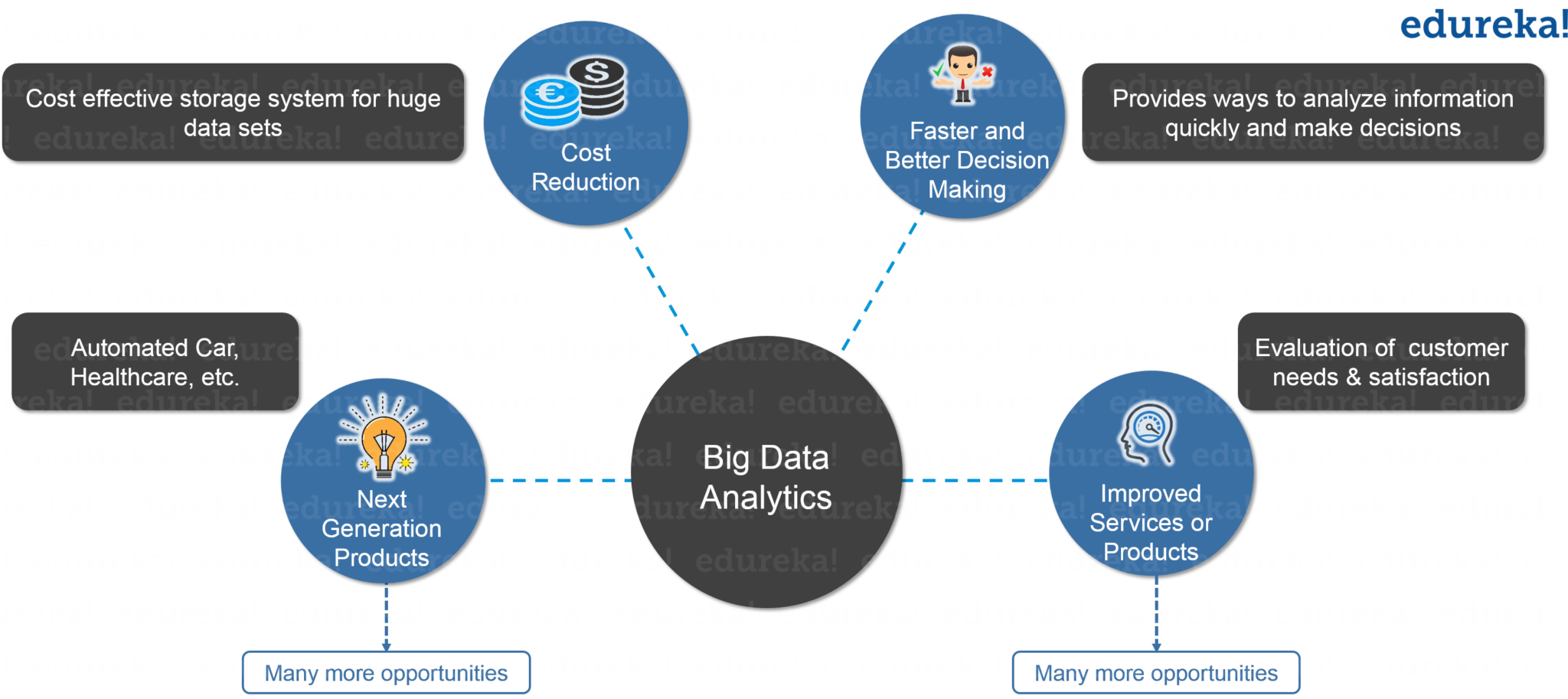
As to understand What is Hadoop, we have to first take a deep dive into issues related to Big Data and traditional system processing. After that, I will tell you what is Hadoop and how Hadoop is a solution to Big Data associated problems. At last, we will look at a CERN case study to highlight the benefits of using Hadoop. In the previous blog i.e. [***Big Data Tutorial***](http://www.edureka.co/blog/big-data-tutorial?utm_source=blog&utm_medium=related-posts&utm_campaign=what-is-hadoop), we already discussed about Big Data in detail and the challenges with Big Data. In this blog, we are going to discuss:

* [Problems with Traditional Approach](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#ProblemsWithBigData)
* [Evolution of Hadoop](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#HadoopHistory)
* [Hadoop](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#WhatIsHadoop)
* [Hadoop-as-a Solution](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#HadoopAsASolution)
* [When to use Hadoop?](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#HadoopUseCases)
* [When not to use Hadoop?](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#NotToUseHadoop)
* [CERN Case Study](https://www.edureka.co/blog/what-is-hadoop?utm_source=blog&utm_medium=left-menu&utm_campaign=big-data-tutorial#CERNCaseStudy)

Technology has evolved a lot, we can use Big data as an opportunity. Earlier, we were spending too much money on storage. Until Big Data came into the picture, we never thought of using commodity hardware to store and manage our data which is both reliable and feasible as compared to the costly servers. But later, organisations realized that they are getting lot of benefits by Big Data Analytics, as you can see in the below image.

