Session: 10 Basics of Map Reduce Programming

1. **Hadoop Data Types (Java and Map Reduce)**
   A. Primitive Data Types
   These data types and their sizes are similar to SQL/Java primitive data types and sizes.

   Primary Data Types are four types as:

   i. **Numeric Data Type**

   ii. **Date/Time Data Type**

   iii. **String Data Type**

   iv. **Miscellaneous Data Type.**

i) Numeric Data Type

It supports both integral and floating data types. These are equivalent to Java's primitive types:

Integral Data Type:

Type	Size	Range
TINYINT	1 BYTE	-128 to 127
SMALLINT	2 BYTE	-32, 768 to 32, 767
INT	4 BYTE	–2,147,483,648to 2,147,483,647
BIGINT	8 BYTE	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

Floating Data Type:

Type	Size
FLOAT	4 BYTE
DOUBLE	8 BYTE
DECIMAL	17 BYTE

ii) Date/Time Data Type

It supports timestamp, date, and interval data types. Hive provides Timestamp and Date data types to UNIX timestamp format.

- **TIMESTAMP**- It uses nanosecond precision and is denoted by yyyy-mm-dd hh:mm: ss format.
- **DATE**- These are represented as YYYY--MM--DD
- **INTERVAL**

iii) String Data Type: It supports string, varchar, and char data types**.**

STRING - Either single or double quotes can be used to enclose characters.

VARCHAR - Maximum length is specified in braces and allowed up to 65355 bytes.

CHAR - 'm' or "m"

iv) Miscellaneous Data Type: it support both Boolean and binary data types.

- **BOOLEAN** - It stores True or false values
- **BINARY -** It is an array of bytes.

2. Complex Data Types

i) ARRAY: an ordered collection of similar types of fields that are indexable using zero-based integers. (These are just like arrays in java).

Syntax: ARRAY<data_type>

E.g. array (5, 6)

ii) MAP:

Unordered collection of key-value pairs. Keys may be values, primitives or any type.

For a specific map, the keys and values must be the same type.

Syntax: MAP<primitive_type, data_type>

E.g. MAP('m', 8, 'n', 9).

iii) STRUCT:

Collection of named fields. This fields may be of various types. (It's similar to STRUCT in C language)

Syntax: STRUCT<col_name : data_type [COMMENT col_comment],…..>

E.g. struct('m', 1 1.0),[n] named_struct('col1', 'm', 'col2', 1, 'col3', 1.0)

iv) UNION:

Union type can hold any data type that may be one of the specified data types.

Syntax: UNIONTYPE<data_type, data_type, …>

E.g. create_union(5, 'b', 60)

3. Columns

In Hive, columns support integral type, string, timestamp, date, decimal, and union data types.

i) Integral type: By default, the integral type is considered as '**int'** unless the range of the number exceeds.

- INT/INTEGER
- **SMALLINT-** Its data range is less than the range of INT.

- **BIGINT-** Its data range exceeds the range of INT.
- **TINYINT-** Its data range is less than SMALLINT

Table	Postfix	Example
BIGINT	L	100L
SMALLINT	S	100S
TINYINT	Y	100Y

ii) Strings: These are represented with either single (‘) or double (“) quotes and classified as:

- **VARCHAR-** These are built with a length specifier (1-65535). It represents the maximum number of characters passed in the character string.
- **Char-** These are same as VARCHAR but of fixed-length. Maximum fixed length is 255.

iii) Timestamp:

It maintains traditional UNIX timestamp with operational nanosecond precision.

The supported Timestamp format is”YYYY-MM-DD HH:MM: SS.(.f...)” in text files.

Timestamps are further categorized as:

Integer numeric type	UNIX timestamp in seconds
Floating-point numeric type	UNIX timestamp in seconds with decimal precision
String	Supports java.sql.Timestamp
	format "YYYY-MM-DD HH:MM: SS.fffffffff"

iv) Dates:

DATE value represents a particular year/month/day as YYYY-MM-DD format. For eg: DATE ‘2019-08-02’

It does not contain a time of day time component. It supports a range of 0000-01-01 to 9999--12-31.

v) Decimals:

The DECIMAL type is similar to Java’s Big Decimal format which denotes immutable arbitrary precision.

