ORACLE Certified Professional

Oracle Certified Professional Java Programmer

Microsoft Certified Professional

sohailimran@yahoo.com

Sohail IMRAN سهيل عمران

intro

Ref: Web

Common systems we run today

The challenges that supporting all these systems brings:

- They have to build the in-house knowledge to administer and keep the systems up and healthy. Systems fail, especially complicated distributed systems, and being open source, many of these systems don't have the tooling to facilitate easy management.

- Data exchange between systems is painful, primarily due to the volume of data and the lack of tooling for the data movement. Large, expensive projects ensue.

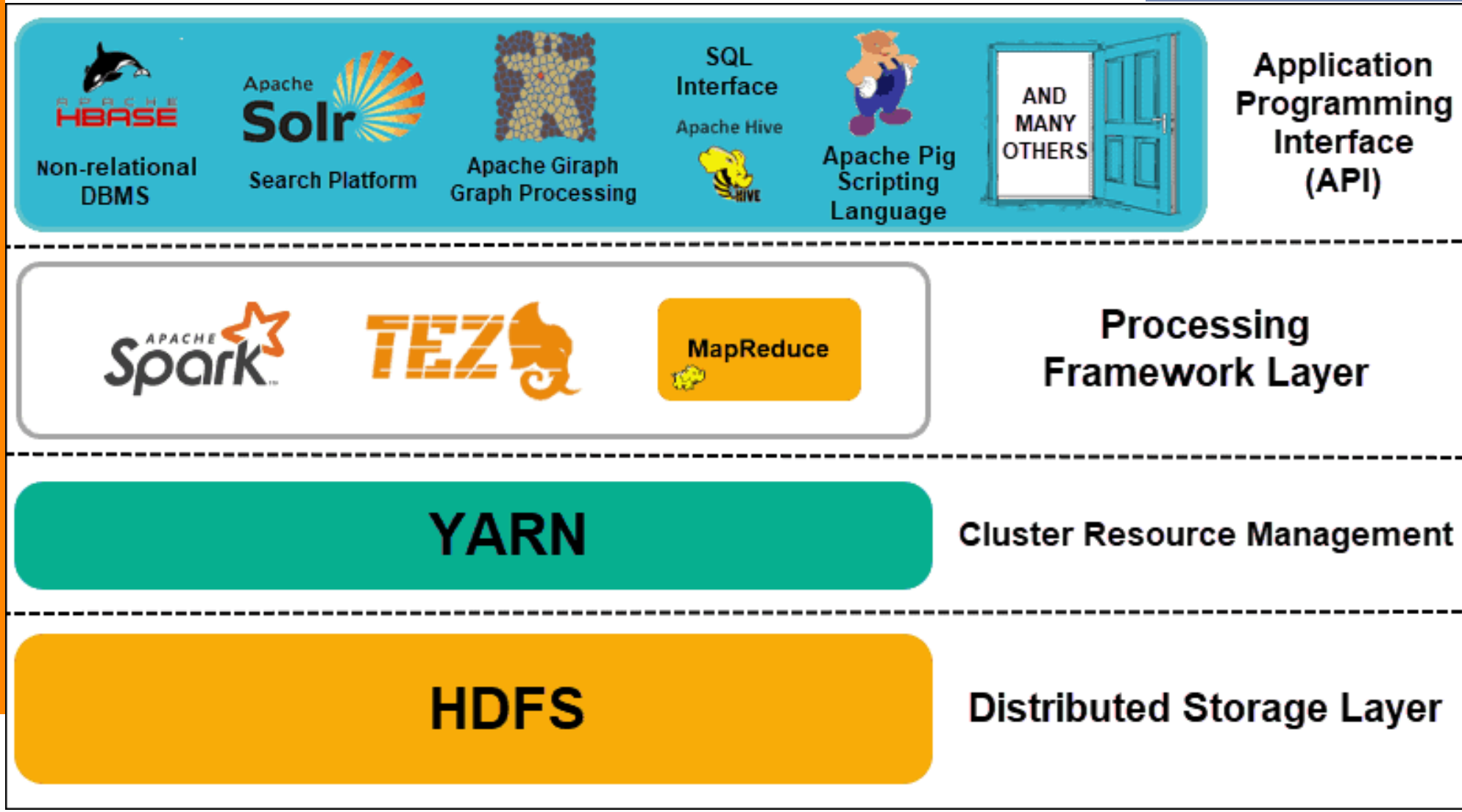
- Each system has to solve the same distributed problems, such as fault tolerance, distributed storage, log handling, and resource scheduling.
- https://www.edureka.co/blog/mapreduce-tutorial/

