ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Big Data Analytics: Approaches and Tools

Московский городской педагогический университет

Лекция. Spark Structured Streaming

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Основные темы

- Особенности Structured Streaming
- Операции

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Введение в Structured Streaming

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Словарь Structured Streaming

- Неограниченная таблица
- Режим выводы
- Запросы
- Приемник (sink)
- Встроенные временные метки
- Watermark

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Особенности Structured Streaming

Structured Streaming обеспечивает быструю, масштабируемую, отказоустойчивую обработку потока с гарантией выполнения «только один раз» (exactly-once).

Обработка построена на движке Spark SQL и используются его механизмы оптимизации

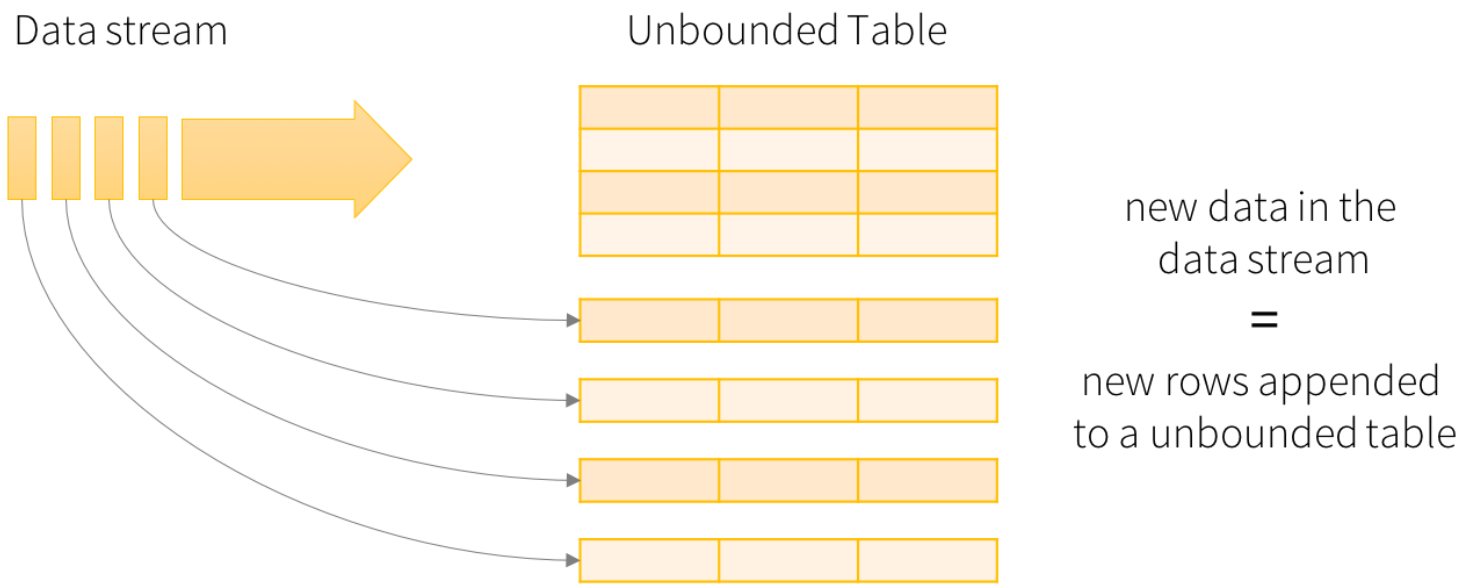
DataFrame API (Scala, Java, Python или R) используется для выполнения агрегации, операций с временными окнами, объединения и пр.

Отказоустойчивость с гарантией выполнения «только один раз» обеспечивается за счёт применения Checkpoint’ов и Write-Ahead Logs

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Модель программирования

Основная идея в Structured Streaming состоит в обработке потоков данных, представленных в виде таблиц с постоянным добавлением поступающих данных.



Data stream as an unbounded table

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Модель обработки

– Мини-пакетами (Mini-batch)

По умолчанию. Обработка потоков происходит сериями работ над небольшими пакетами данных. При этом достигается задержка на сквозную обработку до 100 миллисекунд и отказоустойчивость с гарантией выполнения «только один раз»

