ĐẠI HỌC
BÁCH KHOA

25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Chương 7

Xử lý luồng

Phát trực tuyến có cấu trúc trong Spark

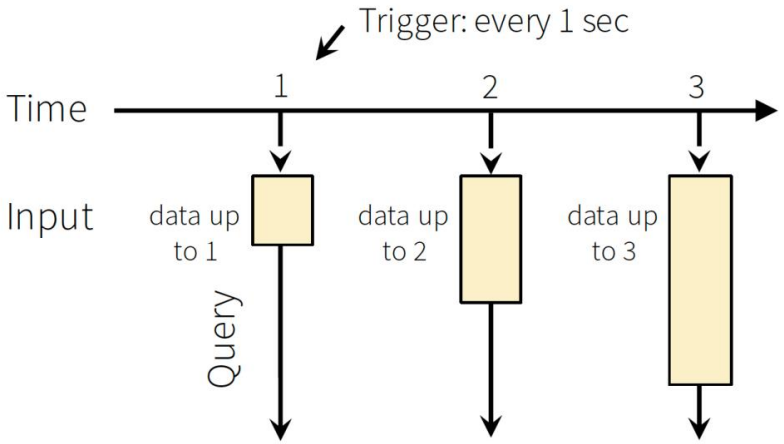
Những điểm khó khăn với DStreams

- Xử lý theo thời gian sự kiện, xử lý dữ liệu trễ
  - API DStream cho thấy thời gian xử lý hàng loạt, khó kết hợp sự kiện thời gian
- Tương tác phát trực tuyến với hàng loạt VÀ tương tác
  - RDD/DStream có API tương tự, nhưng vẫn cần phải biên dịch
- Lý luận về bảo đảm toàn diện
  - Cần phải xây dựng cẩn thận các bồn chứa để xử lý các sự cố đúng cách
  - Tính nhất quán của dữ liệu trong quá trình lưu trữ khi được cập nhật

Mô hình mới

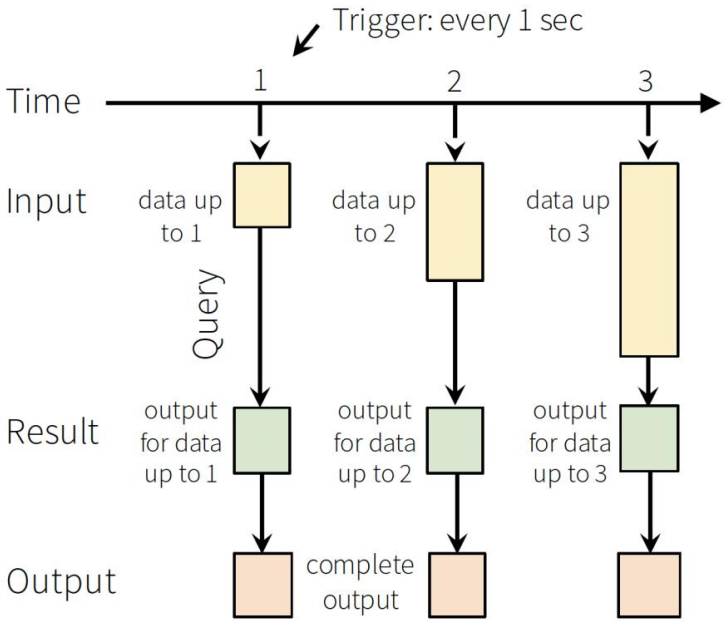
- Đầu vào: dữ liệu từ nguồn dưới dạng bảng chỉ thêm vào
- Kích hoạt: tần suất kiểm tra dữ liệu đầu vào mới • Truy

vấn: các thao tác trên đầu vào thông thường map/filter/reduce new
cửa sổ, phiên hoạt động



Mô hình mới (2)

- Kết quả: bảng hoạt động cuối cùng được cập nhật mỗi lần kích hoạt khoảng thời gian
- Đầu ra: phần nào của kết quả được ghi vào bộ thu dữ liệu sau mỗi lần kích hoạt
- Hoàn thành đầu ra: Viết bảng kết quả đầy đủ mỗi lần

