Traditional Data Warehouse Vs BDB Big Data Pipeline Warehouse with Implementation Use Case

Background

The entire Data Warehouse Architecture has been changed by the evolution of digital footprints of organizations. BDB (Big Data BizViz) Platform claims to adopt this alteration seamlessly through various ways of utilizing Big Data, IOT, Cognitive BI, Machine Learning, and Artificial Intelligence.

Introduction

Our intent is to show how our BDB platform has evolved on data pipeline architectures based on distributed data stores, distributed computing, real-time or near real-time data processing (Streams), use of machine learning, and analytics to support decision-making process in the current rapidly changing business environment.

Today, Big Data has grown really 'Big' to become one of the pillars of any business strategy. It is necessary in a highly competitive and regulated business environment that decision making process is based on data instead of intuition.

To make good decisions, it is necessary to process huge amount of data in an efficient way (the less possible computing resources and a minimum processing latency), add new data sources (structured, semi-structured, and unstructured ones such as UI activities, logs, performance events, sensor data, emails, documents, social media etc.) and support the decisions using machine learning algorithms as well as visualization techniques. The more an organization invests on its proprietary algorithms, the more chances are that it can sustain the storm of the digital era.

Let us see how we can transform our traditional data warehouse architecture into a contemporary one to solve the current challenges related to big data and high computing. Try to understand this situation through the example of an Enterprise customer that has 40 Million active users and overall 100 Million User bases.

Traditional data warehouse architecture

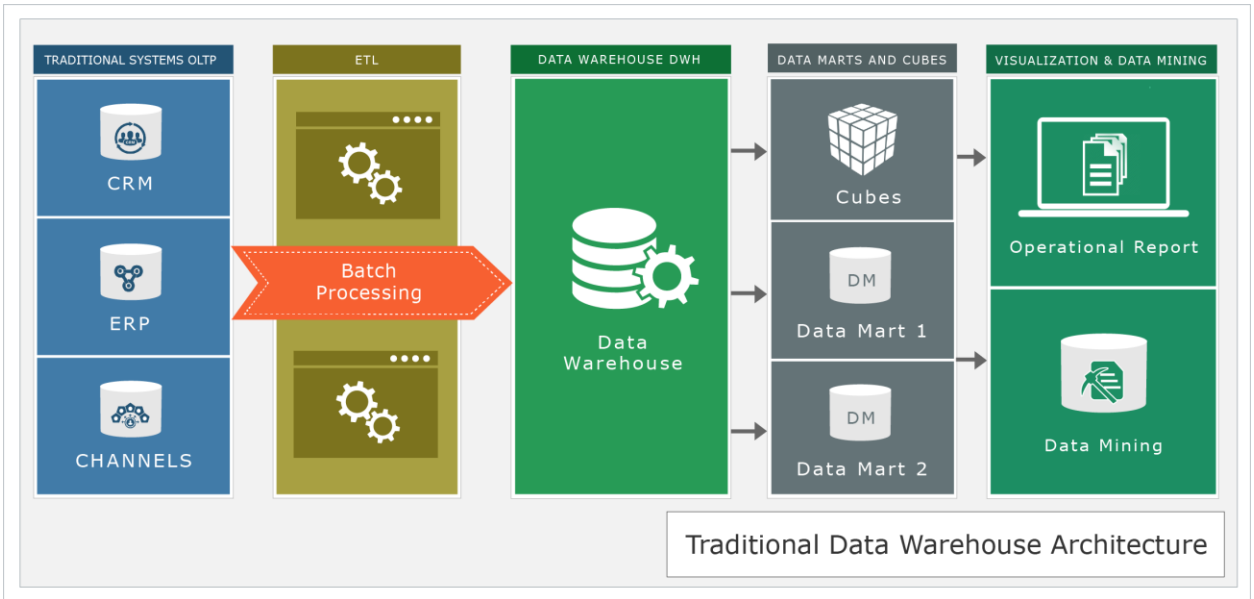
A traditional data warehouse is still good for businesses that are completely off the cloud but slowly they need to change their strategy to adopt digital revolution. Traditional data warehouse architecture comprises of the following elements:

- Transactional systems (OLTP)- Produce and record the transactional data/facts resulted from business operations.
- A data warehouse (DWH)- Centralized data stores that integrate and consolidate the transactional data.
- ETL processes-Batch processes that move the transactional data from OLTP towards DWH (scheduled).