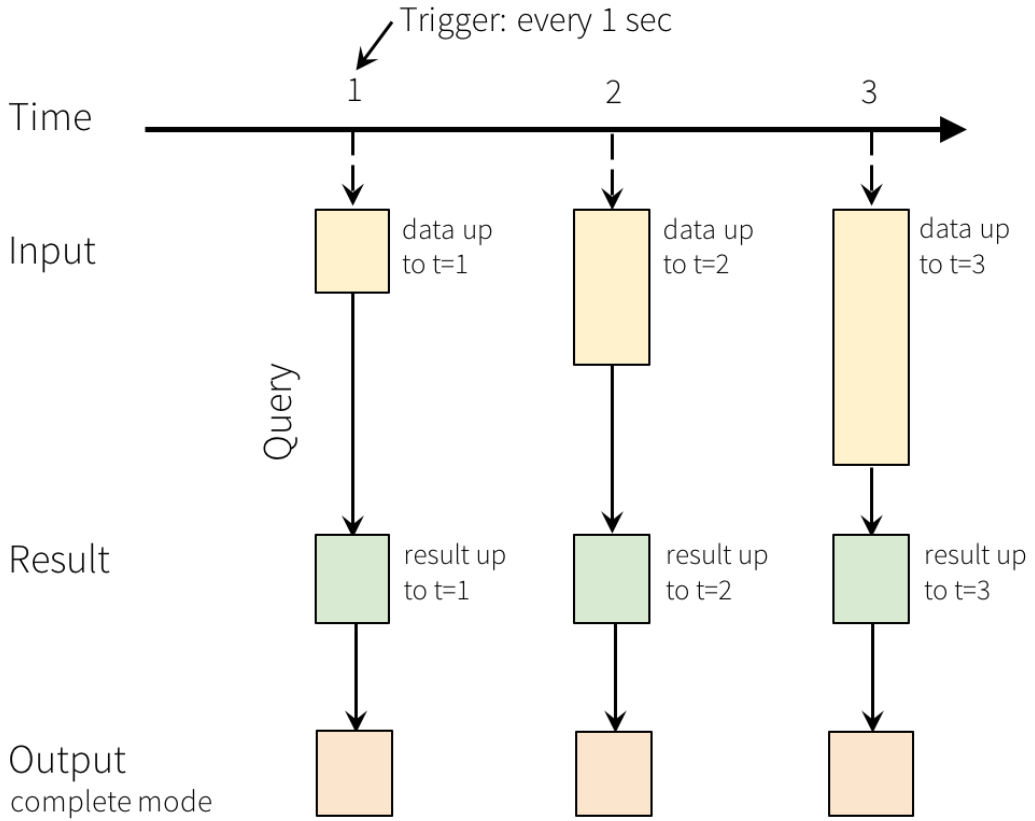
– Непрерывная (Continuous)

Экспериментальная. Начиная с Spark 2.3, появилась непрерывная обработка с низкой сквозной задержкой до 1 миллисекунды с гарантией обработки «по крайней мере один раз» (at-least-once)

В общем виде модель можно менять без изменения кода обработки (операций в запросе)

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Модель обработки. Mini-Batch



Programming Model for Structured Streaming

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Режимы вывода

Вывод (output) – это то, что записывается во внешнее хранилище

- Режим полного вывода (Complete Mode)

  Обновленная таблица полностью записывается во внешнее хранилище

- Режим с добавлением (Append Mode)

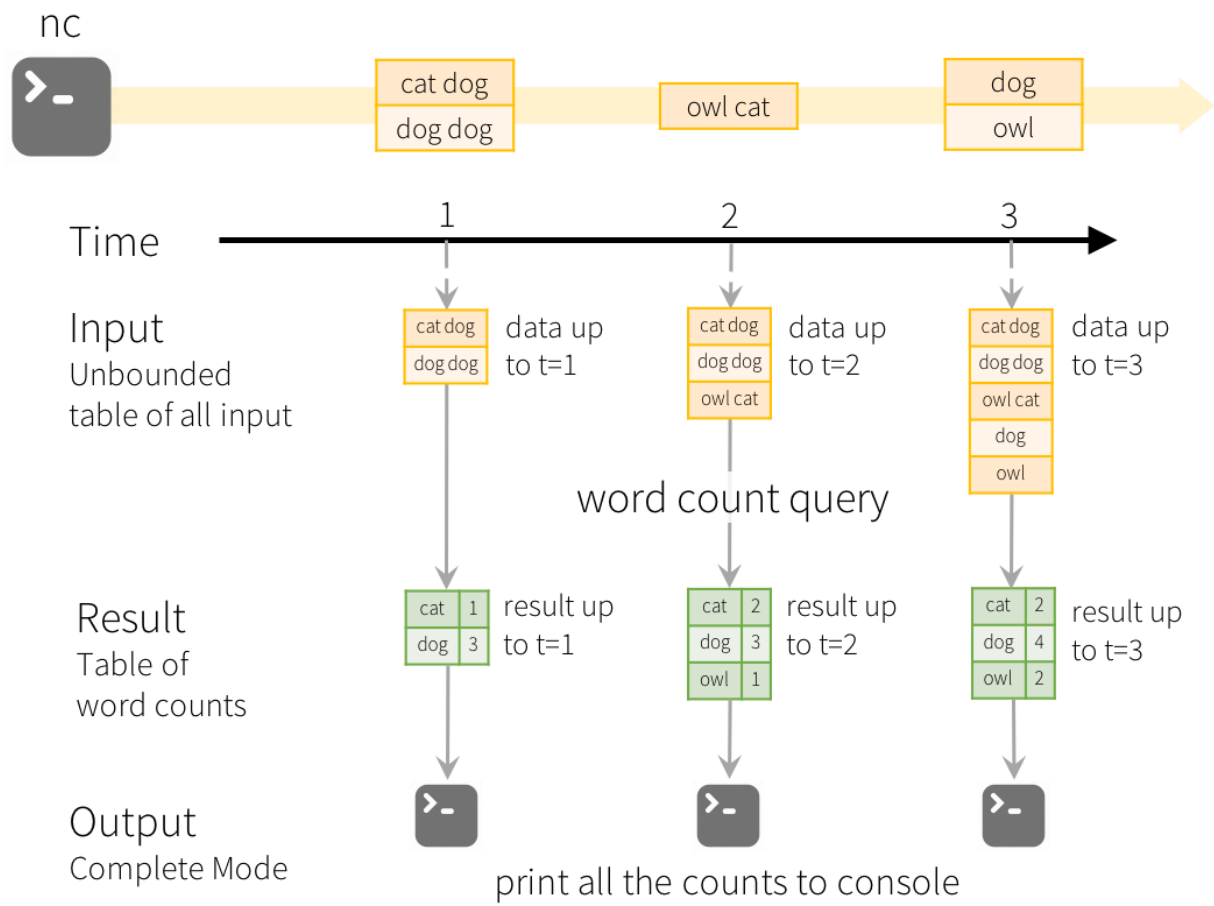
  Только новые строки, добавленные в таблицу результата (Result Table) с момента последнего срабатывания, записываются во внешнее хранилище. Данный режим применим только для запросов, в которых не ожидается изменений в существующих строках таблицы результата