- Apache Hive 0.11 and 0.12 have the precision of the DECIMAL type fixed and defined to 38 digits.

- Apache Hive 0.13 users can specify scale and precision while designing tables with the DECIMAL datatype by a syntax DECIMAL (precision, scale).
- If precision is not specified its default value is 10, and if the scale is not specified its default value is 0.

not specified its default value is 0.

Eg: CREATE TABLE foo (

a DECIMAL, -- Defaults to decimal(10,0)

b DECIMAL(10, 6))

vi) Union Types:

Union Hive data types are the set of independent data types. We can create an instance of this type by create_union UDF.

#4. Literals

Literals support both floating-point types and decimal types.

- **Floating Point Types -** These are assumed to be DOUBLE data types in the Hive.
- **Decimal Types -** These are assumed to be higher value than float point value with a range more than DOUBLE data type. It ranges between -10-308 to 10308.

#5. Null Values

In Hive, missing values are denoted by the specific value called NULL.

2. Map Reduce program structure

MapReduce is a programming model and processing framework designed for processing large datasets in a distributed and parallel manner. It was popularized by Google and is widely used in big data processing.

The MapReduce model consists of two main phases: **Map phase and the Reduce phase**.

The program structure in MapReduce:

1. **Input Data:**

   MapReduce processes input data, which is typically a large dataset stored in a distributed file system like HDFS (Hadoop Distributed File System).

2. **Mapper Function (Map Phase):**

   The input data is divided into chunks or blocks, and each chunk is processed by a separate Mapper task.

   - The Mapper function takes input data and transforms it into key-value pairs.
   - The output of the Mapper function is an intermediate set of key-value pairs.

3. **Shuffle and Sort:**
    - The intermediate key-value pairs from all Mapper tasks are sorted and grouped by key.
    - This phase ensures that all values associated with a particular key are grouped together and ready for the Reduce phase.
4. **Reducer Function (Reduce Phase):**
    - The Reducer function takes the sorted and grouped key-value pairs as input.
    - It processes the data, typically performing aggregations, calculations, or filtering.
    - The output of the Reducer function is the final set of key-value pairs, which may be written to an output file or sent to another system.
5. **Output Data:**
    - The final output of the MapReduce job is stored in the distributed file system as one or more output files.

Characteristics of a MapReduce program structure:

- **Parallelism:** Map tasks can be executed in parallel, and Reduce tasks can also be executed in parallel, making MapReduce a scalable framework for processing large datasets.
- **Fault Tolerance:** MapReduce frameworks, like Hadoop, provide fault tolerance by automatically reassigning failed tasks to other nodes.
- **Data Locality:** Map tasks are typically executed on nodes where the data resides (data locality), reducing data transfer overhead.
- **Replication**: Input data is often replicated across multiple nodes for fault tolerance, and intermediate data can be replicated for performance.
- **Combiner:** In some cases, a Combiner function can be used to perform local aggregation on the output of Mapper tasks before data is shuffled and sent to Reducer tasks, reducing data transfer.
- **Job Control**: A job controller manages the execution of MapReduce jobs, tracks progress, and handles job-level configuration.

3. **Map-only program, Reduce-only program**
   (Map reduce program in notepad file)
4. **Use of combiner and partitioner**
   (Map reduce program in notepad file)

5. **Counters**

Counters in Hadoop are used to keep track of occurrences of events. In Hadoop, whenever any job gets executed, Hadoop Framework initiates Counter to keep track of job statistics like the number of bytes read, the number of rows read, the number of rows written etc. These built-in counters are:

a. **MapReduce Task Counter.**

b. **File System Counters.**

c. **FileInputFormat Counters.**

Counters are useful for:

- Monitoring the progress of a job, such as the number of records processed.
- Tracking errors or exceptions that occurred during job execution.
- Recording custom application-specific metrics or events, like the number of successful logins, specific exceptions, or other relevant information.

