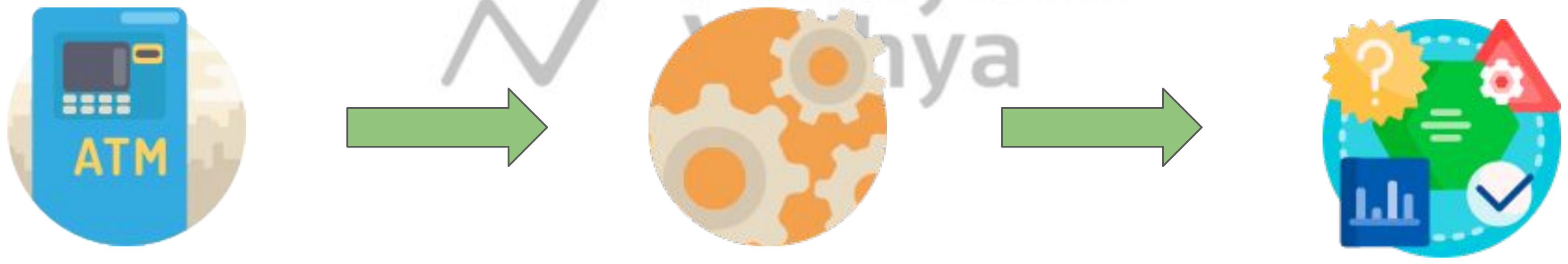
Stream Processing Challenges

Stream Processing

Stream processing is the processing of data in motion, or in other words, computing on data directly as it is produced or received.

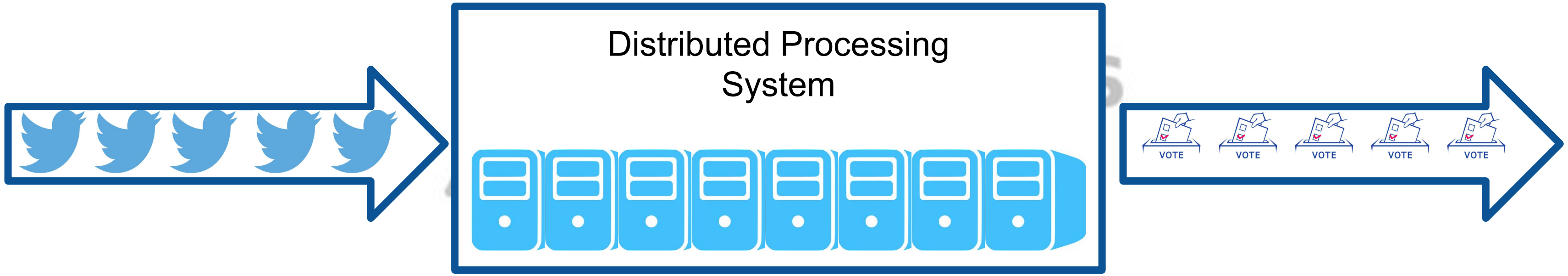

ATM
Customer withdraws money
Application computes something of value
Update results continuously

Tools and Frameworks

Frameworks such as **Spark Streaming** would actually process data in micro-batches.

However, there are some pure-play stream processing tools such as:

How to Process Streaming Data



Analytics Vidhya

Benefits of Processing Streaming Data on Distributed System

- Scales to hundreds of nodes
- Achieves low Latency
- Efficiently recover from failures
- Integrates with batch and interactive processing

Issues with Stream Processing

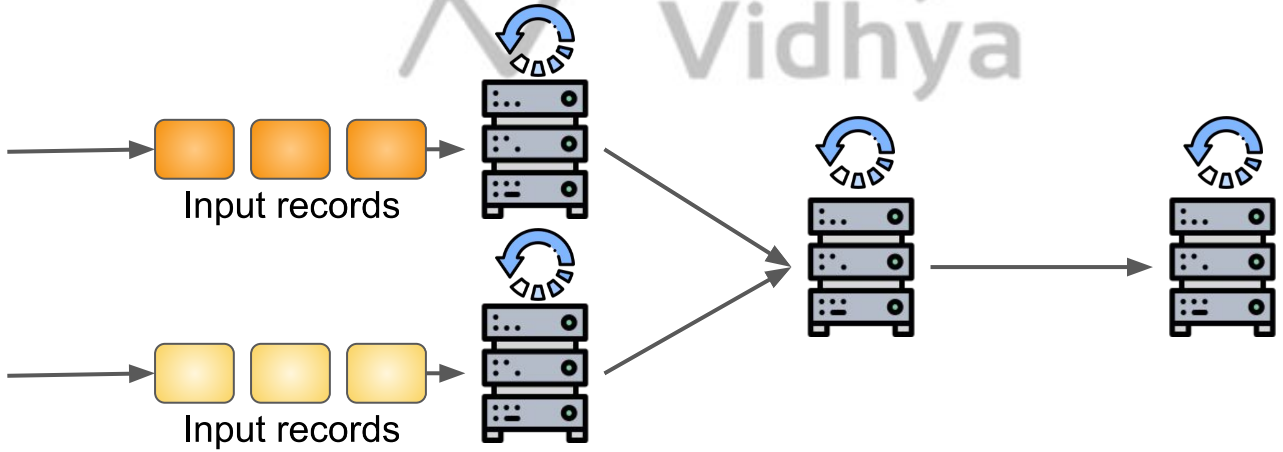
- Build two stacks-
  - One for batch
  - One for streaming
- Often both process same data