- Режим с обновлением (Update Mode)

  Только строки, которые были изменены в таблице результат с момента последнего срабатывания, будут записаны во внешнее хранилище. Если запрос не содержит агрегирований, этот режим будет эквивалентен режиму с добавлением.

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Режим полного вывода



Model of the Quick Example

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Особенности неограниченной таблицы

- Structured Streaming не материализует входную таблицу целиком. Он читает последние доступные данных из источника потоковых данных (Streaming Data Source), обрабатывает их, обновляет таблицу результата и затем отбрасывает исходные данные.

- Хранятся только минимальные промежуточные данные о состояниях, которые необходимы для обновления таблицы результата (например, промежуточное количество слов).

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Учет времени событий и запоздалых данных

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Обработка событий с отметкой времени

- Время события (Event-Time) – это время, которое указано в самих данных. Его можно использовать, когда необходимо учитывать время генерации события, а не время получения их Spark’ом.

- Каждое событие, например, от какого-либо устройства, есть строка таблицы и время события (Event-Time) указывается как значение столбца в строке. Это, например, позволяет выполнять агрегации на основе окон аналогично тому, как это можно выполнить со статическим DataFrame’ом.

- Кроме того, использование встроенного в данные времени позволяет Spark’у обрабатывать события, которые поступили в Structured Streaming с опозданием.

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Семантики отказоустойчивости

Семантики отказоустойчивости

- Для достижения гарантии обработки «только один раз» источники (sources), приемники (sinks) и механизм выполнения (engine) разработаны таким образом, чтобы надежно отслеживать прогресс обработки и в случае любого рода отказа перезапускать процесс и/или повторно обрабатывать данные.

- Предполагается, что каждый источник потока имеет смещения (offsets) (похожие на смещения в Kafka) для отслеживания позиции чтение в потоке с возможностью повторного формирования/получения данных в случае отказа.

- Механизм выполнения использует систему формирования контрольных точек (checkpointing) и журналы упреждающей записи (write-ahead logs) для фиксации диапазона смещения данных, обрабатываемых при каждом срабатывании триггера.

- Приемник является идемпотентным для повторной обработки.

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Операции над потоками

DataFrame API

- Начиная с Spark 2.0, DataFrames и Datesets могут представлять статические (ограниченные данные) и потоковые (неограниченные данные)

- Посредством SparkSession можно создать потоковые DataFrame’ы из потоковых источников и применять аналогичные операции, что используются для статических DataFrame’ов – от SQL подобные операции (например, select, where, groupBy) до RDD подобных операций (например, map, filter, flatMap)

- Большинство наиболее часто используемых операций над DataFrame’ами доступны и для потоков.