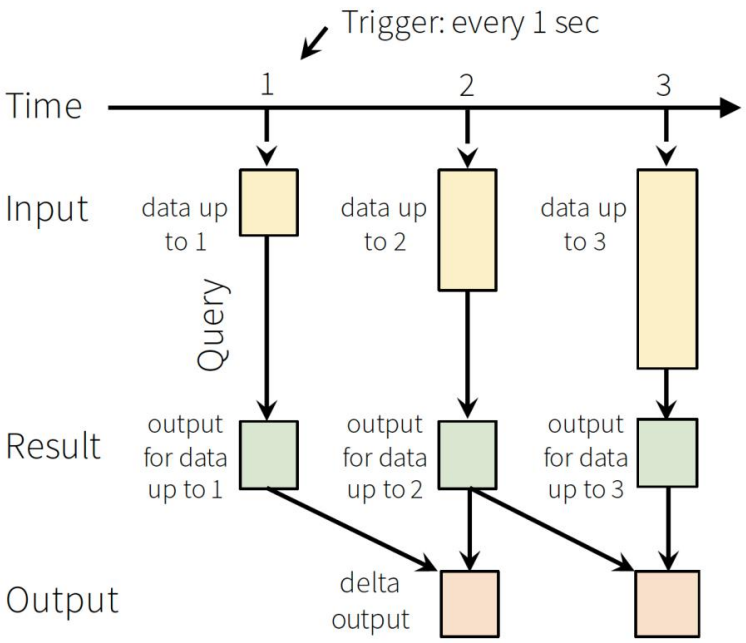


Mô hình mới (3)

- Đầu ra Delta: Chỉ ghi các hàng có kết quả thay đổi từ đợt trước

- Thêm đầu ra: Chỉ ghi các hàng mới • *Không

phải tất cả các chế độ đầu ra đều khả thi với tất cả các truy vấn



ETL hàng loạt với DataFrames

```
đầu vào = spark.read

          .định dạng("json")

          .load("đường dẫn nguồn")

kết quả = đầu vào

          .select("thiết bị",
          "tín hiệu")

          .where("tín hiệu > 15")

kết quả.viết

          .format("sàn gỗ")

          .save("đường dẫn đích")
```

- Đọc từ tệp Json
- Chọn một số thiết bị
- Ghi vào tập tin parquet

Truyền phát ETL với DataFrames

```
đầu vào = spark.read
          .định dạng("json")
          .stream("nguồn-đường
          dẫn")
kết quả = đầu vào
          .select("thiết bị",
          "tín hiệu")
          .ở đâu("tín hiệu >
15")
kết quả.viết
          .format("sàn gỗ")
          .startStream("đường dẫn
đích")
```

- Đọc từ luồng tệp Json
  - Thay thế load() bằng stream()
- Chọn một số thiết bị
  - Mã không thay đổi
- Ghi vào luồng tệp Parquet
  - Thay thế save() bằng startStream()

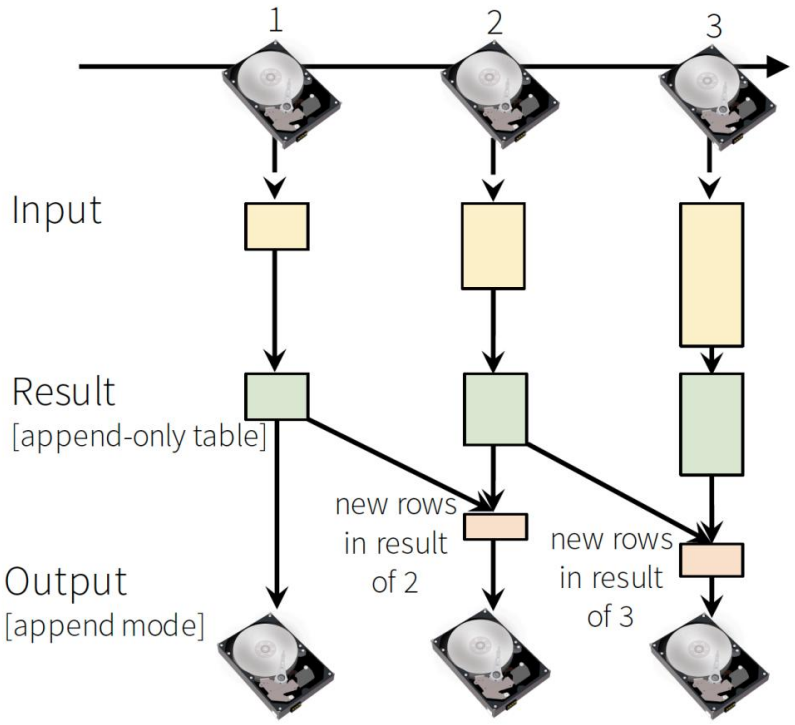
Truyền phát ETL với DataFrames

```
đầu vào = spark.read
          .định dạng("json")
          .stream("nguồn-đường
          dẫn")
kết quả = đầu vào
          .select("thiết bị",
          "tín hiệu")
          .ở đâu("tín hiệu >
15")
kết quả.viết
          .format("sàn gỗ")
          .startStream("đường dẫn
          đích")
```

