Enhanced Lambda Architecture in AWS using Apache Spark

Summary

Many organizations are looking for a cloud-based solution for integrating batch and real-time data while keeping total costs and expenses to a minimum. Lambda Architecture is the answer to this problem. Lambda architecture provides a single framework to handle massive quantities of data. Lambda Architecture can be implemented on Amazon Web Services (AWS) to process large amounts of data and reduce any delay between data collection and availability in dashboards using Apache Spark. The processing of real-time data is possible with lambda architecture, which includes Amazon Simple Storage (S3), Spark Streaming and Spark SQL on top of the Amazon Elastic MapReduce (EMR) cluster.

Traditional Lambda Architecture

The traditional approach is shown below which used Hadoop and Storm together for massive data processing. This approach attempts to balance fault tolerance, latency, throughput by using batch processing to provide batch views and real-time views online simultaneously.

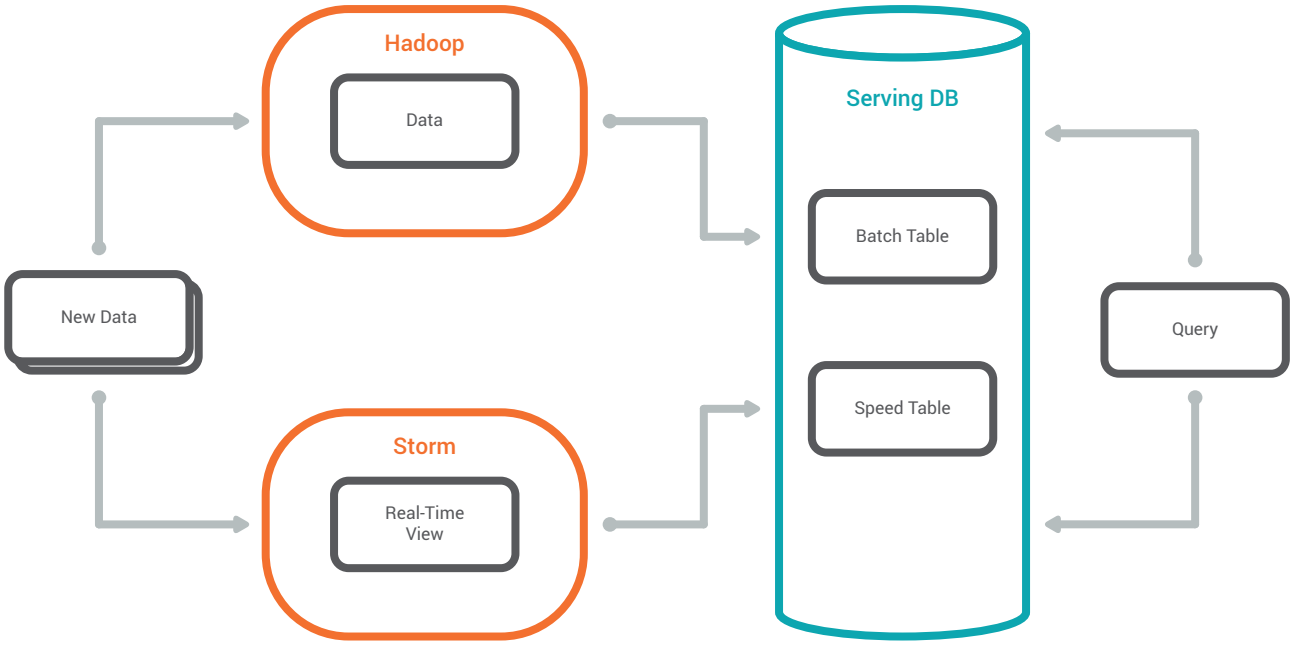


Figure 1: Traditional Lambda Architecture

Enhanced Lambda Architecture Using Spark

Lambda architecture is divided into three processing layers:

- The batch layer
- Serving layer
- Speed layer

All new data will be sent to the batch layer (Amazon EMR, Amazon S3) and then to the speed layer (Spark Streaming). In the batch layer new data is added to the master dataset. In S3 data is appended only immutable set. Similar to the Extract Transform and Load (ETL) operation the batch layer pre-computes all query functions continuously. The product from the batch layer is called a batch view and is stored in Amazon S3 as a tab-separated value file. The serving layer is a scalable database that swaps new batch views with old as new data becomes available. The speed layer uses the Spark engine to process data in the last batch and produces real-time views. The old views are discarded in the speed layer as new data is available. The Spark application resolves queries by merging batch views with real-time views. This approach enhances the traditional Lambda Architecture approach, as the traditional approach requires all code to be maintained in two, complex, distributed systems.

