

As requested, the data system needs to have ability to handle both batch data and streaming data. It needs to have functionality like data ingestion, data storage, data process, data catalogue, and data analytics. Hence, my system architecture is designed with two separated pipelines, one is for batch data processing, and another is for stream data processing. The general process is that data is ingested into batch layer and stream layer through ingestion layer. Then, the output of batch layer and stream layer are delivered to servicing layer for querying purpose. Following, the analytics jobs can be implemented with querying data from servicing layer. Meanwhile, a data management service can be provided for managing the data catalogue. Its advantage is that streaming data always shows real-time info and it can substitute delay info from batch processing data due to HDFS data processing always cause delay on data delivery.

**Data Ingestion**

In ingestion layer, data can be obtained from different sources, like RMDBS database or live streaming data source. The famous platform like Apache Kafka or Apache Flume is always used in this stage for holding data in order to serve following the batch layer and the speed layer. The data is delivered simultaneously to both the batch layer and the speed layer to enable a parallel indexing effort.

**Batch Layer**

For batch processing stage, the batch layer first stores the data in its original format on HDFS to enhance the reliability of the system. Then, use the Hadoop MapReduce job to process the data and save the result. The data after this kind of processing is called a batch view. Hadoop jobs are very robust and will not lose data when various abnormalities occur during the operation. However, the batch layer has a large delay in data processing.

**Stream Layer**

The responsibility of the Speed ​​layer is to meet all real-time processing needs. Speed ​​layer is usually based on a streaming computing platform such as Spark streaming, and uses fast incremental algorithms to read, analyse, and save data in minutes, seconds, or even milliseconds. Streaming processing often uses memory computing, which means that when an exception occurs, it may cause data loss or incorrect calculation results. However, the data system does not need to consider such issues too much, because the next batch layer job will process all the data again and obtain accurate results.

**Servicing Layer**

The responsibility of the Serving layer is to merge the output data of the speed layer to the output data of the batch layer to obtain a complete output data and save it in a NoSQL database such as MongoDB, Hive to serve online retrieval applications. Merger has a small amount of real-time data on top of the batch calculation results, and the results have a good approximation compared with the full batch calculation.

**Application Layer**

In application layer, many build-up applications like ML prediction model, abnormality detection model, or analytic dashboard can be deployed here, and querying data from servicing layer for their own daily operation.