LUPINE PUBLISHERS
Open Access

Advances in Robotics & Mechanical Engineering
DOI: 10.32474/ARME.2018.01.000103

ISSN: 2643-6736

Research Article

Processing and Analysis of Large-Scale Seismic Signal in Hadoop Platform

Shristi Bharti[1], Priyanka S[1], Chinmay Bhushan[1] and Shahrukh Javed[2]*

[1]*Department of Information Technology Engineering, Birla Institute of Technology, India*

[2]*Department of Electronics and Communication Engineering, T John Institute of Technology, India*

Received: September 10, 2018; **Published:** September 20, 2018

***Corresponding author:** Shahrukh Javed, Department of Electronics and Communication Engineering, T John Institute of Technology, Gottigere, Bengaluru-560076, India

Abstract

Through the usage of fifteen noteworthy ventures by the International Seismological Bureau, world has fabricated a seismic observing system, which makes all local and global seismic information that can be observed to published on a week after week for the client download. Given the immense measure of data on this information, Hadoop stage has possessed the capacity to oversee and capacity productively, and to break down more significant data. It has received appropriated storage to enhance the literacy rate and grow the capacity limit, also it has utilized MapReduce to coordinate the information in the HDFS (Hadoop Distributed File System) to guarantee that they are broke down and prepared rapidly. In the interim, it likewise has utilized excess information stockpiling to guarantee information security, in this way making it an instrument for taking care of extensive information.

Introduction

Seismic data are the data extracted from the digital readings of seismic waves. Seismic waves are similar to the recorded echoes what we make on the top of rigged cliff. The only difference is that these seismic waves propagate downwards. In our modern society, information increases in high speed and a large amount of data resides on cloud platform. Over 1/3rd of total digital data are produced yearly which needs to be processed and analyzed. Huge-live digital data like seismic data, where even a small amount of information impacts greatly to human life has to be analyzed and processed to obtain more valuable information [1]. Thus, Hadoop ecological system comes into picture, which is easy to develop & process applications of mass data, has high fault tolerance nature, being developed on java platform and an open source, and ensures deployment of system [2].

Hadoop Architecture

Hadoop supports a traditional hierarchical file organization. HDFS & MapReduce are 2 cores of Hadoop. The Base support of Hadoop is Distributed storage through HDFS and the Program support of Hadoop is Distributed Parallel processing through MapReduce. This HDFS architecture is developed with features like high fault tolerance, expansibility, accessibility, high throughput rate to meet the demand of stream mode and processing super-large files, which can run on cheap commercial servers. It is Master/Slave architecture [3].

Master:

a) It has one Name Node (NN).

b) It manages namespace of file system and client's access operation on file

c) It is responsible for processing namespace operation of file systems (open, close, rename etc.) and also mapping of blocks to Data Node (DN).

Slave:

a) It has several data nodes i.e. one per node in a cluster.

b) It manages storage data.

c) It is responsible for processing file read-and-write requests, create, delete and copy the data block under unified control of NN.

d) The presence of single node NN in a cluster extraordinarily streamlines the structural design of the framework.

e) NN acts like repository for all HDFS metadata.

f) System is designed that never ever the user data flows through NN.

MapReduce Architecture:

a) It is a software structure for effectively composing applications which process immense measure of information like multi-terabytes informational collections in parallel on vast clusters in the sense thousands of nodes of commodity hardware in a dependable adaptation to non-critical failure way [4].

b) A MapReduce job, parts the information into autonomous pieces which are processed by map tasks in a total way.

c) Map task is the input data always is in a key-value pair is sorted by mapper function and resulting key-value pair is fed to reducer.

d) Both input and output undertakings are arranged in a document frameworks and system deals with scheduleling tasks, checking them and re-executing the fizzled tasks [5].

e) MapReduce is a circulated computing with single master node called job - tracker and one slave undertaking tracker per cluster node.

Data Preparing and Processing

Data Collection and Declaration

The data is downloaded from China Earthquake Scientific Share Data Centre. Digital data is stored in the form of excel spread sheets which we are going to download. Before the data is being stored in HDFS, the data should be kept in the CSV format. Over 300000 pieces of data are collected by the observation of various earthquake regions all over China since January 1st, 2015, only to record many small earthquakes every day. This paper counts and analyzes the earthquake statistics according to occurrence time and location with the use of MapReduce framework and pseudo-distributed platform of Hadoop.

