**Understand batch and stream processing**

Completed100 XP

* 9 minutes

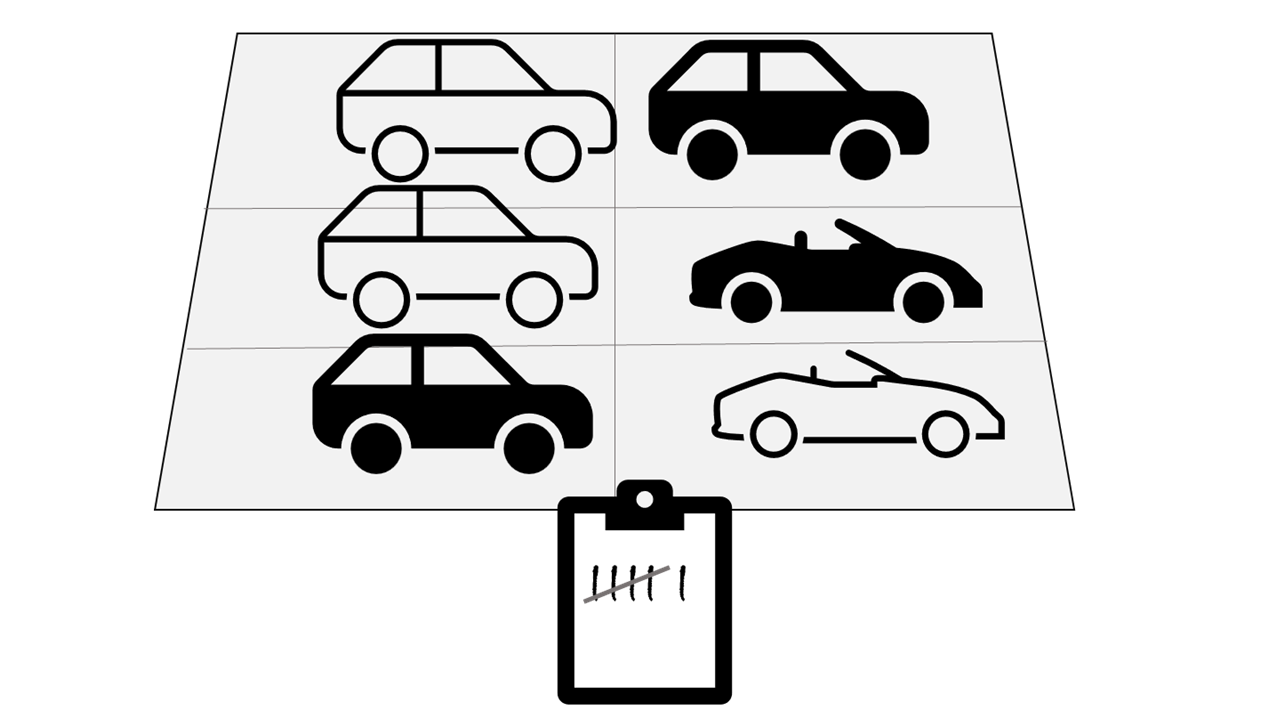
Data processing is simply the conversion of raw data to meaningful information through a process. There are two general ways to process data:

* *Batch processing*, in which multiple data records are collected and stored before being processed together in a single operation.
* *Stream processing*, in which a source of data is constantly monitored and processed in real time as new data events occur.

**Understand batch processing**

In batch processing, newly arriving data elements are collected and stored, and the whole group is processed together as a batch. Exactly when each group is processed can be determined in a number of ways. For example, you can process data based on a scheduled time interval (for example, every hour), or it could be triggered when a certain amount of data has arrived, or as the result of some other event.

For example, suppose you want to analyze road traffic by counting the number of cars on a stretch of road. A batch processing approach to this would require that you collect the cars in a parking lot, and then count them in a single operation while they're at rest.



