EasyChair Preprint

№ 2747

Towards a big data architecture for heterogeneous data sources

Latifa Rassam, Ahmed Zellou and Taoufiq Rachad

February 22, 2020

Towards a big data architecture for heterogeneous data sources

Latifa RASSAM
SPM Team, ENSIAS.
Mohammed V *University in Rabat*
Morocco
rassamlatifa@gmail.com

Ahmed ZELLOU
SPM Team, ENSIAS.
Mohammed V University in Rabat
Morocco
ahmed.zellou@Um5.ac.ma

Taoufiq RACHAD
SPM Team, ENSIAS.
Mohammed V University in Rabat
Morocco
t.rachad@um5s.net.ma

Abstract: The use of freely available online data is rapidly increasing, as companies have detected the possibilities and the value of these data in their activities. In particular, data are seen as interesting and heterogeneous as they can, when properly treated, assist in achieving user insight into activities decision making. However, the unstructured and uncertain nature of this kind of big data presents a new kind of challenge: is there any standard architecture for big data systems? This paper contributes to addressing this challenge by introducing a comparative architectural study to end with a unified architecture that manage data in each processing phase of the big data pipeline.

Keywords: Architecture, Big Data, Comparative study.

I. INTRODUCTION

A Big data architecture [1] is designed to handle the ingestion, processing, and analysis of data that is too large or complex for traditional database systems. [2] The threshold at which organizations enter into the big data realm differs, depending on the capabilities of the users and their tools. For some, it can mean hundreds of gigabytes of data, while for others it means hundreds of terabytes. As tools for working with big data sets advance, so does the meaning of big data. More and more, this term relates to the value we can extract from our data sets through advanced analytics, rather than strictly the size of the data, although in these cases they tend to be quite large[3]. Big data solutions typically involve one or more of the following types of workload: Batch processing of big data sources at rest, Real-time processing of big data in motion and Interactive exploration of big data.

Most big data architectures include some or all of the following components: data sources, data storage, batch processing, real-time message ingestion, stream processing, analytical data store, analysis and reporting and orchestration.

After researches we succeeded in having almost six big data architectures Master / slave architecture, Data Lake, lambda, kappa, Microsoft and IOT architecture, among which we chose the four architectures which we will see later, in detail, in the following section for the following reasons: [4]

- The IOT architecture, kappa, lambda and the Microsoft architecture are among the most trending big data architectures in the Information Technologies domain.

II. BIG DATA ARCHITECTURES

In this article, we will first tackle the six different most big data architectures, and then carry out a comparative study between the four most convincing architectures based on the most powerful criteria in order to develop a new architectural model for heterogeneous data sources.

A. Microsoft's architecture

The following architecture was introduced by Microsoft in 2014 as their big data architecture.

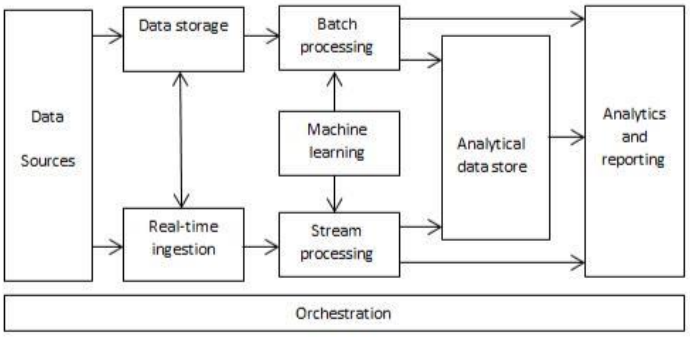

Figure 1: Microsoft architecture

The architecture mentioned above includes all of the following components:

- Data sources : all big data solutions start with one or more data sources, include:
  - ✓ Application data stores such as relational or others databases
  - ✓ Static files produced by applications, such as web server log files
  - ✓ Real-time data sources, such as IOT devices
- Data storage: data for batch processing operations is typically stored in a distributed file store that can hold high volumes of large files in various formats. This kind of store is often called a data lake [5].

- Batch processing: because the data sets are so large, often a big data solution must process data files using long-running batch jobs to filter, aggregate, and prepare the data for analysis. Usually these jobs involve reading source files, processing them and writing the output to new files [5].
- Real-time data ingestion: If the solution includes real-time sources, the architecture must include a way to capture and store real-time data for stream processing. This might be a simple data store, where incoming messages are dropped into a folder for processing [5].
- Stream processing: After capturing real-time data, the solution must process them by filtering, aggregating, and preparing the data for analysis.
- Analytical data store: Many big data solutions prepare data for analysis and then serve the processed data in a structured format that can be queried using analytical tools...
- Analysis: To empower users to analyze the data
- Reporting: the architecture may include a data modeling layer, such as a multidimensional OLAP cube or tabular data model in Azure Analysis Services like Azure
- Orchestration: the main role of this phase is to automate these workflows, we can use an orchestration technology such Apache, Oozie or Sqoop…

One drawback to this approach is that it introduces latency if processing takes a few hours, a query may return results that are several hours old. Ideally, we would like to get some results in real time, perhaps with some loss of accuracy, and combine these results with the results from the batch analytics [6].