Data Processing

Data processing environment is based on pseudo-distributed platform of Hadoop and its Master/Slave architecture. There are 4 major steps [6];

a) Data pretreatment: download the required data and keep it in .csv format.

b) Store data: store the data set into default input path of Hadoop i.e. bin/hadoopfs –put earthquake_data.csv/usr/input

c) Run the program: locally run the MapReduce program to obtain analysis result.

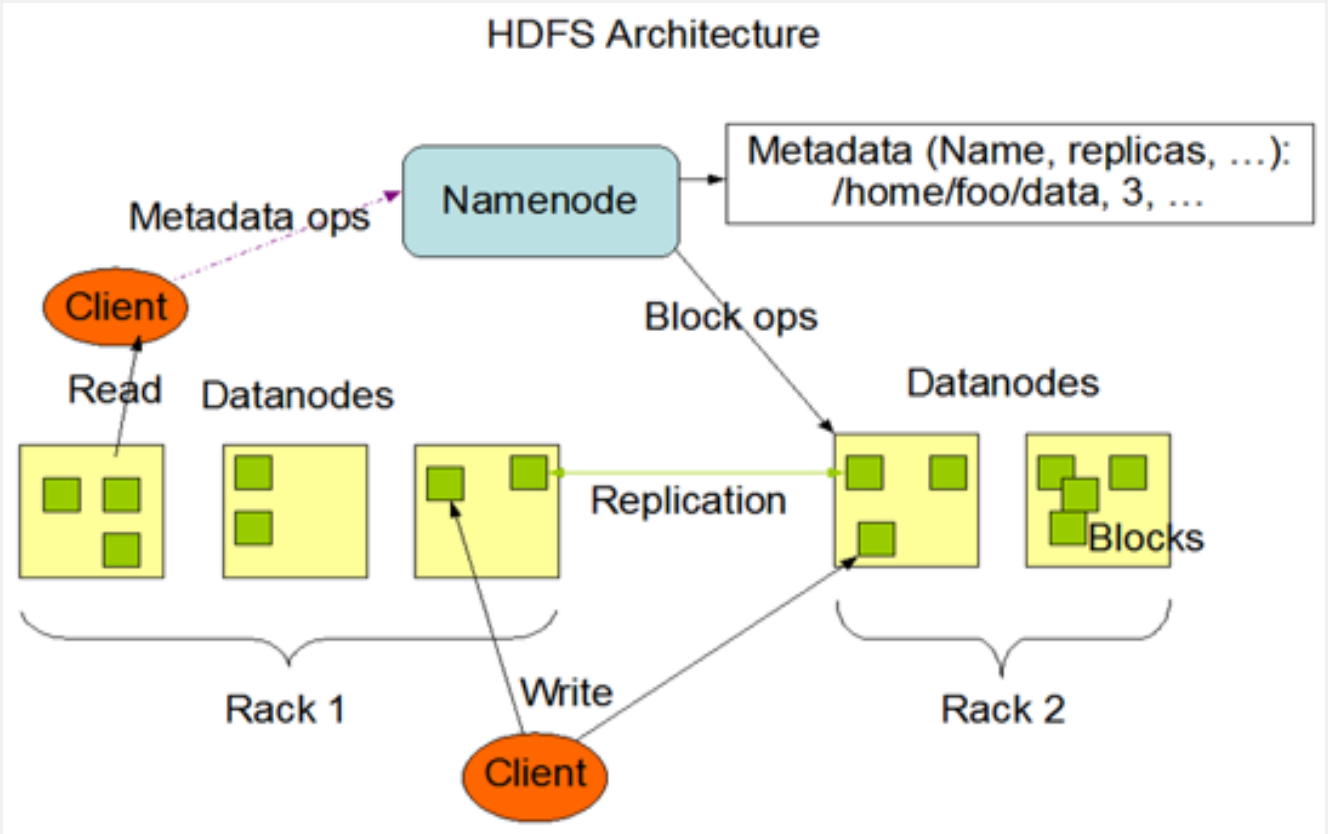

Figure 1: HDFS Architecture.

