MODULE-1 Part-2

COMPILED BY,

DR. PRAKASH KALINGRAO AITHAL

Hadoop's Data Locality

Data locality refers to the pattern of processing data where it resides by bringing the computation to the data rather than the typical pattern of requesting data from its location—for example, a database management system—and sending the data to a remote processing system or host.

Hadoop's Shared Nothing Approach

Hadoop enables large datasets to be processed locally on the nodes of a cluster using a *shared nothing* approach, where each node can independently process a much smaller subset of the entire dataset without needing to communicate with other nodes. This characteristic is enabled through its implementation of a distributed filesystem.

Hadoop's Schemaless Approach

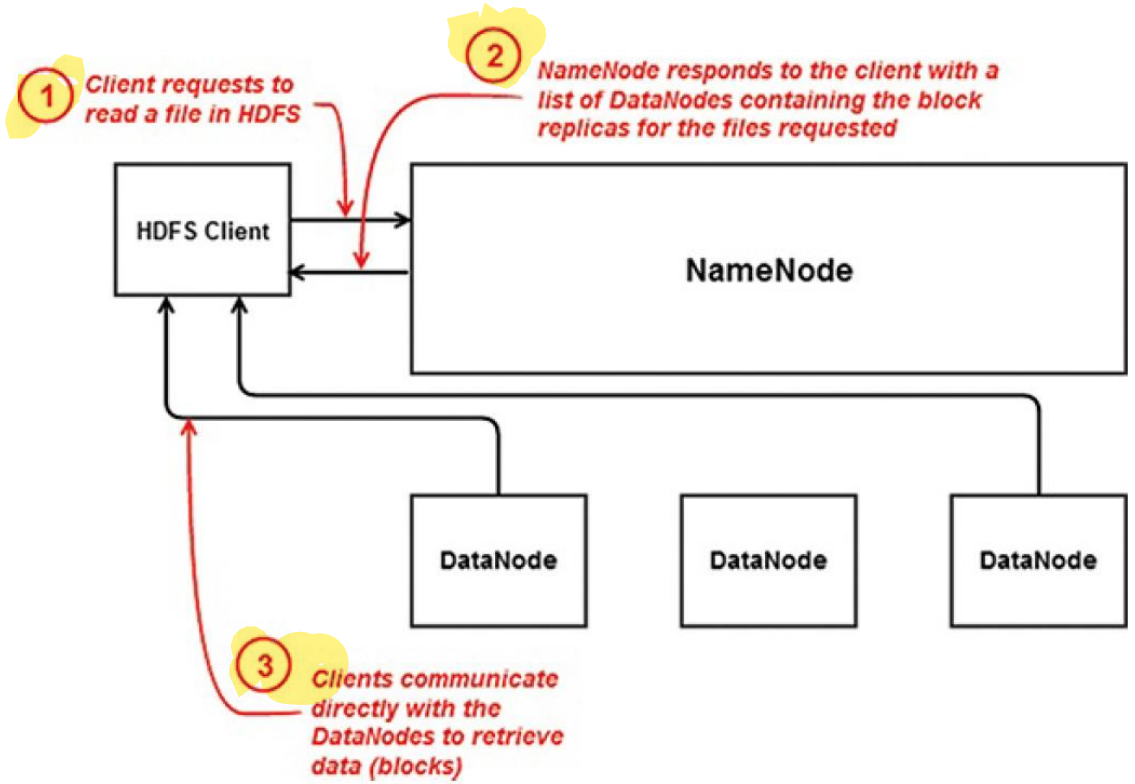
Hadoop is schemaless with respect to its write operations; it is what's known as a *schema-on-read* system. This means it can store and process a wide range of data, from unstructured text documents, to semi-structured JSON (JavaScript Object Notation) or XML documents, to well-structured extracts from relational database systems. Schema-on-read systems are a fundamental departure from the relational databases we are accustomed to, which are, in contrast, broadly categorized as schema-on-write systems, where data is typically strongly typed and a schema is predefined and enforced upon INSERT, UPDATE, or UPSERT operations. NoSQL platforms, such as HBase or Cassandra, are also classified as schema on-read systems.

