ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

BÁO CÁO BÀI TẬP QUÁ TRÌNH

LỚP: IS402.O22.HTCL

Analyzing IoT Data Using Azure Stream Analytics

Giảng viên: ThS. Hà Lê Hoài Trung

Trương Vĩnh Thuận - 21522653

Bùi Văn Thái - 21522577

THÀNH PHỐ HỒ CHÍ MINH, 2024

MỤC LỤC

Chương 1. **GIỚI THIỆU BÀI TOÁN**

Ngày nay, lượng dữ liệu thời gian thực khổng lồ được tạo ra bởi các ứng dụng kết nối, thiết bị và cảm biến Internet of Things (IoT), cùng với nhiều nguồn khác. Sự phát triển của các nguồn dữ liệu trực tuyến đã khiến khả năng tiêu thụ và đưa ra quyết định thông tin từ những dữ liệu này gần như ngay lập tức trở thành một yêu cầu hoạt động đối với nhiều tổ chức.

Dưới đây là một số ví dụ điển hình về khối lượng công việc xử lý dữ liệu thời gian thực:

- Cửa hàng trực tuyến phân tích dữ liệu clickstream thời gian thực để đưa ra gợi ý sản phẩm cho người tiêu dùng khi họ duyệt trang web.
- Các nhà máy sản xuất sử dụng dữ liệu từ các cảm biến IoT để theo dõi các tài sản có giá trị cao từ xa.
- Giao dịch thẻ tín dụng được kiểm tra thời gian thực để phát hiện và ngăn chặn các hoạt động có khả năng gian lận.

Azure Stream Analytics cung cấp một bộ xử lý dữ liệu thời gian thực dựa trên đám mây, chúng ta có thể sử dụng để lọc, tổng hợp và xử lý luồng dữ liệu thời gian thực từ các nguồn khác nhau.