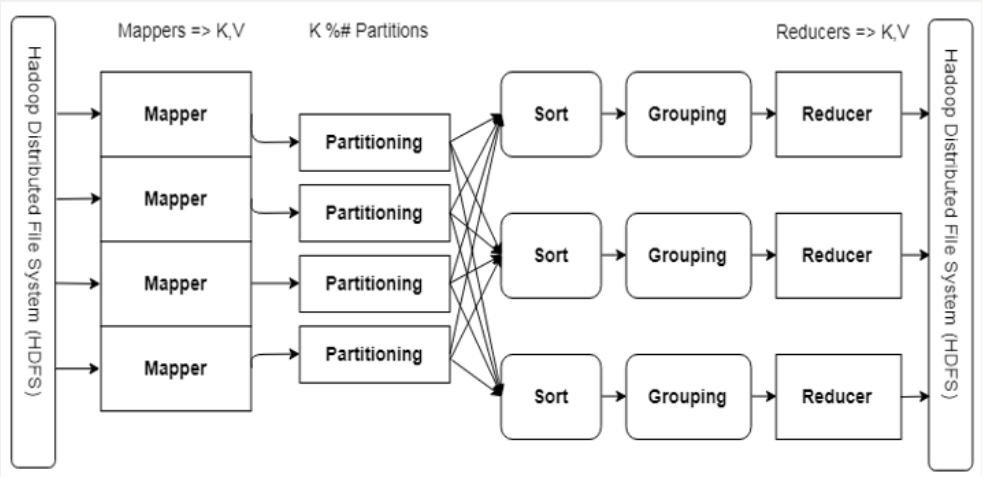

Figure 2: The MapReduce Pipeline.
A mapper receives (Key, Value) and outputs (Key, Value).
A reducer receives (Key, Iterable [Value]) and outputs (Key, Value).
Partioning/Sorting/Grouping provides the Iterable [Value] & Scaling.

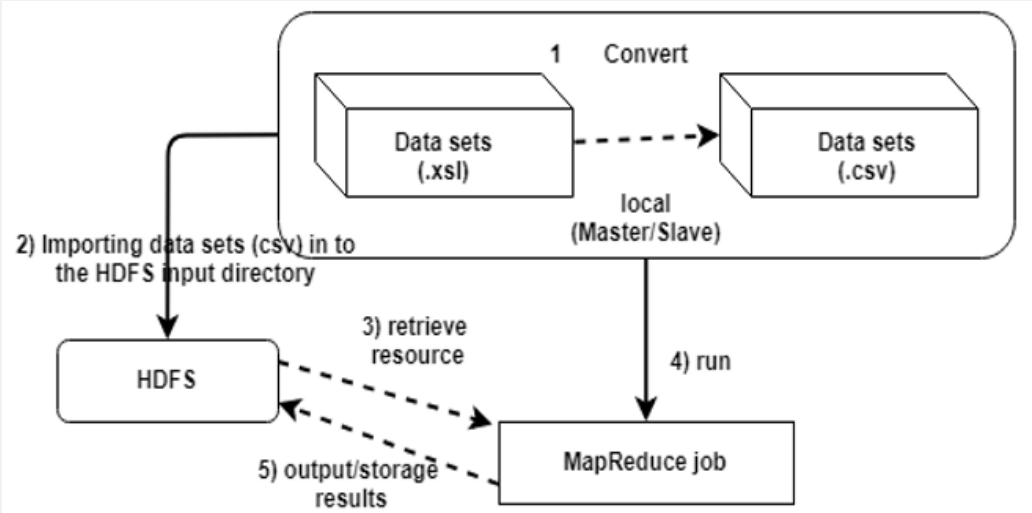

Figure 3: Earthquake data set processing flow chart.

d) Check the results: check the operation results in output directory of HDFS (Figures 1-3).