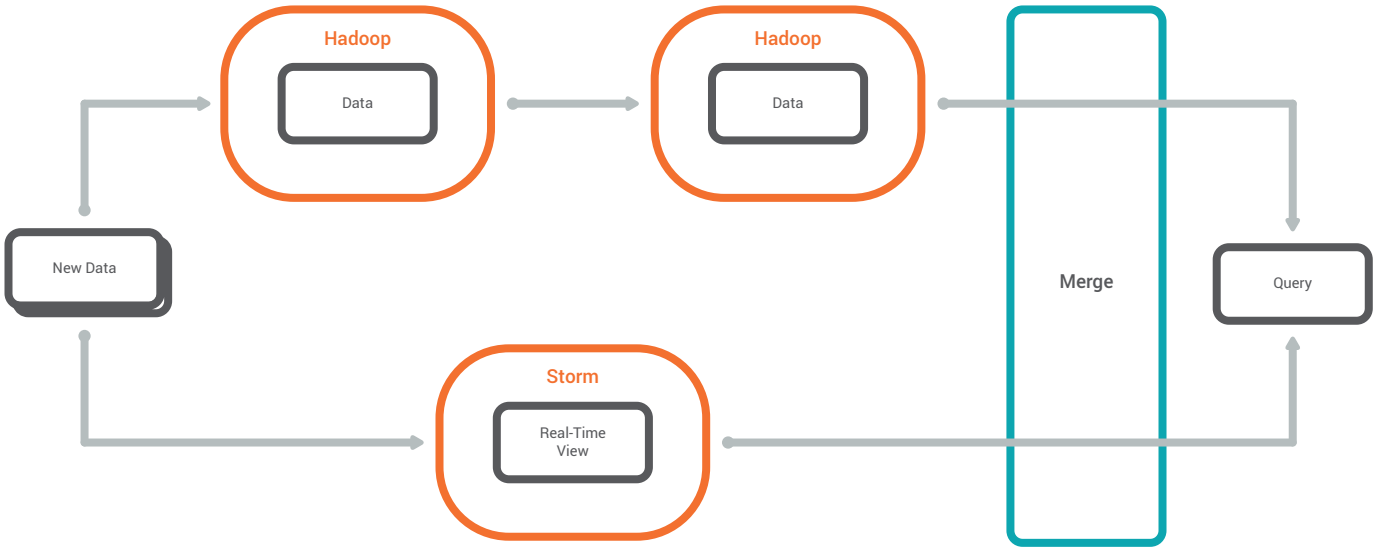
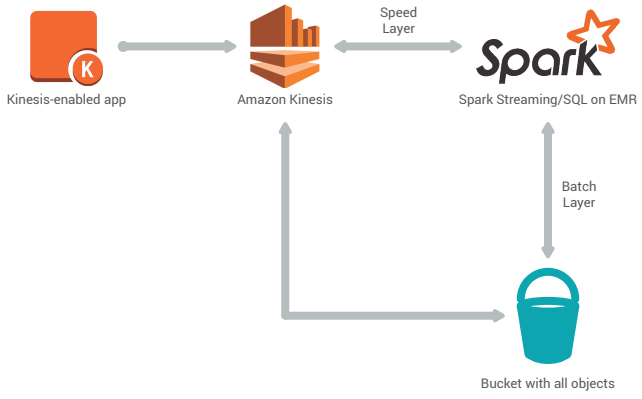


Figure 2: Lambda Architecture

Components of Lambda Architecture on AWS

Amazon Kinesis can be used to feed real-time data flows into AWS. An Amazon S3 bucket is used to store all data. The speed layer is built using Spark Streaming on an Amazon EMR cluster. The batch data is processed using Spark SQL on the Amazon EMR cluster. The application used can be written in Java or Scala or Python. The batch intervals in Spark Streaming can be set based on application requirements to process batches in the micro batch level and provide users with up to date aggregates. This simplifies big data processing and provides a cost effective framework which can be dynamically scalable in Amazon EC2.



Conclusion

By using Spark as the processing engine we can write applications within a single code base. Using the Lambda architecture we can perform real-time data processing and batch data processing. All the business logic code can be implemented using Java or Python or Scala. Through AWS, we can quickly implement the Lambda Architecture, reduce maintenance overhead and reduce costs.