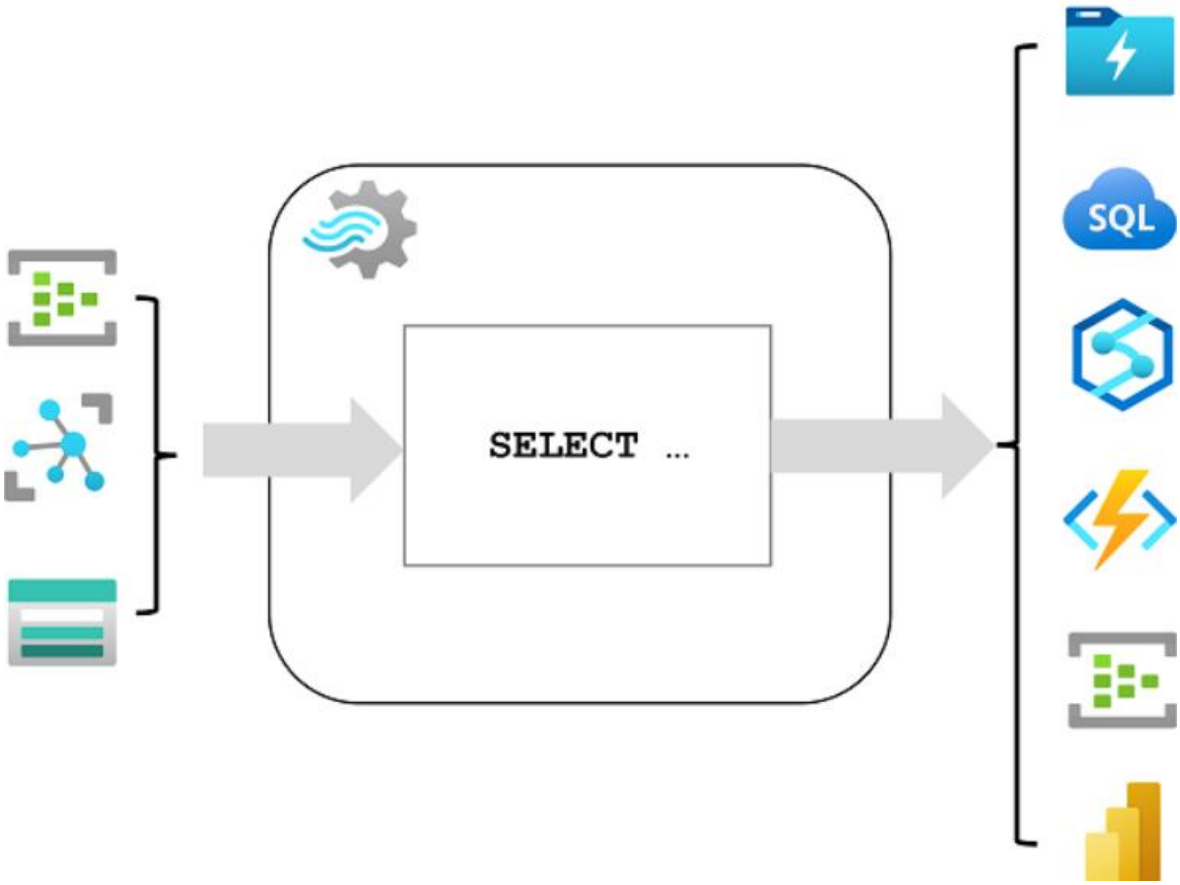
Chương 2. **CƠ SỞ LÝ THUYẾT**

2.1. Giới thiệu chung:

Azure Stream Analytics là dịch vụ xử lý sự kiện phức tạp và phân tích dữ liệu truyền phát. Bao gồm:

- Dữ liệu đầu vào được nhập từ: Azure event hub, Azure IoT Hub, or Azure Storage blob container.
- Xử lý dữ liệu bằng cách sử dụng các câu truy vấn để chọn, chiếu, hoặc tổng hợp dữ liệu.
- Kết quả dữ liệu đầu ra được ghi chẳng hạn như là Azure Data Lake Gen 2, Azure SQL Database, Azure Synapse Analytics, Azure Functions, Azure event hub, Microsoft Power BI,…

Các câu truy vấn luồng(Stream Analysts query) sẽ chạy liên tục, dữ liệu đầu vào mới sẽ được xử lý và lưu trữ kết quả đầu ra.



Một số tính năng của Azure Stream Analytics:

- Exactly Once Event Processing: Azure Stream Analytics đảm bảo rằng mỗi sự kiện chỉ được xử lý đúng một lần duy nhất, tránh việc dữ liệu bị mất hoặc bị xử lý nhiều lần. Điều này đảm bảo tính nhất quán và tin cậy của dữ liệu.
- At-Least-Once Event Delivery: trong quá trình gửi dữ liệu từ nguồn đến đích, mỗi sự kiện sẽ được gửi đi ít nhất một lần là cần thiết. Azure Stream Analytics đảm bảo khi có sự cố xảy ra trong quá trình gửi dữ liệu, sự kiện sẽ được gửi lại để đảm bảo không có sự mất mát dữ liệu.
- Recovery Capabilities: Azure Stream Analytics có khả năng tự động khôi phục và tiếp tục quá trình xử lý dữ liệu mà không mất dữ liệu khi có sự cố xảy ra.
- Checkpointing: Azure Stream Analytics sử dụng tính năng checkpointing để lưu trữ trạng thái của công việc và tiếp tục từ checkpoint gần nhất.

- Azure Stream Analytics là một platform-as-a-service(Paas) nên nó cung cấp một mô hình lập trình linh hoạt và có độ tin cậy cao, và có hiệu suất cao, bởi vì cho phép tính toán trong bộ nhớ. Sử dụng SQL-like-query-language

Có hai cách sử dụng Azure Stream Analytics,

- Sử dụng Azure Stream Analytics job trong Azure subscription, cài đặt đầu vào và đầu ra, định nghĩa các câu truy vấn cần thiết trong quá trình thực hiện. Các câu truy vấn sử dụng cấu trúc truy vấn SQL, có thể kết hợp thêm với tham chiếu tĩnh từ nhiều nguồn dữ liệu để cung cấp các giá trị tra cứu.
- Nếu dữ liệu phức tạp hoặc chuyên sâu, thì có thể sử dụng Stream Analysts cluster.

2.2. Đầu vào(Inputs)

Azure Stream Analytics có thể nhận dữ liệu đầu vào từ nhiều nguồn khác nhau như:

- Azure Event Hubs.
- Azure IoT Hub.
- Azure Blob storage.
- Azure Data Lake Storage Gen2.

