

Cloud Computing Based Hadoop Mapreduce Analytics Using Python Framework

V. Shanmugarajeshwari

Assistant Professor

Department of Computer Applications (UG & PG)

Ayya Nadar Janaki Ammal College Sivakasi, Tamil Nadu, India

v.shanmugarajeshwari@gmail.com

Abstract— Cloud means Internet. Cloud computing is one of the potential research fields regarding interdisciplinary aspects. Hadoop mapreduce is developing discipline in the present scenario. Mapreduce techniques in the cloud computing plays an important role in the various area of virtualization, load balancing, scalability & elasticity, deployment, replication, monitoring, mapreduce, identity and access management, service level agreements and filling. The main goal regarding this paper is used to analysis the relevant features of services, deployment, technologies and applications using cloud computing based hadoop mapreduce process.

Keywords— Cloud Computing; Hadoop MapReduce; Service Model; Deployment Model; Cloud Computing techniques.

1.INTRODUCTION

The U.S. National Institute of Standards and Technology (NIST) defines cloud computing as: Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

a) Characteristics of Cloud Computing

- On-demand self service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service
- Performance
- Reduced costs
- Outsourced Management
- Reliability
- Multi-tenancy

b) Cloud Service Models

1. Software as a Service (SaaS)

Applications, management and user interfaces provided over a network

2. Platform as a Service (PaaS)

Application development frameworks, operating systems and deployment frameworks

3. Infrastructure as a Service (IaaS)

Virtual computing, storage and network resource that can be provisioned on demand

c) Cloud Deployment Models

- Public Cloud
Available for public use or a large industry group
- Private Cloud
Operated for exclusive use of a single organization
- Community Cloud
Available for shared use of several organizations supporting a specific community
- Hybrid Cloud
Combines multiple clouds (public and private) that remain unique but bound together to offer application and data portability

d) Cloud Service Examples

- IaaS:
Amazon EC2
Google Compute Engine
Windows Azure VMs
- PaaS:
Google App Engine
- SaaS:
Salesforce

f) Cloud Computing Applications

- Financial & Banking Applications
- Electronic-Commerce Applications
- Any Social Networking Processing
- Healthcare Monitoring Systems
- Energy Consumption Systems
- Intelligent Transportation Systems (ITS)
- Electronic-Governance
- Educational System
- Mobile/Cell Phone Communications

g) Hadoop MapReduce

MapReduce is a parallel data processing model for processing and analysis of massive scale data.

- MapReduce phases:

1. *Map Phase:* In the Map phase, data is read from a distributed file system, partitioned among a set of computing nodes in the cluster, and sent to the nodes as a set of key-value pairs. The Map tasks process the input records independently of each other and produce intermediate results as key-value pairs. The intermediate results are stored on the local disk of the node running the Map task.

2. *Reduce Phase:* When all the Map tasks are completed, the Reduce phase begins in which the intermediate data with the same key is aggregated [3].

Apache Hadoop is an open source framework for distributed batch processing of big data.

- Hadoop Ecosystem includes:
- Hadoop MapReduce
- HDFS
- YARN
- HBase
- Zookeeper
- Pig
- Hive
- Mahout
- Chukwa
- Cassandra
- Avro
- Oozie
- Flume
- Sqoop

h) Python

Python is a general-purpose high level programming language and suitable for providing a solid foundation to the reader in the area of cloud computing.

The main characteristics of Python are:

- Multi-paradigm programming language
- Interpreted / Interactive Language
- Easy-to-learn, read and maintain
- Object and Procedure Oriented
- Extendable
- Scalable
- Portable
- Broad Library Support

A file represents a sequence of bytes on the disk where a group of related data is stored.

- File is created for permanent storage of data.
- Python has several functions for creating, reading, updating, and deleting files [4].

File Handling Function

- Python is the opening mode name is open() function
- The Opening mode of python open() function takes two parameters; filename, and mode.

```
f = open("D:\MCA\Python\myfile.txt", "x") //To create a new file
```

Step : 1

```
f = open("samplefile.txt")
```

Step : 2

```
f = open("samplefile.txt", "rt")
```

R represents Read , t represents Text

```
f = open("samplefile.txt", "w")
```

```
f.write("WELCOME TO MCA") //Overall Write
```

```
f = open("samplefile.txt", "a")
```

```
f.write("THEN WELCOME TO BCA!")//After,Adding the line
```

Opening and Reading the file

```
>>> f=open("D:\MCA\Python\sample.txt")
>>> print(f.read())
```

```
>>> f=open("D:\MCA\Python\sample.txt")
>>> print(f.read(5)) //Hello
```

```
>>> f=open("D:\MCA\Python\sample.txt")
>>> print(f.readline()) // Read One Line
>>> print(f.readline()) // Read Second Line
```