a distributed framework for big data

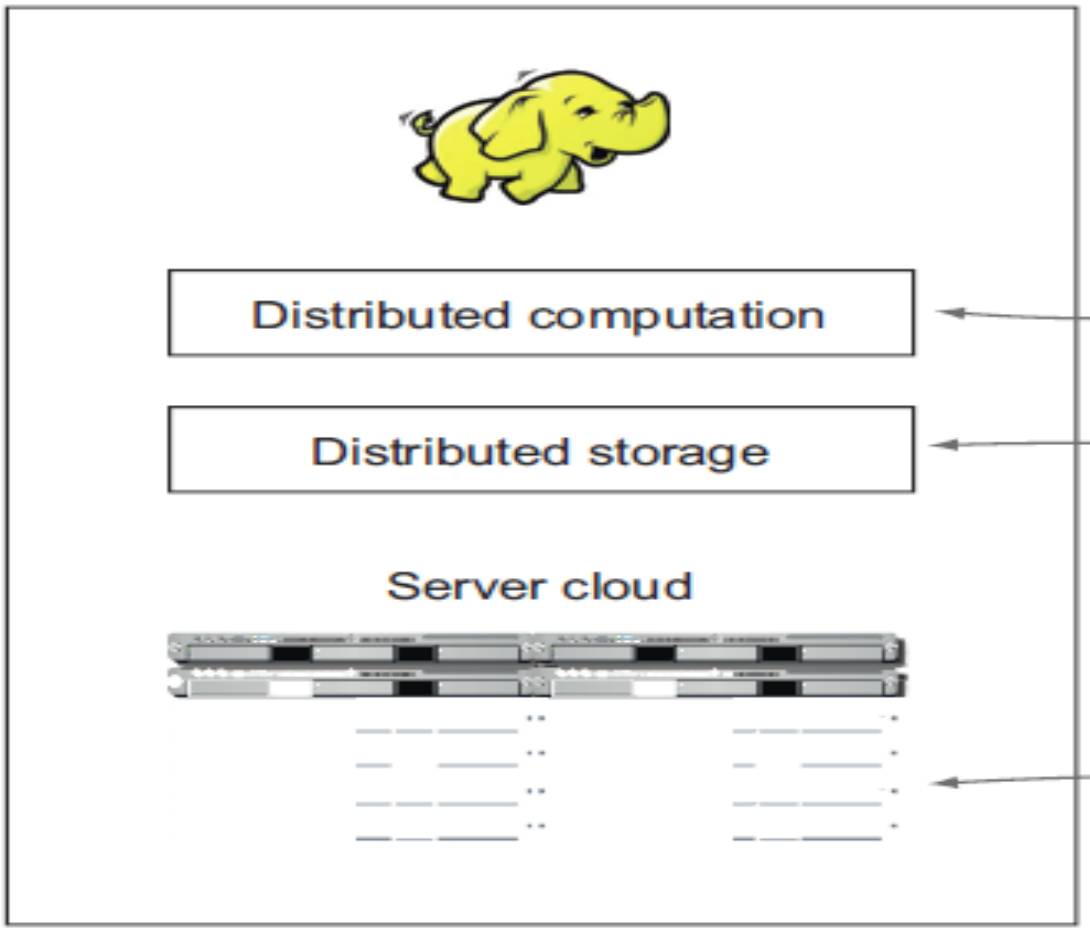
- Apache top level project, open-source implementation of frameworks for reliable, scalable, distributed computing and data storage.
- It is a flexible and highly-available architecture for large scale computation and data processing on a network of commodity hardware.

Designed to answer the question:
"How to process big data with reasonable cost and time?"

layers


Non-relational DBMS
Search Platform
Apache Giraph Graph Processing
SQL Interface
Apache Hive
Apache Pig Scripting Language
AND MANY OTHERS
Application Programming Interface (API)
MapReduce
Processing Framework Layer
YARN
Cluster Resource Management
HDFS
Distributed Storage Layer