Ngoài Ngoài ra thì cũng có thể định nghĩa các đầu vào tham chiếu(reference inputs) dùng để nhập dữ liệu tĩnh bổ sung dữ liệu theo thời gian thực.

2.3. Đầu ra(Outputs)

Là đích đến mà kết quả của quá trình truyền tải dữ liệu được gửi tới. Azure Stream Analytics hỗ trợ một loạt các kết quả đầu ra:

- Duy trì kết quả của quá trình Stream để có thể thực hiện phân tích thêm, bằng cách lưu chúng vào trong Data Lake, hoặc kho dữ liệu.
- Hiển thị trực quan hóa dữ liệu luồng dữ liệu theo thời gian thực, bằng cách thêm dữ liệu vào tập dữ liệu trong Power BI.

- Tạo ra các bộ lọc hay tóm tắt để có thể xử lý tiếp theo. Có thể viết kết quả vào trong Event Hub.

2.4. Truy vấn(Queries)

Logic xử lý dòng dữ liệu được đóng gói trong một truy vấn. Các truy vấn được xác định bằng các câu lệnh SQL SELECT các trường dữ liệu từ(FROM) một hoặc nhiều nguồn dữ liệu, lọc hoặc tổng hợp dữ liệu và ghi kết quả vào(INTO) một đầu ra.

2.5. Window functions

Mục tiêu phổ biến quá trình truyền dữ liệu là tổng hợp các sự kiện vào các khoảng thời gian, hoặc cửa sổ thời gian.

Azure Stream Analytics hỗ trợ nguyên bản cho năm loại hàm cửa sổ thời gian. Các hàm này cho phép xác định các khoảng thời gian mà dữ liệu được tổng hợp trong một truy vấn. Các hàm cửa sổ được hỗ trợ bao gồm Tumbling, Hopping, Sliding, Session và Snapshot.

- Tumbling: dữ liệu được chia thành các cửa sổ thời gian không chồng lên nhau và có độ dài cố định. Điều này có nghĩa là mỗi cửa sổ bắt đầu và kết thúc vào các điểm thời gian cố định và không chồng lên nhau.

- Ví dụ:

```
SELECT DateAdd(minute,-1,System.TimeStamp) AS WindowStart,
    System.TimeStamp() AS WindowEnd,
    MAX(Reading) AS MaxReading
INTO
  [output]
FROM
  [input] TIMESTAMP BY EventProcessedUtcTime
GROUP BY TumblingWindow(minute, 1)
```

Trong ví dụ trên nhằm mục đích ghi nhận lại giá trị đọc lớn nhất trong mỗi phút.

Sử dụng GROUP BY cùng với hàm TumblingWindow chỉ định kích thước của cửa sổ thời gian là 1 phút.

- Hopping: mô hình các cửa sổ chồng lấp được lập lịch, nhảy chuyển tiếp trong thời gian theo một khoảng cố định. Để tạo ra 1 cửa sổ nhảy thì cần 3 tham số bắt buộc là đơn vị thời gian, kích thước cửa sổ, kích thước bước nhảy.
  - Ví dụ:

```
SELECT DateAdd(second,-60,System.TimeStamp) AS
WindowStart,
    System.TimeStamp() AS WindowEnd,
    MAX(Reading) AS MaxReading
INTO
  [output]
FROM
  [input] TIMESTAMP BY EventProcessedUtcTime
GROUP BY HoppingWindow(second, 60, 30)
```

  Trong ví dụ trên sử dụng hàm HoppingWindow() với đơn vị thời gian là giây, kích thước cửa sổ là 60 giây, bước nhảy là 30 giây.

- Sliding: dữ liệu sẽ được chia thành các cửa sổ thời gian có độ dài cố định. Mỗi khi có nội dung thay đổi thì một sự kiện mới sẽ được tạo ra. Tóm lại, mỗi cửa sổ sẽ chứa ít nhất một sự kiện và các sự kiến có thể thuộc về nhiều hơn một cửa sổ. Làm giảm thiểu số lượng cửa sổ cần xem xét, giúp tăng hiệu suất của hệ thống

- Ví dụ:

```
SELECT DateAdd(minute,-1,System.TimeStamp) AS WindowStart,
    System.TimeStamp() AS WindowEnd,
    MAX(Reading) AS MaxReading
INTO
  [output]
FROM
  [input] TIMESTAMP BY EventProcessedUtcTime
GROUP BY SlidingWindow(minute, 1)
```

Trong ví dụ trên để có thể sử dụng Sliding ta có thể sử dụng hàm SlidingWindow ở GROUPBY.