Hadoop 2.0 Components

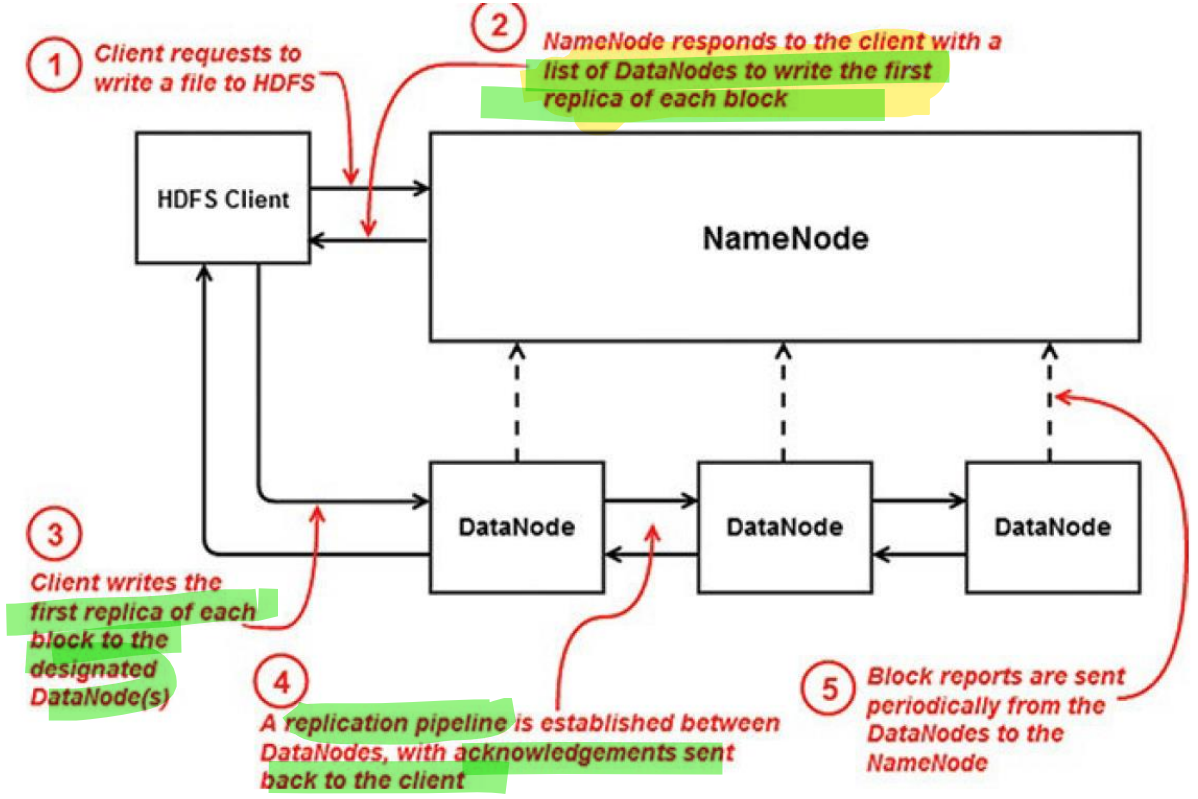
HDFS

YARN

Anatomy of HDFS Read Operations



Anatomy of HDFS Write Operations



Application Scheduling using YARN

The YARN cluster architecture is a master/slave cluster framework like HDFS, with a master node daemon called the *ResourceManager* and one or more slave node daemons called *NodeManagers* running on worker, or slave, nodes in the cluster.

The ResourceManager is responsible for granting cluster compute resources to applications running on the cluster. Resources are granted in units called *containers*, which are predefined combinations of CPU cores and memory. Container allotments, including minimum and maximum thresholds, are configurable on the cluster. Containers are used to isolate resources dedicated to a process or processes.