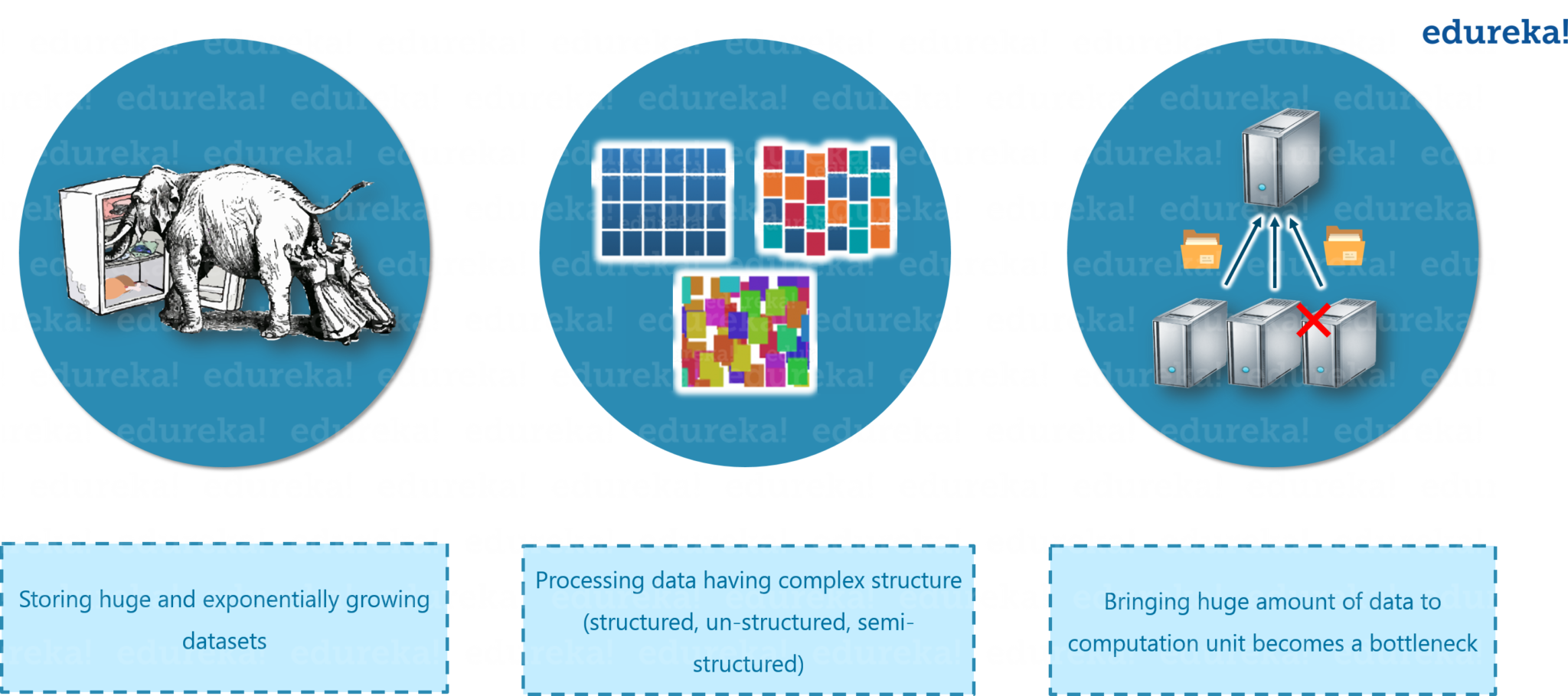
**Figure**: *What is Hadoop – Benefits of Big Data Analytics*

So, let us move ahead and know the problems associated with traditional approach in en-cashing Big data opportunities.

**Problems with Traditional Approach**

In traditional approach, the main issue was handling the heterogeneity of data i.e. structured, semi-structured and unstructured. The RDBMS focuses mostly on structured data like banking transaction, operational data etc. and Hadoop specializes in semi-structured, unstructured data like text, videos, audios, Facebook posts, logs, etc. RDBMS technology is a proven, highly consistent, matured systems supported by many companies. While on the other hand, Hadoop system technology is developed and is in demand due to Big Data, which mostly consists of unstructured data in different formats.

Now let us understand what are the major problems associated with Big Data. So that, moving ahead we can understand how Hadoop emerged as a solution.



**Figure**: *What is Hadoop – Problems with Big Data*

***So, the first problem is storing the colossal amount of data.*** Storing this huge data in a traditional system is not possible. The reason is obvious the storage will be limited to one system and the data is increasing in tremendous rate.