Data Cleanup, Mata Data Management, Data Marts, and Cubes- Representing basically a derived and aggregated view of the DWH.

Analytical and Visualization Tools- Enabling the visualization of data stored in the data marts and cubes for Canned, Ad-Hoc, and dashboards for different types of users. They do first generation analytics using data mining techniques. The BDB Platform does this quite well and has done many successful deployments.

This kind of architecture can be illustrated in the following figure.



This architecture has limitations in today's modern Big Data Era:

- Data sources are limited only to transactional systems (OLTP).
- The major workload is based on ETL batch processing (jobs). It's well-known that there is a loss of data in this step.
- The integration and consolidation of data is very complex due to the rigid nature of ETL jobs.
- Data schema is not very flexible to be extended for new analytics use cases. Maintenance is high with new fields coming into the business processes.
- It's very complex to integrate semi- and un-structure data sources. So, we lose very important information in the form of log files, sensor data, emails and documents.
- It doesn't support naturally real-time and interactive analytics. It's thought to be batch-oriented. With data streams there is a need of Push Analytics.
- The efforts put in scaling the solution is quite a lot, so all the businesses are not able to budget such projects properly.
- It is mainly designed for on-premise environment, so complex to extend and deploy in the cloud or hybrid-based environments.

- Although all the above steps can be done efficiently with BDB Platform there is a need to bring the data in a Pipeline based Architecture to give seamless analytics.

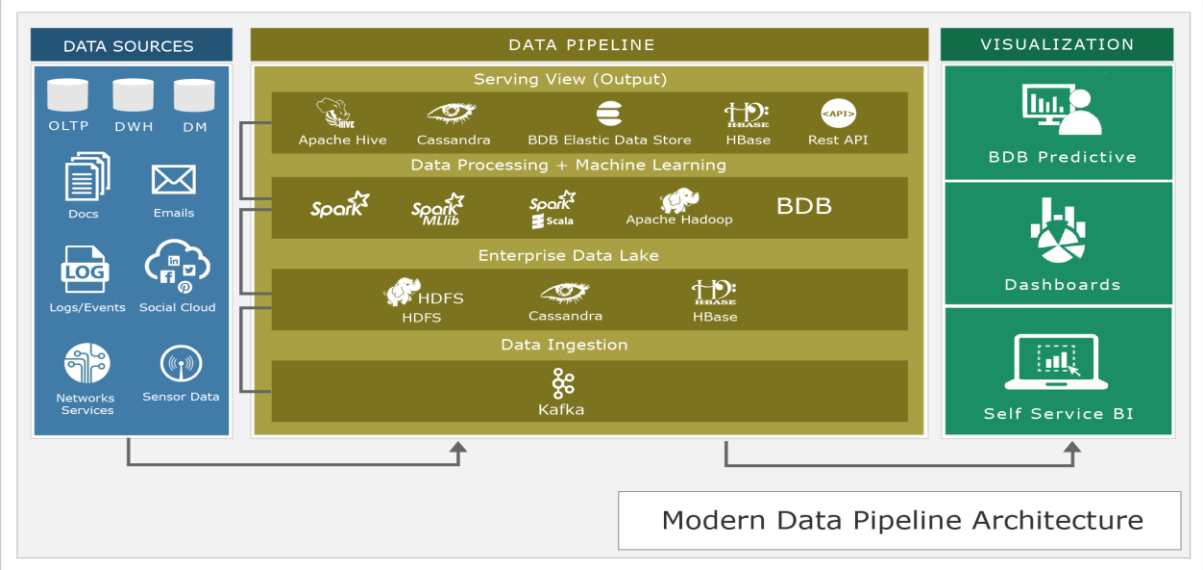
Modern pipeline architectures

The modern pipeline architect is an evolution from the previous one integrating new data sources and using new computing paradigms as well as the integration of IOT data, artificial intelligence, machine learning algorithm, Deep Learning, and cognitive computing.