B. Lambda architecture

The lambda architecture, first proposed by Nathan Marz in 2015, addresses this limitation by creating two paths for data flow. All data coming into the system goes through these two paths as mentioned in the following figure: [7]

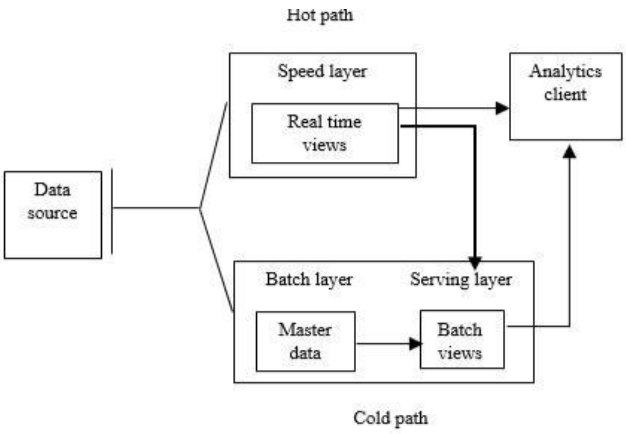

Figure 2: Lambda architecture

This architecture contains the following components:

- A batch layer (cold path) stores all of the incoming data in its raw form and performs batch processing on the data. The result of this processing is stored as a batch view.

- A speed layer (hot path) analyzes data in real time. This layer is designed for low latency, at the expense of accuracy.

The data that enters the hot path is constrained by the latency requirements imposed by the speed layer, so that it can be processed as quickly as possible. This often requires a compromise between a certain level of accuracy and data that is ready to be available as soon as possible. For example, consider an IOT scenario in which a large number of temperature sensors send telemetry data. The speed layer can be used to process a sliding time window of incoming data.

Data entering the cold track, on the other hand, are not subject to the same low latency requirements. This allows for high-precision calculation on large datasets, which can be very time-consuming.

Finally, the hot and cold paths converge at the level of the analysis client application. If the customer needs to display punctual, but potentially less accurate, data in real time, he will get his results from the quick access path. Otherwise, it will select the results of the cold path to display slower but more accurate data. In other words, the hot path contains data for a relatively short time window, after which the results can be updated with more accurate data from the cold path.

A drawback to the lambda architecture is its complexity. Processing logic appears in two different places, the cold and hot paths, using different frameworks. This leads to duplicate computation logic and the complexity of managing the architecture for both paths [8].

C. Kappa architecture

The kappa architecture was proposed by Jay Kreps as an alternative to the lambda architecture in 2016. It has the same basic goals as the lambda architecture, but with an important distinction: All data flows through a single path, using a stream processing system [9].

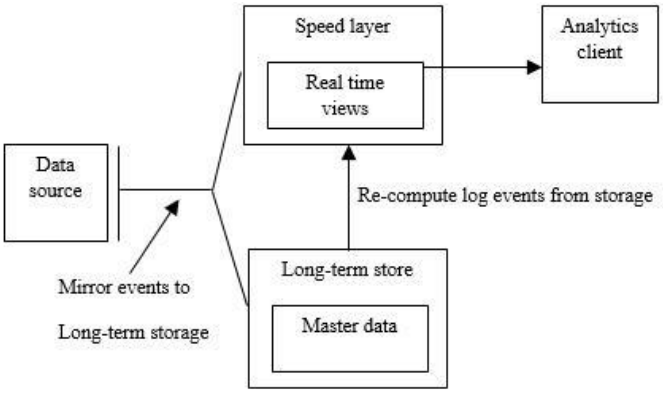

Figure 3: Kappa architecture

There are some similarities to the lambda architecture's batch layer, in that the event data is immutable and all of it is collected, Instead of a subset. The data is ingested as a stream of events into a distributed and fault tolerant unified log.

These events are ordered, and the current state of an event is changed only by a new event being appended. Similar to a lambda architecture's speed layer, all event processing is performed on the input stream and persisted as a real-time view.