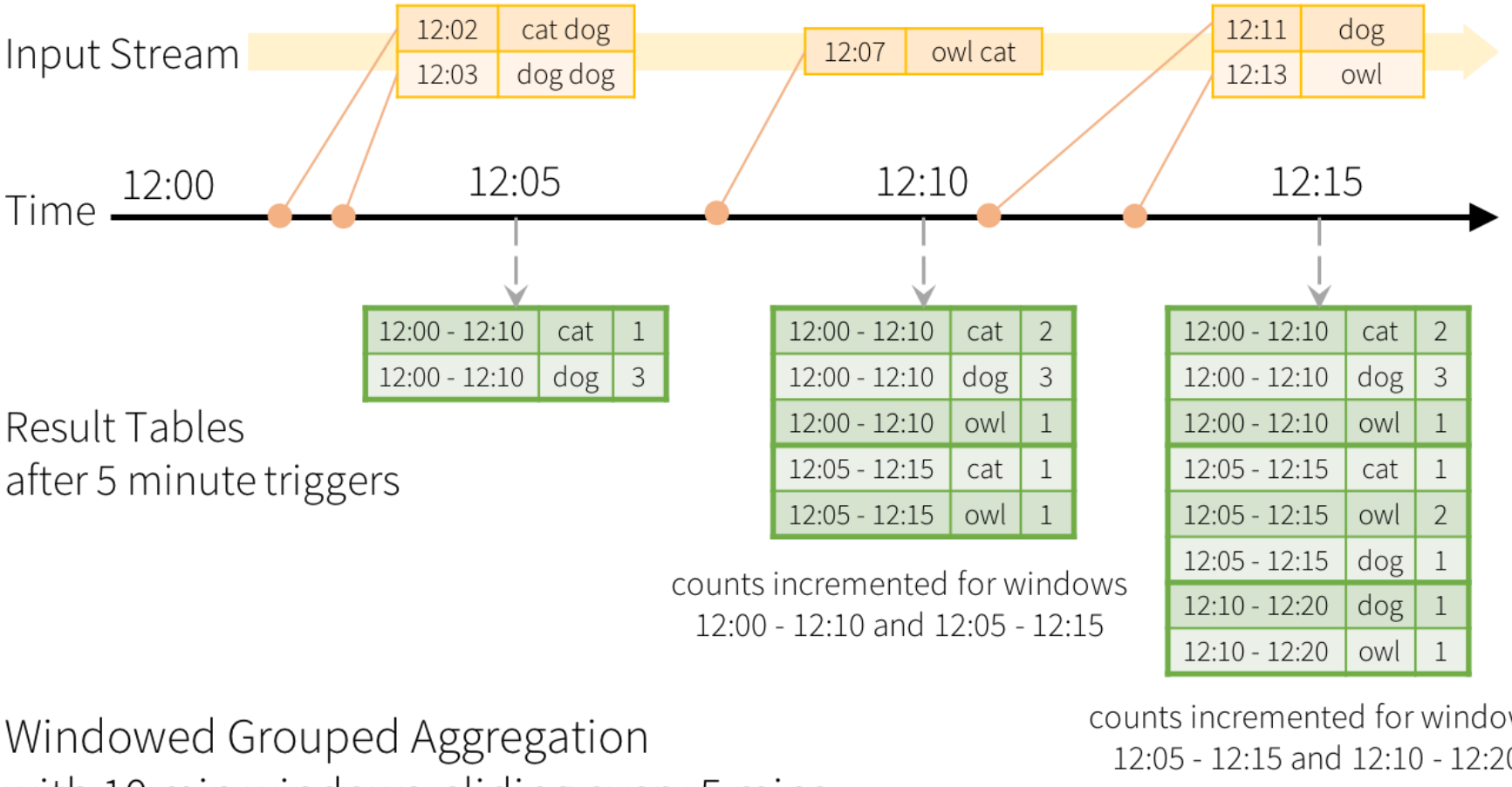
- Можно зарегистрировать потоковый DataFrame как временное представление (temporary view) и затем применять SQL команды.

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Операции с окнами

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Операции с окнами для событий с указанием времени