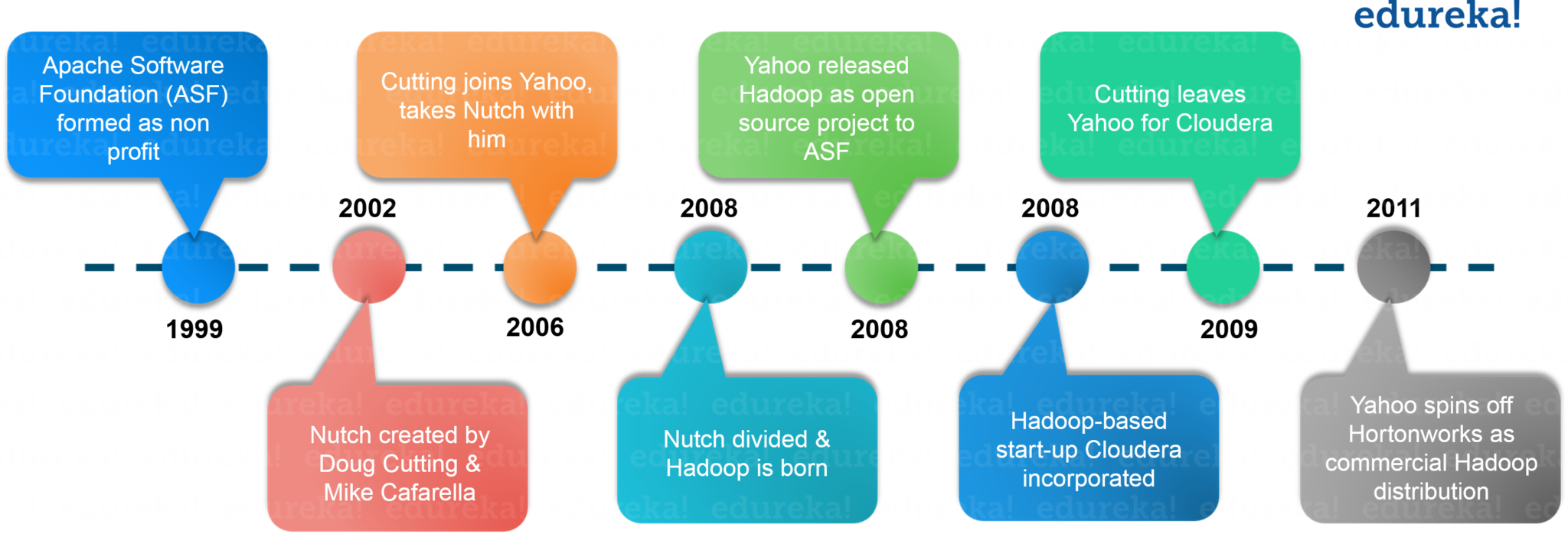
***Second problem is storing heterogeneous data.*** Now we know storing is a problem, but let me tell you it is just one part of the problem. Since we discussed that the data is not only huge, but it is present in various formats as well like: Unstructured, Semi-structured and Structured. So, you need to make sure that you have a system to store these varieties of data, generated from various sources.

***Now, let’s focus on third problem, which is accessing and processing speed***. The hard disk capacity is increasing but disk transfer speed or the access speed is not increasing at similar rate. Let me explain you this with an example: If you have only one 100mbps I/O channel and you are processing say 1TB of data, it will take around 2.91 hours. Now, if you have four machines with four I/O channel for the same amount of data, then it will take 43 minutes approx. Thus for me, accessing and processing speed is the bigger problem than storage of Big Data.

Before understanding what is Hadoop, let us first look at the evolution of Hadoop over a period of time.

[LEARN HADOOP WITH EXPERTS](https://www.edureka.co/big-data-and-hadoop?utm_source=blog&utm_medium=cta-blog&utm_campaign=what-is-hadoop)

**Evolution of Hadoop**

  
In 2003, Google launches project Nutch to handle billions of searches and indexing millions of web pages. Later in Oct 2003 – Google releases papers with GFS (Google File System). In Dec 2004, Google releases papers with MapReduce. In 2005, Nutch used GFS and MapReduce to perform operations. In 2006, Yahoo created Hadoop based on GFS and MapReduce with Doug Cutting and team. You would be surprised if I would tell you that, in 2007 Yahoo started using Hadoop on a 1000 node cluster.

Later in Jan 2008, Yahoo released Hadoop as an open source project to Apache Software Foundation. In Jul 2008, Apache tested a 4000 node cluster with Hadoop successfully. In 2009, Hadoop successfully sorted a petabyte of data in less than 17 hours to handle billions of searches and indexing millions of web pages. Moving ahead in Dec 2011, Apache Hadoop released version 1.0. Later in Aug 2013, Version 2.0.6 was available.

When we were discussing about the problems, we saw that a distributed system can be a solution and Hadoop provides the same. Now, let us understand what is Hadoop.

**What is Hadoop?**

Hadoop is a framework that allows you to first store Big Data in a distributed environment so that you can process it parallely. There are basically two components in Hadoop:

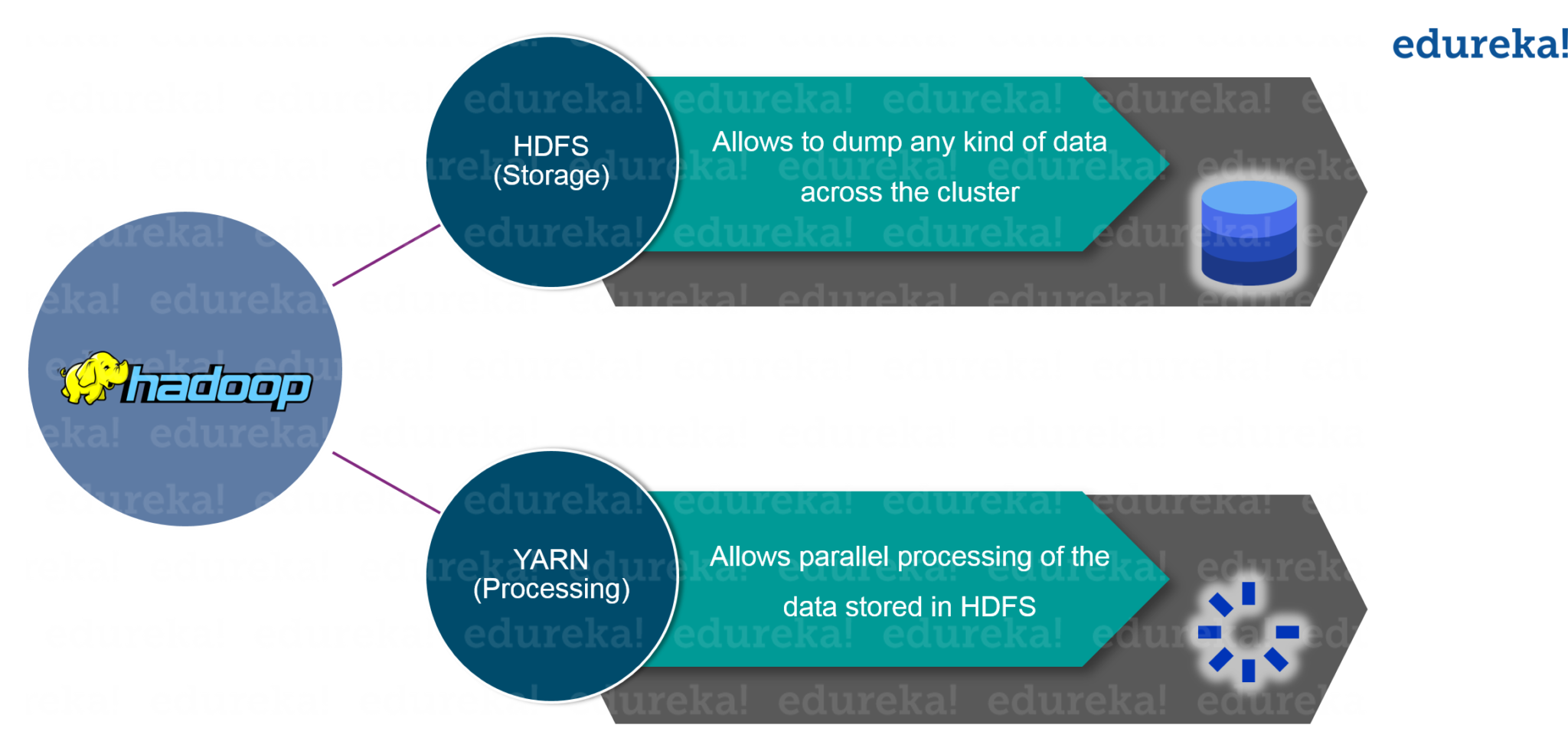


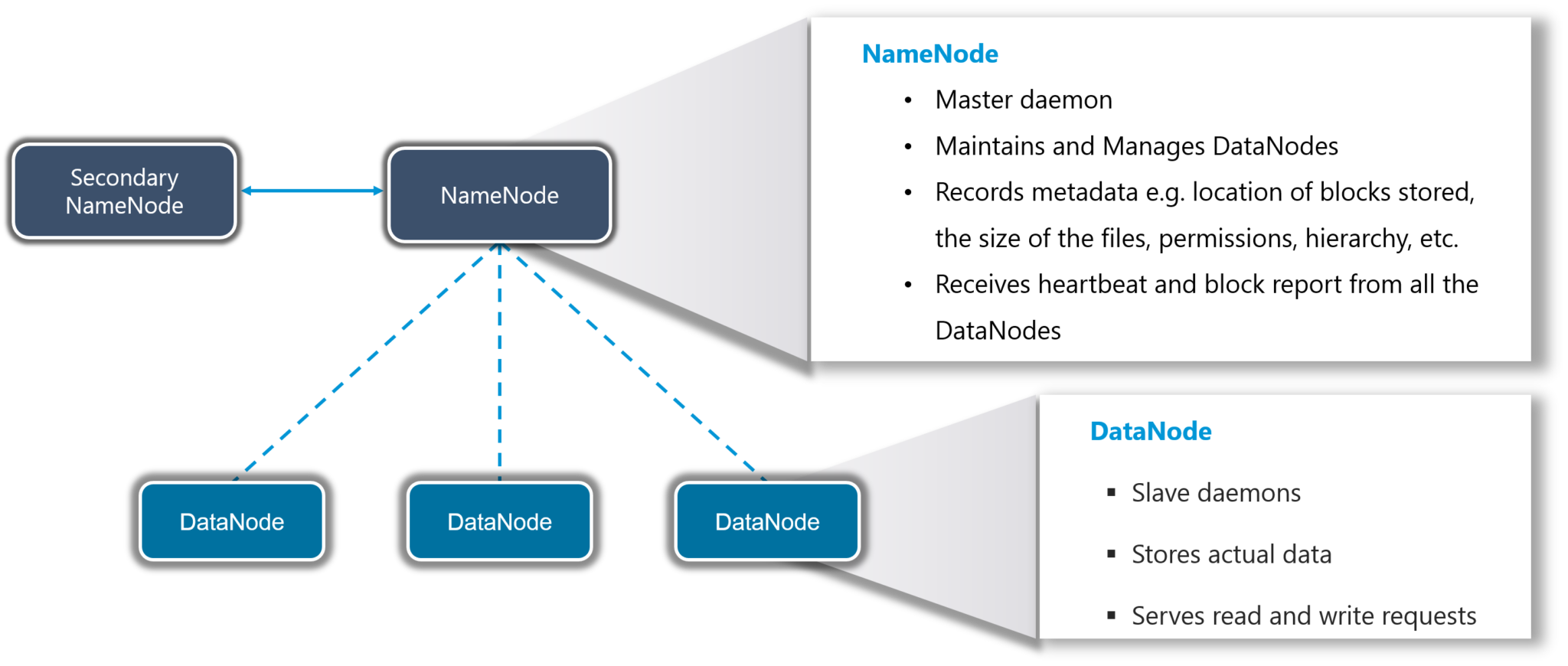
Figure: What is Hadoop – Hadoop Framework