6. **Schedulers (Job Scheduling)**

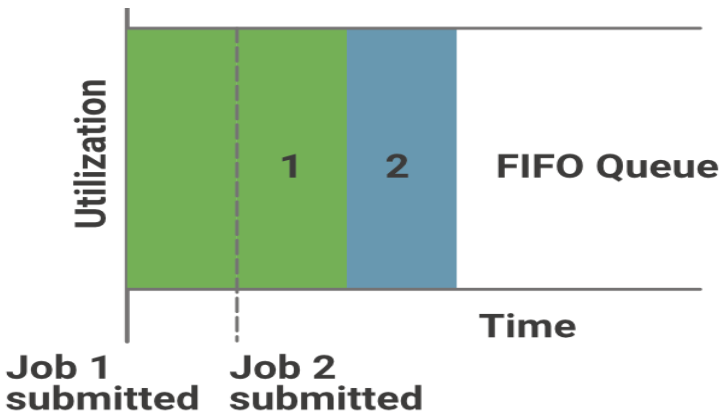
Schedulers and Applications Manager are the 2 major components of resource Manager. The Scheduler in YARN is totally dedicated to scheduling the jobs, it cannot track the status of the application.

A Job queue is nothing but the collection of various tasks that we have received from our various clients. The tasks are available in the queue and we need to schedule this task on the basis of our requirements.

There are mainly 3 types of Schedulers in Hadoop:

1. FIFO Scheduler

FIFO i.e. First In First Out, so the tasks or application that comes first will be served first. This is the default Scheduler we use in Hadoop. Sometimes the high-priority process has to wait for a long time since the priority of the task does not matter in this method.



Advantage:

- No need for configuration
- First Come First Serve
- Simple to execute

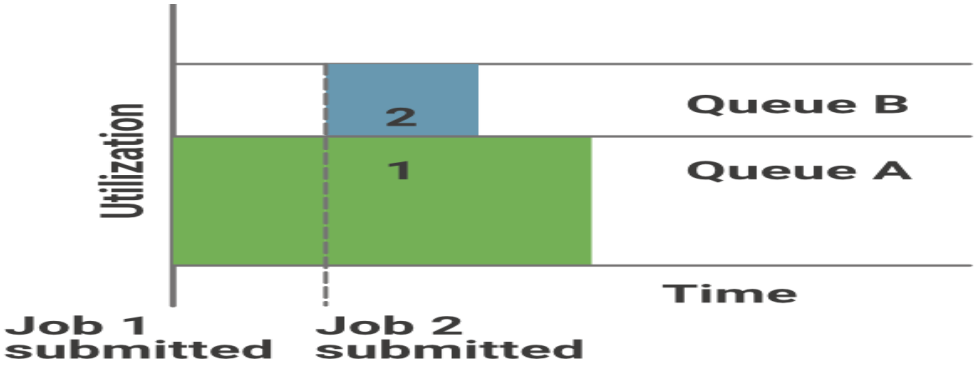
Disadvantage:

- Priority of task doesn't matter, so high priority jobs need to wait
- Not suitable for shared cluster

2. Capacity Scheduler.

In Capacity Scheduler we have multiple job queues for scheduling our tasks. The Capacity Scheduler allows multiple occupants to share a large size Hadoop cluster.

The capacity Scheduler mainly contains 3 types of the queue that are root, parent, and leaf which are used to represent cluster.



Advantage:

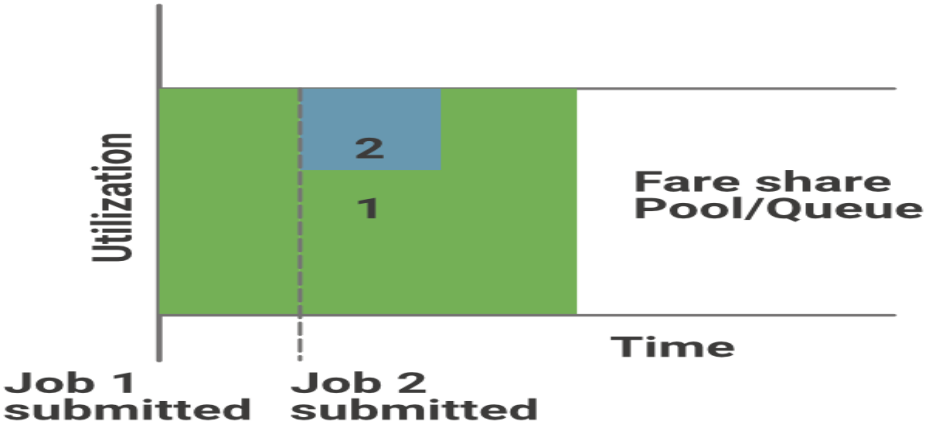
- Best for working with multiple clients or priority jobs in a Hadoop cluster
- Maximizes throughput in the Hadoop cluster