In this new approach, we have a pipeline engine with the following features:

- Unified data architecture for all the data sources irrespective of the original structure. Integration with existing data marts and data warehouses help to keep the original data as well as transformed data with the advent of Data Lakes.
- Flexible data schemas designed to be changed frequently. Use of NoSQL-based data stores.
- Unified computing platform for processing any kind of workload (batch, interactive, real- time, machine learning, cognitive, etc.). Use of distributed platforms such as Hadoop, HDFS, Spark, Kafka, Flume etc.
- Deployable on the hybrid- and cloud-based environments. Private Clouds for Enterprises are the key.
- Horizontal scalability, so we can process unlimited data volume by just adding new nodes to the cluster. (BDB platform gives horizontal and vertical scalability)

Although, there are a huge amount of technologies related to big data and analytics, we have shown here how various Big Data Technologies are being used with BDB Platform to provide a working modern data pipeline architecture (see the figure 02). Using BDB platform reduces the risk of experimenting with too many options but to use the architecture that has been working.



Use Case 01. Data warehouse offloading

The Data warehouse should be in major companies for many years. With the exponential growth of data, the DWHs are reaching their capacity limits and batch windows are also increasing putting at risk the SLA. One approach is to migrate heavy ETL and calculation workloads into Hadoop to achieve faster processing time, lower costs per stored data, and free DWH capacity to be used in other workloads.

Here we have two major options:

- One option is to load raw-data from OLTP into Hadoop, then transform the data into the required models, and finally move the data into the DWH. We can also extend this scenario by integrating semi-structured, un-structured, and sensor-based data sources. In this sense, the Hadoop environment acts as an Enterprise Data Lake.
- Another option is moving data from the DWH into Hadoop using Kafka (real Tim messaging services) to do pre-calculations, and then the result is stored in data marts to be visualized using traditional tools. Because the storage cost on Hadoop is much lower than on a DWH, we can save money and keep the data for a longer time. We can also extend this use case by taking advantage of analytical power and creating predictive models using Spark ML Lib or R language or cognitive computing using BDB Predictive Tool to support future decisions of the business.

Use Case 02. Near real-time analytics on data warehouse

In this scenario, we have used Kafka as a streaming data source (front-end for real-time analytics) to store the incoming data in the form of events/messages. As part of the ingestion process, the data is stored directly on the Hadoop file system, or some scalable, fault tolerant, and distributed databases such as Cassandra (provided in BDB) or HBase or something else. Then the data is computed and some predictive models are created using BDB Predictive Tool. The result is stored in BDB's Data Store (Elastic Search based tool) for improving the searching capabilities of the platform, the Predictive models can be stored in BDB Predictive Tool (R, Scala, Shell Scripts, Python based code can be written) and the result of calculations can be stored in Cassandra. The data can be consumed by traditional tools as well as by Web and mobile applications via API Rest.

This architectural design has the following benefits:

- Add near real-time processing capabilities over batch-oriented processing
- Lower the latency for getting actionable insights, impacting positively on the agility of the business on making decisions.
- Lower the storage cost significantly comparing to traditional data warehousing technologies because we can create commodity and low-cost cluster of data and processing nodes on premise and in the cloud.

- This architecture is prepared to be migrated to the cloud and take advantage of elasticity, so adding computing resources when the workload is increasing and relieving computing resources when they're not needed.
- Once this migrates to the cloud it can address the analytics needs of Hundreds and thousands of my customers with much lower efforts. Horizontal scalability is extremely high.
- Business can think of increasing more number of users by providing subscription base services.
-

BDB Data Pipeline Use Case (02) for Education Industry.