hadoop environment - a distributed system

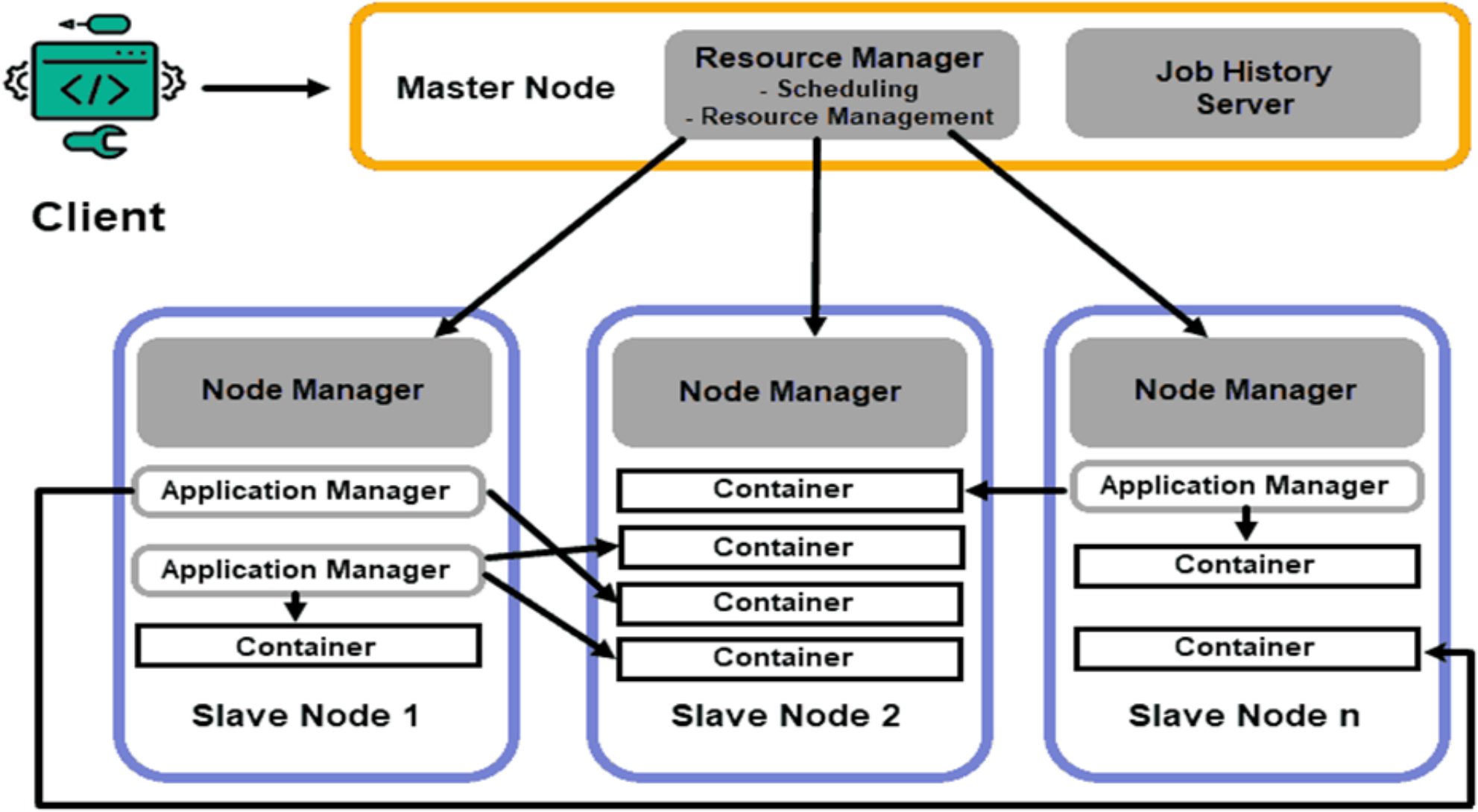


The computation tier is a general-purpose scheduler and a distributed processing framework called MapReduce.

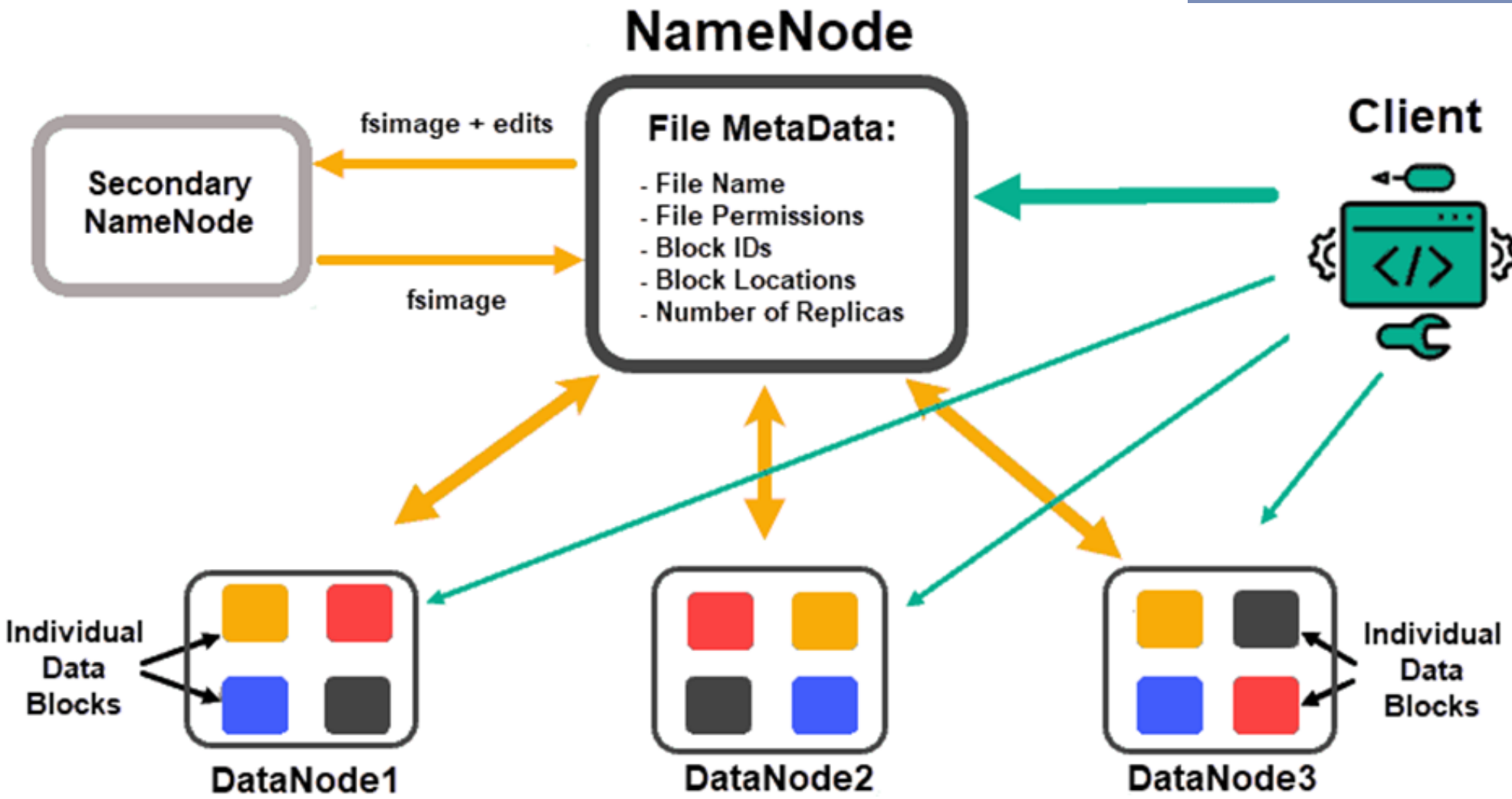
Storage is provided via a distributed filesystem called HDFS.

