R P SARATHY INSTITUTE OF TECHNOLOGY

BIG DATA ANALYTICS – CASE STUDY

ASSIGNMENT - 05

- *REG NO : 611721104063*
- *NAME : R.Poovarasan*
- *DEPT : B.E - CSE – B*
- *YEAR / SEM : III / V*
- *DATE : 11.09.2023*

Contents

- Hadoop I/O
- Data Integrity
- Compression
- Serialization
- File-Based Data Structures

Hadoop I/O

- Hadoop Comes with a set of primitives for data I/O.

- Some of these are techniques that are more general than Hadoop, such as data integrity and compression, but deserve special consideration when dealing with multiterabyte datasets.

- Others are Hadoop tools or APIs that form the building blocks for developing distributed system, such as serialization frameworks and on-disk data structures.

Data Integrity

- When the volumes of data flowing through the system are as large as the ones Hadoop is capable of handling, the chance of data corruption occurring is high

- Checksum
  - Usual way of detecting corrupted data
  - Technique for only error detection (cannot fix the corrupted data)
  - CRC-32 (cyclic redundancy check)
    - Compute a 32-bit integer checksum for input of any size

- Two major benefits of file compression
  - Reduce the space needed to store files
  - Speed up data transfer across the network

- When dealing with large volumes of data, both of these savings can be significant, so it pays to carefully consider how to use compression in Hadoop

Serialization

- Process of turning structured objects into a byte stream for transmission over a network or for writing to persistent storage
- Deserialization is the reverse process of serialization
- Requirements
  - Compact
    - To make efficient use of storage space
  - Fast
    - The overhead in reading and writing of data is minimal
  - Extensible
    - We can transparently read data written in an older format
  - Interoperable
    - We can read or write persistent data using different language

File-Based Data Structure

- For some applications, you need a specialized data structure to hold your data. For doing MapReduce-based processing, putting each blob of binary data into its own file doesn't scale, so Hadoop developed a number of higher-level containers for these situations.

- Higher-level containers
  - SequenceFile
  - MapFile

PIG- HADOOP RELATED TOOLS

CONTENTS

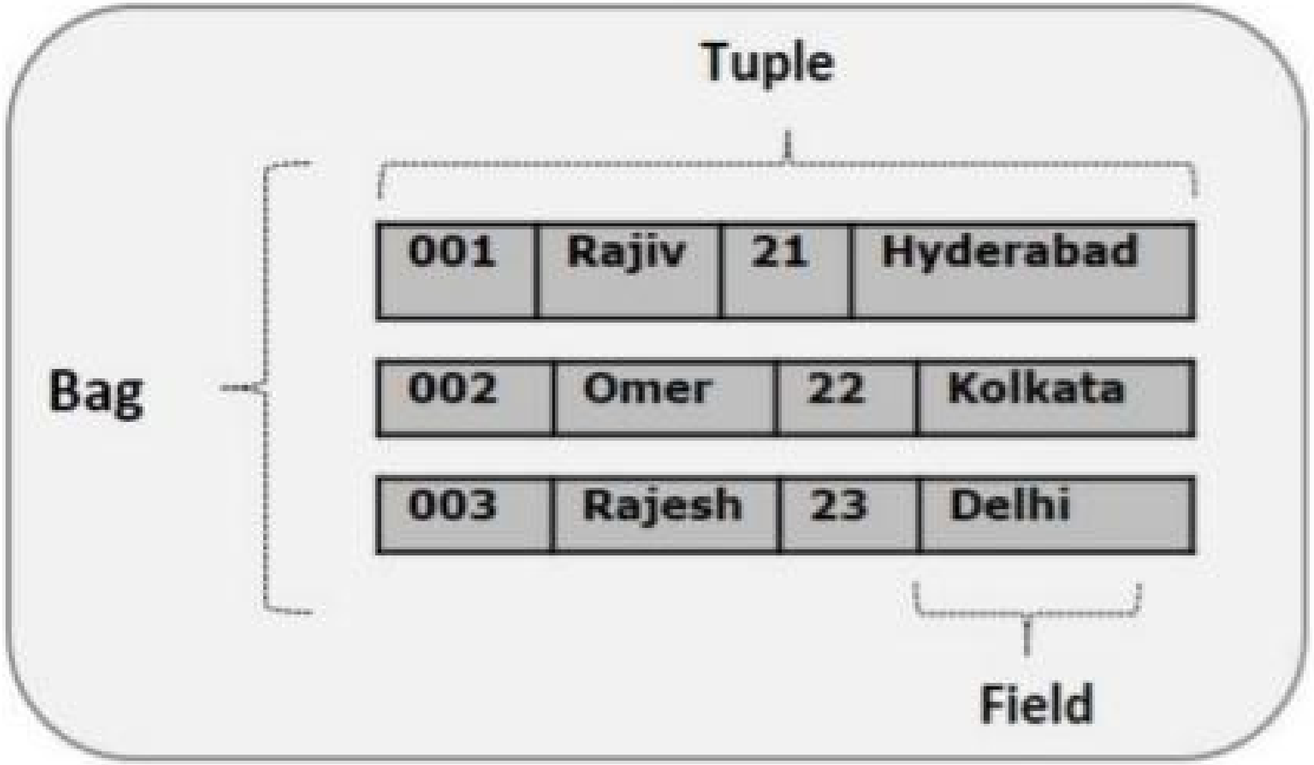
What is Pig?

- Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows.
- Pig is generally used with Hadoop; we can perform all the data manipulation operations in Hadoop using Apache Pig.
- To write data analysis programs, Pig provides a high-level language known as Pig Latin.
- This language provides various operators using which programmers can develop their own functions for reading, writing, and processing data.

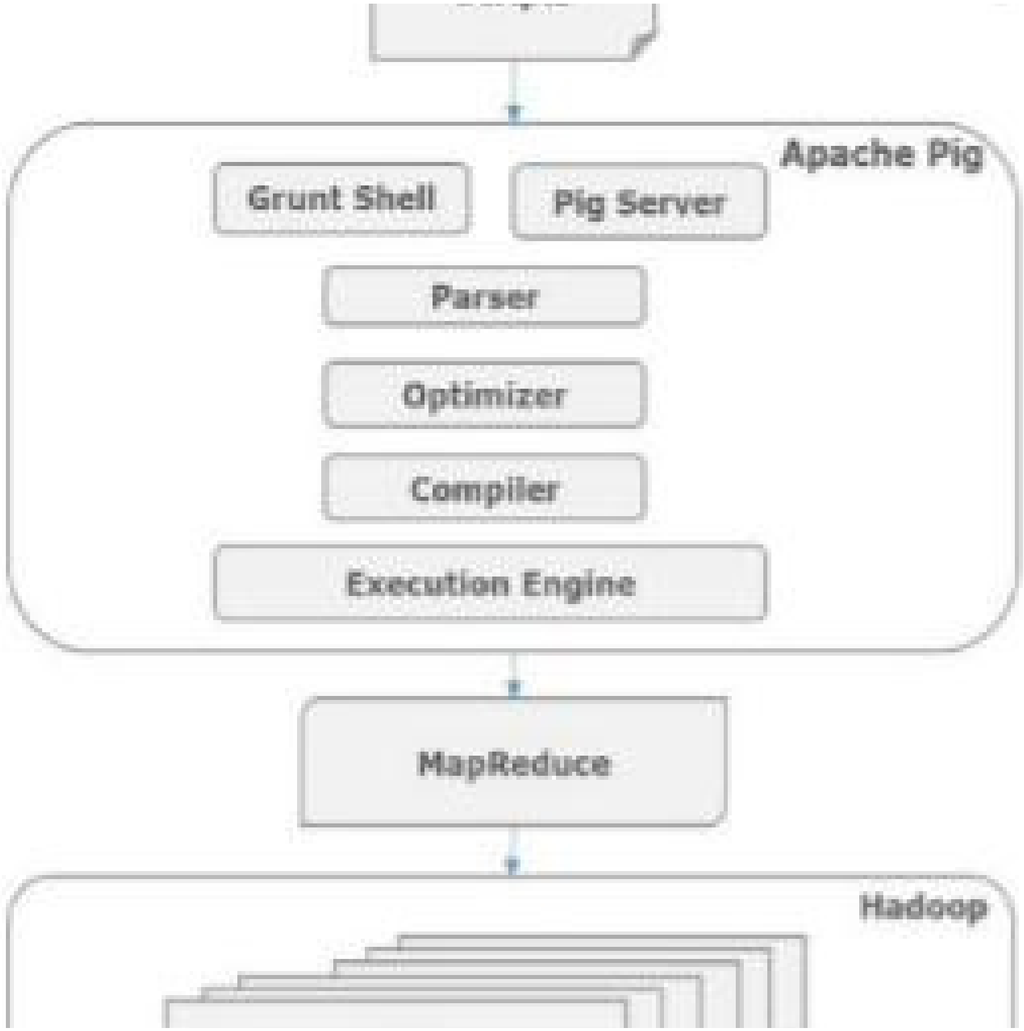
Features of Pig

- Rich set of operators: It provides many operators to perform operations like join, sort, filer, etc.
- Ease of programming: Pig Latin is similar to SQL and it is easy to write a Pig script if you are good at SQL.
- Optimization opportunities: The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on semantics of the language.
- Extensibility: Using the existing operators, users can develop their own functions to read, process, and write data.
- UDF's: Pig provides the facility to create User-defined Functions in other programming languages such as Java and invoke or embed them in Pig Scripts.
- Handles all kinds of data: Apache Pig analyzes all kinds of data, both structured as well as unstructured. It stores the results in HDFS.

Pig – Data Model



Pig Architecture



Applications of Pig

- To process huge data sources such as web logs.
- To perform data processing for search platforms.
- To process time sensitive data loads.