Windowed Grouped Aggregation
with 10 min windows, sliding every 5 mins

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

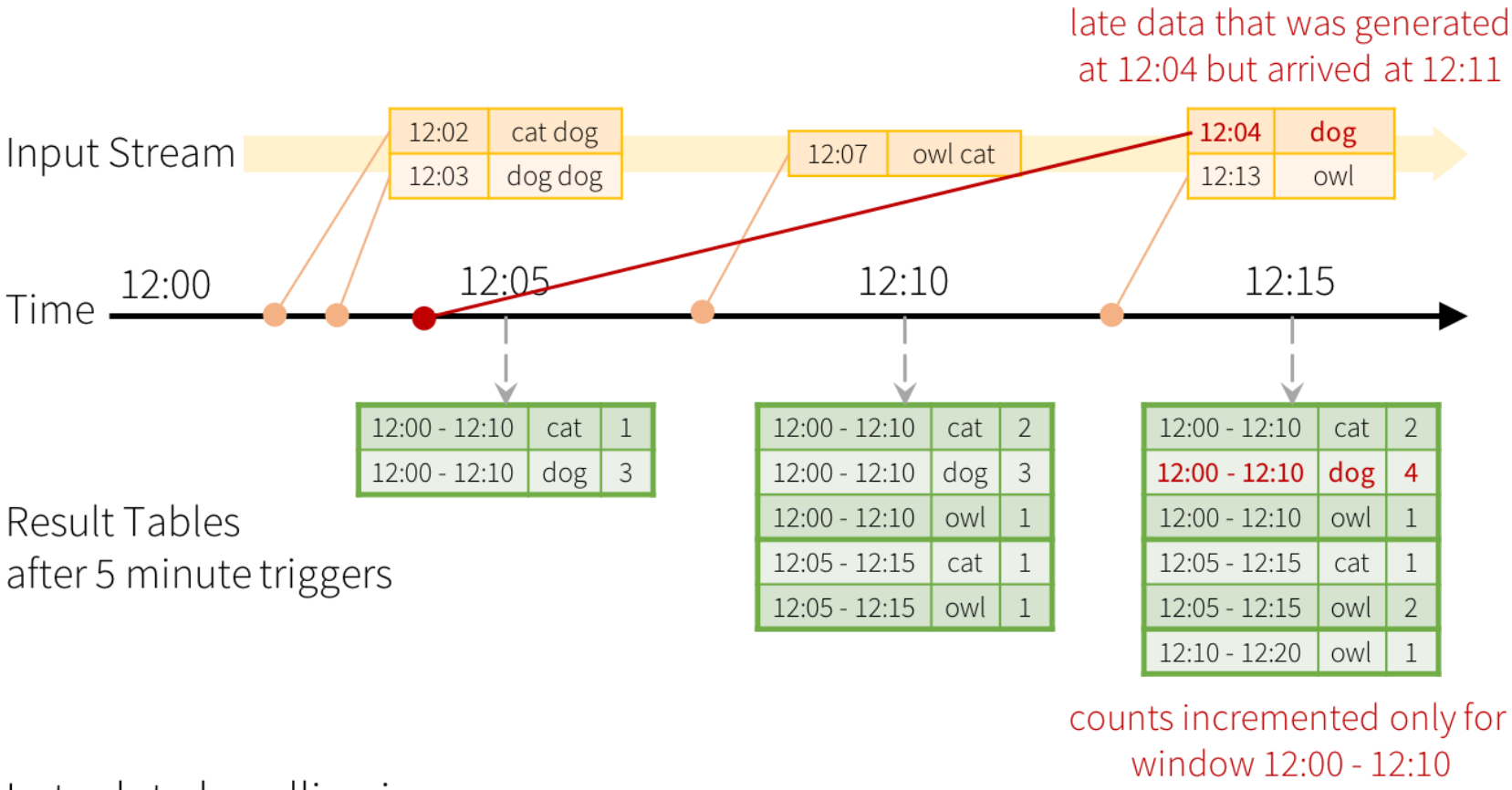
Операции с окнами для событий с указанием времени

```
words = ...  # streaming DataFrame of schema { timestamp: Timestamp, word: String }

# Group the data by window and word and compute the count of each group
windowedCounts = words.groupBy(
    window(words.timestamp, "10 minutes", "5 minutes"),
    words.word
).count()
```

Операции с окнами для запоздавших событий

Structured Streaming может поддерживать промежуточное состояние в течение длительного периода времени таким образом, что пришедшие запоздалые данные могут корректно обновить агрегаторы старых окон.



Late data handling in
Windowed Grouped Aggregation

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Операции с окнами для запоздавших событий

- Structured Streaming может поддерживать промежуточное состояние в течение длительного периода времени таким образом, что пришедшие запоздалые данные могут корректно обновить агрегаторы старых окон.

- Системе можно сообщить, как долго она может хранить старые значения агрегаторов, в случае если приложение больше не собирается получать запоздалые данные для этих агрегаторов.

- В Spark 2.1 был добавлен механизм **watermarking**, который позволяет Spark'у автоматически отслеживать текущее время события в данных и пытается очистить старые состояния.

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Операции с окнами для запоздавших событий

- **Wartermark** запроса устанавливается указав столбец времени событий и предел запаздывания на основе времени событий.

- Для конкретного окна, заканчивающегося в момент времени T, движок Spark будет поддерживать состояние и позволять запоздалым данным обновлять состояние до

  Максимальное время события, обработанное движком – предел запаздывания > T

- Другими словами, запоздалые данные в пределах интервала запаздывания будут агрегированы, но данные более поздние станут отбрасываться.

