HADOOP DISTRIBUTED FILE SYSTEM ~&~ HBASE

Guided By:
Dr. Ying jin

Presenters:
Jaidipkumar Patel
Jay Bibodi

AGENDA

- Introduction to HDFS and HBASE
- HDFS Architecture
- Operations and Goals of HDFS
- Pros and cons of HDFS
- HBASE Architecture
- Storage Mechanism of HBASE
- Application of HBASE
- Difference between column-oriented vs Row-oriented and HBASE vs RDBMS
- Difference between HDFS vs HBASE
- Summary
- References
- Questions

INTRODUCTION TO HDFS AND HBASE

HDFS

- HDFS is a file system specially designed for storing huge dataset on cluster of commodity hardware with streaming access pattern.
- HDFS works with principle of "write once, read any number of times ."
- Key features of HDFS are fault tolerance and high throughput.
- File replication method is used to avoid fault in the HDFS.

HBASE

- HBase is NoSQL column oriented distributed database built on top of Hadoop distributed file system.
- It can be scaled horizontally to many commodity servers.
- All the data will be taken as key-value pair of byte-array.
- HBase allows fast random read and writes in optimized manner.

HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

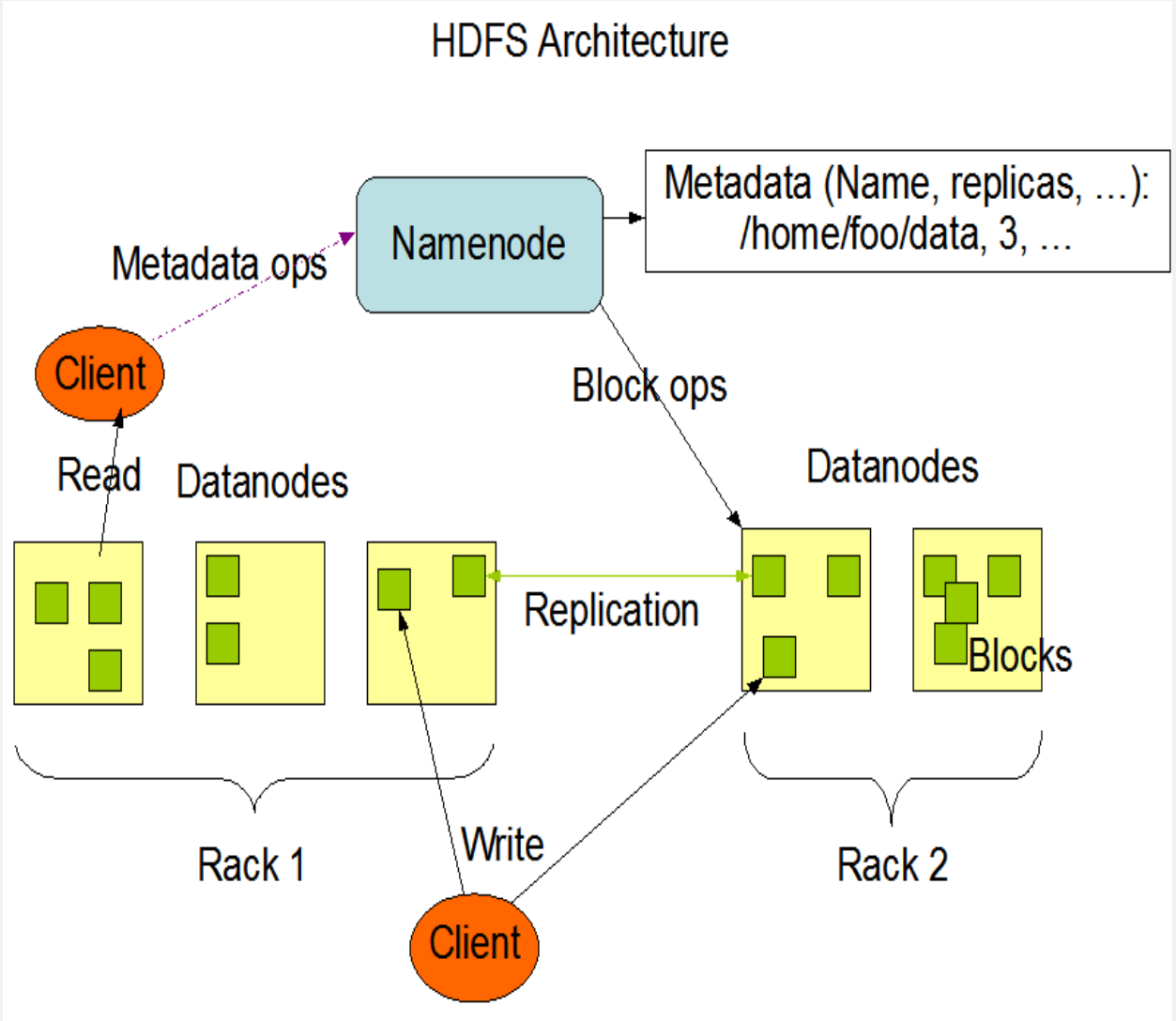
HDFS Architecture

- 3 Main components of this architecture

A. **Name Node:** It will keep the details such as a number of blocks required to store the file, size and type of the file and also store the replicated nodes detail which is used in case of node failure or data loss.

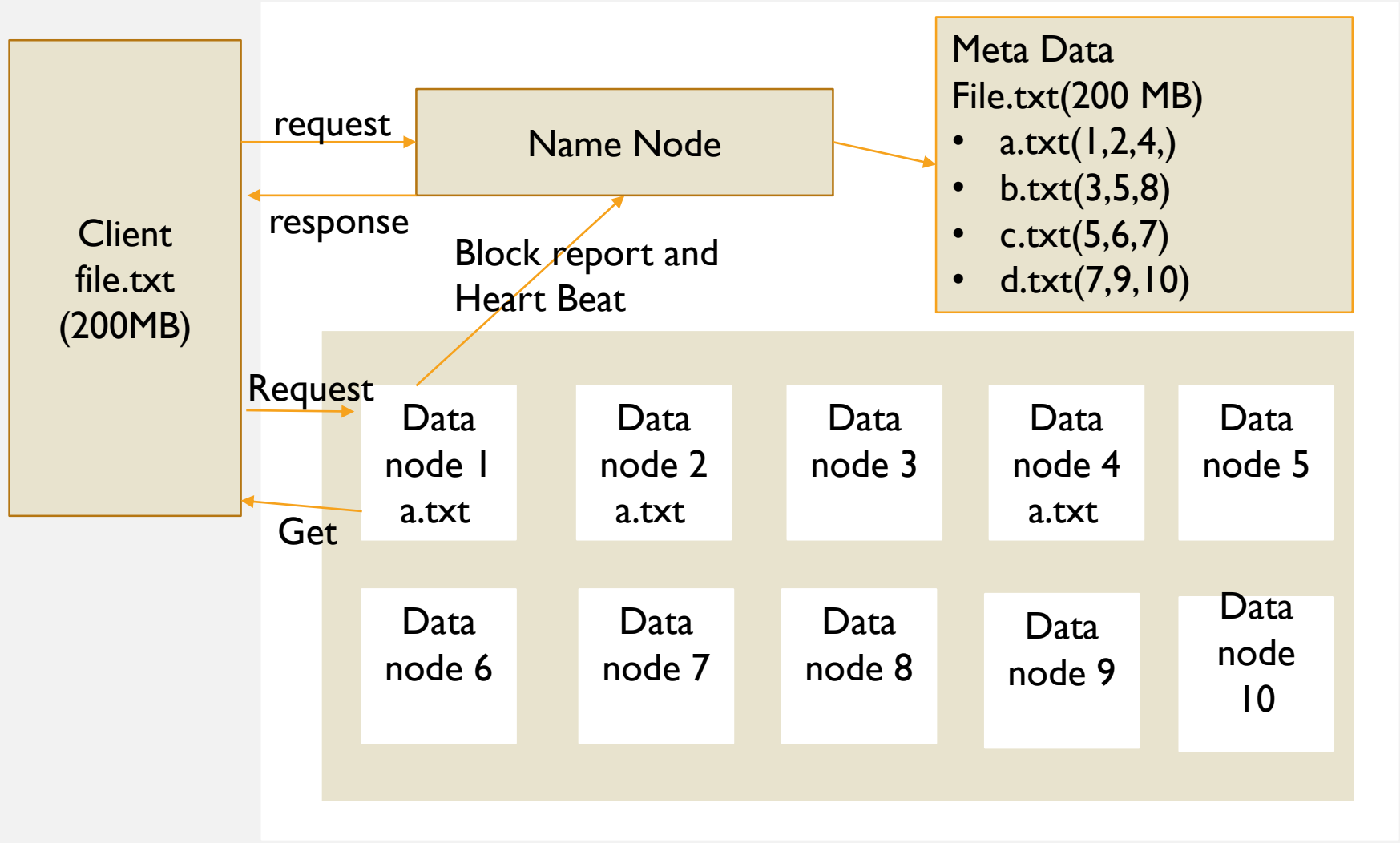
B. **Data Node:** Data node establish a connection with Name node and takes instructions from it to perform certain operations such as the read and write operation on file system to serve the client request.