If the road is busy, with a large number of cars driving along at frequent intervals, this approach may be impractical; and note that you don't get any results until you have parked a batch of cars and counted them.

A real world example of batch processing is the way that credit card companies handle billing. The customer doesn't receive a bill for each separate credit card purchase but one monthly bill for all of that month's purchases.

Advantages of batch processing include:

* Large volumes of data can be processed at a convenient time.
* It can be scheduled to run at a time when computers or systems might otherwise be idle, such as overnight, or during off-peak hours.

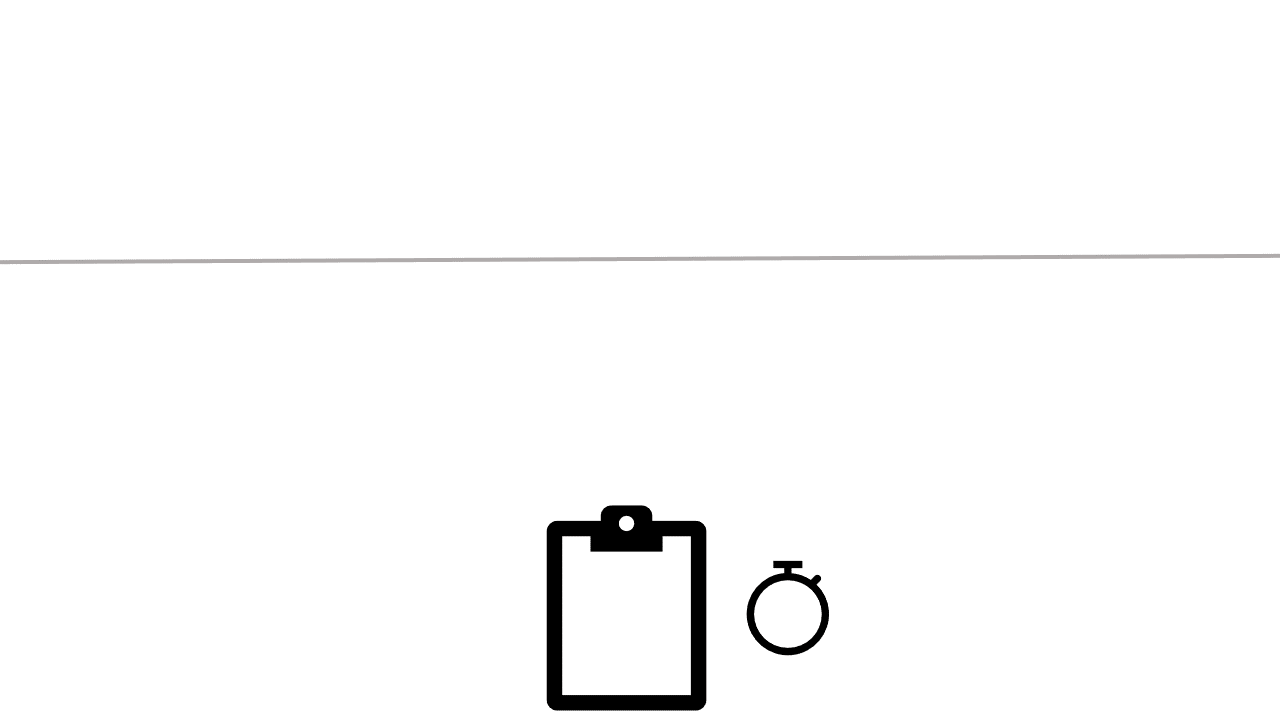
Disadvantages of batch processing include:

* The time delay between ingesting the data and getting the results.
* All of a batch job's input data must be ready before a batch can be processed. This means data must be carefully checked. Problems with data, errors, and program crashes that occur during batch jobs bring the whole process to a halt. The input data must be carefully checked before the job can be run again. Even minor data errors can prevent a batch job from running.