D. IOT *architecture*

From a practical viewpoint, Internet of Things (IOT) represents any device that is connected to the Internet. This includes PC, mobile phone, smart watch, smart thermostat, smart refrigerator, connected automobile, heart monitoring implants, and anything else that connects to the Internet and sends or receives data [10]. The number of connected devices grows every day, as does the amount of data collected from them. Often this data is being collected in highly constrained, sometimes high-latency environments. In other cases, data is sent from low-latency environments by thousands or millions of devices, requiring the ability to rapidly ingest the data and process accordingly.

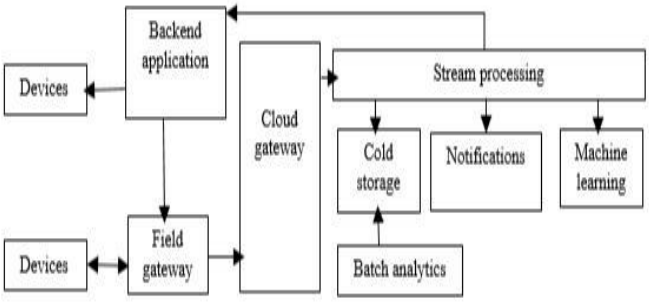

Figure 4: IOT architecture

The cloud gateway: it ingests device events at the cloud boundary, using a reliable, low latency messaging system.

Devices might send events directly to the cloud gateway, or through a field gateway. A field gateway is a specialized device or software, usually collocated with the devices, that receives events and forwards them to the cloud gateway. The field gateway might also preprocess the raw device events, performing functions such as filtering, aggregation, or protocol transformation [11]. After ingestion, events go through one or more stream processors that can route the data for example, to storage or perform analytics and other processing.

The following are some common types of processing.

- Writing event data to cold storage, for archiving or batch analytics.
- Hot path analytics, analyzing the event stream in (near) real time, to detect anomalies, recognize patterns over rolling time windows, or trigger alerts when a specific condition occurs in the stream [13].
- Handling special types of data from devices such as notifications and alarms.
- Machine learning: to simplify the tasks asked by a process.

E. Master / slave architecture

It's an architecture of distributed processing in which a machine called master node acts like a central machine, while a set of machines called slave nodes execute the tasks which are sent by the master[14].

The architecture below describes decently the slave node's architecture during the data process phase.

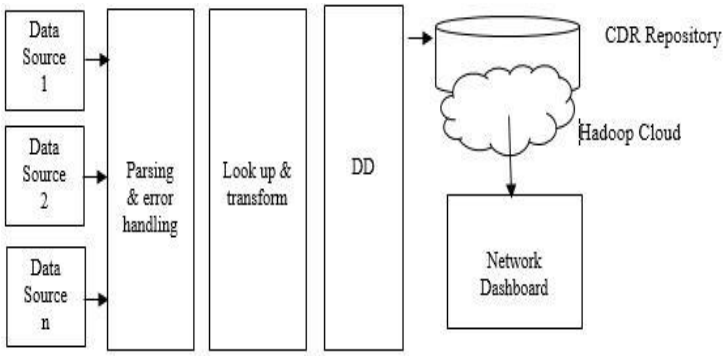

Figure5: Master/Slave Architecture

The architecture cited above contains the following components and activities:

- Data sources: This includes all types of data.
- Parsing and error handling: this phase allows the analysis of the input data and the handling of the various errors established during the execution of this process.
- Lookup and transformation: the objective of this phase is to research, filter and transform the data in question.
- De-Duplication: it allows data duplication in order to prepare it for ingestion.
- CDR Repository: a Compact Disc Recordable Repository in which data is stored.
- Hadoop cloud: it is the component that takes care of the part of data processing.
- Network Dashboard: it manages analytic and reporting results.
- DD: De-Duplication of data.

F. Data lake architecture

It's an architecture based firstly, on a store repository that can store a large amount of unstructured, semi-structured and structured data, secondly on HDFS as the main storage component [15].