- Existing frameworks cannot do both simultaneously-
  - Either stream processing of 100s of MB/s with low latency or
  - Batch processing of TBs of data with high latency

- Extremely painful to maintain two different stacks because-
  - Different programming models
  - Doubles implementation effort
  - Doubles operational effort

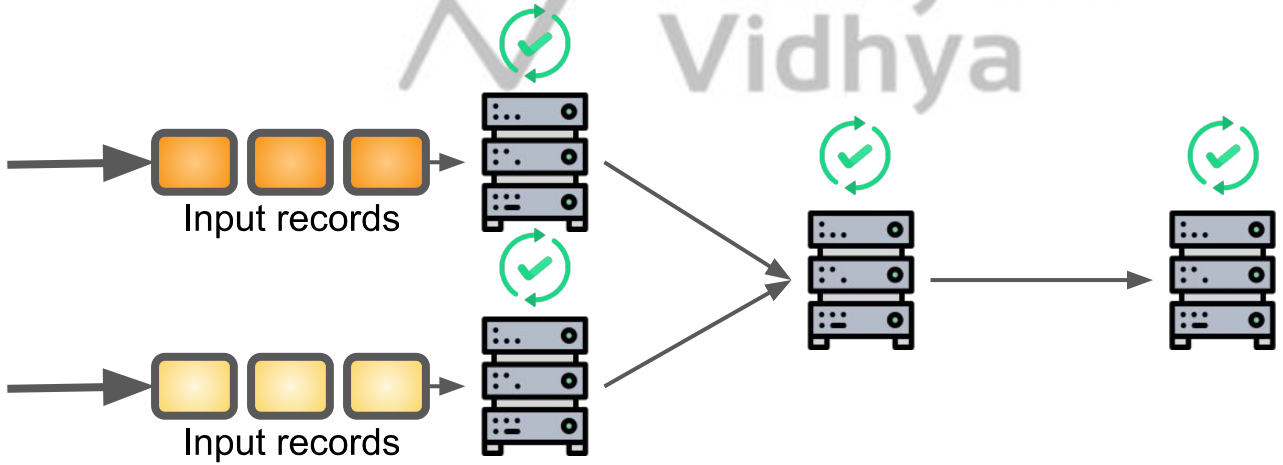
Issues with Stream Processing

- In Traditional Processing Model there are-
  - Pipeline of Nodes
  - Each node maintains mutable state



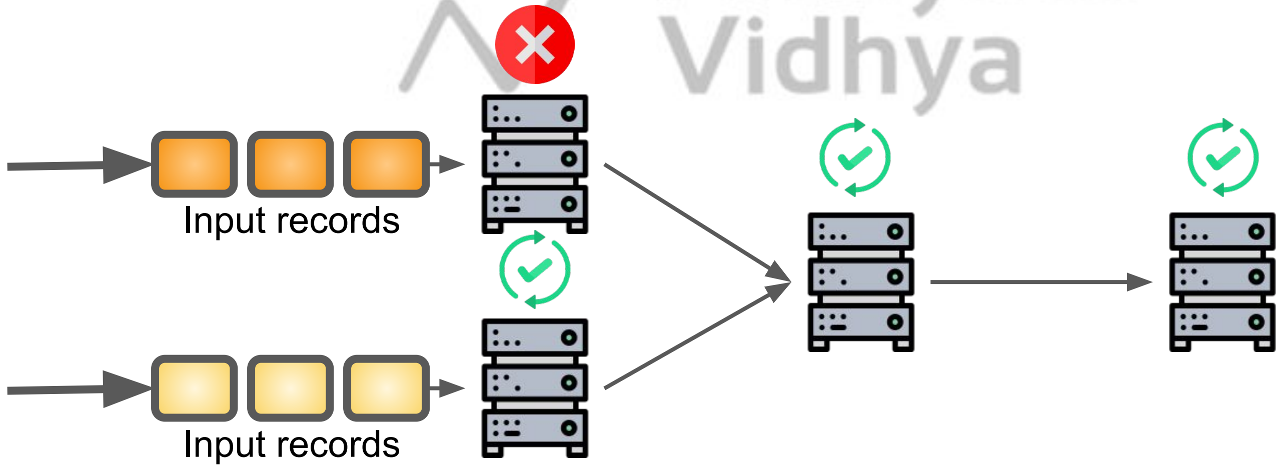
Issues with Stream Processing

- In Traditional Processing Model there are
    - A Pipeline of Nodes
    - Each node maintains mutable state
    - Each input record updates the state and new records are sent out



Issues with Stream Processing

- Mutable state is lost if node fails
- Making stateful stream processing fault-tolerant is challenging!



Additional Challenges

Scalability

Ordering

Consistency and Durability

Fault Tolerance

Thank You

Streaming Data Use Cases

Gain insights from historic data

Add in real-time data streams

Adopt machine learning

Drive further business opportunities