Deleting the file

```
>>> import os
>>> os.remove("D:\MCABCARAJI\Python\sample.txt")
```

Deleting the directory

```
>>> import os
>>> os.rmdir("D:\Sample")//Only Delete the empty folder
```

Python (Classes):

```
class hello():
    def dis(self):
        print 'helloshana'

if __name__=='__main__':
    app=hello()
    app.dis()
```

OUTPUT: helloshana

Python Function:

```
def addtwo(a, b):
    added = a + b
```

```
return added
x = addtwo(3, 5)
print ("ARITHMETIC OPERATION ADD VALUE:",+ x)
```

OUTPUT: 8

Python (Classes- Inheritance-Multiple):

```
class study1:
    def s1(self):
        print 'BCA RAJI'

class study2():
    def s2(self):
        print 'MCA RAJI'

class study3(version1,version2):
    def s3(self):
        print 'M.Phil. RAJI'

if __name__=='__main__':
    obj=study3()
    obj.s1()
    obj.s2()
    obj.s3()
```

OUTPUT: BCA RAJI, MCA RAJI, M.Phil. RAJI

Python (Classes- Inheritance-Multi Level):

```
Class course1:
    def c1(self):
        print 'BCA'

class course2(course1):
    def c2(self):
        print 'MCA'

class course 3(course 2):
    def c3(self):
        print' MHPIL'

if __name__=='__main__':
    obj=course3()
    obj.c1()
    obj.c2()
    obj.c3()
```

OUTPUT:

BCA
MCA
BCA
MPHIL

Python Modules:

```
>>> import sys
```

```
>>> sys.path
['', 'C:\\Users\\hp\\AppData\\Local\\Programs\\Python\\Python37-32\\Lib\\idlelib',
'C:\\Users\\hp\\AppData\\Local\\Programs\\Python\\Python37-32\\python37.zip',
'C:\\Users\\hp\\AppData\\Local\\Programs\\Python\\Python37-32\\DLLs',
'C:\\Users\\hp\\AppData\\Local\\Programs\\Python\\Python37-32\\lib',
'C:\\Users\\hp\\AppData\\Local\\Programs\\Python\\Python37-32',
'C:\\Users\\hp\\AppData\\Local\\Programs\\Python\\Python37-32\\lib\\site-packages']
```

```
>>> import random
>>> random.randint(0,5)
4
>>> List = [1, 4, True, 800, "python", 27, "hello"]
>>> random.choice(List)
'python'
```

Python Packages:

1. FUNCTION

```
def funraji123():
    print ("HI HELLO PACKAGE RAJI")
```

2. PACKAGE

```
import mypackageraji.first as mpr
mpr.funraji123()
```

OUTPUT: HI HELLO PACKAGE RAJI

1. FUNCTION

```
def funraji123():
    print ("HI HELLO SUB PACKAGE RAJI")
```

2. PACKAGE

```
import mypackageraji.subpackageraji.subpack as spr
spr.funraji123()
```

OUTPUT: HI HELLO SUB PACKAGE RAJI

3. CLOUD APPLICATION DEVELOPMENT IN PYTHON

a) MapReduce App – Component Design

1. Functionality:

This application allows users to submit MapReduce jobs for data analysis. This application is based on the Amazon Elastic MapReduce (EMR) service. Users can upload data files to analyze and choose/upload the Map and Reduce programs. The selected Map and Reduce programs along with the input data are submitted to a queue for processing [5].

2. Component Design

Web Tier: The web tier for the MapReduce app has a front end for MapReduce job submission.

Application Tier: The application tier has components for processing requests for uploading files, creating MapReduce jobs and enqueueing jobs, MapReduce consumer and the component that sends email notifications.

Analytics Tier: The Hadoop framework is used for the analytics tier and a cloud storage is used for the storage tier.

Storage Tier: The storage tier comprises of the storage for file [2].