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

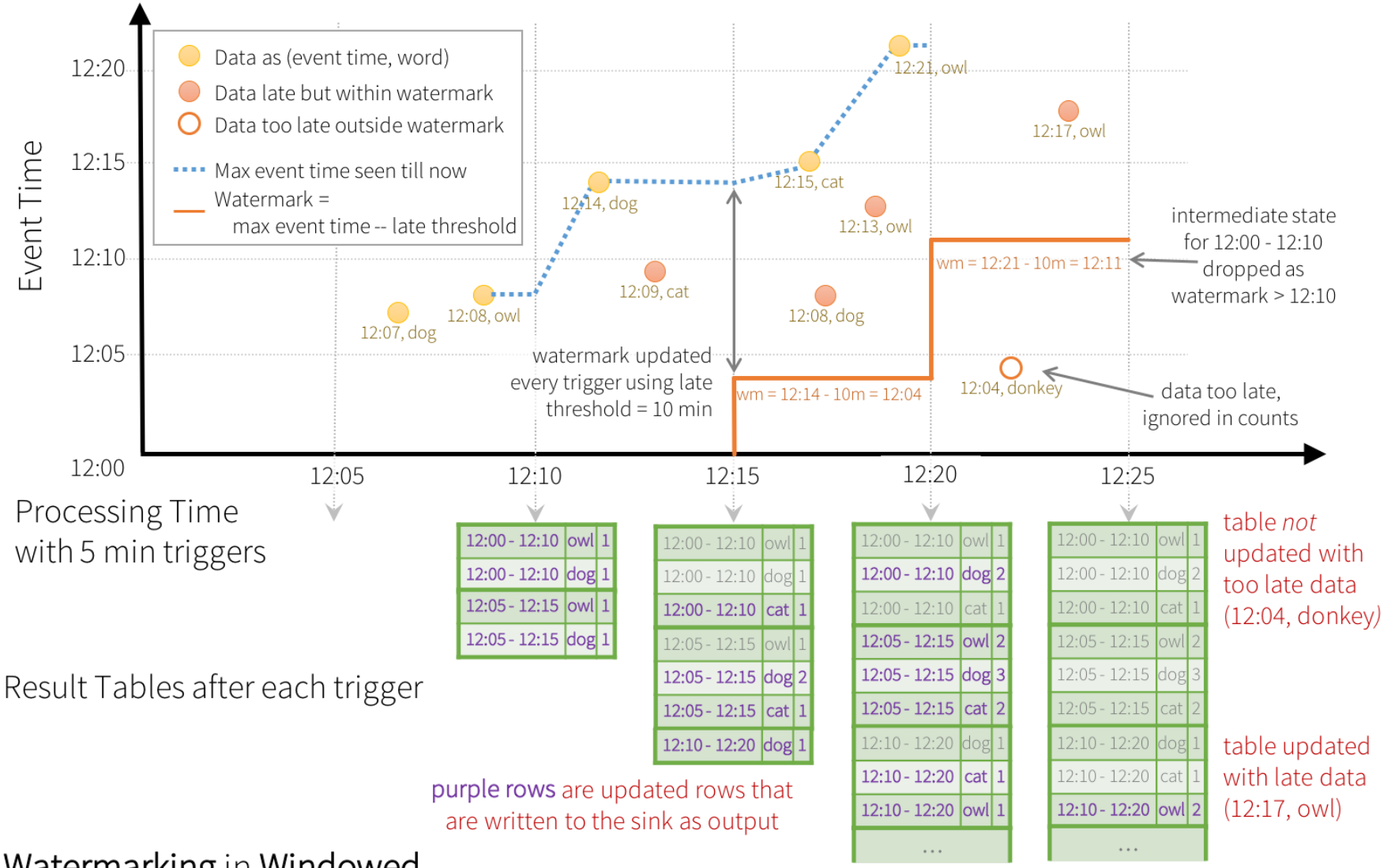
Операции с окнами для запоздавших событий

```
words = ...  # streaming DataFrame of schema { timestamp: Timestamp, word: String }

# Group the data by window and word and compute the count of each group
windowedCounts = words \
    .withWatermark("timestamp", "10 minutes") \
    .groupBy(
        window(words.timestamp, "10 minutes", "5 minutes"),
        words.word) \
    .count()
```