- read.stream() tạo ra một DataFrame phát trực tuyến, không bắt đầu bất kỳ phép tính nào
- viết.startStream() xác định nơi và cách xuất dữ liệu và bắt đầu xử lý

Truyền phát ETL với DataFrames

```
đầu vào = spark.read
          .định dạng("json")
          .stream("nguồn-đường
          dẫn")
kết quả = đầu vào
          .select("thiết bị",
          "tín hiệu")
          .ở đâu("tín hiệu >
15")
kết quả.viết
          .format("sàn gỗ")
          .startStream("đường dẫn
          đích")
```



Tổng hợp liên tục

```
đầu vào.avg("tín hiệu")
```

- Tính toán liên tục

  tín hiệu trung bình trên tất cả các thiết bị

```
input.groupBy("kiểu thiết bị")
        .avg("tín hiệu")
```

- Liên tục tính toán tín hiệu trung bình của từng loại thiết bị

Tổng hợp cửa sổ liên tục

```
đầu vào.groupBy(
          $"loại thiết bị",
          cửa sổ($"sự kiện-thời
gian-cột", "10 phút"))
          .avg("tín hiệu")
```

- Liên tục tính toán tín hiệu trung bình của từng loại thiết bị trong 10 phút qua bằng cách sử dụng thời gian sự kiện

- Đơn giản hóa thời gian sự kiện xử lý luồng (không thể thực hiện trong DStreams)
  Hoạt động trên cả hai luồng công việc và công việc hàng loạt

Kết nối các luồng với dữ liệu tĩ nh

```
kafkaDataset = spark.đọc
        .kafka("cập nhật iot")
        .suối()

staticDataset = ctxt.đọc
        .jdbc("jdbc://", "iot-
thông tin thiết bị")

joinDataset =
        kafkaDataset. tham gia(
        staticDataset,
"kiểu thiết bị")
```

- Kết hợp dữ liệu phát trực tuyến từ Kafka với dữ liệu tĩ nh thông qua JDBC để làm giàu dữ liệu phát trực tuyến.
- . mà không cần phải nghĩ rằng bạn đang tham gia truyền dữ liệu

Chế độ đầu ra

Xác định những gì được xuất ra mỗi khi có kích hoạt Các chế độ xuất ra khác nhau có ý nghĩa đối với các truy vấn khác nhau

- Thêm chế độ với các truy vấn không tổng hợp
- Chế độ hoàn chỉnh với các truy vấn tổng hợp

```
input.select("device", "signal")
    .viết
    .outputMode("thêm") .format("sàn gỗ
đầu vào.agg(đếm("*")) .write
    .outputMode("hoàn thành"
)
    .format("sàn
gỗ") .startStream("đường dẫn đích")
```

Quản lý truy vấn

```
truy vấn = kết quả.viết
          .format("sàn gỗ")
          .outputMode("thêm vào"
)
          .startStream("đích-
con đường")


truy vấn.dừng()
truy vấn.awaitTermination()
truy vấn.ngoại lệ()


truy vấn.sourceStatuses()
truy vấn.sinkStatus()
```

- truy vấn: một xử lý cho tính toán phát trực tuyến đang chạy để quản lý nó
  - Dừng lại, chờ cho nó kết thúc
  - Nhận trạng thái
  - Nhận lỗi nếu bị chấm dứt

- Nhiều truy vấn có thể hoạt động cùng một lúc

- Mỗi truy vấn có tên duy nhất để theo dõi

Thực hiện truy vấn

- Về mặt logic
  - Các thao tác tập dữ liệu trên bảng (tức là dễ hiểu như hàng loạt)
- Về mặt vật lý
  - Spark tự động chạy truy vấn theo kiểu phát trực tuyến (tức là tăng dần và liên tục)