Hadoop runs on commodity hardware.

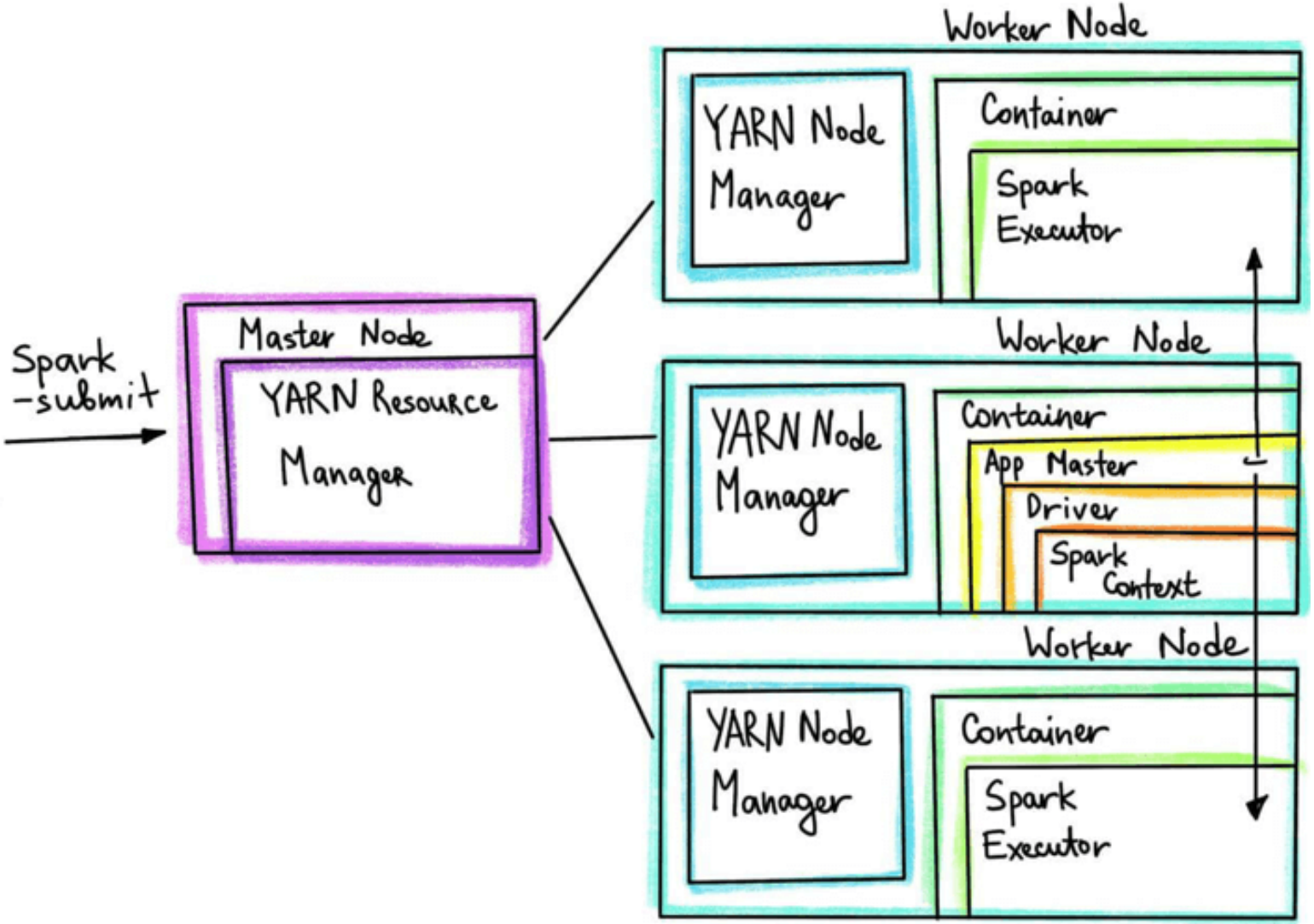
architecture



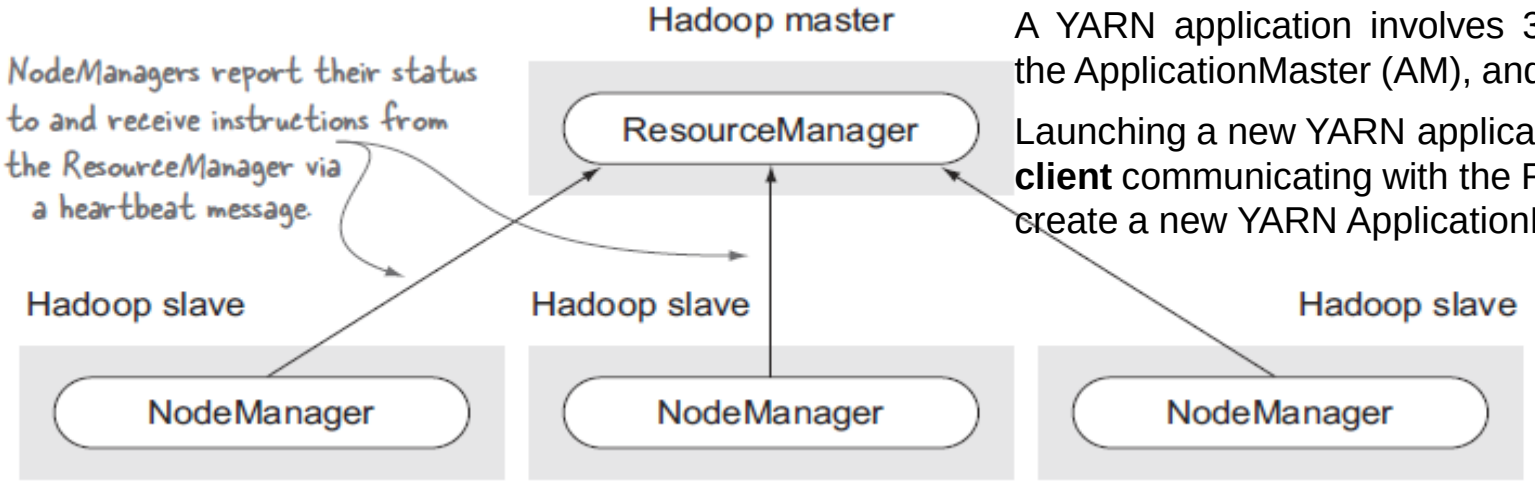
hdfs


NameNode
Secondary NameNode
fsimage + edits
fsimage
File MetaData:
- File Name
- File Permissions
- Block IDs
- Block Locations
- Number of Replicas
Client
Individual Data Blocks
DataNode1
DataNode2
DataNode3
Individual Data Blocks

yarn


Worker Node