**Understand stream processing**

In stream processing, each new piece of data is processed when it arrives. Unlike batch processing, there's no waiting until the next batch processing interval - data is processed as individual units in real-time rather than being processed a batch at a time. Stream data processing is beneficial in scenarios where new, dynamic data is generated on a continual basis.

For example, a better approach to our hypothetical car counting problem might be to apply a *streaming* approach, by counting the cars in real-time as they pass:



In this approach, you don't need to wait until all of the cars have parked to start processing them, and you can aggregate the data over time intervals; for example, by counting the number of cars that pass each minute.

Real world examples of streaming data include:

* A financial institution tracks changes in the stock market in real time, computes value-at-risk, and automatically rebalances portfolios based on stock price movements.
* An online gaming company collects real-time data about player-game interactions, and feeds the data into its gaming platform. It then analyzes the data in real time, offers incentives and dynamic experiences to engage its players.
* A real-estate website that tracks a subset of data from mobile devices, and makes real-time property recommendations of properties to visit based on their geo-location.

Stream processing is ideal for time-critical operations that require an instant real-time response. For example, a system that monitors a building for smoke and heat needs to trigger alarms and unlock doors to allow residents to escape immediately in the event of a fire.

**Understand differences between batch and streaming data**

Apart from the way in which batch processing and streaming processing handle data, there are other differences:

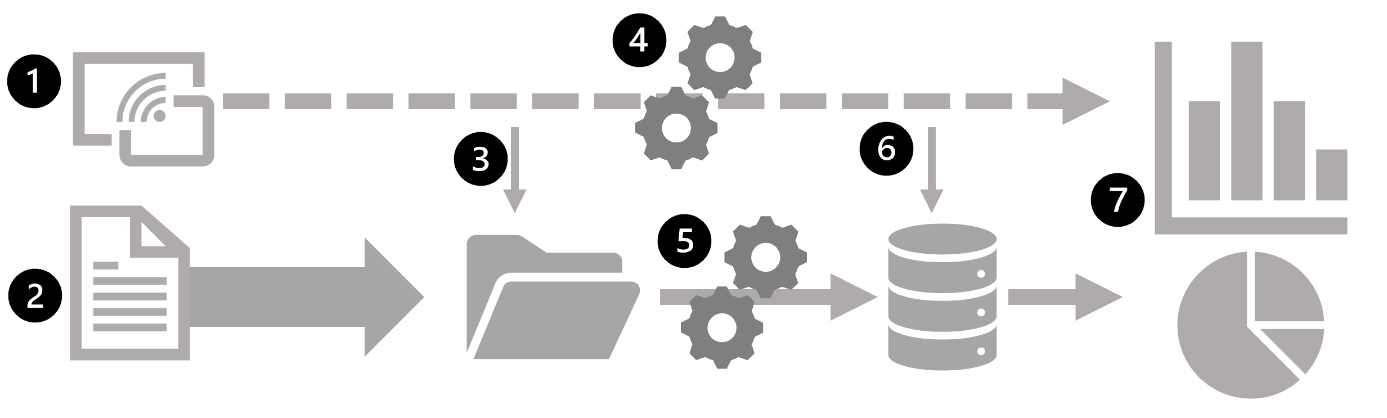
* *Data scope*: Batch processing can process all the data in the dataset. Stream processing typically only has access to the most recent data received, or within a rolling time window (the last 30 seconds, for example).
* *Data size*: Batch processing is suitable for handling large datasets efficiently. Stream processing is intended for individual records or *micro batches* consisting of few records.
* *Performance*: *Latency* is the time taken for the data to be received and processed. The latency for batch processing is typically a few hours. Stream processing typically occurs immediately, with latency in the order of seconds or milliseconds.
* *Analysis*: You typically use batch processing to perform complex analytics. Stream processing is used for simple response functions, aggregates, or calculations such as rolling averages.

