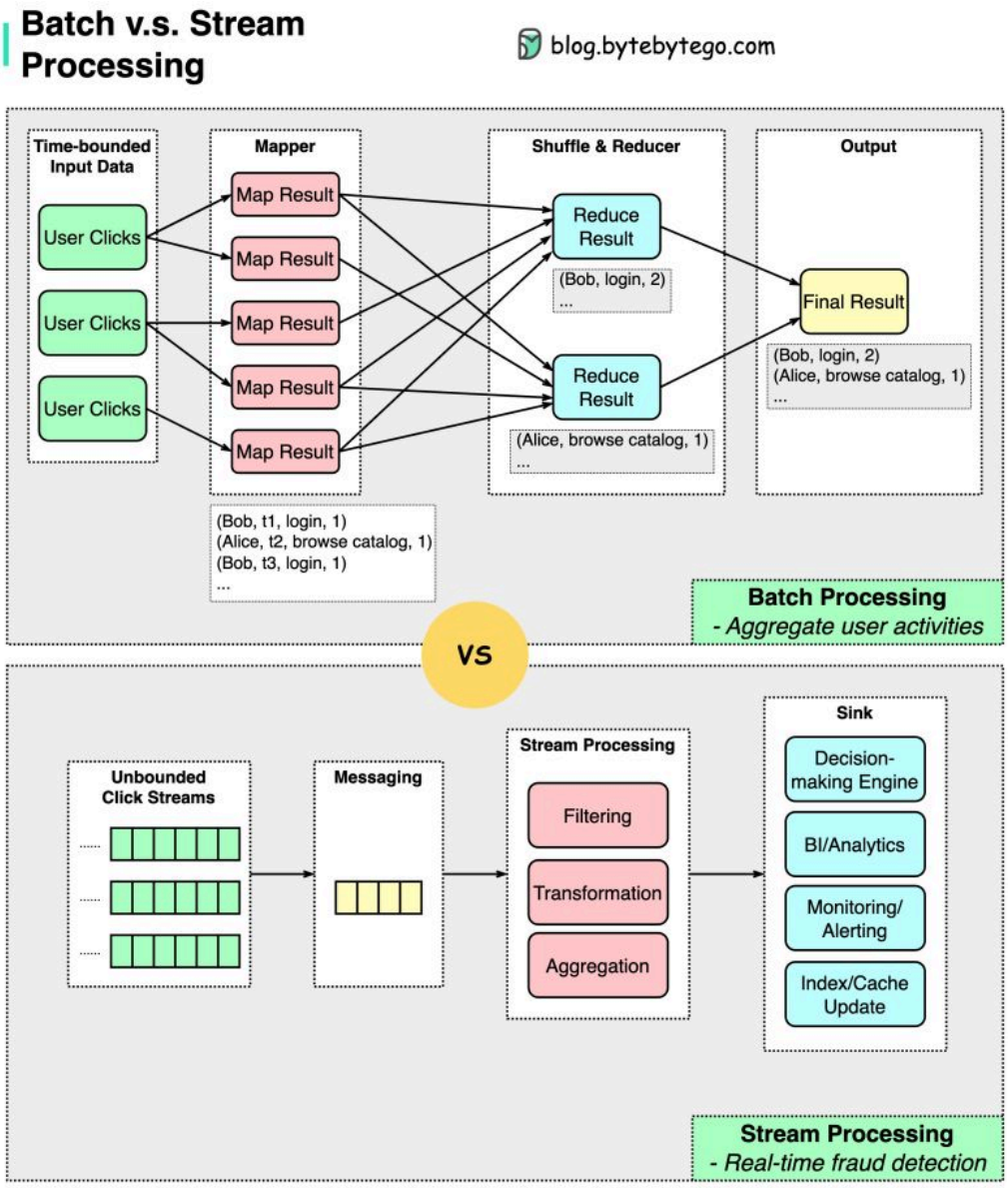
Two common data processing models: Batch v.s. Stream Processing. What are the differences?

The diagram below shows a typical scenario with user clicks:

- Batch Processing: We aggregate user click activities at end of the day.
- Stream Processing: We detect potential frauds with the user click streams in real-time.



Both processing models are used in big data processing. The major differences are:

1. Input

Batch processing works on time-bounded data, which means there is an end to the input data.

Stream processing works on data streams, which doesn't have a boundary.

2. Timelineness
   Batch processing is used in scenarios where the data doesn't need to be processed in real-time.

   Stream processing can produce processing results as the data is generated.

3. Output
   Batch processing usually generates one-off results, for example, reports.

   Stream processing's outputs can pipe into fraud decision-making engines, monitoring tools, analytics tools, or index/cache updaters.

4. Fault tolerance
   Batch processing tolerates faults better as the batch can be replayed on a fixed set of input data.

Stream processing is more challenging as the input data keeps flowing in. There are some approaches to solve this:

a. Microbatching which splits the data stream into smaller blocks (used in Spark);
b. Checkpoint which generates a mark every few seconds to roll back to (used in Flink).

👉Over to you: Have you worked on stream processing systems?