The Lambda Architecture [1] provides a general-purpose approach to implementing

an arbitrary function on an arbitrary dataset and having the function return its results with low latency. The main idea of the Lambda Architecture is to build Big Data systems as a series of three layers, named as “batch layer”, “speed layer” and “serving layer”, which will be discussed in details in the following sections.

Batch layer

The batch layer tackle the high volume characteristics of big data analytics. Data processing and query tool in Hadoop like hive and spark is capable to handle pera-bytes of data, however it stills does not guarantee low latency response of query. An obvious approach is to precompute the query function to meet the performance requirement of the use cases, like instant recommendation engine in this project. He propose to use store the immutable but constantly growing data in HDFS, and implement the batch pre-processing system using Hive(meta-store) and Spark.

Speed layer

Batch processing is run in a regular basis, implies that data represented in the batch view does not cover the most up-to-dated data emerged after the last executed batch. The objective of speed layer is to process the recent data incrementally to facilitate a fully real time low latency data analysis system. In our project we propose to use spark streaming to capture live streaming data, process the data in micro-batch and output the result to downstream serving layer.

Serving layer

Batch view and speed view are saved in the consolidated serving layer. A specialized data model will facilitate query patterns of the recommendation system use case. SparkSQL is chosen in our project to support high performance and flexible queries.

# **References**

|  |  |
| --- | --- |
| [1] | J. W. Nathan Marz, big data - principles and best practices of scalable realtime data systems, 2015. |