Disadvantage:

- More complex
- Not easy to configure for everyone

3. Fair Scheduler.

The Fair Scheduler is very much similar to that of the capacity scheduler. The priority of the job is kept in consideration. With the help of Fair Scheduler, the YARN applications can share the resources in the large Hadoop Cluster and these resources are maintained dynamically so no need for prior capacity.



Advantages:

- Resources assigned to each application depend upon its priority.
- it can limit the concurrent running task in a particular pool or queue.

Disadvantages: The configuration is required.

7. Custom Writables

Writable is an interface in Hadoop. Writable in Hadoop acts as a wrapper class to almost all the primitive data type of Java. That is how int of java has become IntWritable in Hadoop and String of Java has become Text in Hadoop. Writables are used for creating serialized data types in Hadoop.

- BooleanWritable.
- ByteWritable.
- IntWritable.
- VIntWritable.
- FloatWritable.
- LongWritable.
- VLongWritable.
- DoubleWritable.

Hadoop comes with a useful set of Writable implementations that serve most purposes; however, on occasion, you may need to write your own custom implementation. nWith a custom Writable, you have full control over the binary representation and the sort order.

8. Compression

Compression is a technique used to reduce the size of data files, making them smaller and more efficient for storage and transmission. The primary goal of compression is to

decrease the amount of space required to store or transmit data while retaining as much of the original information as possible.

In the context of big data, compression is often used to store and process large datasets efficiently. Formats like Parquet and ORC for columnar storage and Apache Avro for data serialization support compression.

Compression can significantly reduce storage costs in distributed storage systems like Hadoop HDFS.

Common Compression Algorithms:

- **Run-Length Encoding (RLE)**: This algorithm replaces sequences of repeated characters with a single character and a count. It's simple but is most effective for highly repetitive data.
- **Huffman Coding**: Huffman is a variable-length encoding algorithm that assigns shorter codes to more frequent characters or sequences of characters.
- **Lempel-Ziv-Welch (LZW)**: LZW is a dictionary-based compression algorithm used in formats like GIF and ZIP.
- **DEFLATE**: DEFLATE is a combination of LZ77 (Lempel-Ziv 1977) and Huffman coding used in formats like PNG and Gzip.
- **JPEG and MPEG**: These are lossy compression standards used for images and videos, respectively.
- **MP3**: A lossy audio compression format that reduces the size of audio files while preserving reasonable audio quality.

Session: 11 Map Reduce Streaming

1. Complex Map Reduce programming

Complex MapReduce programming refers to the development of intricate and sophisticated MapReduce jobs to process and analyze large datasets. MapReduce is a programming model and an associated implementation for processing large data sets.