C. **Data Replication:** To enable the Fault Tolerance feature in the HDFS, HDFS replicates the file blocks



HDFS(CONTINUE..)

- Example HDFS
- Client and Name node initiate Request/Response call.
- Name node maintains metadata of client's data. In response to the client request it provides data node Information to the client.
- Client can use information given by the Name node to store and get data from the Data Node.



HADOOP OPERATIONS AND GOALS

OPERATIONS:

- Format and Start HDFS
- List Files in HDFS
- Insert data into HDFS
- Retrieval from HDFS
- Shutting down HDFS

GOALS:

- Hardware Failure
- Streaming Data Access
- Large Datasets
- Hardware at Data

HDFS PROS AND CONS

PROS:

- Data Source Range
- Cost Effective
- Speed
- Multiple Copies

CONS:

- Security Measures
- Data Concerns
- Difficult Integration
- Lack of interactive access

HBASE ARCHITECTURE

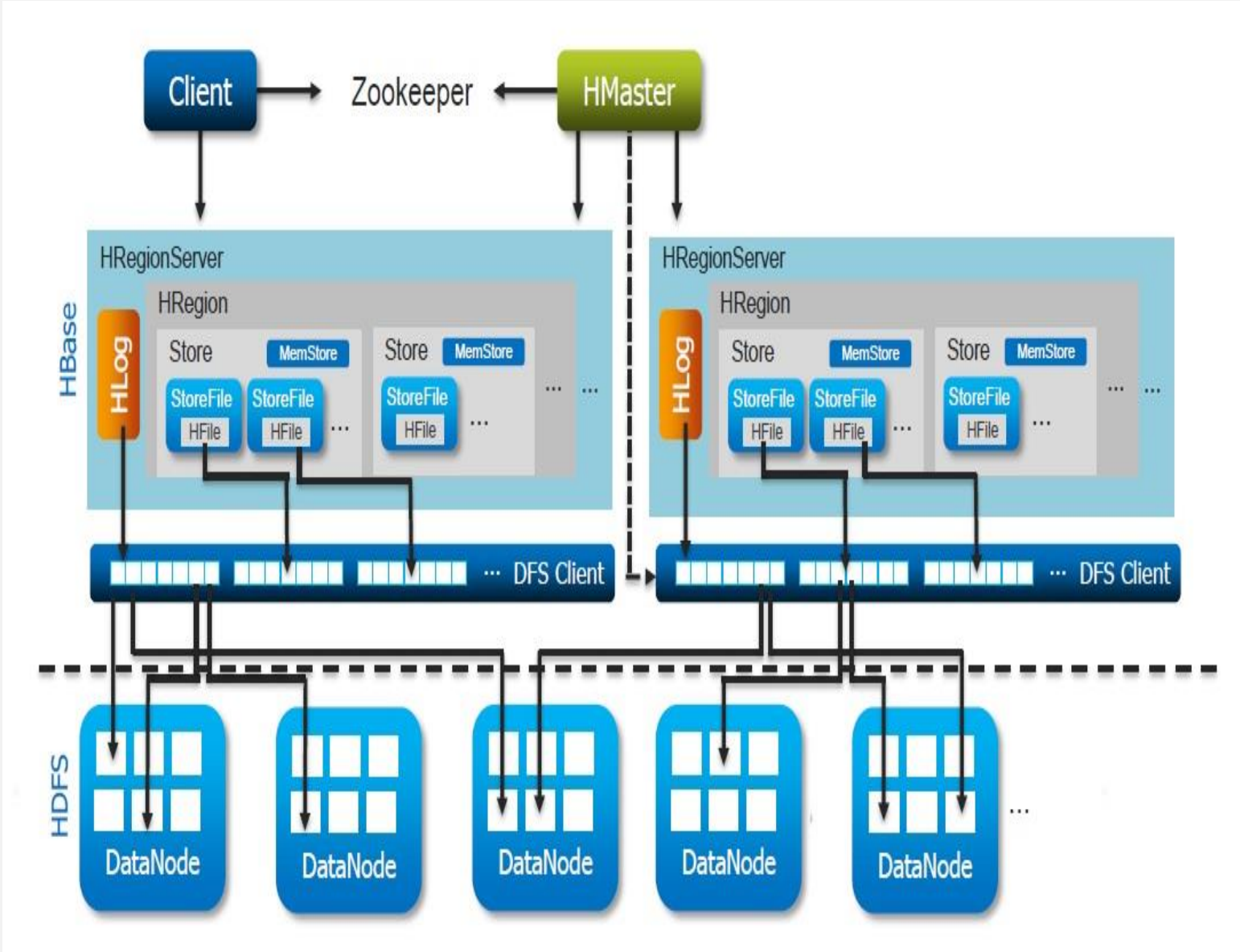
HBASE Architecture

- 3 Main components of this Architecture

A. **HMaster:** Assigns the region to the Hregion server with help of zookeeper. responsible for managing and coordinating activities of schema design.

B. **Zookeeper:** provide services such as naming, synchronization and maintain configuration information.

C. **Region Server:** It has regions which were used for read-write request handling.



STORAGE MECHANISM OF HBASE

- HBase is designed to work with huge data.
- it follows the column-oriented database approach in which tables sorted by row.
- Column family can have any number of column with their respective key value pairs.

RowID	**Column Family**			**Column Family**		
1	Col1	Col2	Col3	Col1	Col2	col3
2						
3						

APPLICATION OF HBASE

- Adobe ---- It use it to write data to HBase and run MapReduce jobs to process then store it back to Hbase or external systems.
- Facebook ---- It use Hbase for Messages Infrastructures.
- Yahoo ---- It uses HBase to store document fingerprint for detecting near-duplications.
- Twitter ---- HBase provides distributed, read/write a backup of all MySQL tables in Twitter's production backend, allowing engineers to run MapReduce jobs over the data while maintaining the ability to apply periodic row updates.
- Project Astro
- Caree.rs
- Filmweb
- Infolinks
- Mahalo
- NGDATA
- OpenLogic

DIFFERENCE BETWEEN COLUMN-ORIENTED AND ROW-ORIENTED DATABASES