YARN Node Manager
Container
Spark Executor
Spark -submit
Master Node
YARN Resource Manager
Worker Node
YARN Node Manager
Container
App Master
Driver
Spark Context
Worker Node
YARN Node Manager
Container
Spark Executor

interactions of a yarn application

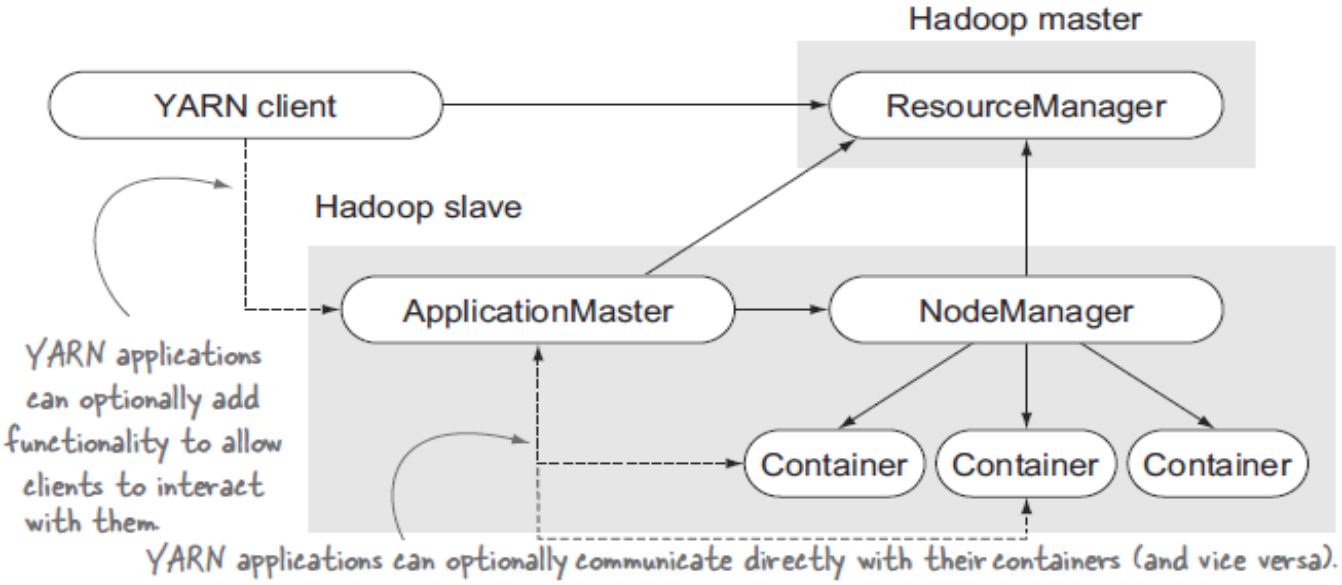


A YARN application involves 3 components—the client, the ApplicationMaster (AM), and the container.