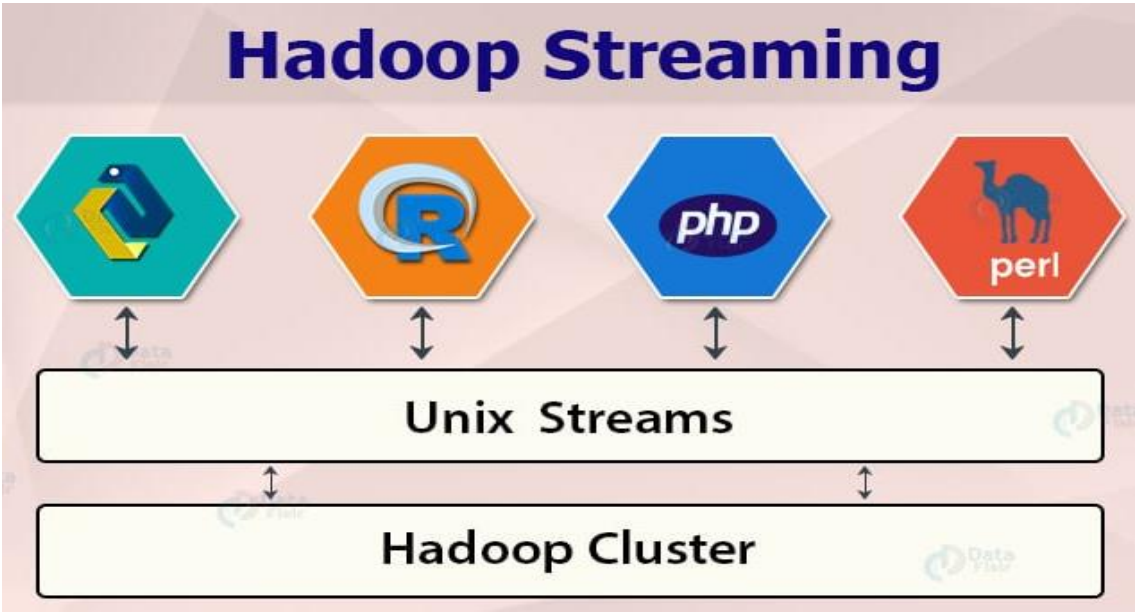
```
from mrjob.job import MRJob
from mrjob.step import MRStep
 class RatingsBreak(MRJob):
  def steps(self):
```

```
    return [
        MRstep(mapper=self.mapper_get_ratings,
               reducer=self.reducer_count_ratings)
    ]
    # MAPPER CODE
  def mapper_get_ratings(self, _, line):
    (User_id, Movie_id, Rating, Timestamp) = line.split('/t')
    yield rating,
    # REDUCER CODE
  def reducer_count_ratings(self, key, values):
    yield key, sum(values)
```

2. Map Reduce streaming

Hadoop streaming is a utility that comes with the Hadoop distribution. The utility allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer.

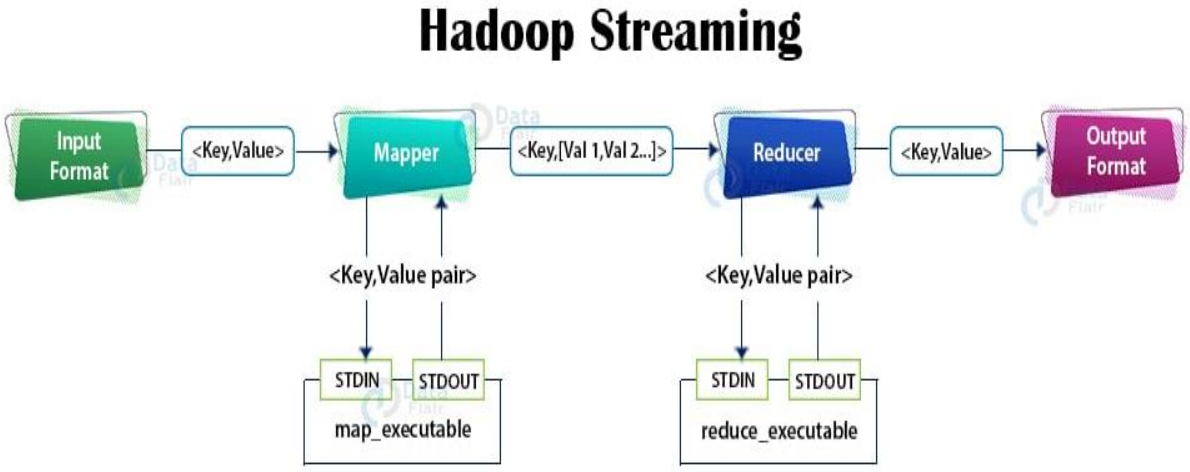
- **Python and Map Reduce**



By default, the Hadoop MapReduce framework is written in Java and provides support for writing map/reduce programs in Java only. But Hadoop provides API for writing MapReduce programs in languages other than Java.

Hadoop Streaming is the utility that allows us to create and run MapReduce jobs with any script or executable as the mapper or the reducer. It uses Unix streams as the interface between the Hadoop and our MapReduce program so that we can use any language which can read standard input and write to standard output to write for writing our MapReduce program.

How Streaming Works



Map reduce on image dataset

A.MIPr provides the ability to process images in Hadoop.