- Session: gom nhóm các sự kiện lại gần với nhau, loại bỏ các khoảng thời gian không có dữ liệu. Có hai tham số chính đó là thời gian chờ và thời lượng tối đa.
  - Ví dụ:

```
SELECT DateAdd(second,-60,System.TimeStamp) AS
WindowStart,
    System.TimeStamp() AS WindowEnd,
    MAX(Reading) AS MaxReading
INTO
  [output]
FROM
  [input] TIMESTAMP BY EventProcessedUtcTime
GROUP BY SessionWindow(second, 20, 60)
```

Ở ví dụ ta cài đặt thời gian chờ là 20 giây và thời gian tối đa là 60 giây.

- Snapshot: nhóm các sự kiện theo các giá trị timestamp giống nhau.
  - Ví dụ:

```
SELECT System.TimeStamp() AS WindowTime,
    MAX(Reading) AS MaxReading
INTO
  [output]
FROM
  [input] TIMESTAMP BY EventProcessedUtcTime
GROUP BY System.Timestamp()
```

Chương 3. **MÔ HÌNH DỮ LIỆU**

3.1. Định nghĩa câu truy vấn

3.1.1. SELECT INTO

Cách đơn giản nhất để nhập dữ liệu đang phát vào Azure Synapse Analytics là ghi lại các tên cột cần thiết cho mỗi sự kiện bằng một truy vấn SELECT...INTO:

```
SELECT
    EventEnqueuedUtcTime AS ReadingTime,
    SensorID,
    ReadingValue
INTO
    [synapse-output]
FROM
    [streaming-input] TIMESTAMP BY EventEnqueuedUtcTime
```

3.1.2. Lọc dữ liệu (WHERE)

Trong một số trường hợp, bạn có thể muốn lọc dữ liệu để chỉ bao gồm các sự kiện cụ thể bằng cách thêm một mệnh đề WHERE. Ví dụ, truy vấn sau đây ghi dữ liệu chỉ cho các sự kiện có giá trị trường ReadingValue là âm.

```
SELECT
    EventEnqueuedUtcTime AS ReadingTime,
    SensorID,
    ReadingValue
INTO
    [synapse-output]
```

```
FROM
    [streaming-input] TIMESTAMP BY EventEnqueuedUtcTime
WHERE ReadingValue < 0
```

3.1.3. Tổng hợp qua các cửa sổ thời gian

Thông thường các truy vấn dữ liệu đang phát tổng hợp dữ liệu qua các khoảng thời gian, hoặc cửa sổ thời gian. Để thực hiện điều này, chúng ta có thể sử dụng một mệnh đề GROUP BY kèm theo một hàm Window định nghĩa loại cửa sổ mà mình muốn xác định (ví dụ, tumbling, hopping, hoặc sliding).

Ví dụ sau nhóm các cảm biến đang phát vào cửa sổ tumbling (tuần tự, không chồng chéo) 1 phút, ghi lại thời gian bắt đầu và kết thúc của mỗi cửa sổ và giá trị tối đa cho mỗi cảm biến. Mệnh đề HAVING lọc kết quả để chỉ bao gồm các cửa sổ nơi ít nhất một sự kiện đã xảy ra.

```
SELECT
    DateAdd(second, -60, System.TimeStamp) AS StartTime,
    System.TimeStamp AS EndTime,
    SensorID,
    MAX(ReadingValue) AS MaxReading
INTO
    [synapse-output]
FROM
    [streaming-input] TIMESTAMP BY EventEnqueuedUtcTime
GROUP BY SensorID, TumblingWindow(second, 60)
HAVING COUNT(*) >= 1
```

3.2. Quản lý hiệu suất

3.2.1. Streaming units

Azure sử dụng các đơn vị xử lý dòng (SUs) để kiểm soát số lượng tài nguyên xử lý có sẵn để xử lý nhập, biến đổi và đầu ra. Các công việc ASA tính phí dựa trên số lượng SUs được chọn. Bắt đầu với một SU duy nhất giảm thiểu chi phí hàng giờ cho Stream Analytics. Không phải tất cả các công việc có thể sử dụng nhiều hơn một SU.

