Lesson 3

Data Stream Architecture and Processing Languages

Queries

- Static data in a relational database
- Applications send queries to the database and obtain the results
- Queries are one-time or transient.

Stream Queries Types

- Stream data changes frequently
- The results of the queries against the stream also change
- Two types of Queries: continuous queries or persistent queries. They process continuously as data continue to arrive
-

Stream Queries Results

- Obtained in two forms:

1. Store and update when data arrives, Aggregation queries
2. They make data stream for the results themselves: join queries

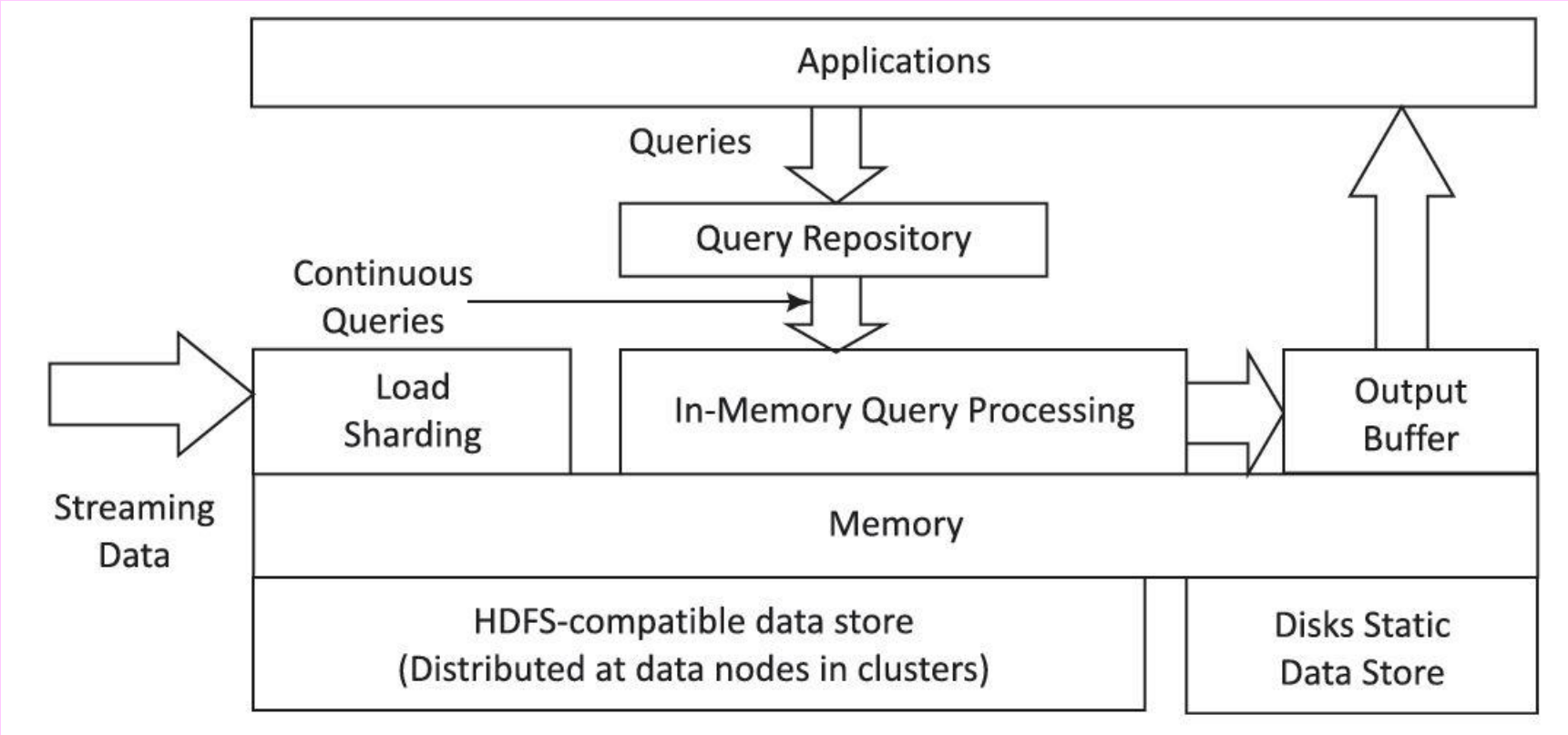
Query Processing Method

- A user application functions as query repository and continuously sending queries for processing of the shards in-memory
- The responses of queries save at an output buffer before they are finally retrieved by the application

Query Processing Method

- Streaming data shards load at memory in real-time applications
- Large data blocks in received stream then store at HDFS compatible data store or static data at disk
- Data shards load at memory from data store or disk for future uses

Figure 7.3 Data stream Query Processing architecture



Lambda Architecture

- A hybrid architecture for processing streaming data and back-end jobs at the same time
- The system manages stream flow over real-time data until the data elements pushed to a batch system
- The data then accessed and processed

Relation-based query languages (QLs)

- Based on SQL-like syntax, providing better support for windows and ordering
- Examples: Esper, CQL (STREAM), StreaQuel (TelegraphCQ), AQuery and GigaScope

Object-based QLs

- Object based QL: Classify the stream-elements according to type hierarchy
- Examples: Tribeca and COUGAR.

Procedure-based QLs

- User functions (procedures) specify the dataflow
- For example, Aurora provides graphical interface to users for constructing query plans

STREAM Continuous Query Language (CQL) [Stanford]

- An extension of SQL
- Example 7.1: usages of CQL syntaxes

Truviso syntaxes

- Example 7.2

TelegraphCQ Timestamp Column modifier

- Example 7.3

Stream Processing Issues

1. Size of the streaming data not fixed
2. Need of scalable processing
3. Variation in the frequency of data stream
4. Need of near real-time processing

Stream Processing Issues

5. Need of processing large data streams from different domains
6. Need of events-processing
7. Need of filtering to eliminate undesirable data elements

Stream Processing Types

1. Real-time Processing
2. Stream Processing and
3. Batch Processing

Streaming Processing Needs

(i) Computations: a function of a single data element or a smaller piece of recent data, no access to the complete data.

(ii) Processing algorithms must be relatively fast and simple

Streaming Processing Needs

(iii) Need to complete each computation in near real-time while static processing has more latency

(iv) Generally independent Computations

(v) Asynchronous processing: source of data does not interact with the stream processing system directly

Streaming Processing Needs

(vi) Requirements of high-volume processing with low latency because no information exists about when the next data will arrives.

Summary

We learnt:

- Stream Queries, Types and Results
- Queries Processing Architecture
- Relation-based query language
- Object-based query languages
- Procedure-based languages

Summary

We learnt:

- Stream Processing Seven Issues
- Stream Processing Three Types
- Streaming Processing Six Needs

End of Lesson 3 on

Data Stream Architecture and Processing Languages