Introduction

Big data architectures are reliable, scalable, and completely automated data pipelines. Choosing architecture and building the appropriate big data solution has never been easier.

Lambda Architecture

We are following Lambda architecture here because it is fault-tolerant and linearly scalable. We choose this because of the below given reasons:

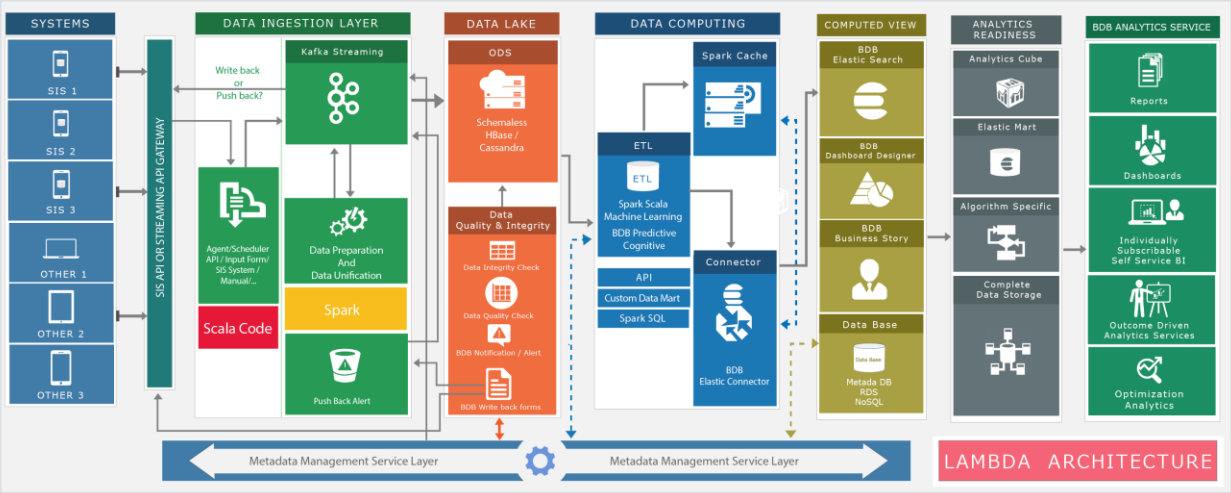
- Data is served by different applications to both the batch layer and the speed layer
- The batch layer manages an append-only dataset and pre-computes the batch views
- The batch views are indexed by the serving layer for low-latency, ad-hoc queries
- The speed layer processes recent data and counterbalances the lower speed of the batch layer
- Queries are answered by combining results from both batch views and real-time views

Let us assume you have different Source Systems and you want to bring those systems data to Data Lake through Streaming engines. Also, most of your systems are in the same domain but their data format is different.

In this case, you want to transform all those data to a unified format based on domain and store it in Data Lake. We can call it as organized data storage in Data Lake.

Once data is there in Data Lake, you may want to do different computation on top of that data and store the result somewhere for analysis or analytical or reporting purpose. Our system provides Computation framework to do this operation. This computation framework helps you to read data from Data Lake, and perform necessary calculations (if needed). The result can be stored in the Analytics store or Report Store. From analytics store, various tools/applications will be reading data to create reports/dashboards and generate data exports. API is provided to read data directly from the Data Lake (if required).

All the modules are API driven so that technology is hidden behind the API and this helps us to design the system loosely coupled.



How different Systems data can be send to this platform

Data can be sent to this platform through the below given options:

- BDP (BizViz Big Data Platform) provides different buckets for accepting data from customer. The customer can export the data as CSV/Excel and dump those data to our FTP/SFTP/S3 buckets provided for the same. These buckets are then monitored by our system to read the data and broadcast it to our streaming engine.

Note: In this case, the customer should follow certain naming conventions recommended by our system cookbook.