Application Scheduling using YARN

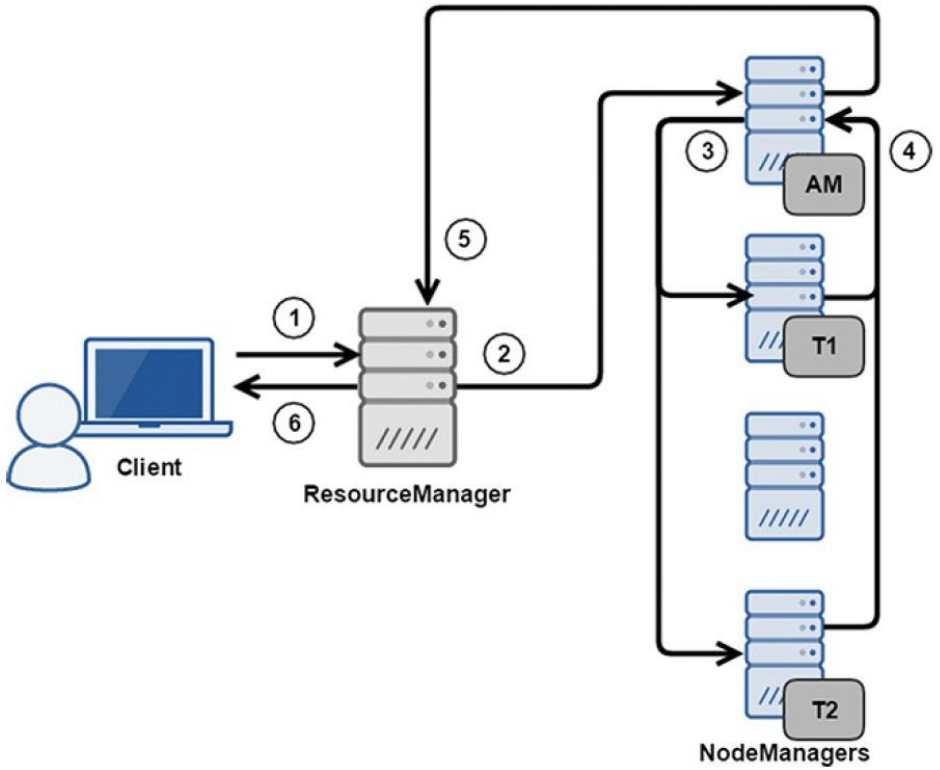
The ResourceManager also tracks available capacity on the cluster as applications finish and release their reserved resources, and it tracks the status of applications running on the cluster. The ResourceManager serves an embedded web UI on port 8088 of the host running this daemon, which is useful for displaying the status of applications running, completed, or failed on the cluster

Application Scheduling using YARN

Clients submit applications, such as Spark applications, to the ResourceManager; the ResourceManager then allocates the first container on an available NodeManager in the cluster as a delegate process for the application called the *ApplicationMaster*; the ApplicationMaster then negotiates all further containers required to run tasks for the application The NodeManager is the slave node YARN daemon that manages containers on the slave node host. Containers are used to execute the tasks involved in an application. As Hadoop's approach to solving large problems is to "divide and conquer," a large problem is deconstructed into a set of tasks, many of which can be run in parallel; recall the concept of shared nothing. These tasks are run in containers on hosts running the NodeManager process. Most containers simply run tasks. However, the ApplicationMaster has some additional responsibilities for managing an application

The ApplicationMaster is the first container allocated by the ResourceManager to run on a NodeManager. Its job is to plan the application, including determining what resources are required—often based on how much data is being processed—and to work out resourcing for application stages The ApplicationMaster requests these resources from the ResourceManager on behalf of the application. The ResourceManager grants resources on the same or other NodeManagers to the ApplicationMaster.

Application Scheduling using YARN



Application Scheduling using YARN

1. A client submits an application to the ResourceManager.

2. The ResourceManager allocates an ApplicationMaster process on a NodeManager with sufficient capacity to be assigned this role.

3. The ApplicationMaster negotiates task containers with the ResourceManager to be run on NodeManagers—which can include the NodeManager on which the ApplicationMaster is running as well—and dispatches processing to the NodeManagers hosting the task containers for the application.

Application Scheduling using YARN

4. The NodeManagers report their task attempt status and progress to the ApplicationMaster.

5. The ApplicationMaster reports progress and the status of the application to the ResourceManager.

6. The ResourceManager reports application progress, status, and results to the client.

Apache Spark

A major disadvantage of Hadoop's MapReduce implementation was its persistence of intermediate data to disk between the Map and Reduce processing phases.

As an alternative to MapReduce, Spark implements a distributed, fault-tolerant, in-memory structure called a Resilient Distributed Dataset (RDD). Spark maximizes the use of memory across multiple machines, significantly improving overall performance. Spark's reuse of these in-memory structures makes it well suited to iterative machine learning operations as well as interactive queries.

Uses of Spark

- Extract-transform-load (ETL) operations
- Predictive analytics and machine learning
- Data access operations, such as SQL queries and visualizations
- Text mining and text processing
- Real-time event processing
- Graph applications
- Pattern recognition
- Recommendation engines

Python Functional Programming

- Functions as first-class objects and the fundamental unit of programming
- Functions with input and output only (Statements, which could result in side effects, are not allowed.)
- Support for higher-order functions
- Support for anonymous functions