СБЕР УНИВЕРСИТЕТ

Школа Java Middle Developer

Kafka

Надёжность доставки данных

Содержание

1. Настройка брокера;
2. Использование производителей и потребителей в надёжной системе;
3. Проверка надёжности системы.

Надёжность доставки данных

Надёжная доставка данных

- ✓Надёжность это свойство всей системы;
- ✓Apache Kafka очень гибка в том, что касается надежной доставки данных;
- ✓Kafka спроектирована в расчете на довольно широкие возможности настройки, а её клиентский API достаточно гибок для любых компромиссов.

Гарантии надёжности

- Упорядоченность сообщений в партиции;
- Сообщения от производителей считаются зафиксированными, когда они записаны во все согласованные реплики партиции, но не обязательно уже сброшены на диск;
- Зафиксированные сообщения не будут потеряны, если функционирует хотя бы одна реплика;
- Потребители могут читать только зафиксированные сообщения.

Репликация

- Kafka обеспечивает сохраняемость сообщений в случае аварийного сбоя благодаря записи сообщений в несколько реплик;
- Реплика считается согласованной, если она является ведущей репликой партиции или ведомой, которая:
  - отправляла в ZooKeeper контрольный сигнал в последние 6 секунд;
  - извлекала сообщения из ведущей реплики в последние 10 секунд;
  - извлекала наиболее свежие сообщения из ведущей реплики в последние 10 секунд.
- Состояние системы, при котором одна или несколько реплик быстро перепрыгивают из согласованного состояния в рассогласованное и наоборот, - верный признак проблем с кластером;
- Отстающая согласованная реплика может замедлять работу производителей и потребителей, поскольку они считают сообщение зафиксированным только после его получения всеми согласованными репликами

Настройка брокера

- На поведение Kafka в смысле надёжного хранения сообщений влияют три параметра конфигурации:
  - Коэффициент репликации
  - «Грязный» выбор ведущей реплики
  - Минимальное число согласованных реплик
- Эти параметры можно использовать на уровне брокера для управления настройками всех топиков системы, а также на уровне отдельных топиков.

Коэффициент репликации

- Соответствующий параметр уровня топика называется replication.factor;
- На уровне брокера для автоматически создаваемых топиков используется параметр default.replication.factor;
- Коэффициент репликации N означает возможность потери N - 1 брокеров при сохранении надежности чтения из топика и записи в него;
- Доступность повышается за счёт дополнительного аппаратного обеспечения.

Как определить правильное число реплик для топика?

- Ответ зависит от степени его важности и от ресурсов, которые вы готовы потратить для повышения доступности;
- Рекомендуется использовать коэффициент репликации 3 для всех топиков, для которых важна доступность;
- Есть проекты, в которых для критично важных топиков создаются пять и более реплик;
- При заданных названиях стоек (broker.rack) Kafka обеспечит распределение партиции по нескольким стойкам, что гарантирует еще более высокую доступность.

«Грязный» выбор ведущей реплики

- Параметр, доступный только на уровне брокера, называется unclean.leader.election.enable. По умолчанию его значение равно true;
- Если ведущая реплика партиции становится недоступной, одна из согласованных реплик выбирается новой ведущей («чистый» способ);
- Разрешение рассогласованным репликам становиться ведущими увеличивает риск потери данных и того, что они станут противоречивыми;
- Если же запретить это, уменьшится доступность из-за необходимости ждать, пока первоначальная ведущая реплика станет доступной и можно будет восстановить работу партиции.

Минимальное число согласованных реплик

- Соответствующий параметр уровня как топика, так и брокера называется min.insync.replicas;
- Если в топике три реплики и для параметра min.insync.replicas установлено значение 2, то записывать в партицию топика можно будет только тогда, когда по крайней мере две из трех реплик согласованы.

Использование производителей в надёжной системе

При разработке приложений, которые служат производителями, необходимо учитывать две вещи:

- Использование соответствующего требованиям надежности значения параметра acks;
- Правильная обработка ошибок как в настройках, так и исходном коде.

Отправка подтверждений

Производители могут выбрать один из трёх режимов подтверждения:

- acks=0 - означает, что сообщение считается успешно записанным в Kafka, если производитель сумел отправить его по сети
- acks=1 - означает, что ведущая реплика в момент получения сообщения и записи его в файл данных партиции (но необязательно на диск) отправит подтверждение или сообщение об ошибке
- acks=all - означает, что ведущая реплика, прежде чем отправлять подтверждение или сообщение об ошибке, дождётся получения сообщения всеми согласованными репликами.