- BDP provides API for accepting data to the platform. The customer can write data to our API that can be forwarded to the streaming engine by our API.

Note: In this case, the customer should follow certain rules recommended by our system cookbook.

- BDP provides different Agents for fetching data directly from customer database. If the customer allows, system agents can fetch the data from their database and broadcast to our streaming engine.

Note: In this case, the customer should follow certain rules recommended by BDP cookbook.

How the Unification, Data quality has been taken care

Data ingestion to the Data Lake should be there once customer data is available in stream engine. In BDP we are calling it as data unification process.

Sometimes data comes with references and before moving this data to Data Lake, we may need a lookup process to find out rest of the part of data and polish it. Sometimes we may need to add some extra columns to the data as a part of data polishing. All these processes are happening at unification engine.

Sometimes data comes with some lookup reference, which is not available in Data Lake. In this scenario, system can't push the data to Data Lake and system will keep this information in the raw area. There will be a scheduled process which will be working on top of these raw data and move it to refined area later.

Sometimes data came with missing information, which is very important for the Data Lake. This data also will keep in raw area and will move to refined area later once information is available.

All these processes are handled by unification and Data Quality Engines. BDP Data Quality Engine will generate alert to the customer about raw data, which will help them to correct their data in their system.

How Meta Data Management layer works here

All above process need certain rules and instructions. BDP is a framework which works as per the input rules and given instructions to the system.

E.g. Attendance of Students is coming to Stream engine which has "Student ID" as a reference. In Data Lake, Attendance object needs "Student Name" and "Student Gender" as standard. MDS provides facility to configure lookup to "Student" object and fetch this two information. In another side, Attendance data came without "Student Id" which is mandatory information for the Data Lake. We need a quality check on data to make sure Data Lake has polished data. MDS provides facility to do this as configuration.

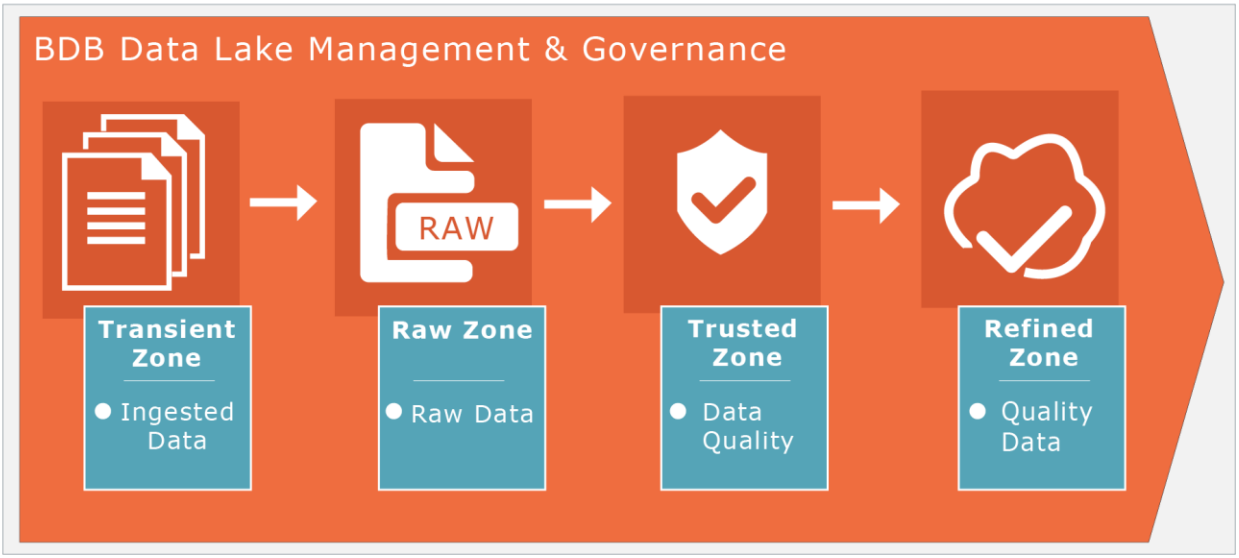
These rules and instructions are called as Metadata. BDP has an API driven Metadata Service Layer (MDS) with the facility to accept the rules and instructions from the administrator. BDP framework has the capability to work with these rules and instructions.