MIPr includes:

- **Writable Wrappers for images**
- **InputFormat and OutputFormat for images**
- **Several Jobs for image processing**
- **OpenCV and OpenIMAJ support**

1. **Copy image files to HDFS:**

   $ hadoop fs -copyFromLocal local_image_folder hdfs_image_folder

2. **Run test MIPr Job which converts color images to grayscale:**

   $ hadoop jar mipr-core-0.1-jar-with-dependencies.jar experiments.Img2Gray hdfs_image_folder hdfs_output_folder

3. **Copy processed images back from HDFS to the local filesystem:**

   $ hadoop fs -copyToLocal hdfs_output_folder local_output_folder

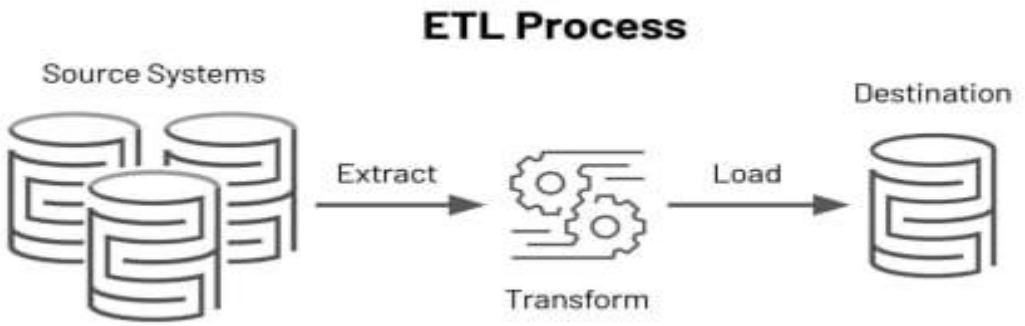
4. **Check that images were converted correctly.**

2. HIPI: Hadoop Image Processing Interface (HIPI) is an image processing library designed to be used with Apache Hadoop MapReduce.

Hadoop ETL Session: 12

1. **Hadoop ETL Development & Process**

   ETL stands for Extract, Transform and Load. The ETL process typically extracts data from the source / transactional systems, transforms it to fit the model of data warehouse and finally loads it to the data warehouse. Hadoop isn't an ETL tool, but it can help you manage your ETL projects.



   Source data may be in structured transaction formats, such as relational databases, XML, JSON, and flat files, or unstructured data, such as internet clickstream records, web server records, mobile application logs, social media posts, customer emails, and sensor data from internet of things (IoT) devices.

   Setting up ETL in Hadoop

   1. **Setting up a Hadoop Cluster**
   2. **Connecting Data Sources**
   3. **Defining the Metadata**
   4. **Creating Jobs for ETL in Hadoop**
   5. **Creating the Workflow for ETL in Hadoop**

   Types of ETL Tools and their function

   ETL tools have been around for over 30 years. As technology has evolved in that time, different types of solutions have entered the market

Enterprise Software ETL

Several software companies sell and support commercial ETL software products. All of these products provide graphical interfaces for designing and executing ETL pipelines. They all connect to most relational databases. Some have support for non-relational data sources such as JSON and XML. A few have support for event streaming sources such as Apache Kafka.

- **Informatica PowerCenter**. This product is arguably the most mature ETL product in the market. It is used by many large companies and is well-regarded by industry analysts. It is part of a large portfolio of products, bundled as Informatica Platform. These products are perceived as IT-centric, and they are also very expensive. Informatica is less mature than some other products for semi-structured and unstructured sources.
- **IBM InfoSphere DataStage**. DataStage is a very mature ETL product that was acquired from the company Ascential. It is especially popular with IBM shops. Unlike many other ETL tools, it provides strong capabilities for working with mainframe computers. DataStage is perceived as expensive, complex to license, and overlaps with other products in the same family.
- **Oracle Data Integrator (ODI)**. Oracle's ETL product has been available for many years. It uses a fundamentally different architecture from other ETL products. Instead of performing transformations in the ETL tool itself using a dedicated process and hardware resources, ODI moves data into the destination, then performs transformations using the features of the database or Hadoop cluster.
- **Microsoft SQL Server Integration Services (SSIS)**. SSIS is very popular among users of SQL Server. It is lower in cost than other enterprise ETL tools, and easier to use.
- **Ab Initio**. Ab Initio is a highly secretive company based in Lexington, MA. The product began as ETL and evolved to provide other offerings, such as metadata catalogs. Ab Initio is proprietary and very expensive. It claims to be higher in performance and easier to use than traditional ETL tools.
- **SAP Data Services**. SAP's ETL tool is designed primarily for moving data between SAP applications. It is not widely used outside of these environments.
- **SAS Data Manager**. SAS has developed an ETL product with strong support for Hadoop, streaming data and machine learning, but limited support for bulk loading