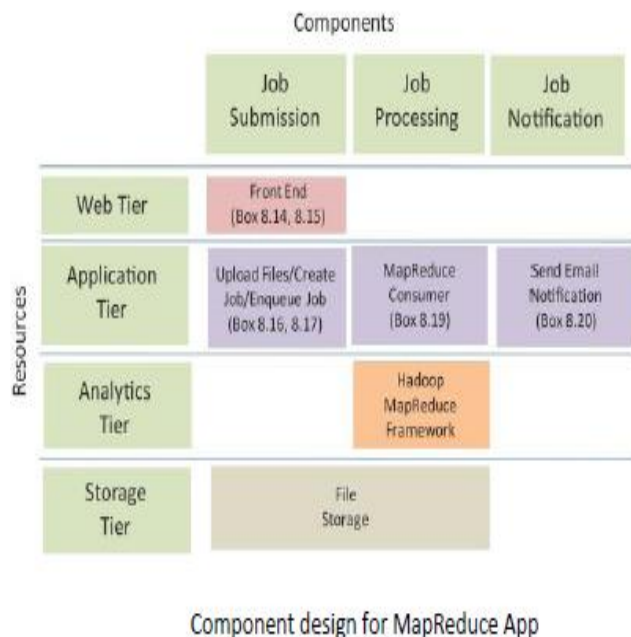


Fig 1. Component design for MapReduce App Block Diagram

b) MapReduce App – Architecture Design

Architecture design step which defines the interactions between the application components.

This application uses the Django framework, therefore, the web tier components map to the Django templates and the application tier components map to the Django views. For each component, the corresponding code box numbers are mentioned. To make the application scalable the job submission and job processing components are separated. The MapReduce job requests are submitted to a queue. A consumer component that runs on a separate instance retrieves the MapReduce job requests from the queue and creates the MapReduce jobs and submits them to the Amazon EMR service. The user receives an email notification with the download link for the results when the job is complete [6].

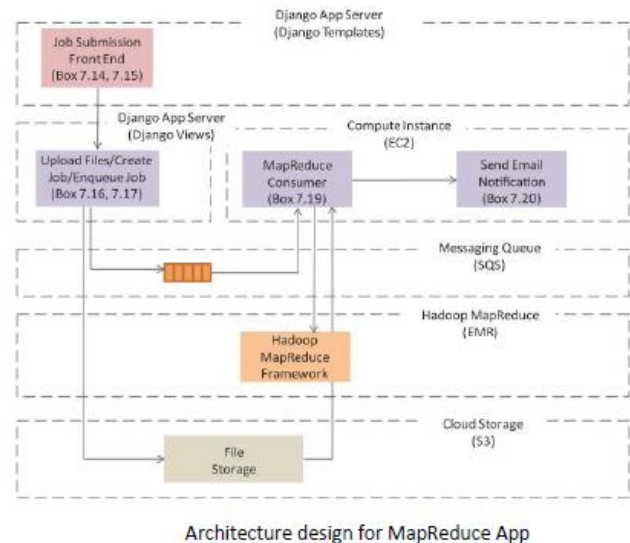


Fig 2. Architectural design for MapReduce App Block Diagram

c) MapReduce App – Deployment Design

Deployment for the app is a multi-tier architecture comprising of load balancer, application servers and a cloud storage for storing MapReduce programs, input data and MapReduce output. For each resource in the deployment the corresponding Amazon Web Services (AWS) cloud service is mentioned [7].

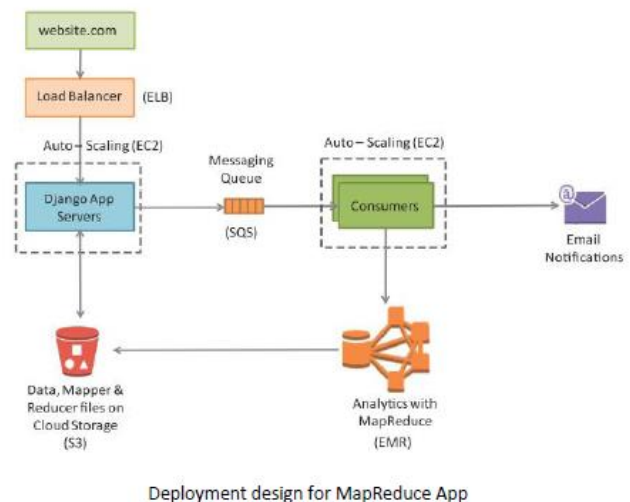


Fig 3. Deployment design for MapReduce App Block Diagram

3. APACHE HADOOP

A Hadoop cluster comprises of a Master node, backup node and a number of slave nodes.

The master node: It runs the name node and job tracker processes and the slave nodes run the data node and task tracker components of Hadoop.

The backup node: It runs the secondary name node process.

The name node: It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself. Client applications talk to the name node whenever they wish to locate a file, or when they want to add/copy/move/delete a file [10].

Secondary name node: Name node is a Single Point of Failure for the HDFS Cluster. An optional Secondary name node which is hosted on a separate machine creates checkpoints of the namespace.

Job tracker: It is the service within Hadoop that distributes MapReduce tasks to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack.