- Metadata information is key for end to end data governance.

How data lake design and implementation is planned

Data Lake is the major part of our platform. BDP Data Lake is a storage repository that holds a vast amount of raw data in its native format until it is needed. While a hierarchical data warehouse stores data in files or folders, a data lake uses a flat architecture to store data. Each data element in a lake is assigned a unique identifier and tagged with a set of extended metadata tags. When a business question arises, the Data Lake can be queried for relevant data, and that smaller set of data can then be analyzed to help answer the question.

BDP Data Lake is often associated with Hadoop-oriented object storage.An organization's data is first loaded into the Hadoop platform, and then BDP compute engine is applied to the data where it resides on Hadoop's cluster nodes of commodity computers.



How ETL and Compute layers (Data Pipeline) are being worked

Compute engine follows UBER architecture. We provide components like SCALA, JAVA, TALEND, SHELL, BDB ETL etc. which help end users to come up with their on-computation logic and integrate with our system.

We use SPARK ecosystem in BDP. SPARK provides distributed computing capability for BDP.

Compute Engine can read data from the data Lake through API and it can perform all the required transformations and calculations which are necessary for you and allow you to store the result at any analytical or computing storage.

This framework follows pluggability, which allow 3rd party engines to plug into it and complete the computation. Further, it can also be used as ETL.

All the rules and instructions needed by Compute Engine are managed in MDS layer.

Compute engine needs certain rules and instruction to do all these operations. All these rules and instructions are managed in MDS layer. In short, BDP is a highly scalable and powerful system completely driven through MDS.

We have robust MDS UI, which provides facility to create Compute (Data Pipeline) workflows and schedule it on top of Data Lake. All the MDS governance can be managed through the MDS UI.