**Combine batch and stream processing**

Many large-scale analytics solutions include a mix of batch and stream processing, enabling both historical and real-time data analysis. It's common for stream processing solutions to capture real-time data, process it by filtering or aggregating it, and present it through real-time dashboards and visualizations (for example, showing the running total of cars that have passed along a road within the current hour), while also persisting the processed results in a data store for historical analysis alongside batch processed data (for example, to enable analysis of traffic volumes over the past year).

Even when real-time analysis or visualization of data is not required, streaming technologies are often used to capture real-time data and store it in a data store for subsequent batch processing (this is the equivalent of redirecting all of the cars that travel along a road into a parking lot before counting them).

The following diagram shows some ways in which batch and stream processing can be combined in a large-scale data analytics architecture.



1. Data events from a streaming data source are captured in real-time.
2. Data from other sources is ingested into a data store (often a *data lake*) for batch processing.
3. If real-time analytics is not required, the captured streaming data is written to the data store for subsequent batch processing.
4. When real-time analytics is required, a stream processing technology is used to prepare the streaming data for real-time analysis or visualization; often by filtering or aggregating the data over temporal windows.
5. The non-streaming data is periodically batch processed to prepare it for analysis, and the results are persisted in an analytical data store (often referred to as a *data warehouse*) for historical analysis.
6. The results of stream processing may also be persisted in the analytical data store to support historical analysis.
7. Analytical and visualization tools are used to present and explore the real-time and historical data.

**Note**

Commonly used solution architectures for combined batch and stream data processing include *lambda* and *delta* architectures. Details of these architectures are beyond the scope of this course, but they incorporate technologies for both large-scale batch data processing and real-time stream processing to create an end-to-end analytical solution.

**Hiểu về xử lý dữ liệu theo lô và theo luồng**

Xử lý dữ liệu đơn giản là quá trình chuyển đổi dữ liệu thô thành thông tin có ý nghĩa. Có hai cách chính để xử lý dữ liệu:

* Xử lý theo lô (batch processing): Thu thập và lưu trữ nhiều bản ghi dữ liệu trước khi xử lý chúng cùng nhau trong một hoạt động duy nhất.
* Xử lý theo luồng (stream processing): Liên tục theo dõi và xử lý nguồn dữ liệu theo thời gian thực khi các sự kiện dữ liệu mới xảy ra.

**Hiểu về xử lý theo lô**

Trong xử lý theo lô, các phần tử dữ liệu mới đến được thu thập và lưu trữ, và toàn bộ nhóm được xử lý cùng nhau thành một lô. Thời điểm chính xác từng nhóm được xử lý có thể được xác định theo một số cách. Ví dụ, bạn có thể xử lý dữ liệu dựa trên khoảng thời gian theo lịch (ví dụ: mỗi giờ), hoặc nó có thể được kích hoạt khi một lượng dữ liệu nhất định đến hoặc do kết quả của một số sự kiện khác.

Ví dụ, giả sử bạn muốn phân tích lưu lượng đường bộ bằng cách đếm số xe trên một đoạn đường. Cách tiếp cận xử lý theo lô đối với việc này sẽ yêu cầu bạn tập hợp các xe trong bãi đậu xe,然后在它们静止时一次性计数它们。

Nếu đường đông đúc, với một lượng lớn xe chạy dọc theo các khoảng thời gian thường xuyên, thì cách tiếp cận này có thể không thực tế; và lưu ý rằng bạn không nhận được bất kỳ kết quả nào cho đến khi bạn đỗ một lô xe và đếm chúng.

Một ví dụ thực tế về xử lý theo lô là cách các công ty thẻ tín dụng xử lý hóa đơn. Khách hàng không nhận được hóa đơn cho từng giao dịch mua thẻ tín dụng riêng lẻ mà là một hóa đơn tháng cho tất cả các giao dịch mua trong tháng đó.

