

Lambda Architecture with Spark



Narayan Kumar
Software Consultant
Knoldus Software LLP

Agenda

- What is Lambda Architecture ?
- Components of Lambda Architecture
- Advantages of Lambda Architecture
- Implementation with Spark and it's Benefits
- Code Review & Demo

Agenda

- What is Lambda Architecture ?
- Components of Lambda Architecture
- Advantages of Lambda Architecture
- Implementation with Spark and it's Benefits
- Code Review & Demo

What is Lambda Architecture ?

“ Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch- and stream-processing methods.”

wikipedia

Coined by Nathan marz

- Ex- Twitter Engineer
- Creator of Apache Storm



Agenda

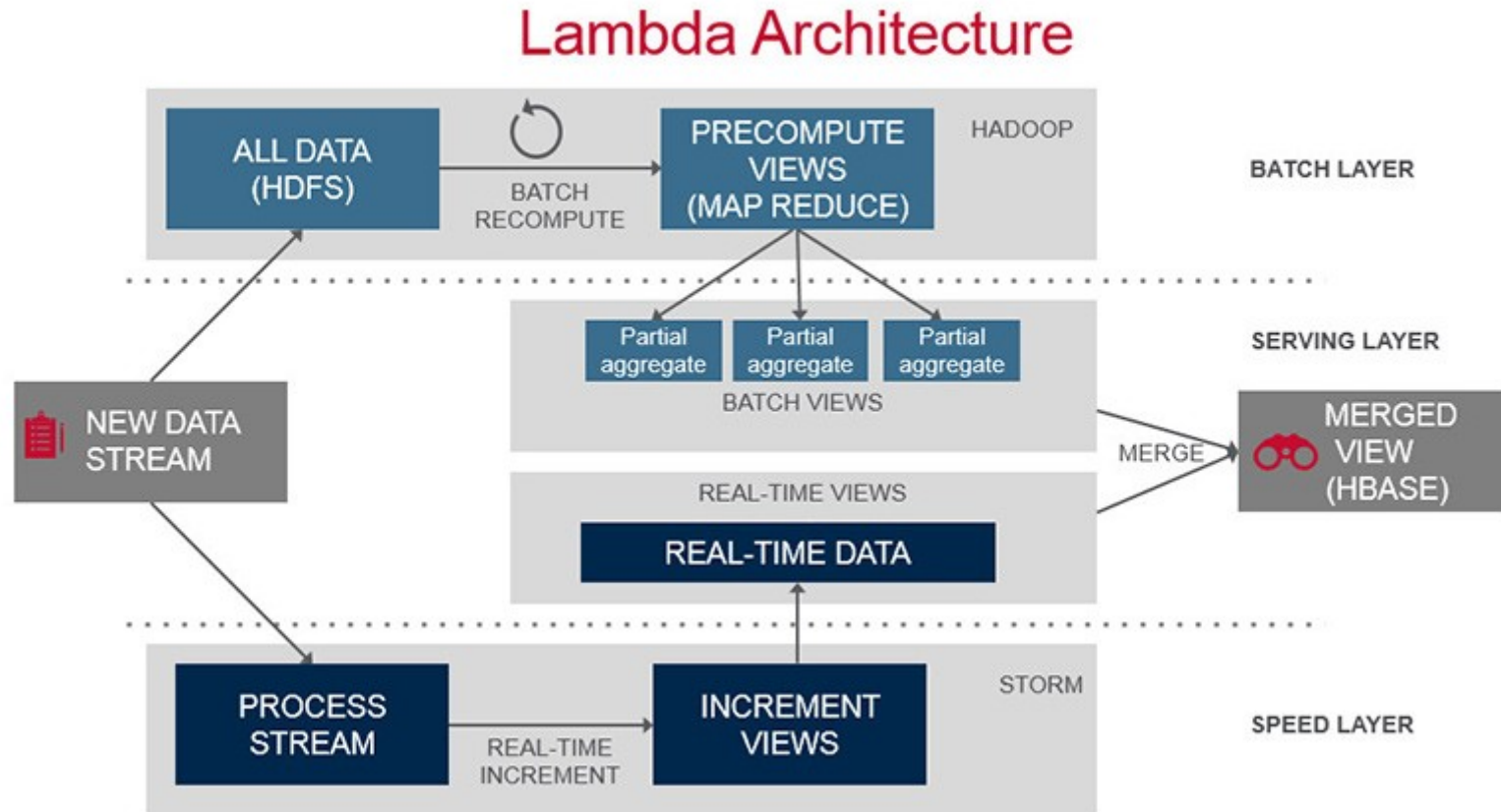
- What is Lambda Architecture ?
- Components of Lambda Architecture
- Advantages of Lambda Architecture
- Implementation with Spark and it's Benefits
- Code Review & Demo