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

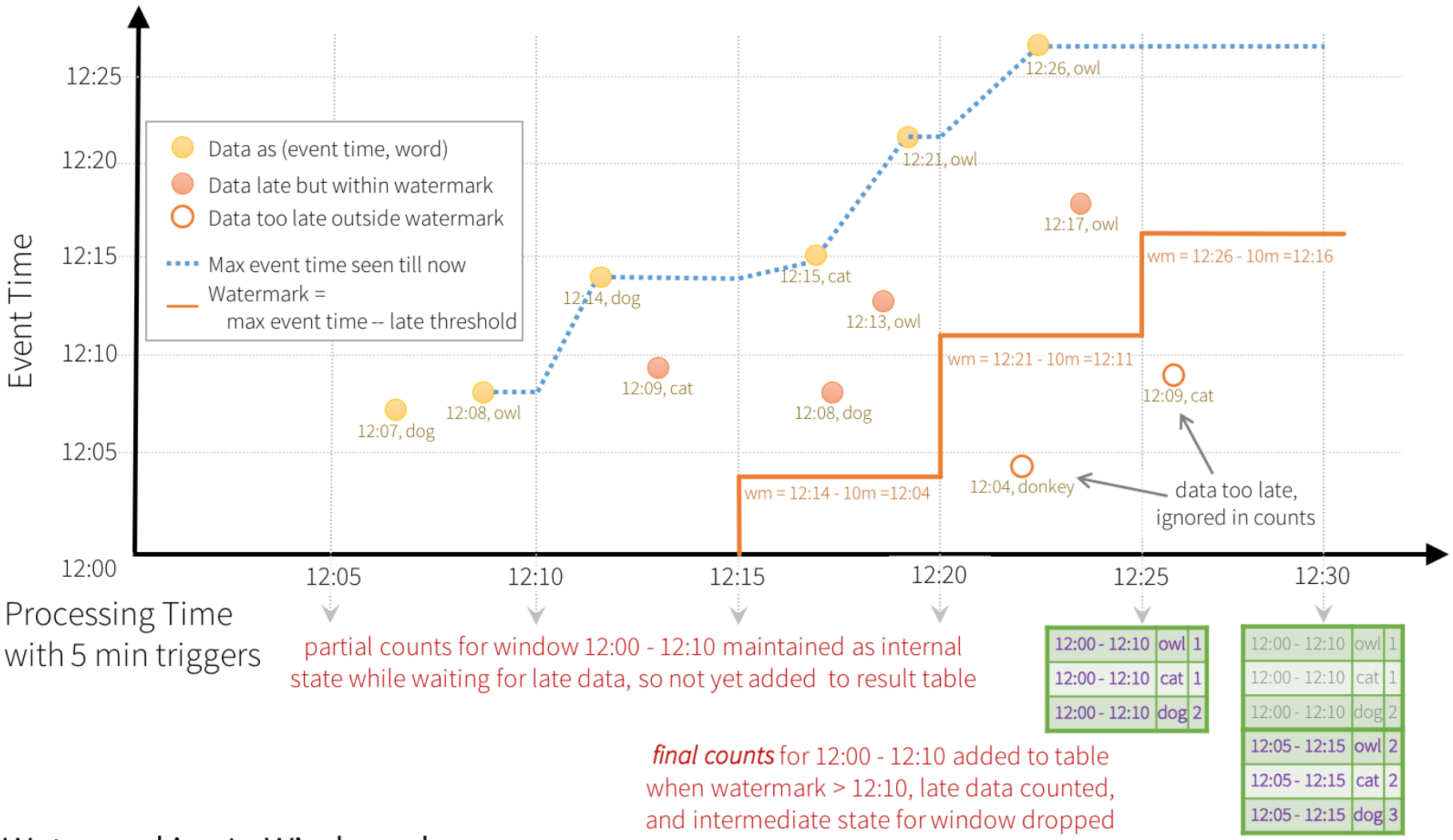
Операции с окнами. Режим с обновлением



Watermarking in **Windowed Grouped Aggregation** with **Update Mode**

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Операции с окнами. Режим с добавлением



Watermarking in **Windowed Grouped Aggregation** with **Append Mode**

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Условия использования Watermarking

- Режим выводы Append или Update.
- Для агрегирования необходим столбец времени события (event-time) или окно по столбцу времени события.
- withWatermark необходимо вызывать по тому же столбцу времени события, что и для столбца агрегации

```
df.withWatermark("time", "1 min") \
    .groupBy("time") \
    .count()
```

- withWatermark указывается до операций агрегирования.

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Условия использования Watermarking

– Интервал запаздывания, установленный посредством withWatermark, гарантирует, что Spark никогда не отбросит события, запоздавшие на время меньшее чем установленный интервал.

– Однако, гарантия строгая только в одном направлении. Данные, поступившие с задержкой более установленного значения, не гарантированно отбрасываются. Они могут быть агрегированы, а могу и нет. Чем больше задержка, тем меньше вероятность, что данные будут обработаны.

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Операция соединения (Join)

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Операции Join

- Structured Streaming поддерживает соединение (joining) потокового DataFrame’ам со статическим DataFrame’ои и с другим потоковым DataFrame’ом.

- В Spark 2.3 была добавлена поддержка соединения потоков, т.е. двух потоковых DataFrame’ов

Операции Join

- Проблемой использования join для двух потоков данных заключается в том, что в любой момент времени данные могут быть неполными с обоих сторон операции join. Любая строка, полученная от одного входного потока, может иметь сопоставление со строкой, которая ещё не поступила из другого входного потока.

- Таким образом, для обоих потоков необходимо сохранять предыдущие данные в виде состояния потока, чтобы была возможность сопоставить каждое будущее входное событие с предыдущими и, соответственно, сгенерировать результат соединения join.

- Для автоматической обработки запоздалых событий и ограничения накапливаемого состояния используются watermarks.

