**Batch data - SparkSQL with Hive**  
  
To produce the batch data, we will extract the MovieLens dataset to Hadoop environment. Hadoop Distributed File System (HDFS) is a distributed file system that is reliable, scalable and fault-tolerant. It can separate the data and distribute them into different nodes which enable highly efficient parallel processes.  
  
However, Hadoop low level MapReduce framework poses a huge challenge to programmer to query the data. Even for a simple partitioning of data, it required a custom program which is extremely difficult to maintain and reuse. To make big data analytic possible, we will build a data warehouse infrastructure with Hive on top of Hadoop which can transform the dataset to star schema. With the SQL like interface provided by Hive, we can perform data warehousing tasks such as extract, transform and load.  
  
To handle large dataset, it is important to reduce the time and resources required aggregate data and compute complicated queries in an efficient way. Therefore, we will use Spark SQL to process the queries in optimized and fast pace. We can use the HiveMetastore to get the metadata stored in HDFS to let Spark SQL perform optimization of the queries.   
  
**Streaming data – Spark Streaming with Hbase**  
  
Spark Streaming is good at process time series data to get results in near real time. It divide the incoming data into DStream micro-batches of seconds. The DStream is a series of Resilient Distributed Datasets (RDD) that are operated in parallel on clusters. After transforming the data to DStream, the processed results will be outputted in batches. We will save the result in Hbase and then combine the batch data in Hive to complete the Lambda Architecture.   
  
Hbase is a distributed, scalable, column-based big datastore that is good at read and write processes in real time. Hbase is running on HDFS with key-value store to make real-time data access scalable. It is then possible to query data from Hbase and combine with the result from Spark SQL with Hive. For example, we could make use of the Spark-Hbase Connector to solve the issue of simple key value store in Hbase.  
  
**Data Analysis – MLlib**  
  
Finally we can analyze the data using MLlib, which is a Spark’s machine learning library that is RDD-based contains high-quality algorithm. This enable us to run iterative computation much faster than Hadoop MapReduce.

<https://github.com/mionisation/BI_BigData_2_HiveDatasetAnalysis>

<https://mapr.com/resources/videos/spark-streaming-hbase/>