The first one is ***HDFS*** for storage (Hadoop distributed File System) that allows you to store data of various formats across a cluster. The second one is ***YARN***, it is nothing but a processing unit of Hadoop. It allows parallel processing of data i.e. stored across the HDFS.

**What Is Hadoop | Introduction to Hadoop | Hadoop Training | Edureka**

Now, let us dig deep into HDFS and understand it in a better way.

HDFS creates an abstraction of resources, let me simplify it for you. Similar as virtualization, you can see HDFS logically as a single unit for storing Big Data, but actually you are storing your data across multiple nodes in a distributed fashion. Here, you have master-slave architecture.

****

**Figure**: *What is Hadoop – HDFS*

In HDFS, Namenode is a master node and Datanodes are slaves. Namenode contains the metadata about the data stored in Data nodes, like which data block is stored in which data node, where are the replications of the data block kept etc. The actual data is stored in Data Nodes. I also want to add, we actually replicate the data blocks present in Data Nodes, and by default, the replication factor is 3. Since we are using commodity hardware and we know the failure rate of these hardwares are pretty high, so if one of the DataNodes fails, HDFS will still have the copy of those lost data blocks. That’s the reason we need to replicate the data block. You can configure replication factor based on your requirements. You can go through [***HDFS tutorial***](https://www.edureka.co/blog/hdfs-tutorial?utm_source=blog&utm_medium=content-link&utm_campaign=what-is-hadoop) to know *HDFS* in detail.

[CHECK OUT HADOOP BATCHES](https://www.edureka.co/big-data-and-hadoop?utm_source=blog&utm_medium=cta-blog&utm_campaign=what-is-hadoop)

**Hadoop-as-a-Solution**

Let’s understand the how Hadoop provided the solution to the Big Data problems that we just discussed.

