Introduction to Data Stream concepts and its model

by

Prof. Sartaj Ahmad

Department of Information Technology

KIET Group of Institutions, Delhi-NCR

Contents

- Stream Data
- Model and architecture
- Queries types

Introduction to streams concepts

Stream Data:

Such data flow in and out of a computer system continuously and with varying updates rates. Such data is temporarily ordered, fast changing, massive and potentially infinite.

It may be impossible to store an entire data stream or to scan through it multiple times due to tremendous volume.

Examples of streaming data

- Sensors in transportation vehicles, industrial equipment, and farm machinery send data to a streaming application. The application monitors performance, detects any potential defects in advance, and places a spare part order automatically preventing equipment down time.

- 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.

- A real-estate website tracks a subset of data from consumers' mobile devices and makes real-time property recommendations of properties to visit based on their geo-location.

- A solar power company has to maintain power throughput for its customers, or pay penalties. It implemented a streaming data application that monitors of all of panels in the field, and schedules service in real time, thereby minimizing the periods of low throughput from each panel and the associated penalty payouts.

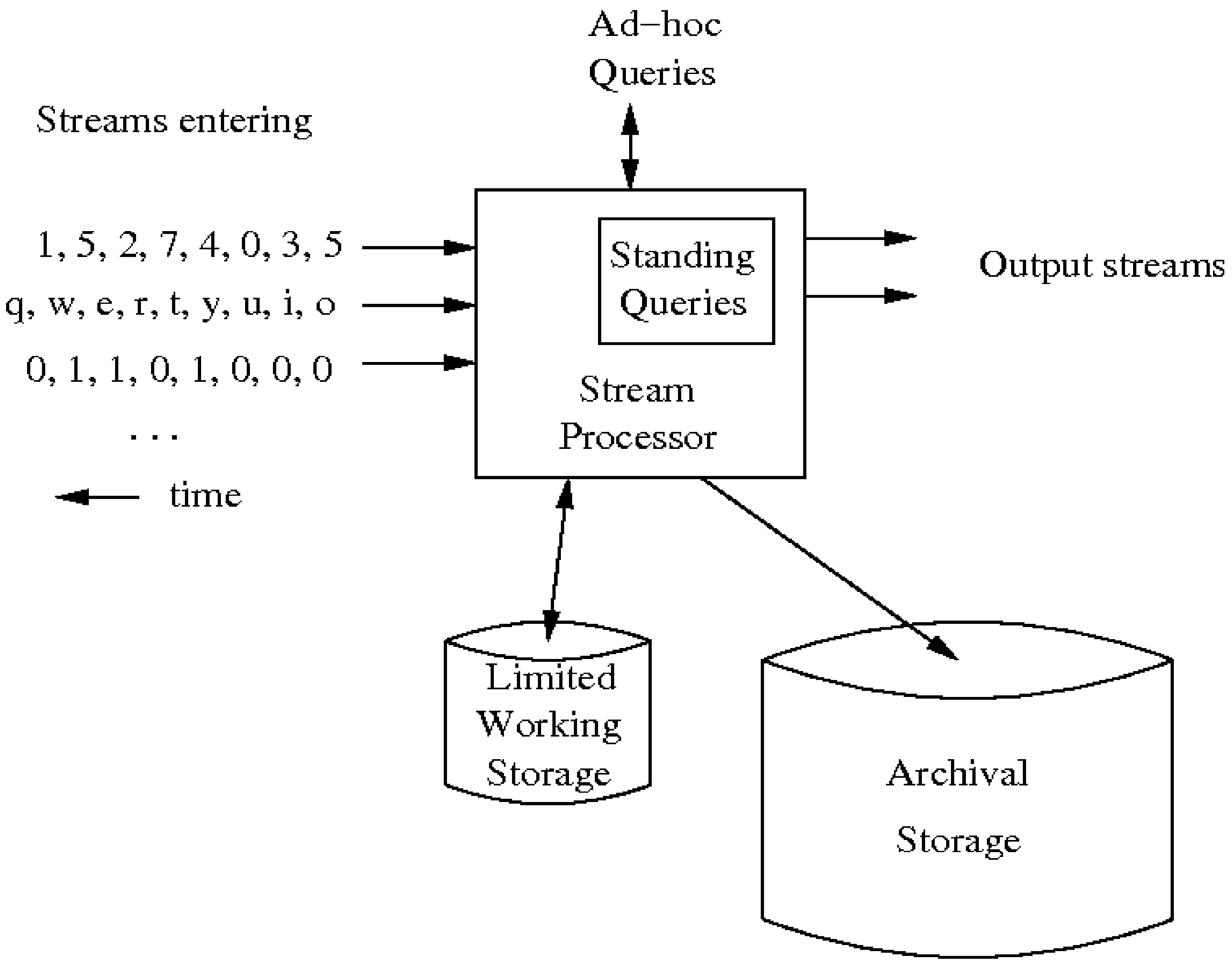
- A media publisher streams billions of clickstream records from its online properties, aggregates and enriches the data with demographic information about users, and optimizes content placement on its site, delivering relevancy and better experience to its audience.

- An online gaming company collects streaming 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.

Stream data model and Architecture

In the data stream model, some or all of the input data that are to be operated on are not available for random access from disk or memory, but rather arrive as one or more continuous data streams. **Data streams differ from the conventional stored relation model in several ways**:

- ❖The data elements in the stream arrive online.
- ❖The system has no control over the order in which data elements arrive to be processed, either within a data stream or across data streams.
- ❖Data streams are potentially unbounded in size.
- ❖ Once an element from a data stream has been processed it is discarded or archived — it cannot be retrieved easily unless it is explicitly stored in memory, which typically is small relative to the size of the data


Ad–hoc
Queries
Streams entering
1, 5, 2, 7, 4, 0, 3, 5
q, w, e, r, t, y, u, i, o
0, 1, 1, 0, 1, 0, 0, 0
. . .
time
Standing
Queries
Stream
Processor
Output streams
Limited
Working
Storage
Archival
Storage

Queries:

One-time queries: These(a class that includes traditional DBMS queries) are queries that are evaluated **once over a point-in-time snapshot of the data set**, with the answer returned to the user.

Continuous queries, on the other hand, **are evaluated continuously as data streams** continue to arrive. Continuous queries are the more interesting class of data stream queries, and it is to them that we will devote most of our attention. **The answer to a continuous query is produced over time**, always reflecting the stream data seen so far. Continuous query answers may be stored and updated as new data arrives, or they may be produced as data streams themselves.

The second distinction is between **predefined queries and ad hoc queries**. A predefined query is one that is supplied to the data stream management system **before any relevant data has arrived**. Predefined queries are generally continuous queries, although scheduled one-time queries can also be predefined.

Ad hoc queries, on the other hand, are issued **online after the data streams have already begun**. Ad hoc queries can be either one-time queries or continuous queries. Ad hoc queries complicate the design of a data stream management system, both because they are not known in advance for the purposes of query optimization.

In the data stream model of computation, once a data element has been streamed by, it cannot be revisited. This limitation means that ad hoc queries that are issued after some data has already been discarded may be impossible to answer accurately. One simple solution to this problem is to stipulate that ad hoc queries are only allowed to reference future data: they are evaluated as though the data streams began at the point when the query was issued, and any past stream elements are ignored (for the purposes of that query).

Thank you