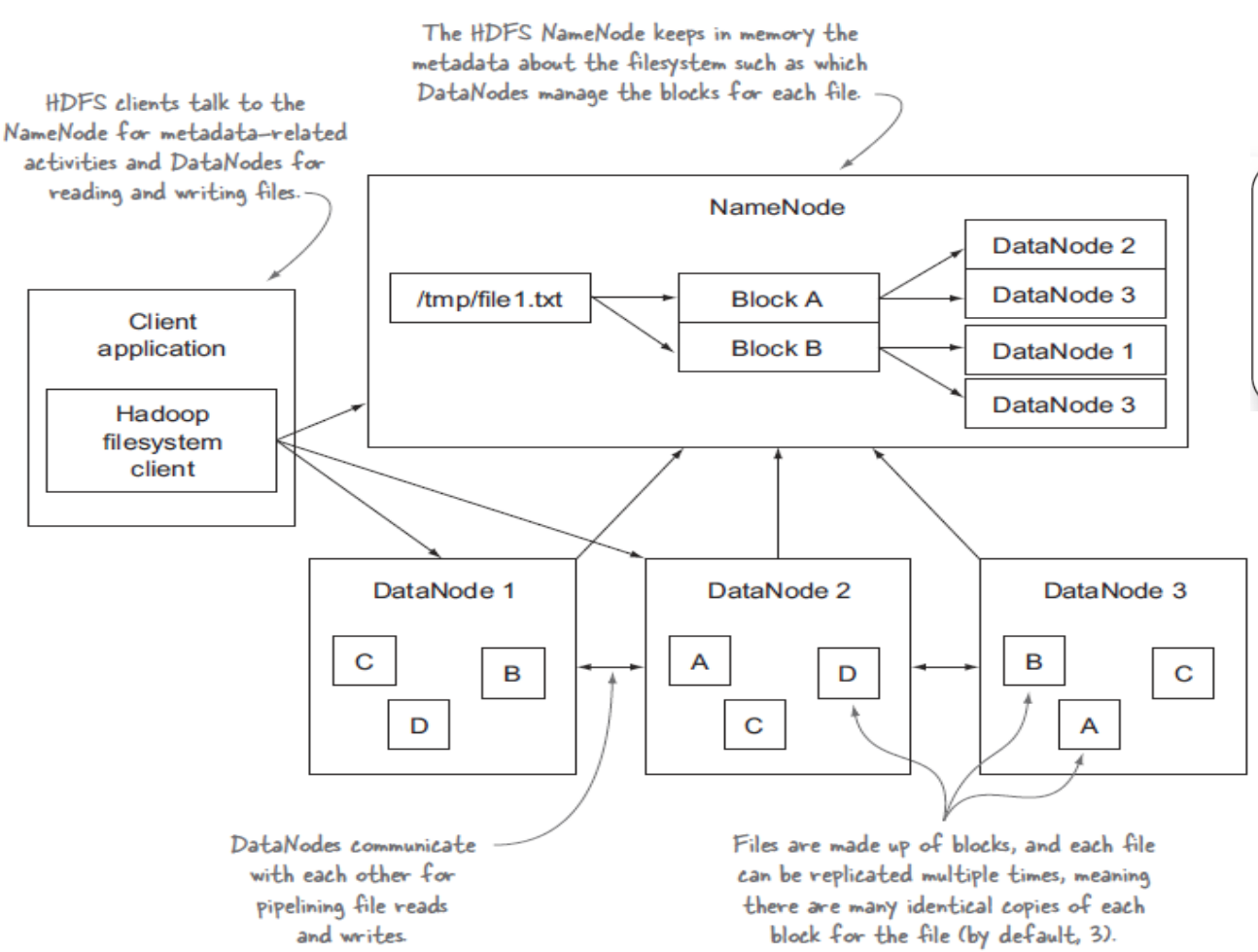
Launching a new YARN application starts with a YARN **client** communicating with the ResourceManager to create a new YARN ApplicationMaster instance.

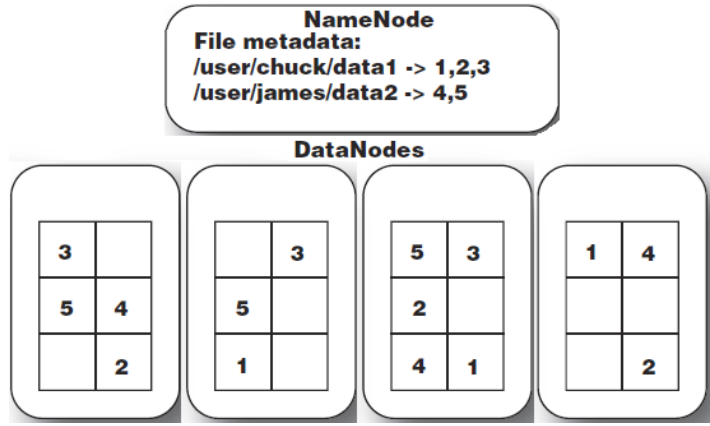
the **ApplicationMaster** must specify the resources that each container requires in terms of which host should launch the container and what the container's memory and CPU requirements are.

A ***container*** is an application-specific process that's created by a NodeManager on behalf of an ApplicationMaster.