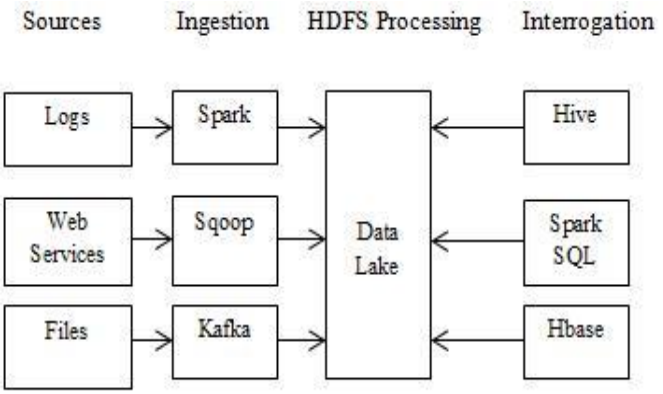

Figure 6: Data Lake architecture

- Data sources: which includes log files, web services and normal files.
- Ingestion phase: it includes multiple data ingestion tools depending on the data source type, to act as a buffer for messages.
- HDFS Processing: the abbreviation of the Hadoop Distributed File System, its role is to ensure the data storage within the data lake, in this phase, each piece of data is stored in several places and an therefore be retrieved under any circumstances. Likewise, this replication helps to combat potential data corruption.
- Interrogation: this phase refers to the manual process of querying the source data, to do that there multiple tools depending on the type of data, such as Hive, spark sql and Hbase.

III. COMPARATIVE STUDY

A. *Criteria's choice*

According to researches, the concept of Big Data can be defined by the four Vs: Volume, Variety, Veracity and velocity [16]. These four dimensions characterize it and distinguish big data from ordinary data, there are also some others criteria that could help in characterizing a big data system such as: Latency, Data storage, Analytics and reporting, Adapter between data source and data storage, two way information processing, Validity, Visualization, and Information processing, Vulnerability and Volatility… We will work in this part by only eight criteria among the fourteen criteria already established above, for this, it has been important to choose only the most characterizing criteria in terms of storage optimization, request processing and time Answer.

- Latency: defines the time interval between the stimulation and the response of a request.
- Storage: the capacity to store processed data.
- Veracity: defines the data reliability.
- Variety: the type of the used data source as an input for big data systems.
- Volume : the quantity of data to process
- Adapter: a component placed between data source and data storage that could help in adapting data sources during the phase before aggregation.
- Processing: the possibility of processing different types of data based on two paths, a cold one which refers to batch processing and a hot one which means the speed processing.
- Analytics: include the inside/outside systems helping in analysis and reports generation.

B. *Benchmarking Study*

The table below shows the result of the comparative study lead by eight of the criteria quoted previously:

TABLE I. **BENCHMARKING STUDY OF BIG DATA'S ARCHITECTURES**

	Microsoft architecture	Lambda architecture	Kappa architecture	IOT architecture
Latency	Very high latency	Normal latency	Very high latency	It depends on the number of connected devices
Storage	Dual Storage	Dual Storage	Dual Storage	single storage
Veracity	Reliable Data	Not really reliable	Very high latency	Reliable Data
Variety	All sources	All sources	All sources	IOT Devices
Volume	Big	Standard	Big	Big
Adapter	No adapter	No adapter	Mirror events to long term storage	A field gateway and a cloud gateway
Processing	One way	Two ways: Cold and hot paths	One way	One way
Analytics	Included system	Included system	Included system	Back end application

C. *Feedback*

After the comparative study carried out on the four architectures, and based on the criteria chosen previously we can conclude that each architecture among the four has its own advantage, notwithstanding some drawbacks. We found that the latency of the IOT and lambda architecture is much less reduced than that of the two other architectures, but for the data storage we note that the IOT architecture is the only one which tends to carry out a single storage since it processes data in real time unlike other architecture.

However it does not have an internal reporting tool like other architectures, yet it can guarantee the reliability of the data transmitted and this is the case for kappa and Microsoft architectures thanks to the concept of one-way processing unlike lambda architecture which processes data on two ways, as for the factor of variety, the IOT architecture is the only one who could deal with just IOT devices sources as input unlike the three others that's accept all types of data in their system inputs.

With regard to the adapter between the data source and the data storage, we can conclude from the previous comparative study that only the kappa and IOT architectures have this adapter unlike other architectures.

By way of conclusion, the IOT architecture has many advantageous points in the face of its drawbacks, such as the fact that this architecture does not have an internal analysis and reporting tool.

To carry out the improvement of the IOT architecture, adding an adapter between the data source, which include the IOT devices inputs and other data, and the field gateway appeared the most appropriate to our context, in order to include all the types of data, this solution gives rise to two different paths, the first path, through which we added a batch processing component and an analysis tool, will be devoted to all types of data except the IOT ones which must follow the second path as mentioned bellow.