destination systems. Data Manager is perceived as expensive, while SAS is perceived as a vendor with high customer satisfaction.

Open Source ETL and Data Extraction

Over the past 10 years, software developers have created several open source ETL products. These products are free to use. Their source code is also freely available, which allows you to extend or enhance their capabilities. These tools vary significantly in quality, integrations, ease of use, adoption and availability of support. Like the enterprise ETL tools, many of these open source ETL tools provide a graphical interface for designing and executing pipelines.

- **Talend Open Studio**. Talend's ETL tool is the most popular open source ETL product. Open Studio generates Java code for ETL pipelines, rather than running pipeline configurations through an ETL engine. This approach gives it some performance advantages.
- **Pentaho Data Integration (PDI)**. Formerly known as Kettle, PDI is an open source ETL tool well known for its graphical interface called Spoon. PDI generates XML files to represent pipelines, and executes pipelines through its ETL engine. Pentaho was acquired by Hitachi Data Systems in 2015.

- **Hadoop**. Hadoop is a general purpose distributed computing platform. It is used to store, manipulate and analyze data of any structure. Hadoop is a complex ecosystem of open source projects, comprising over 20 different technologies. Some of these projects are used to perform ETL tasks, such as Pig, MapReduce and Spark. When used for ETL, data is typically first loaded into Hadoop's Distributed File System (HDFS) "as is" from source systems. Sqoop is a tool used to move data from relational databases into HDFS. Once the data is stored in Hadoop, any of the projects can be used to transform and store the cleansed data in HDFS. Hive is a popular project for using SQL to define these transformations (a Hive query is compiled into MapReduce). Pig and Spark can be used as well.

Custom ETL

Many companies use general purpose programming languages to write their own ETL tools. This approach has the greatest flexibility, but also requires the most effort. This approach also requires users to perform their own maintenance, build their own documentation, test and perform ongoing development.

SQL. If your data source and destination are the same, then using SQL is a very powerful option. SQL is very efficient at reading and writing data, as well as performing basic transformations. SQL is not very effective at complex transformations that involve decision trees, or calling external sources, for example. SQL is also built into the database, so no additional license fees or technologies are required. SQL is widely understood by DBAs and developers.

- **Python**. Python is a general purpose programming language. It has become a popular tool for performing ETL tasks due to its ease of use and extensive libraries for accessing databases and storage technologies. Python can be used instead of ETL tools for ETL tasks. Many data engineers use Python instead of an ETL tool because it is more flexible and more powerful for these tasks.
- **Java**. Java is another general purpose programming language that can be used to build ETL processing. Java is one of the most popular programming languages and has extensive support for different data sources and data transformation. Java and Python each present different trade-offs for ETL. When choosing between them existing skills may be the most important factor.
- **Spark & Hadoop**. Spark and Hadoop work with large datasets on clusters of computers. They make it easier to apply the power of many computers working together to perform a job on the data. This capability is especially important when the data is too large to be stored on a single computer. Today Spark and Hadoop are not as easy to use as Python, and there are far more people who know and use Python.

ETL Cloud Services

Amazon AWS, Google Cloud Platform and Microsoft Azure offer their own ETL capabilities as cloud services. If your data is already in one of these cloud platforms, there are a number of advantages to using their ETL services.

AWS EMR. Elastic MapReduce (EMR) is the Hadoop offering provided by AWS. Companies running on AWS who like to use Hadoop for ETL or ELT may use EMR. As with any Hadoop distribution, there are several tools available to perform ETL, including Hive and Spark. This solution has the advantages of being very powerful and scalable, and capable of working with structured and unstructured data. It is also elastic and users only pay for what they use. It has the downside of being very difficult to use.