hdfs client communication



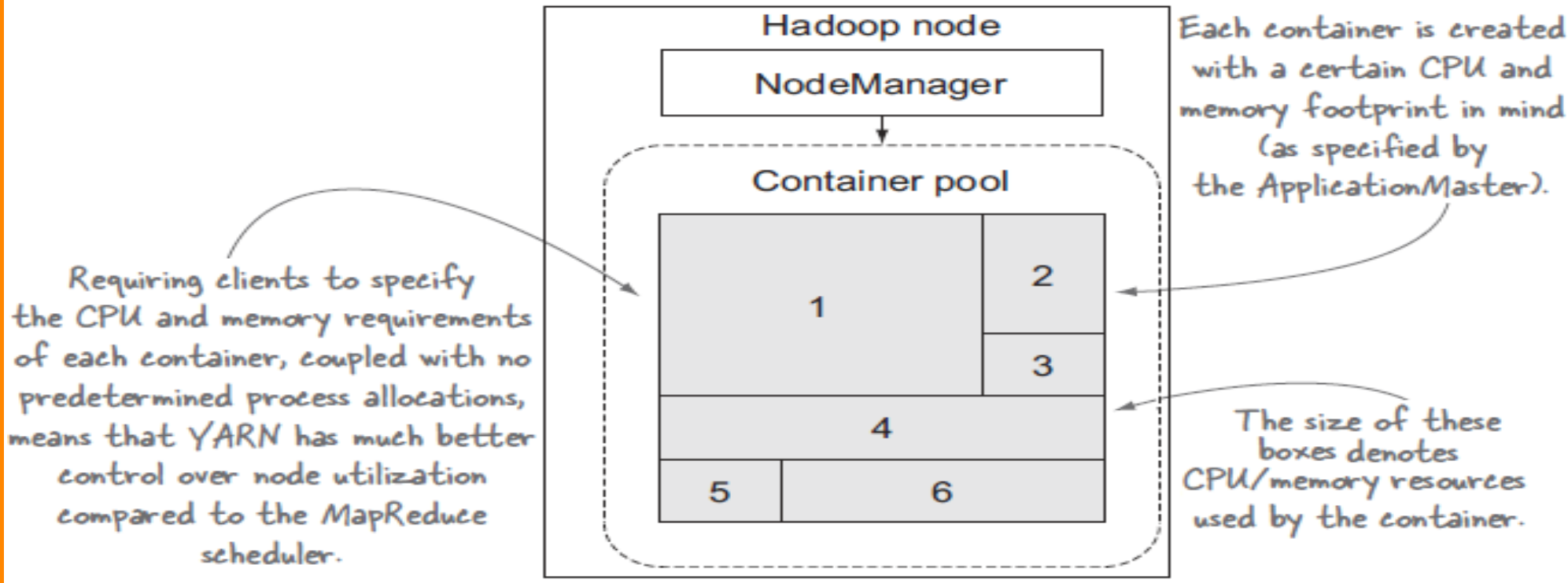


Hadoop 2 introduced two significant new features for HDFS—Federation and **High Availability** (HA):

- Federation allows HDFS **metadata** to be **shared** across **multiple NameNode** hosts, which aides with HDFS scalability and also provides data isolation, allowing different applications or teams to run their own NameNodes without fear of impacting other NameNodes on the same cluster.
- High Availability in HDFS removes the single point of failure that existed in Hadoop 1, wherein a NameNode disaster would result in a cluster outage. HDFS HA also offers the ability for failover (the process by which a standby Name-Node takes over work from a failed primary NameNode) to be automated.