Настройка повторов отправки производителями

- Обработка ошибок на стороне производителя состоит из двух частей: автоматической обработки производителем и обработки с помощью вашей пользовательской реализации производителя;
- Коды ошибок делятся на две категории: коды ошибок, которые можно разрешить путем повтора отправки, и коды ошибок, которые разрешить нельзя;
- Если ваша цель — не терять ни одного сообщения, то лучше всего настроить производитель на повтор отправки сообщений в тех случаях, когда это имеет смысл.

Дополнительная обработка ошибок

Разработчику нужно иметь возможность обрабатывать разные типы ошибок, в том числе включающие:

- ошибки брокеров, которые нельзя разрешить путем повтора отправки, например, ошибки, связанные с размером сообщений, ошибки авторизации и т. п.;
- ошибки, произошедшие до отправки сообщения брокеру, например, ошибки сериализации;
- ошибки, связанные с тем, что производитель достиг предельного количества попыток повтора отправки или исчерпал во время этого доступную ему память на хранение сообщений.

Использование потребителей в надёжной системе

- Потребители должны фиксировать обработанные смещения;
- Потребители в основном теряют сообщения, когда фиксируют смещения для прочитанных, но ещё не полностью обработанных событий;
- Чрезвычайно важно тщательно отслеживать, когда и как фиксируются смещения;
- Зафиксированное сообщение (committed message) представляет собой сообщение, записанное во все согласованные реплики и доступное потребителям;
- Зафиксированные смещения (committed offsets) — это смещения, отправленные потребителем в Kafka в подтверждение получения и обработки ею всех сообщений в разделе вплоть до этого конкретного смещения.

Конфигурации потребителей для надёжной доставки

Существует четыре параметра конфигурации потребителей, без понимания которых не получится настроить надежное поведение:

- group.id - если необходимо, чтобы отдельный потребитель увидел каждое из сообщений топика, у него должен быть уникальный group.id
- auto.offset.reset - определяет, что потребитель будет делать, если никаких смещений не было зафиксировано или когда потребитель запросил смещения, которых нет в брокере
- enable.auto.commit – автоматическая фиксация смещений
- auto.commit.interval.ms – частота автоматической фиксации смещений

Явная фиксация смещений в потребителях

- Всегда фиксируйте смещения после обработки событий;
- Частота фиксации - компромисс между производительностью и числом дубликатов, возникающих при аварийном сбое;
- Убедитесь, что фиксируете правильные смещения;
- Перераспределение;
- Потребителям может понадобиться повторить попытку;
- Потребителям может потребоваться сохранение состояния;
- Длительная обработка записей;
- Строго однократная доставка.

Проверка надежности системы

Рекомендуется выполнять три уровня проверки:

- проверку конфигурации;
- проверку приложения;
- мониторинг приложения при промышленной эксплуатации.

Проверка конфигурации

- Kafka содержит две утилиты, предназначенные для проверки брокера и клиента независимо от логики приложения;
- Пакет org.apache.kafka.tools включает классы VerifiableProducer и VerifiableConsumer, которые можно запускать в виде утилит командной строки или встраивать во фреймворк автоматизированного тестирования;
- Необходимо подумать что стоит проверить:
  - Выбор ведущей реплики
  - Выбор контроллера
  - Плавающий перезапуск
  - Грязный» выбор ведущей реплики

Проверка приложений

- Проверка выполнения гарантий: пользовательский код обработки ошибок, фиксация смещений, перераспределение потребителей и других мест, в которых логика приложения взаимодействует с клиентскими библиотеками Kafka;
- Рекомендуется в тестах проверять следующие проблемные кейсы:
  - Потеря клиентами соединения с сервером;
  - Выбор ведущей реплики;
  - Плавающий перезапуск брокеров;
  - Плавающий перезапуск потребителей;
  - Плавающий перезапуск производителей.

Мониторинг надежности при промышленной эксплуатации

- Клиенты Kafka включают показатели JMX, позволяющие выполнять мониторинг событий и состояния клиентов;
- Два наиболее важных для производителей показателя - число ошибок и число повторов в секунду (агрегированные);
- По журналам производителей также стоит отслеживать ошибки отправки событий, помеченные как WARN, которые выглядят примерно так: «Got error produce response with correlation id 5689 on topic-partition [topic-1, 3], retrying (two attempts left). Error…. ». Вы можете или увеличить число повторов, или первым делом устранить вызывающую ошибки проблему.

Мониторинг надежности при промышленной эксплуатации

- Важнейшим показателем является задержка потребителя, показывающая, насколько он отстаёт от последнего зафиксированного в партиции на брокере сообщения;
- В идеале задержка всегда должна быть равна 0;
- Вы должны убедиться в том, что все сформированные производителями сообщения были потреблены за приемлемое время.

Спасибо за внимание