Операции Join

Чтобы избежать неограниченного накапливания состояния, применяют следующие подходы:

- Устанавливают watermark задержки на оба входных потока

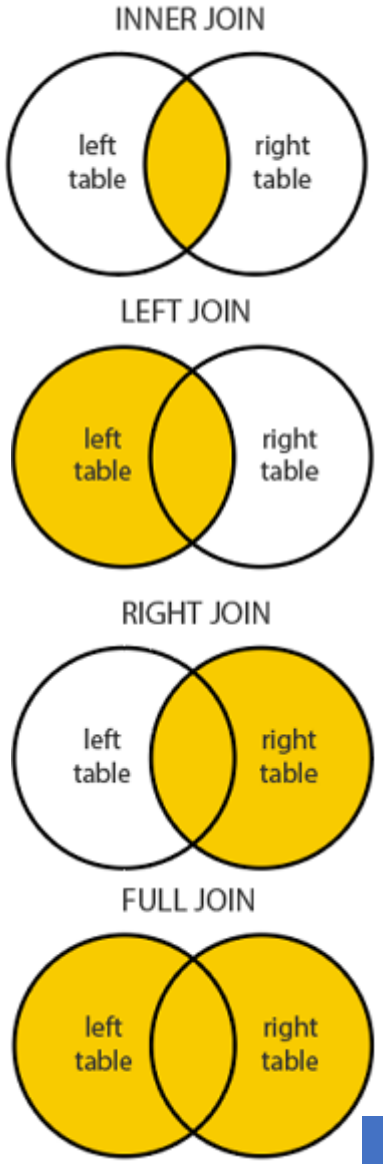
- Указывают ограничение на время событий двух потоков, чтобы Spark мог определить, когда старые строки уже не понадобятся для сопоставления с событиями второго потока. Данное ограничение может быть определено двумя способами:

  - Временной интервал для join (например, ...JOIN ON leftTime BETWEEN rightTime AND rightTime + INTERVAL 1 HOUR)

  - Join по временному окну (например, ...JOIN ON leftTimeWindow = rightTimeWindow)

Операции Join

Left Input	Right Input	Join Type	
Static	Static	All types	Supported, since its not on streaming data even though it can be present in a streaming query
Stream	Static	Inner	Supported, not stateful
		Left Outer	Supported, not stateful
		Right Outer	Not supported
		Full Outer	Not supported
Static	Stream	Inner	Supported, not stateful
		Left Outer	Not supported
		Right Outer	Supported, not stateful
		Full Outer	Not supported
Stream	Stream	Inner	Supported, optionally specify watermark on both sides + time constraints for state cleanup
		Left Outer	Conditionally supported, must specify watermark on right + time constraints for correct results, optionally specify watermark on left for all state cleanup
		Right Outer	Conditionally supported, must specify watermark on left + time constraints for correct results, optionally specify watermark on right for all state cleanup
		Full Outer	Not supported



ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Неподдерживаемые операции

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Неподдерживаемые операции

- Несколько последовательных потоковых агрегаций
- Извлечение первых N строк
- Distinct операции на потоках
- Операции сортировки поддерживаются только после агрегации и в режим полного выводы
- Некоторые типы внешнего join

ИНСТИТУТ
ЦИФРОВОГО
ОБРАЗОВАНИЯ
МГПУ

Запуск потоковых запросов

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Параметры запуска

- Источник данных

  File source, Kafka source, Socket source (for testing), Rate source (for testing)

- Приемник (формат данных, расположение и пр.)

  File Sink, Kafka Sink, Foreach Sink, ForeachBatch Sink, Console Sink, Memory Sink

- Режим вывода

  Complete mode, Append mode, Update mode

- Имя запроса (опционально)

- Интервал триггера (опционально)

- Расположение контрольных точек (checkpoint) (опционально)

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

Параметры запуска

```
spark = SparkSession. ...

# Read text from socket
socketDF = spark \
    .readStream \
    .format("socket") \
    .option("host", "localhost") \
    .option("port", 9999) \
    .load()

aggDF = socketDF.groupBy("value").count()

# Print updated aggregations to console
aggDF \
    .writeStream \
    .outputMode("complete") \
    .format("console") \
    .start()
```

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

foreachBatch и foreach

Foreach и foreachBatch операции позволяют выполнять произвольные пользовательские операции и логику записи выходного потока.

Применяются для:

– Реализации приемников (sink) под хранилища данных, которые ещё не поддерживаются Structured Streaming

– Записи в несколько локаций

– Выполнение дополнительных операций над DataFrame’ом

ИНСТИТУТ ЦИФРОВОГО ОБРАЗОВАНИЯ МГПУ

foreachBatch и foreach

- foreach для каждой записи (строки)
- foreachBatch для каждого мини-пакета (mini-batch)

```
def foreach_batch_function(df, epoch_id):
    # Transform and write batchDF
    pass

streamingDF.writeStream \
    .foreachBatch(foreach_batch_function) \
    .start()
```

Источники

Structured Streaming Programming Guide (official website)