Task tracker: Task tracker is a node in a Hadoop cluster that accepts Map, Reduce and shuffle tasks from the Job tracker. Each task tracker has a defined number of slots which indicate the number of tasks that it can accept.

Data node: A Data node stores data in an HDFS file system. A functional HDFS file system has more than one Data node, with data replicated across them. Data nodes respond to requests from the name node for file system operations. Client applications can talk directly to a data node, once the name node has provided the location of the data. Similarly, MapReduce operations assigned to task tracker instances near a data node, talk directly to the data node to access the files. Task tracker instances can be deployed on the same servers that host data node instances, so that MapReduce operations are performed close to the data [8].

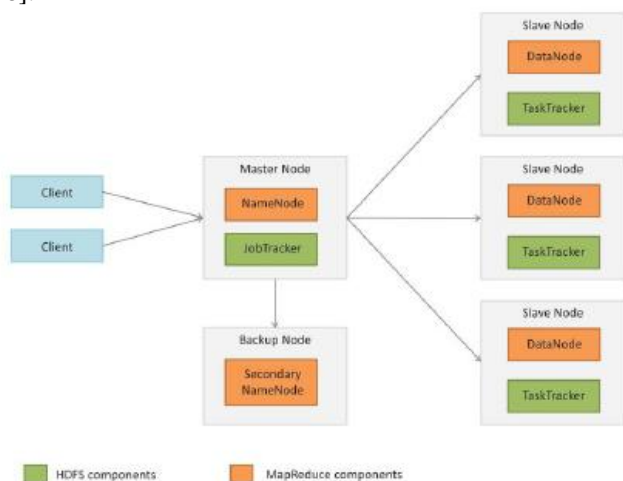


Fig 4. Components of Hadoop cluster

4. MAPREDUCE JOB EXECUTION WORKFLOW

MapReduce job execution starts when the client applications submit jobs to the Job tracker. The job tracker returns a job ID to the client application. The job tracker talks to the name node to determine the location of the data. The job tracker locates task tracker nodes with available slots at/or near the data. The task trackers send out heartbeat messages to the job tracker, usually every few minutes, to reassure the job tracker that they are still alive. These messages also inform the job tracker of the number of available slots, so the job tracker can stay up to date with where in the cluster, new work can be delegated [9].

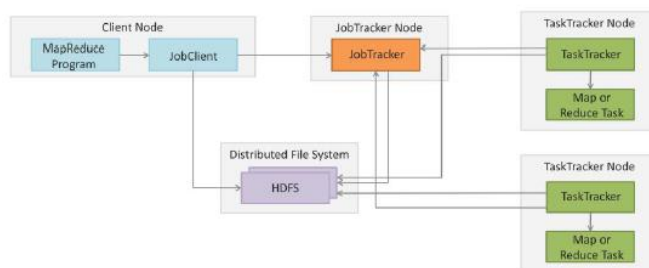


Fig 4. Hadoop MapReduce Job Execution

The job tracker submits the work to the task tracker nodes when they poll for tasks. To choose a task for a task tracker, the job tracker uses various scheduling algorithms (default is FIFO). The task tracker nodes are monitored using the heartbeat signals that are sent by the task trackers to job tracker. The task tracker spawns a separate JVM process for each task so that any task failure does not bring down the task tracker. The task tracker monitors these spawned processes while capturing the output and exit codes. When the process finishes, successfully or not, the task tracker notifies the job tracker. When the job is completed, the job tracker updates its status [1].

5. CONCLUSION

Cloud computing produces the better decisions for the big data using Apache hadoop methods. This paper is mainly used to various cloud computing techniques using python based hadoop mapreduce process. Python language is support any framework. The various methods of python classes, objects, inheritance, modules, function, packages and file handling coding are discussed in this paper.

REFERENCES

- [1] Bahga, A., & Madiseti, V., "CLOUD COMPUTING A HANDS ON APPROACH", UNIVERSITIES PRESS(INDIA) 2014.
- [2] <http://www.cloudcomputingbook.info>
- [3] <https://www.edureka.co/blog/hadoop-streaming-mapreduce-program/>
- [4] <https://www.dezyre.com/hadoop-tutorial/hadoop-mapreduce-tutorial->

- [5] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4224309/>
- [6] <https://www.newgenapps.com/blog/python-for-big-data-science-projects-benefit-uses>
- [7] <https://www.knowledgehut.com/blog/big-data/5-best-data-processing-frameworks>
- [8] <https://www.analyticsindiamag.com/10-hadoop-alternatives-consider-big-data/>
- [9] <https://searchcloudcomputing.techtarget.com/definition/MapReduce>
- [10] <https://cloud.google.com/appengine/docs/standard/python/dataprocessing/>