**Ưu điểm của xử lý theo lô:**

* Có thể xử lý khối lượng lớn dữ liệu vào thời điểm thuận tiện.
* Nó có thể được lên lịch chạy vào thời điểm máy tính hoặc hệ thống có thể nhàn rỗi, chẳng hạn như qua đêm hoặc trong giờ ngoài cao điểm.

**Nhược điểm của xử lý theo lô:**

* Khoảng thời gian trễ giữa việc nhập dữ liệu và nhận kết quả.
* Tất cả dữ liệu đầu vào của作业批处理 (batch job) phải sẵn sàng trước khi có thể xử lý một lô. Điều này có nghĩa là dữ liệu phải được kiểm tra kỹ lưỡng. Các vấn đề về dữ liệu, lỗi và sự cố chương trình xảy ra trong các作业批处理 sẽ khiến toàn bộ quá trình dừng lại. Dữ liệu đầu vào phải được kiểm tra kỹ lưỡng trước khi có thể chạy lại作业. Ngay cả lỗi dữ liệu nhỏ cũng có thể ngăn chặn việc chạy作业批处理.

**Hiểu về xử lý theo luồng**

Trong xử lý theo luồng, mỗi phần dữ liệu mới được xử lý khi nó đến. Không giống như xử lý theo lô, không cần phải đợi đến khoảng thời gian xử lý theo lô tiếp theo - dữ liệu được xử lý thành các đơn vị riêng lẻ theo thời gian thực thay vì được xử lý theo lô. Xử lý dữ liệu theo luồng có lợi trong các tình huống mà dữ liệu mới, động được tạo ra liên tục.

Ví dụ, một cách tiếp cận tốt hơn cho bài toán đếm xe giả định của chúng tôi có thể là áp dụng phương pháp xử lý theo luồng, bằng cách đếm xe theo thời gian thực khi chúng đi qua:

Trong cách tiếp cận này, bạn không cần phải đợi cho đến khi tất cả xe đã đỗ để bắt đầu xử lý chúng và bạn có thể tổng hợp dữ liệu theo các khoảng thời gian; ví dụ, bằng cách đếm số xe đi qua mỗi phút.

Các ví dụ thực tế về dữ liệu theo luồng bao gồm:

* Một tổ chức tài chính theo dõi sự thay đổi của thị trường chứng khoán theo thời gian thực, tính toán giá trị theo rủi ro và tự động tái cân bằng danh mục đầu tư dựa trên biến động giá cổ phiếu.
* Một công ty game online thu thập dữ liệu theo thời gian thực về tương tác giữa người chơi và trò chơi, và cung cấp dữ liệu vào nền tảng chơi game của mình. Sau đó, nó phân tích dữ liệu theo thời gian thực, cung cấp các ưu đãi và trải nghiệm năng động để thu hút người chơi.
* Một trang web bất động sản theo dõi một tập hợp con dữ liệu từ các thiết bị di động và đưa ra các đề xuất bất động sản theo thời gian thực về các bất động sản cần đến thăm dựa trên vị trí địa lý của họ.

**Hiểu về những khác biệt giữa xử lý dữ liệu theo lô và theo luồng**

Ngoài cách xử lý dữ liệu của xử lý theo lô và xử lý theo luồng, còn có những khác biệt khác:

* **Phạm vi dữ liệu:** Xử lý theo lô có thể xử lý tất cả dữ liệu trong tập dữ liệu. Xử lý theo luồng thường chỉ truy cập được vào dữ liệu nhận được gần đây nhất hoặc trong một cửa sổ thời gian di động (ví dụ: 30 giây trước).
* **Kích thước dữ liệu:** Xử lý theo lô phù hợp để xử lý các tập dữ liệu lớn một cách hiệu quả. Xử lý theo luồng dành cho các bản ghi riêng lẻ hoặc các **gói nhỏ (micro batches)** bao gồm một vài bản ghi.
* **Hiệu suất:** **Độ trễ (Latency)** là thời gian cần thiết để nhận và xử lý dữ liệu. Độ trễ cho xử lý theo lô thường là vài giờ. Xử lý theo luồng thường diễn ra ngay lập tức, với độ trễ theo thứ tự của giây hoặc mili giây.
* **Phân tích:** Bạn thường sử dụng xử lý theo lô để thực hiện phân tích phức tạp. Xử lý theo luồng được sử dụng cho các hàm phản hồi đơn giản, tổng hợp hoặc các phép tính như trung bình di động.