Proposed System

After the collection of required data there are two major steps for implementation. They are;

Analysis of CSV file:

a) Excel file looks like a table format but when it is converted to CSV it has only 3 lines.

b) First 2 lines are headers and third line have actual data separated by commas.

c) To analyze this file, this paper has used open source library called "opencsv", this works like;

```
//Copy a row from CSV file
String LINE=
"2015-10.27 02:11:23.4,30.14,98.02,8, Ms,4.2,
eq,The
Tibet autonomous region to prosperous cty ChaYa
county";
testReadingOneLine (){
//analysis of opencsv
String [ ]lines=ewCSVParser().parseLine(LINE);
//Print resolution resultfor (String line: lines) {
System.out.println(line);}
}
```

The result of the test analysis is shown below;

```
<terminated>CSVProcessingTest [Java Application]
2015-10-27
02:11:23.4
30.14
98.02
8
Ms
4.2
Eq
The Tibet Autonomous Region Dhangdu City Chaya
County
```

The Processing of Map and Reduce Functions [7]:

The Processing of map and reduce When processing data with MapReduce, firstly the data set file should be led into HDFS file system, and then the program will automatically divide the file into several pieces (default size 64MB) and read line by line [8-10]. Function map will analyze, preset the keyword in advance, and form into intermediate key-value pair. The program will automatically combine the key-value pairs of same key value, several corresponding values packaged in iterator, and the combination has been taken as the input key value of reduce processing. Reduce function accumulates to accumulate intermediate key/value pair which has been outputted at the form of, finally the total times of keyword in data set has been obtained [11].

Existing System

```
map function:
map(LongWritable key, Text value, Context context)
{
//Parse a CSV file data sets
String[] lines = new CSVParser(). parseLine(value.
toString());
//map process ,mapsent to the output datareduce
context.write(new Text(lines[8]), new IntWri table(1));
}

reduce function:
reduce(Text key, Iterablevalues,Context context)
{
int count = 0;
 //Query the iterator
for (IntWritable value : values) count++;
 //times of statistic reduce process
context.write(key,newIntWritable(count));
//reduce output
}
```

Proposed System

```
map function:
map(LongWritable key, Text value, Context context)
{
// csvresolverCSVParser
parser = new CSVParser();
// analysis of csv
String[] lines = parser.parseLine(value.toString()); String
dtstr = lines[0];
 //map process ,map and sent to the output datareduce
context.write(new Text(dtstr), new IntWri table(1));
}

reduce function:
reduce(Text key, Iterable values, Context context)
{
int count = 0;
 //Query the iterator
for (IntWritable value : values) count++;
//times of analysis reduce process
context.write(key, new IntWritable(count));
 //reduce output
 }
```

Table 1: A copy of one row data in data set.

Date	Time	Longitude	Depth	Magnitude Type	Magnitude Value	Event Type	Location	
2015- 10-27	2:11 23.4	30.14	98.02	8	Ms	4.2	eq	Tibet-Dhangdu-Chaya

Experimental Analysis and Results

Environment of Experiment:

a) Hardware configuration – CPU= Intel® Core™ i7- 4510U @2.00GHz 8.00GB of memory.

b) The virtual machine environment configuration [12]:

installing OS – Ubuntu12.04.

a) Hadoop version- Hadoop2.7.1

b) IDE – eclipse 4.3.0

Result: Based on region to region & on daily basis

Graphical Representation:

a) Graph on daily basis statistics graph from the data of (Table 2)

Table 2: Statistics experiment on total no. of earthquake from region to region.

Indian ocean 11	Sichuan mabian 34	The kuril island 38
Russia 29	Sichuan Gaoxian 25	India 16
Yunnan dongchuan 44	Sichuan heishui 20	Inner Mongolia liang city 10
Yunnan gejiu 42	Santa cruz island 14	Jiangsu donghai 26
Yunnan linxiang 37	The tower's island 15	Jiangxi dingnan 12
Yunnan yunxian 19	Tajikistan 89	Xinjiang qinghe 108
Yunnan yunlong 37	Ningxia zhongning 11	Taiwan hualian 12
Yunnan huize 73	Ningxia wuzhing 10	Sichuan ma'erkang 33
Yunnan yuanjiang 33	In northers Chille 11	Mianlan oldisland 25
Yunnan yuandie 33	Hokkaido area 24	The Hindu kush region 61
In southern Iran 12	The kuril island 38	South Atlantic ridge 18
The south china sea 27	HaiNan dongfang 28	……

b) Regional basis statistics graph from the data of (Table 3)

Table 3: Statistics experiment on total no. of earthquakes on daily basis.