Monitoring

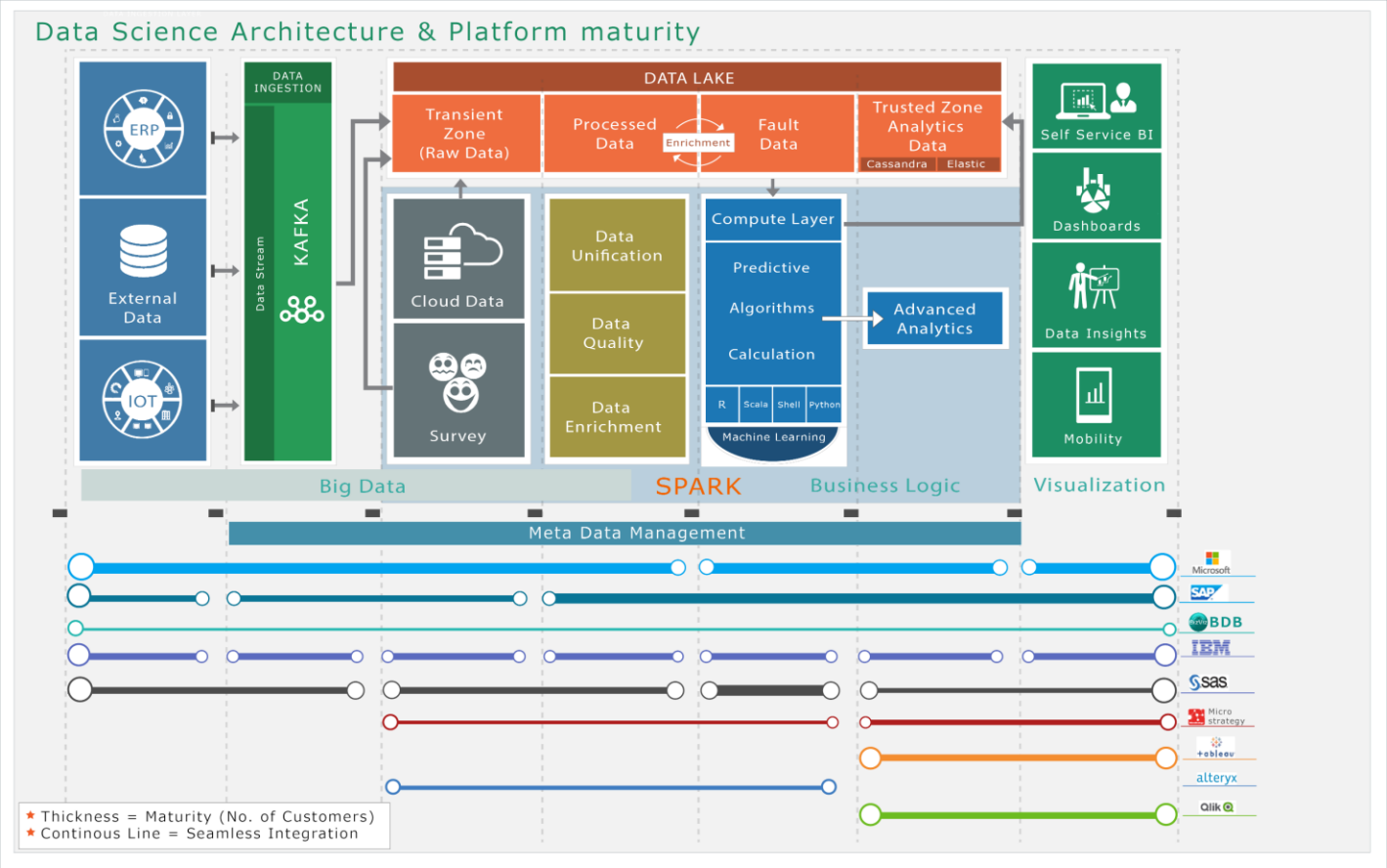
BDB has very good ecosystem monitoring, error handling, and troubleshooting facility with the help of various frameworks like AMBARI.

Scalability, Benefits & ROI of the above Solution

- BDB system works in one platform/integrated ecosystem on Cloud rather than working in SILOS like other vendors. There is no need for any other product with BDB Platform, it comes with Big Data Pipeline, ETL, Query Services, Predictive Tool with Machine Learning, Survey Platform, Self Service BI, Advanced Visualization and Dashboards.
- Provided real-time attendance & assessment reporting to its entire 40 million active user bases – Teachers, Students, Administrators etc.
- Substantial Reduction in Cost of Operation
    - Converted all Compliance Reporting into BizViz Architecture to reduce the Costs by ¼
    - Saved 2 years of time and multi - million Dollars of effort to create an Analytics Platform.
- Market Opportunity
    - Instead of selling others licensing package, customers can sell their own Subscription Services and Licenses – Market opportunity 10x of investment.
- Additional Analytics Services revenue with BDB as Implementation team.
- The solution can be extended for other 60 million passive Users, also Data Lake Services or Data as Services can be provided to other School Users. The entire solution can be extended with Big Data Analytics, Predictive, and Prescription Analytics like Student Dropout and Teacher Attritions by trapping critical Student and Teacher related parameters that are built into the system. This itself gives another 20x opportunity to the customer.

Mapping BDB into the BI & Analytics Ecosystem –

A picture is equivalent to 1000 words; the diagram given below is self-expressive of where BDB stands in the BI and Analytics Ecosystem. The other products have been shown to use them as a template to place BDB in the right perspective. Our Pre-Sales team that has studied multiple BI products over a period has created the diagram. The diagram depicts that BDB owns a single installation to provide for all elements of Big Data.



The following image displays our current set of thought process and feature based maturity vis-à-vis global names. We would like to repeat that we are competing against the best names in the industry and our closure after POC round is 100%.