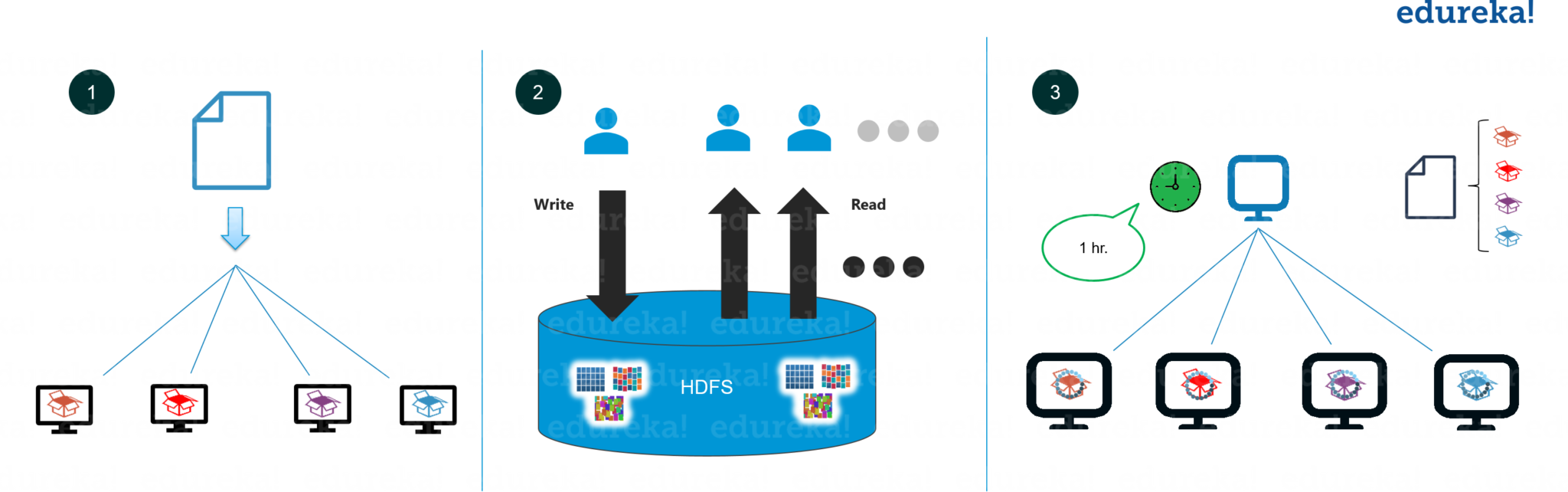
******

Figure: What is Hadoop – Hadoop-as-a-Solution

***The first problem is storing Big data***. HDFS solved it, let’s know how.

HDFS provides a distributed way to store Big data. Your Data is stored in blocks in data nodes and you specify the size of each block. Basically, if you have 512MB of data and you have configured HDFS such that it will create 128 MB of data blocks. So HDFS will divide data into 4 blocks as 512/128=4 and store it across different DataNodes, it will also replicate the data blocks on different DataNodes. Now, as we are using commodity hardware, hence storing is not a challenge.

It also solves the scaling problem. It focuses on ***horizontal scaling*** instead of vertical scaling. You can always add some extra data nodes to HDFS cluster as and when required, instead of scaling up the resources of your data nodes. Let me summarize it for you basically for storing 1 TB of data I don’t need a 1TB system. You can instead do it on multiple 128GB systems or even less.

***Next problem was storing the variety of data***. This problem is also addressed by HDFS.

With HDFS you can store all kinds of data whether it is structured, semi-structured or unstructured. Since in HDFS, there is *no pre-dumping schema validation*. And it also follows write once and read many model. Due to this, you can just write the data once and you can read it multiple times for finding insights.

If you can recall, the ***third challenge was accessing & processing the data faster***. Yes, this is one of the major challenges with Big Data. In order to solve it, we move processing to data and not data to processing. What does it mean? Instead of moving data to the master node and then processing it. In YARN, the processing logic is sent to the various slave nodes & then data is processed parallely across different slave nodes. Then the processed results are sent to the master node where the results is merged and the response is sent back to the client.

In YARN architecture, we have ResourceManager and NodeManager. ResourceManager might or might not be configured on the same machine as NameNode. But, NodeManagers should be configured on the same machine where DataNodes are present.

**YARN** performs all your processing activities by allocating resources and scheduling tasks.

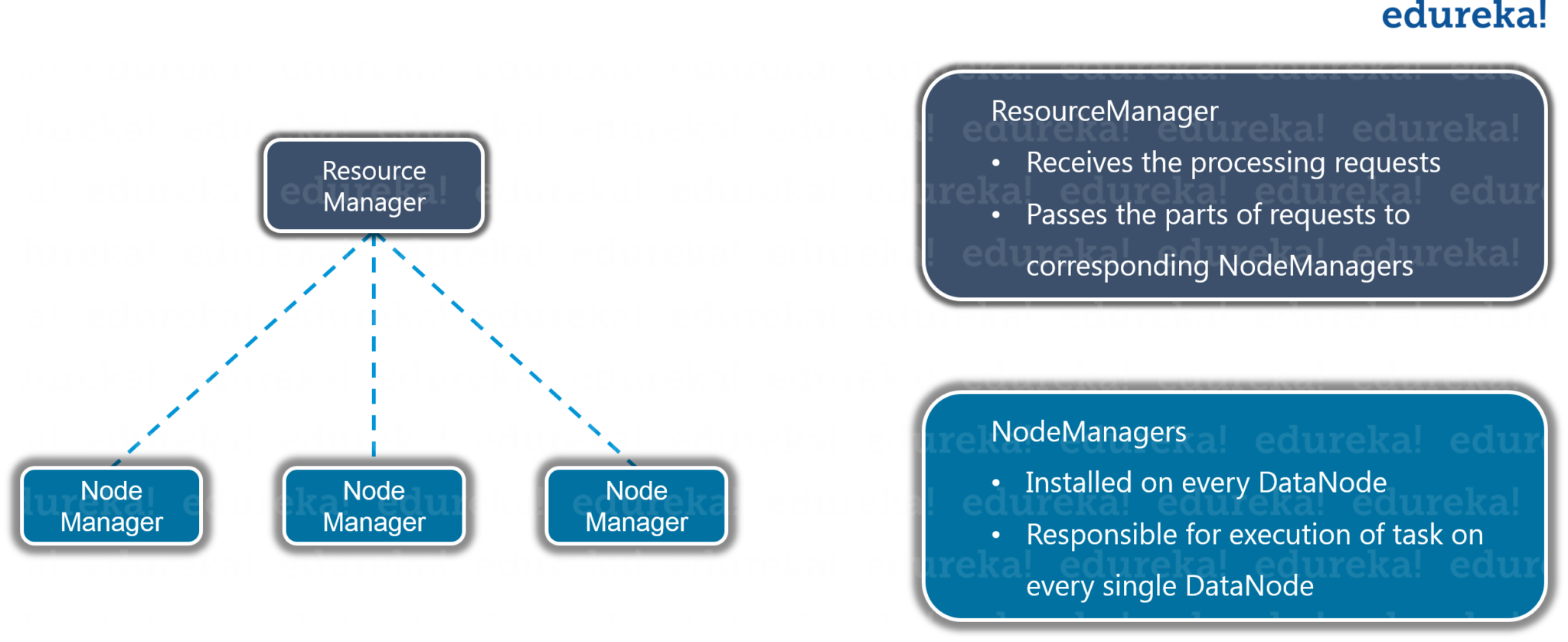


Figure: What is Hadoop – YARN

It has two major components, i.e. ResourceManager and NodeManager.