**Kết hợp xử lý theo lô và theo luồng**

Nhiều giải pháp phân tích quy mô lớn bao gồm kết hợp xử lý theo lô và theo luồng, cho phép phân tích dữ liệu lịch sử và thời gian thực. Điển hình là các giải pháp xử lý theo luồng để thu thập dữ liệu theo thời gian thực, xử lý nó bằng cách lọc hoặc tổng hợp và hiển thị nó thông qua các bảng điều khiển và hình ảnh hóa theo thời gian thực (ví dụ: hiển thị tổng số xe đã đi qua đường trong giờ hiện tại), đồng thời duy trì kết quả đã xử lý trong kho lưu trữ dữ liệu để phân tích lịch sử cùng với dữ liệu được xử lý theo lô (ví dụ: để cho phép phân tích lưu lượng giao thông trong năm qua).

Ngay cả khi không cần phân tích hoặc trực quan hóa dữ liệu theo thời gian thực, các công nghệ xử lý theo luồng thường được sử dụng để thu thập dữ liệu theo thời gian thực và lưu trữ nó trong kho lưu trữ dữ liệu để xử lý theo lô sau đó (điều này tương đương với việc chuyển hướng tất cả các xe chạy trên đường vào bãi đậu xe trước khi đếm chúng).

Biểu đồ sau đây cho thấy một số cách kết hợp xử lý theo lô và theo luồng trong kiến trúc phân tích dữ liệu quy mô lớn.

1. **Các sự kiện dữ liệu từ nguồn dữ liệu theo luồng được ghi lại theo thời gian thực.**
2. **Dữ liệu từ các nguồn khác được đưa vào kho lưu trữ dữ liệu (thường là data lake) để xử lý theo lô.**
3. **Nếu không cần phân tích theo thời gian thực, dữ liệu theo luồng đã thu thập được sẽ được ghi vào kho lưu trữ dữ liệu để xử lý theo lô sau đó.**
4. **Khi cần phân tích theo thời gian thực, công nghệ xử lý theo luồng được sử dụng để chuẩn bị dữ liệu theo luồng cho phân tích hoặc trực quan hóa theo thời gian thực; thường bằng cách lọc hoặc tổng hợp dữ liệu theo cửa sổ thời gian.**
5. **Dữ liệu không theo luồng được xử lý theo lô định kỳ để chuẩn bị cho phân tích và kết quả được lưu trữ trong kho lưu trữ dữ liệu phân tích (thường được gọi là data warehouse) để phân tích lịch sử.**
6. **Kết quả của xử lý theo luồng cũng có thể được lưu trữ trong kho lưu trữ dữ liệu phân tích để hỗ trợ phân tích lịch sử.**
7. **Các công cụ phân tích và trực quan hóa được sử dụng để trình bày và khám phá dữ liệu theo thời gian thực và lịch sử.**

**Lưu ý:** Kiến trúc giải pháp thường được sử dụng để xử lý dữ liệu theo lô và theo luồng kết hợp bao gồm kiến trúc lambda và delta. Chi tiết của các kiến trúc này vượt quá phạm vi của khóa học này, nhưng chúng kết hợp các công nghệ cho cả xử lý dữ liệu theo lô quy mô lớn và xử lý theo luồng thời gian thực để tạo ra giải pháp phân tích tổng thể.