2015-01-27 100	2015-1-01 78	……
2015-01-28 138	2015-04-02 92	2015-07-27 127
2015-01-29 111	2015-04-03 95	2015-07-28 118
2015-01-30 112	2015-04-04 105	2015-07-29 123
2015-01-31 134	2015-04-05 119	2015-07-30 108
2015-02-01 133	……	2015-08-01 118
2015-02-02 115	2015-05-27 126	2015-08-02 125
2015-02-03 127	2015-05-27 81	2015-08-03 91
2015-02-04 93	2015-05-29 105	2015-08-04 158
2015-02-05 128	2015-05-30 98	2015-08-04 151
……	2015-05-30 113	……
2015-03-27 96	2015-06-01 102	2015-09-08 108
2015-03-28 82	2015-06-02 103	2015-09-08 98
2015-03-29 97	2015-06-03 85	2015-09-10 80
2015-03-30 128	2015-06-4 95	……
2015-03-31 97	2015-06-05 104	

c) Geographical representation of statistics.

Conclusion

Hadoop is broadly notable system for information investigation for vast datasets that gives execution because of its capability of datasets examination in parallel and distributed environment [13]. Hadoop Distributed File System (HDFS) and the MapReduce are the modules of Hadoop. HDFS is responsible of information stockpiles while MapReduce is responsible of information handling. Tremendous informational index, such as web logs can be handled for investigation by Hadoop [14]. Here the paper utilizes the Hadoop Pseudo disseminated framework stage to break down and deal with the seismic data released by the National Earthquake Monitoring Station. The examination and testing are taken in the Hadoop. In other words, the procedure of Hadoop is taken by isolated Java. Local host node is as the NameNode and DataNode [15]. With the assistance of Hadoop MapReduce, it is conceivable to process the real time huge digital data and analyze effortlessly. It can get the number of the earthquake in all districts from the outcome since 2015, which helps us to think about where the earthquake inclined zones in that period are and furthermore the season of seismic tremor from 2015, which encourages us to know the season of earthquake in a year [16]. It additionally demonstrates the season of earthquake and the level of seismic tremor, in Figure 4.

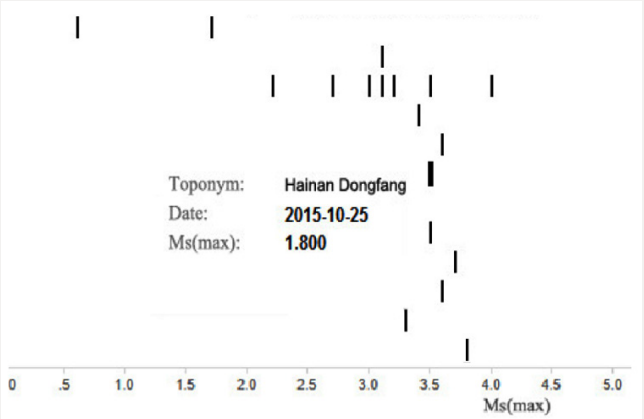

Figure 4: Number of earthquakes everywhere.

Figure 5: The information of earthquake everywhere.

It can likewise specifically demonstrate the territories of nation from 2015 [17]. The more profound shading implies the more circumstances of seismic tremor there. Else, it can demonstrate the data the biggest level of earthquake and the most profound quake as in the Figure 5. Results are to such an extent that it can be seen easily to fundamental man by its direct section wise portrayal depiction of yield. One can undoubtedly send out Hadoop yield records to few apparatuses like R, Tableau and so on to produce reasonable graphs and report [18]. The investigation made by Hadoop stage is

extremely encouraging with higher productivity and down to earth esteem and are anything but difficult to extend. Otherwise, the theory and practical application of Hadoop, yet in addition mirrors the high unwavering quality and productivity of the Hadoop stage to manage information [19]. In outline, the utilization of Hadoop stage to analyze and process huge informational indexes has higher effectiveness and reasonable esteem, and simple to grow [20].