Components of Lambda Architecture

Lambda architecture broadly classified into three layer :-

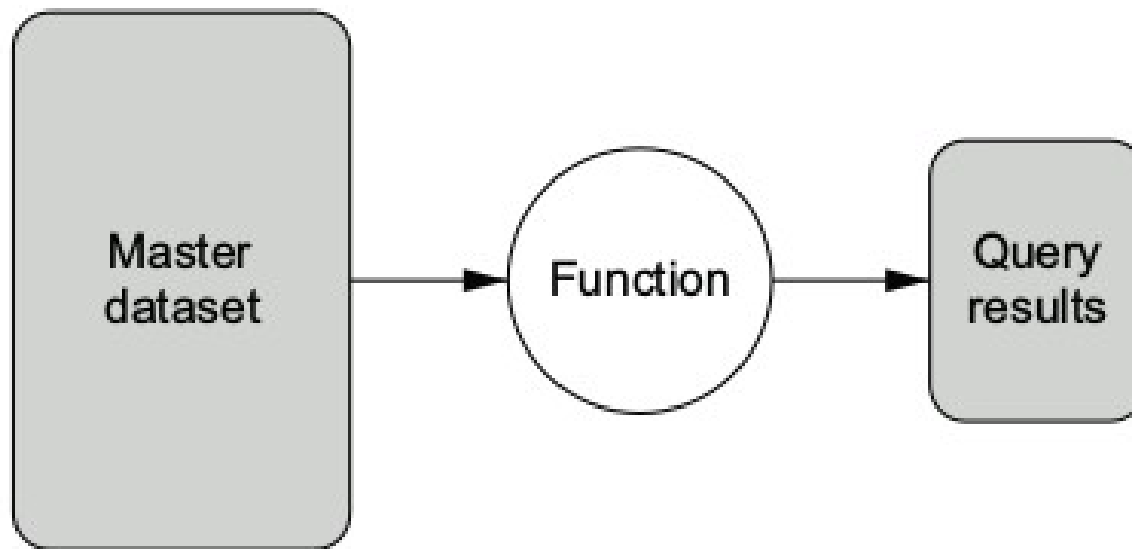
- Batch Layer
- Speed Layer
- Serving Layer

Overview of Lambda Architecture



Batch Layer

In the Lambda Architecture, the batch layer precomputes the master dataset into batch views so that queries can be resolved with low latency.



Master DataSet

The master dataset is the source of truth in the Lambda Architecture. Even if you were to lose all your serving layer datasets and speed layer datasets, you could reconstruct your application from the master dataset.

Data in master dataset must hold three properties :-

- Data is raw
- Data is immutable
- Data is eternally true



Computing functions on the batch layer

As our master dataset is continually growing, we must have a strategy for updating our batch views when new data becomes available.

Here we have two suitable computing algorithm :-

- **Recomputation algorithms** : Throwing away the old batch views and recomputing functions over the entire master dataset.
- **Incremental algorithms** : An incremental algorithm will update the views directly when new data arrives.

Speed Layer

There are two major facets of the speed layer: storing the realtime views and processing the incoming data stream so as to update those views.

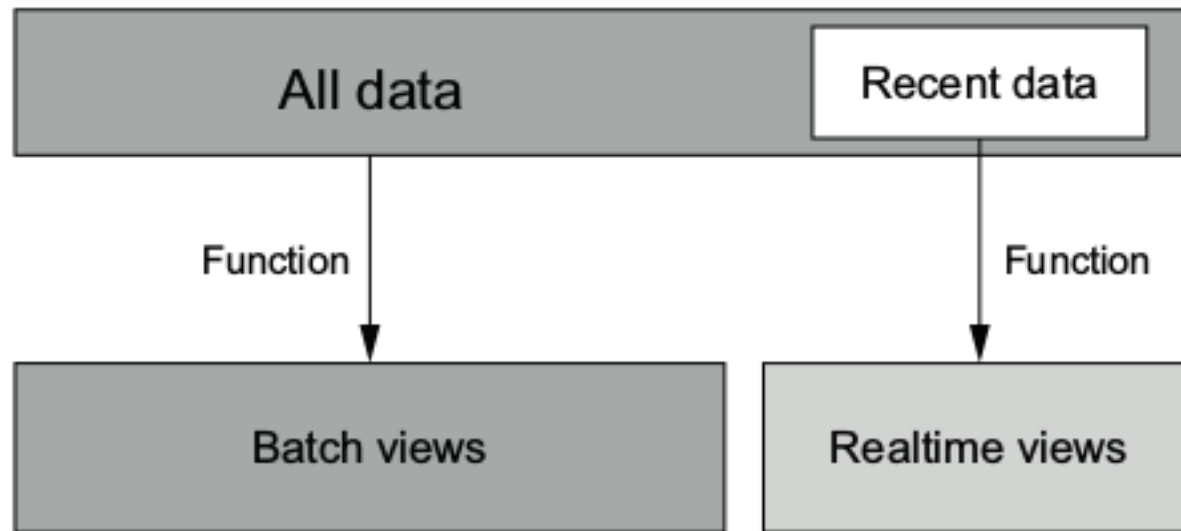


Figure 12.2 Strategy: realtime view = function(recent data)

Storing real time views

The underlying storage layer must meet the following requirements: -

Random reads : A realtime view should support fast random reads to answer queries quickly.