- **AWS Glue**. This is a new fully managed ETL service AWS announced in late 2016. Glue is targeted at developers. It is tightly integrated into other AWS services, including data sources such as S3, RDS, and Redshift, as well as other services, such as Lambda. Glue can connect to on-premises data sources to help customers move their data to the cloud. ETL pipelines are written in Python and executed using Apache Spark and PySpark.
- **AWS Data Pipeline**. AWS Data Pipeline is cloud-based ETL. It can be used to schedule regular processing activities such as distributed data copy, SQL transforms, MapReduce applications, or even custom scripts, and is capable of running them against multiple destinations, like Amazon S3, RDS, or DynamoDB.
- **Azure Data Factory**. Data Factory is a fully managed service that connects to a wide range of cloud and on-prem data sources. It is capable of copying, transforming and enriching data, then writing the data to Azure data services as a destination. Data Factory also supports Hadoop, Spark and machine learning as transformation steps.
- **Google Cloud Dataflow**. Dataflow is a fully managed service targeted at developers for designing batch and continuous ETL jobs. Dataflow provides APIs for Java and Python for developers to connect to Google Cloud sources, apply transformations and write data into other Google Cloud destinations. Dataflow APIs are based on Apache Beam. Unlike other cloud services, Dataflow does not connect to on-prem sources.
- **Segment**. Segment is a SaaS technology that moves data between systems based on events. Companies use Segment to move data into their data warehouses, and also to move data from one application to another, such as from a marketing automation system into Salesforce. This is not strictly an ETL tool. Segment connects to many sources, including dozens of popular applications as sources and destinations, including data warehouses.
- **Stitch**. Stitch Data is a SaaS provider of ETL that is new to the market and focused on developers. This is built on an open source core called Singer.

2. Need of ETL tools

ETL tools automate the data integration process, reducing the time and effort required to build and maintain data pipelines. They also help ensure data accuracy and consistency, improve data quality, and enable faster decision-making.

3. Advantages of ETL Tool

ETL tools support solid data management by letting you apply and maintain complex universal formatting standards and semantic consistency to all data sets as you move and integrate them. There are following advantages of ETL tools:

1. Reduce Delivery Time

ETL tools create workflows using a visual interface with ready-made components. The building of data processes that are required becomes faster.

2. Reduce Unnecessary Expenses

Data migration is an iterative method. This process can easily be modified and repeated, thus saving a considerable amount of time and effort. You can examine changes quickly on the whole data set.

3. Automate Complex Processes

Automated data migration saves time, and energy, and results in better delivery. Automation reduces the hassles of manual work and human error.

4. Validate Data Before Migration

Daton development team provides an effective data quality check to cleanse the data before moving from one system to the other..

Discard the irrelevant part of the data in the data migration process. This not only reduces the storage costs, but also improves the overall data quality, and accelerates data processing speed.

5. Build Data Quality Feedback Loops

You can automate the error handling by exporting any values that don't adhere to the pre-defined data rules and set repeatable processes to fix errors. This technique also helps to feed cleaner data to your systems.

6. Transform Data

Exporting data from one place to another generally involves some transformations in between. The data transformations are required to feed the data into the destination system properly.

The basic transformations which ETL tools perform are:

- Splitting or merging multiple fields
- Validating fields
- Converting currencies or time zones

- Altering product codes
- Updating naming conventions

7. Making the Process Transparent

Manual data migration in Excel or data wrangling tools did not have any way to keep track of edits done in the data other than lengthy documentation and constantly keeping it updated.

Automated data migration tools automatically record all the steps in the workflow. As a result, the whole data migration process is transparent and can be traced back.

8. Repeatability for data migrations

Manual data migration causes several problems. One such can be while modifying the records. If there is a small change in your destination system, you might have to start the process all over again. With a repeatable and customized system, you can easily edit data sets and re-run an automated data migration process.

9. Data Cleansing

While performing a complex transformation during data migration, such as de-duplicating your customer list, ETL tools can help you the most by providing more useful cleansing functions than those available in SQL.

10. Big Data Handling

ETL Tools are now developed enough to handle Big Data efficiently. The structure imposed by an ETL platform makes it easier for a developer to build an enhanced system hence, the overall performance during the data migration process is improved.