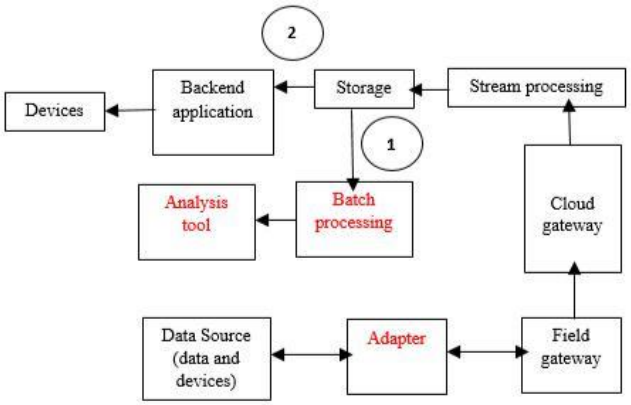

Figure 7: the proposed architecture

IV. CONCLUSION AND PERSPECTIVES

In fact the objective of this paper consisted on the comparative study of big data architectures based on the most characterizing criteria so that we can conclude to a unified architecture for all data types. To achieve these goals, we started with the definition of a big data architecture. Then, we have discussed four among seven architectures by defining each one of them based on their schema formalism and different components, ending with a benchmarking study of the fourth architectures that makes the choice of the IOT architecture as the most powerful one.

As a result of this study, we came up with a new IOT architecture that contains more components so that all heterogeneous data sources can benefit from this system.

As for the perspectives, we will improve more the IOT architecture basing on just one component, if so, we will look for a cloud gateway that will allow us to ingest heterogeneous sources at the edge of the cloud, using a low latency.

REFERENCES

[1] Gunasegaram Manogaran and Daphne Lopez "A survey of big data Architectures and machine learning algorithms in healthcare" Int. J. Biomedical Engineering and Technology, Vol. 25, Nos. 2/3/4, 2017

[2] Jeffrey S. Saltz "The Need for New Processes, Methodologies and Tools to Support Big Data Teams and Improve Big Data Project Effectiveness" in Big Data (Big Data), 2016 IEEE International Conference on, 2015, pp. 2066- 2071: IEEE.

[3] Xing He, Qian Ai, Member, IEEE, Robert C. Qiu, Fellow, IEEE, Wentao Huang, Longjian Piao and Haichun Liu "A Big Data Architecture Design For Smart Grids Based on Random Matrix Theory". arXiv:1501.07329v4 [stat.ME] 2015

[4] Anne Immonen, Pekka Paakkonen and Eila Ovaska "Evaluating the Quality of Social Media Data in Big Data Architecture" 2015. IEEE Access, the journal for rapid open access publishing.

[5] Docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/# components-of-a-big-data-architecture.

[6] Maryam Najafabadi1, Flavio Villanustre, Taghi Khoshgoftaar, Naeem Seliya1, RandallWald1 and Edin Muharemagic "Deep learning applications and challenges in big data analytics", page two 2015. Journal of big data, a springer open journal.

[7] Mariam Kiran , Peter Murphy, Inder Monga, Jon Dugan and Sartaj Singh Baveja "Lambda architecture for cost-effective batch and speed big data processing" 2015. IEEE International Conference on Big Data.

[8] Maryam Najafabadi1, Flavio Villanustre, Taghi Khoshgoftaar, Naeem Seliya1, RandallWald1 and Edin Muharemagic "Deep learning applications and challenges in big data analytics", page Three 2015. Journal of big data, a springer open journal.

[9] Theo Zschörnig, Robert Wehlitz and Bogdan Franczyk "A Personal Analytics Platform for the Internet of Things Implementing Kappa Architecture with Microservice-based Stream Processing" first page ICEIS 2016 - 19th International Conference on Enterprise Information System.

[10] Theo Zschörnig, Robert Wehlitz and Bogdan Franczyk "A Personal Analytics Platform for the Internet of Things Implementing Kappa Architecture with Microservice-based Stream Processing «third page. ICEIS 2016 - 19th International Conference on Enterprise Information System.

[11] Hamid Bagheri "Big Data: challenges, opportunities and cloud based solutions" 2015. International Journal of Electrical and Computer Engineering (IJECE).

[12] Bas Geerdink "A Reference Architecture for Big Data Solutions: Introducing a model to perform predictive analytics of enterprise data, combined with open data sources, using big data technology" book, 2013.

[13] Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani, (Senior Member, IEEE), Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, AND Ibrar Yaqoob "Big IOT Data Analytics: Architecture, Opportunities, and Open Research Challenge" 2017 IEEE Access, open access journal.

[14] Rajeev Gupta, Himanshu Gupta, and Mukesh Mohania Cloud "Computing and Big Data Analytics: What Is New from Databases Perspective?" Springer-Verlag Berlin Heidelberg 2012.

[15] Bill Inmon "Data Lake architecture: designing the data lake and avoiding the garbage dump" Book, 2016.

[16] Valero Sperli "Benchmarking Big Data Architectures for Social Networks Data processing using Public Cloud Platforms" 2016. Future Generation Computer System Conference.