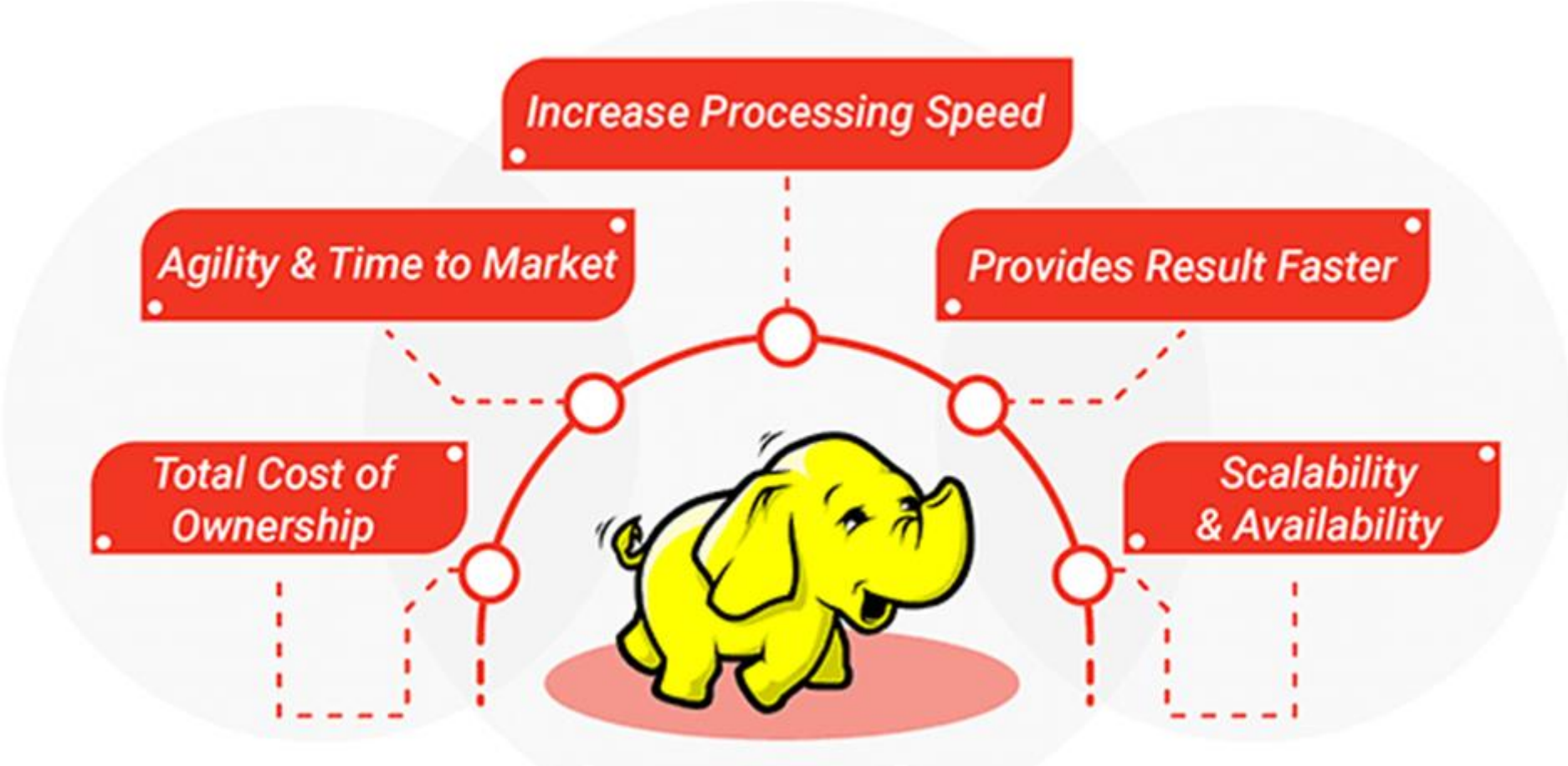
Container configurations

The ability of the ResourceManager to schedule work based on exact resource requirements is a key to YARN's flexibility, and it enables hosts to run a mix of containers.

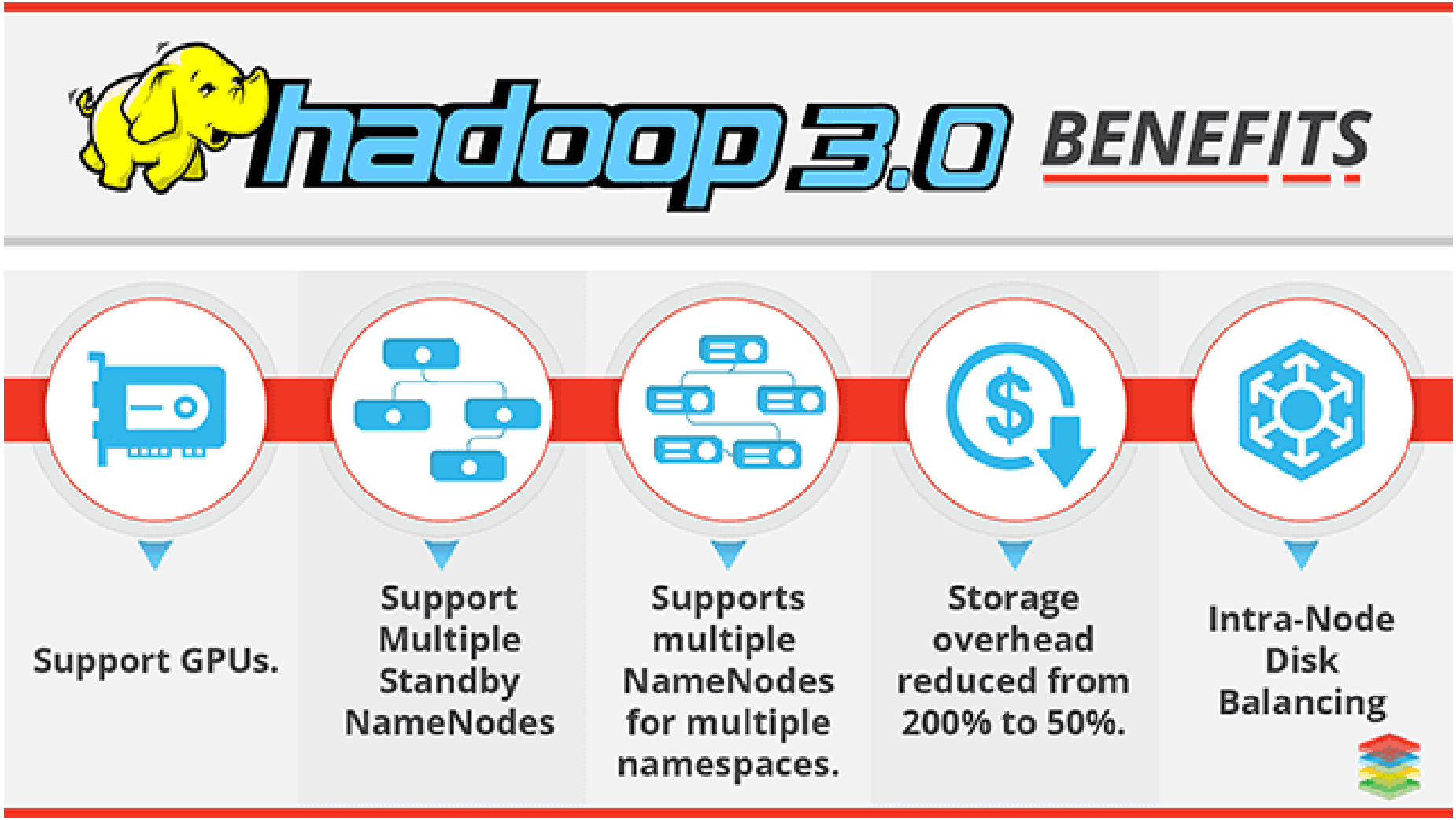


A container created by an ApplicationManager can be an arbitrary process—for example, a container process could simply be a Linux command such as awk, a Python application, or any process that can be launched by the operating system. This is the power of YARN—the ability to launch and manage any process across any node in a Hadoop cluster.

hadoop with gpu



benefits


hadoop 3.0 BENEFITS
Support GPUs.
Support Multiple Standby NameNodes
Supports multiple NameNodes for multiple namespaces.
Storage overhead reduced from 200% to 50%.
Intra-Node Disk Balancing