My Learning Curve

Learning is my passion, curiosity is my trainer/ teacher, motivation is my master/mentor.

Yes, it’s true, I have been learning new technologies since last 6 years, when I started my professional journey I learned J2EE from NIIT, that was my first technical milestone later I never had to look back. I have learned JBOSS, WebLogic, PowerShell, Batch Scripting.

Recently, 6 months ago, I started learning Power BI which is a data visualization tool initially. Microsoft corporation continuously trying to make it more scalable and robust by adding more features. Now we can use Power BI as an ETL tool for OLAP databases and it can act as an analysis tool.

Initially I started learning power BI from “**Pluralsight”** website and I took a subscription for variety of courses such as Azure and google Big query. I do go through the videos and do practice.

Moreover, I do follow the blogs from authors because following blogs would be a great help for techies. If you get struck or you may need a suggestion to implement complex logic, in those scenarios blogs are very helpful.

I do join **MSDN** forums where experts exchange their knowledge, it’s quite interesting and helpful for the people who are having a basic knowledge of subject area.

**What is a Lambda architecture? Which problems does it solve?**

A Lambda architecture is more about data processing than data storage. It's an architectural pattern designed to process large data volumes using both batch and streaming methods. Batch processing is typically pull-oriented, whereas streaming data is push-oriented. A Lambda architecture implementation involves a batch layer, a speed layer, and a serving layer. The serving layer, which handles data access/consumption, may serve consolidated data which was ingested through the different batch and speed layers.

**Serving Layer**

**Batch Layer**

Batsch View

Master Dataset

Query

Batch View

Query

**Speed Layer**

Real-time view

Real-time view

**Lambda solves these long-standing challenges:**

1. The Lambda Architecture solves the problem of computing arbitrary functions on arbitrary data in Realtime by decomposing the problem into three layers: the batch layer, the serving layer, and the speed layer.
2. Preserving data in its original form and never changed or destroyed.
3. Keeping data raw, rather than converting it into an arbitrary format or schema. Then, if you decide you need a component of the data later, it’s still there.
4. Data is “engineered to allow it to be as easily reinterpreted as you learn.” Why does this matter? It makes it reinterpreting fast and fault-tolerant.
5. Supporting real time with two points of view: Just in time and a deep cross-sectional view. This lets you make decisions quickly without sacrificing the 100 percent loss-less accuracy needed for important business areas (such as finance, medicine, or mission-critical operations).
6. It’s an architecture, meaning a smart way to combine Big Data technologies.
7. Lambda defines a Big Data architecture that allows arbitrary queries and computations on both fast-moving data as well as historical data. As noted earlier, the Lambda architecture’s main goal is to execute OLAP-type processing faster – without enabling interesting new applications like real time application of user segments/scoring.