Stream Analytics sử dụng phân vùng trong các đầu vào và đầu ra để chia dữ liệu dòng. Sự chia này cho phép xử lý song song của các bước đầu vào, đầu ra và biến đổi. Mỗi bước biến đổi trong truy vấn công việc phải sử dụng PARTITION BY để tận dụng việc phân vùng. Sau đó, việc xử lý được phân phối trên các SUs được cấp.

3.2.2. Event odering

ASA hỗ trợ xử lý micro-batch trên dữ liệu phụ thuộc vào thời gian. Dữ liệu phụ thuộc vào thời gian chứa một hoặc nhiều dấu thời gian. Ở mức cơ bản, một dấu thời gian đơn lẻ chứa ngày và giờ. Các ứng dụng thông thường của dấu thời gian bao gồm thời gian tạo thông báo và thời gian nộp. Mặc dù thời gian tạo và thời gian nộp nên gần như giống nhau, đôi khi thời gian nộp trễ so với thời gian tạo. Trong trường hợp này, cách tốt nhất là lặp lại việc gửi thông báo cho đến khi việc nộp thành công. Trong tình huống này, các dấu thời gian tạo và nộp có thể khác biệt đáng kể. Hình 6.11 cho thấy cách phân công thời gian có thể biến đổi giữa các bước xử lý.

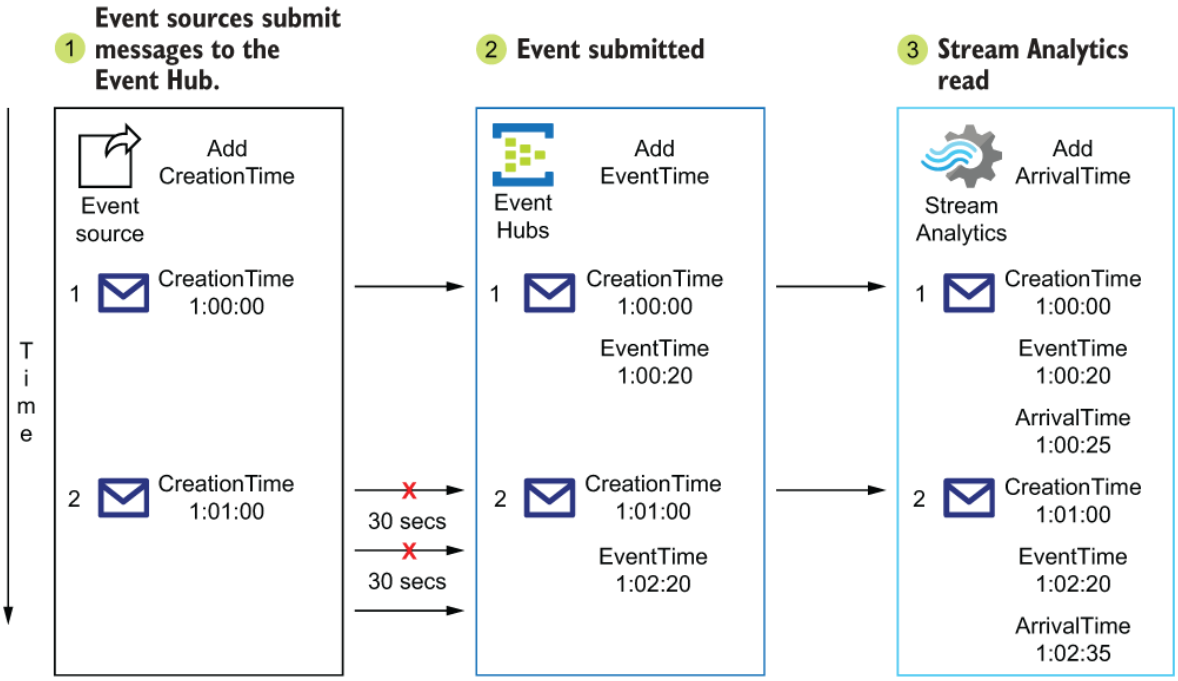


Figure **Event time assignment in data stream**

Chương 4. **DEMO**