References

1. Yao X, Zhu D, Ye S, Yun W, Zhang N A field survey system for land consolidation based on 3S and speech recognition technology. Comput Electron Agric 127: 659-668.
2. Yao X, Yang J, Li L, Yun W, Zhao Z, et al. (2017) LandQv1: A GIS cluster-based management information system for arable land quality big data. In Proceedings of the 6th International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Fairfax, VA, USA pp. 1-6.
3. Huang QY, Yang CW, Liu K, Xia JZ, Xu C, et al. (2013) Evaluating open-source cloud computing solutions for geosciences. Comput Geosci 59: 41-52.
4. Li Z, Yang C, Liu K, Hu F, Jin B (2016) Automatic scaling Hadoop in the cloud for efficient process of big geospatial data. ISPRS Int Geo-Inf 5(10): 173.
5. Eldawy A, Mokbel MF (2013) A demonstration of spatialhadoop: An efficient mapreduce framework for spatial data. Proc. VLDB Endow 6: 1230-1233.
6. Alarabi L (2017) St-Hadoop: A mapreduce framework for big spatio-temporal data. In Proceedings of the ACM International Conference on Management of Data 14-19.
7. Mueller N, Lewis A, Roberts D, Ring S, Melrose R, et al. (2016) Water observations from space: Mapping surface water from 25 years of landsat imagery across Australia. Remote Sens. Environ 174:341-352.
8. Li J, Meng L, Wang FZ, Zhang W, Cai Y (2014) A map-reduce-enabled solap cube for large-scale remotely sensed data aggregation. Comput Geosci 70: 110-119.
9. Magdy A, Mokbel MF, Elnikety S, Nath S, He Y (2016) Venus: Scalable real-time spatial queries on microblogs with adaptive load shedding. IEEE Trans. Knowl. Data Eng 28(2): 356-370.
10. Zou ZQ, Wang Y, Cao K, Qu TS, Wang ZM (2013) Semantic overlay network for large-scale spatial information indexing. Comput. Geosci 57: 208-217.
11. Yao X, Li G (2018) Big spatial vector data management: A review. Big Earth Data 2: 108-129.
12. Zhao L, Chen L, Ranjan R, Choo KKR, He J (2015) Geographical information system parallelization for spatial big data processing: A review. Clust Comput 19(1): 139-152.
13. Singh H, Bawa S (2017) A mapreduce-based scalable discovery and indexing of structured big data. Future Gener. Comput. Syst 73: 32-43.
14. Yao X, Mokbel MF, Alarabi L, Eldawy A, Yang J, et al. (2017) Spatial coding-based approach for partitioning big spatial data in Hadoop. Comput Geosci 106: 60-67.
15. Hadjieleftheriou M, Manolopoulos Y, Theodoridis Y, Tsotras VJ (2008) R-trees-A dynamic index structure for spatial searching. In Encyclopedia of GIS. Shekhar S, Xiong H (Eds.) Springer: Boston, MA, USA pp. 993-1002.
16. Eldawy A, Alarabi L, Mokbel MF (2015) Spatial partitioning techniques in spatialhadoop. Proc. VLDB Endow 8: 1602-1605.
17. Zhang J, You S (2013) High-performance quadtree constructions on large-scale geospatial rasters using GPGPU parallel primitives. Int J Geogr Inf Sci 27: 2207-2226.
18. Eldawy A, Mokbel MF, Jonathan C (2016) Hadoop viz: A mapreduce framework for extensible visualization of big spatial data. In Proceedings of the 32nd IEEE International Conference on Data Engineering, Helsinki, Finland pp. 601-612.
19. Liu Y, Chen L, Jing N, Xiong W (2013) Parallel batch-building remote sensing images tile pyramid with mapreduce. Wuhan Daxue Xuebao (Xinxi Kexue Ban)/Geomat. Inf Sci Wuhan Univ 38: 278-282.
20. Lin W, Zhou H, Xia P (2016) An effective NOSQL-based vector map tile management approach. ISPRS Int. Geo-Inf 5(11): 215.

DOI: 10.32474/ARME.2018.01.000103