Column-oriented	Row-Oriented
More used with Analytical Processing	More used with Transactional Processing
Only relevant columns are required	All the columns are required
Strong Compression due to single datatype and range of values.	Poor compression due to multiple types in the row.
Efficient used when working with Huge dataset.	Efficient when working with a small dataset.
Provides Fast Access to data	Slower than Column Oriented Database.

DIFFERENCE BETWEEN HBASE AND RDBMS

HBASE	RDBMS
It defines Column Families. It is schema-less.	It has a schema which describes the structure of tables.
It is Horizontally Scalable.	It is Hard to Scale.
Not Transactional.	It is Transactional.
It is good for both semi-structured as well as structured data.	It is well suited for Structured Data.

DIFFERENCE BETWEEN HDFS AND HBASE

HDFS	HBASE
It is a Distributed file system suitable for storing huge data.	It is a Database run on top of HDFS.
Not Supports Fast record lookups.	It supports fast lookups for large tables.
It provides sequential access of data.	It uses key, value pair and provides random access.
High latency batch processing.	It provides the Low latency (Random access).

SUMMARY

- A basic idea of the HDFS, HDFS architecture and details about its main components such as Name node, Data node, Data Replication.
- Also basic HDFS operations, Goals, Pros and Cons of HDFS.
- Explained what is HBase, it's architecture, and it's storage mechanism.
- Application of HBase.
- The difference between HDFS and HBase.
- Please view Hadoop operation performed in windows environment.

REFERENCES

- [1] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
- [2] https://www.tutorialspoint.com/hadoop/hadoop_hdfs_operations.htm
- [3] https://www.tutorialspoint.com/hadoop/hadoop_hdfs_overview.htm
- [4] http://www.knowledgehut.com/blog/bigdata-hadoop/top-pros-and-cons-of-hadoop
- [5] https://www.edureka.co/blog/overview-of-hbase-storage-architecture/
- [6] https://www.edureka.co/blog/hbase-architecture/
- [7] http://hbase.apache.org/poweredbyhbase.html
- [8] http://www.webopedia.com/TERM/C/commodity_hardware.html
- [9] https://en.wikipedia.org/wiki/Apache_Hadoop
- [10] https://en.wikipedia.org/wiki/Apache_HBase
- [11] https://www.tutorialspoint.com/hbase/hbase_overview.htm

QUESTIONS…??

Thank You…!!!