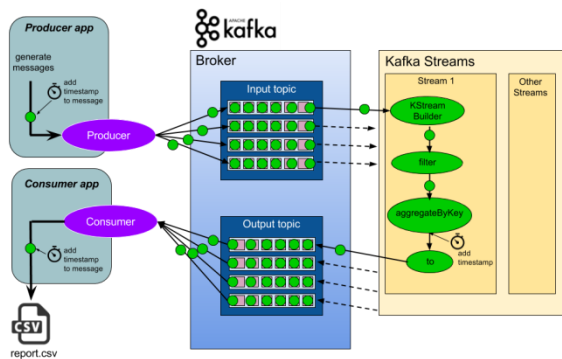


Рис. 2. Схема передачи и обработки данных

Каждое передаваемое JSON сообщение помечалось временной отметкой (timestamp), затем оно помечалось после агрегации и после получения для определения скорости передачи, а так же задержки.

На рисунке 3 представлен график скорости

передачи и приема сообщений в зависимости от общего числа переданных сообщений.

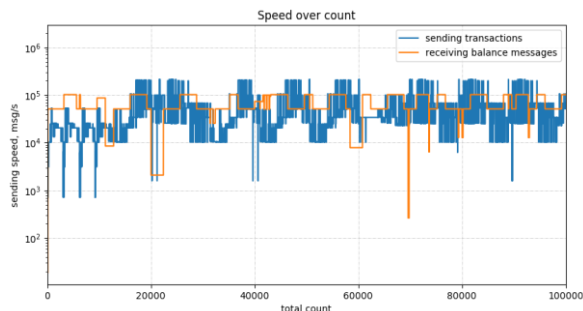


Рис. 3. Скорость передачи и приема сообщений в Apache Kafka

Средняя скорость по 100000 тыс. переданным сообщениям составила ~132345 сообщений в секунду. При этом скорость получения сообщений составляла 31340 сообщений в секунду. Такая разница связана с особенностью агрегации сообщений в данном конкретном примере. Полученное сообщение (transaction) содержало в себе идентификатор пользователя, а так же сумму транзакции. Каждый пользователь состоит в различных «группах» и в случае осуществления транзакции в баланс «каждой» группы засчитывалась эта сумма. Таким образом, каждое полученное сообщение порождало по несколько операций обработок. Среднее время задержки между отправкой и обработкой сообщения составило 0,15 с, а между обработкой и получением 2,18 с.

### Заключение

Рассмотрена архитектура и основные особенности распределенного брокера сообщений Kafka. Оценены параметры пропускной способности и задержки. В данном случае, скорость передачи сообщений ограничилась максимальной пропускной способностью канала.

### Список использованных источников

- 1 Nguyen C. N., Kim J. S., Hwang S. КОНА: Building a Kafka-Based Distributed Queue System on the Fly in a Hadoop Cluster //Foundations and Applications of Self\* Systems, IEEE International Workshops on. – IEEE, 2016. – С. 48-53.
- 2 Bridging two worlds: Integration of SAP and Hadoop Ecosystems. [Электронный ресурс]. – URL: <https://blogs.sap.com/2017/07/19/bridging-two-worlds-integration-of-sap-and-hadoop-ecosystems> (дата обращения 22.11.2017).
- 3 Apache Kafka Documentation. [Электронный ресурс]. – URL: <https://kafka.apache.org/documentation/> (дата обращения 22.11.2017).