ResourceManager is again a master node. It receives the processing requests and then passes the parts of requests to corresponding NodeManagers accordingly, where the actual processing takes place. NodeManagers are installed on every DataNode. It is responsible for the execution of the task on every single DataNode.

I hope now you are clear with What is Hadoop and its major components. Let us move ahead and understand when to use and when not to use Hadoop.

**When to use Hadoop ?**

Hadoop is used for:

* *Search* – Yahoo, Amazon, Zvents
* *Log processing* – Facebook, Yahoo
* *Data Warehouse* – Facebook, AOL
* *Video and Image Analysis* – New York Times, Eyealike

Till now, we have seen how Hadoop has made handling big data possible. But there are some scenarios where Hadoop implementation is not recommended.

**When to not to use Hadoop ?**

Following are some of those scenarios :

* *Low Latency data access* : Quick access to small parts of data
* *Multiple data modification* : Hadoop is a better fit only if we are primarily concerned about reading data and not writing data.
* *Lots of small files* : Hadoop is a better fit in scenarios, where we have few but large files.

After knowing the best suitable use-cases, let us move on and look at a case study where Hadoop has done wonders.

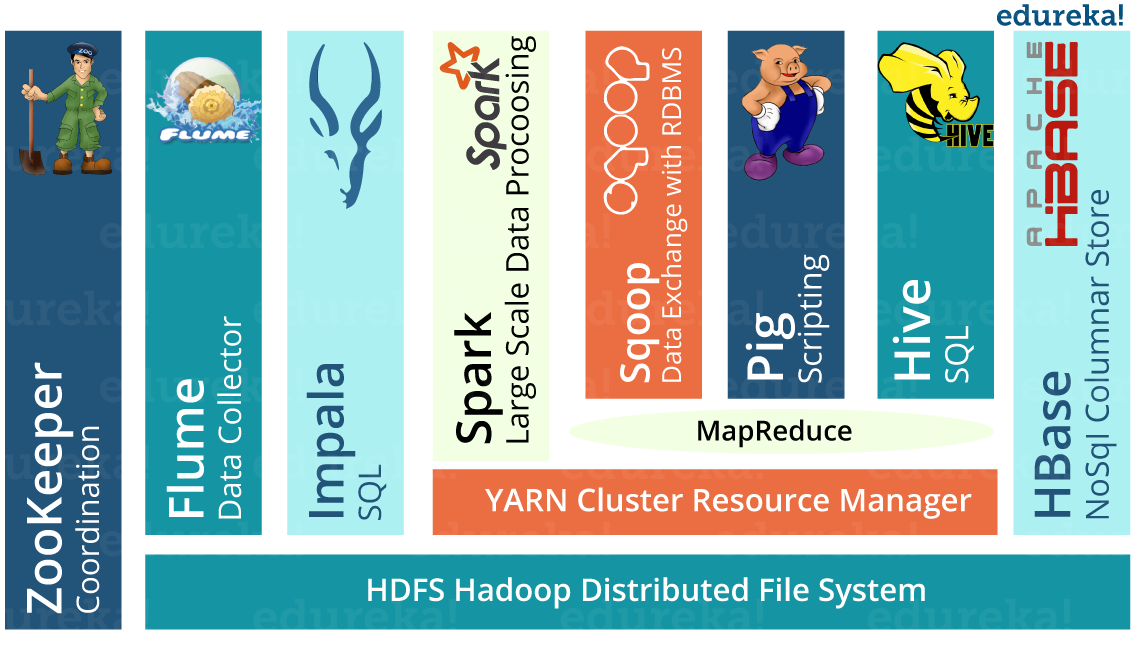
**Hadoop-CERN Case Study**

The *Large Hadron Collider* in Switzerland is one of the largest and most powerful machines in the world. It is equipped with around 150 million sensors, producing a petabyte of data every second, and the data is growing continuously.

CERN researches said that this data has been scaling up in terms of amount and complexity, and one of the important task is to serve these scalable requirements. So, they setup a Hadoop cluster. By using Hadoop, they limited their cost in hardware and complexity in maintenance.

***They integrated Oracle & Hadoop and they got advantages of integrating***.Oracle, optimized their Online Transactional System & Hadoop provided them scalable distributed data processing platform. They designed a hybrid systems, and first they moved data from Oracle to Hadoop. Then, they executed query over Hadoop data from Oracle using Oracle APIs. They also used Hadoop data formats like *Avro* & *Parquet* for high performance analytics without need of changing the end-user apps connecting to Oracle.

The main Hadoop components they are using at the CERN-IT Hadoop service:

You can learn each of these Hadoop ecosystem tool in [***Hadoop ecosystem blog***](https://www.edureka.co/blog/hadoop-ecosystem?utm_source=blog&utm_medium=content-link&utm_campaign=what-is-hadoop).