Random writes : To support incremental algorithms, it must also be possible to modify a realtime view with low latency.

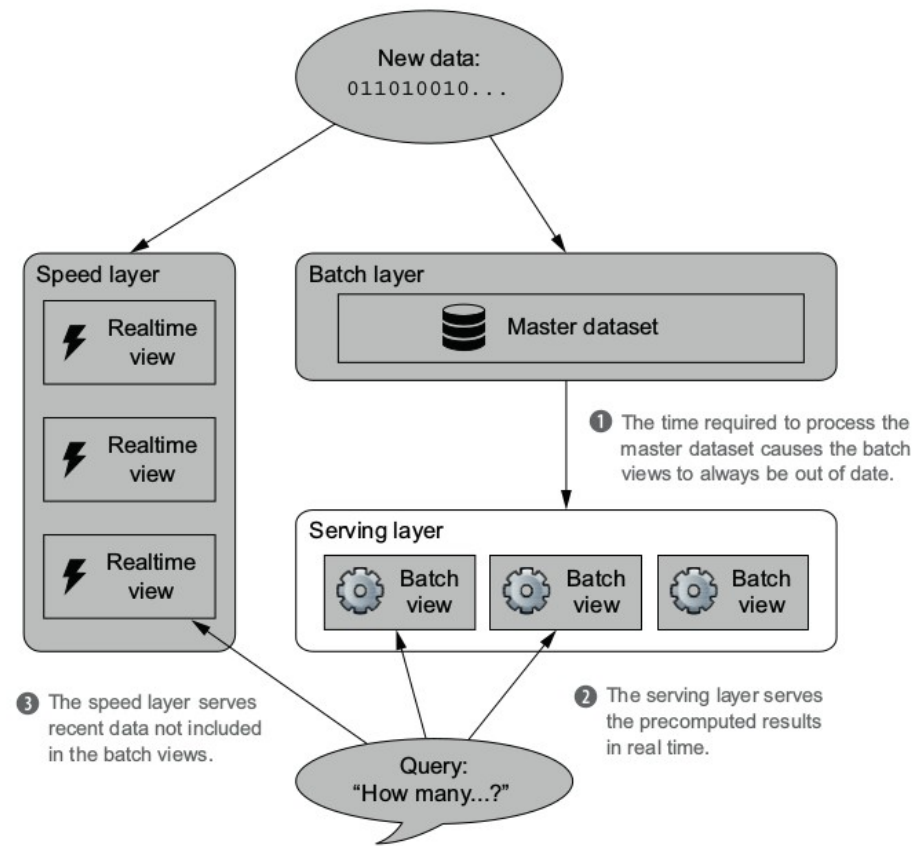
Scalability : As with the serving layer views, the realtime views should scale with the amount of data they store and the read/write rates required by the application.

Fault tolerance : If a disk or a machine crashes, a realtime view should continue to function normally.



Serving Layer

In the Lambda Architecture, the serving layer provides low-latency access to the results of calculations performed on the master dataset. The serving layer views are slightly out of date due to the time required for batch computation.



Requirements for a serving layer database

Similar to speed layer these are following requirements: -

Random reads : A serving layer database must support random reads, with indexes providing direct access to small portions of the view.

Batch writable : The batch views for a serving layer are produced from scratch. When a new version of a view becomes available, it must be possible to completely swap out the older version with the updated view.

Scalability : A serving layer database must be capable of handling views of arbitrary size.

Fault tolerance : Because a serving layer database is distributed, it must be tolerant of machine failures.



Agenda

- What is Lambda Architecture ?
- Components of Lambda Architecture
- Advantages of Lambda Architecture
- Implementation with Spark and it's Benefits
- Code Review & Demo

Advantages of Lambda Architecture

These are following advantages of lambda architecture: -

Human fault tolerance : LA is provides human fault tolerance capability to the Big data system.

Operational complexity : It resolved operational complexity issue of big historical query by divide into precomputed query and on fly query.

Resilience : LA is fully resilience,because it is difficult for human errors or hardware faults to corrupt data stored in the system since the system does not allow update or delete operations in existing data.

Simple & Maintainable : It is simple in nature so we can easily understand and it's flexible architecture is helpful in maintainance.



Agenda

- What is Lambda Architecture ?
- Components of Lambda Architecture
- Advantages of Lambda Architecture
- Implementation with Spark and it's Benefits
- Code Review & Demo

Implementation with Spark and it's Benefits

There are following benefits to implement LA with Spark : -

- Spark gave us unified stack like Spark Core, Spark SQL, Spark Streaming, Mllib, and GraphX, so that we can easily implement LA.
- Spark has clean and easy-to-use APIs (far more readable and with less boilerplate code than MapReduce).
- Biggest advantage Spark gave us in this case is Spark Streaming, which allowed us to re-use the same aggregates we wrote for our batch application on a real-time data stream.



References

Big Data Principles and best practices of scalable real-time data systems
Nathan Marz WITH James Warren

https://en.wikipedia.org/wiki/Lambda_architecture

<https://www.mapr.com/developercentral/lambda-architecture>

<http://lambda-architecture.net/>

Thank you