**Techniques for integrating Oracle and Hadoop:**

* *Export data from Oracle to HDFS*

Sqoop was good enough for most cases and they also adopted some of the other possible options like custom ingestion, Oracle DataPump, streaming etc.

* *Query Hadoop from Oracle*

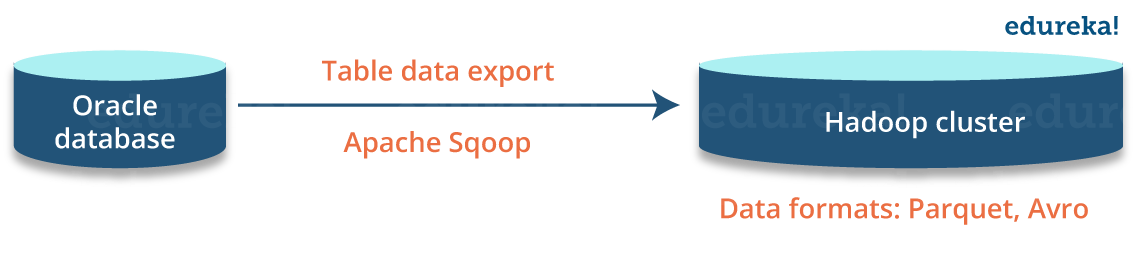
They accessed tables in Hadoop engines using DB links in Oracle. That also build hybrid views by transparently combine data in Oracle and Hadoop.

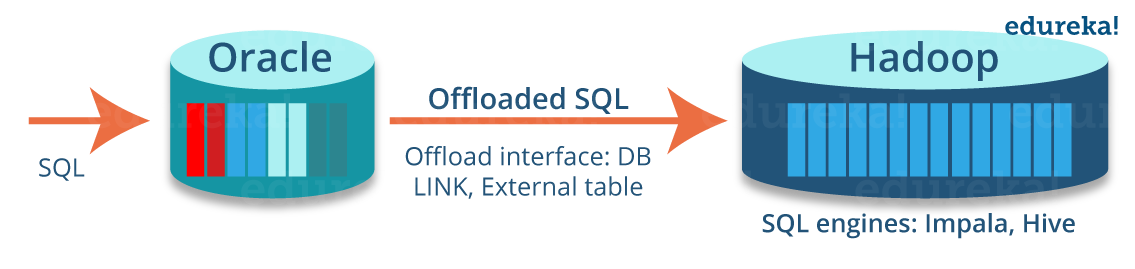
* *Use Hadoop frameworks to process data in Oracle DBs*

They used Hadoop engines (like Impala, Spark) to process data exported from Oracle and then read data in a RDBMS directly from Spark SQL with JDBC.

**Offloading from Oracle to Hadoop**

*Step1: Offload data to Hadoop*

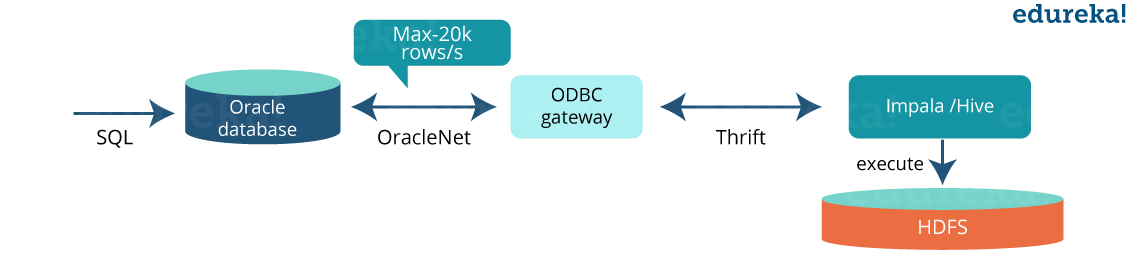
*Step2: Offload queries to Hadoop*

*Step 3: Access Hadoop from an Oracle query*

* Query Apache Hive/Impala tables from Oracle using a database link

|  |  |
| --- | --- |
| 1  2  3 | create database link my\_hadoop using 'impala-gateway';    select \* from big\_table@my\_hadoop where col1= :val1; |

* Query offloaded via ODBC gateway to Impala (or Hive)

**Example of creating hybrid view on oracle**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | create view hybrid\_view as              select \* from online\_table where date > '2016-10-01'              union all              select \* from archive\_table@hadoop where date <= '2016-10-01' |

Based on CERN case study, we can concludes that:

* Hadoop is scalable and excellent for data analytics
* Oracle is proven for concurrent transactional workloads
* Solutions are available to integrate Oracle and Hadoop
* There is a great value in using hybrid systems (Oracle + Hadoop):
  + Oracle APIs for legacy applications and OLTP workloads
  + Scalability on commodity Hardware for analytic workloads

I hope this blog was informative and added value to your knowledge. In our next blog of Hadoop Tutorial Series, i.e. [***Hadoop Tutorial***](https://www.edureka.co/blog/hadoop-tutorial/?utm_source=blog&utm_medium=content-link&utm_campaign=what-is-hadoop), we will discuss about Hadoop in more detail and understand task of HDFS & YARN components in detail.