Big Data Milestone 1

Student Name: Mark Magdy Nasr Said

ID: 18011304

Student Name: Mohamed Yasser Mohamed

ID: 18011648

Student Name: Mina Henen Shafik

ID: 18011939

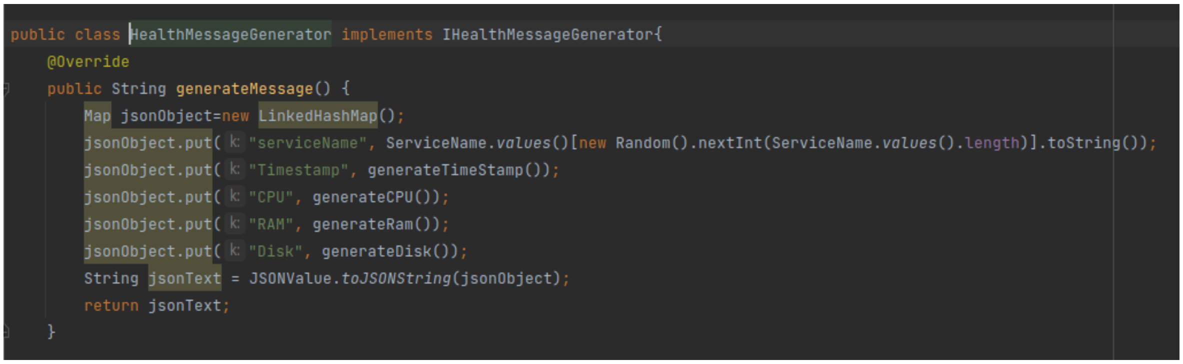
Student Name: Mark Nader Fathy

ID: 18011305

Components explaining:

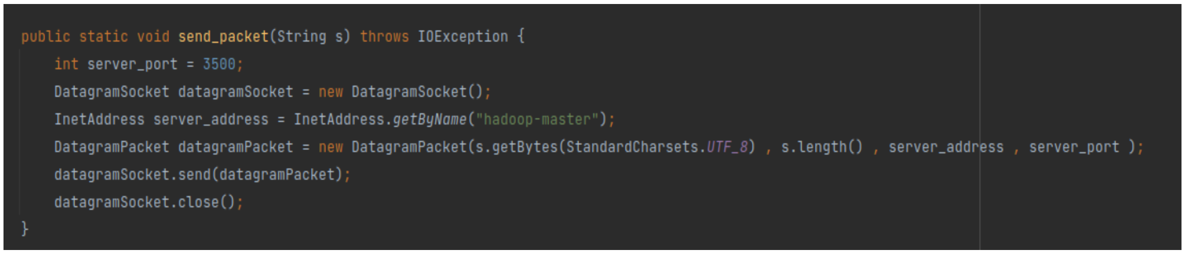
➢ Client:

Message generator



```
public class HealthMessageGenerator implements IHealthMessageGenerator{
    @Override
    public String generateMessage() {
        Map jsonObject=new LinkedHashMap();
        jsonObject.put( k: "serviceName", ServiceName.values()[new Random().nextInt(ServiceName.values().length)].toString());
        jsonObject.put( k: "Timestamp", generateTimeStamp());
        jsonObject.put( k: "CPU", generateCPU());
        jsonObject.put( k: "RAM", generateRam());
        jsonObject.put( k: "Disk", generateDisk());
        String jsonText = JSONValue.toJSONString(jsonObject);
        return jsonText;
    }
```

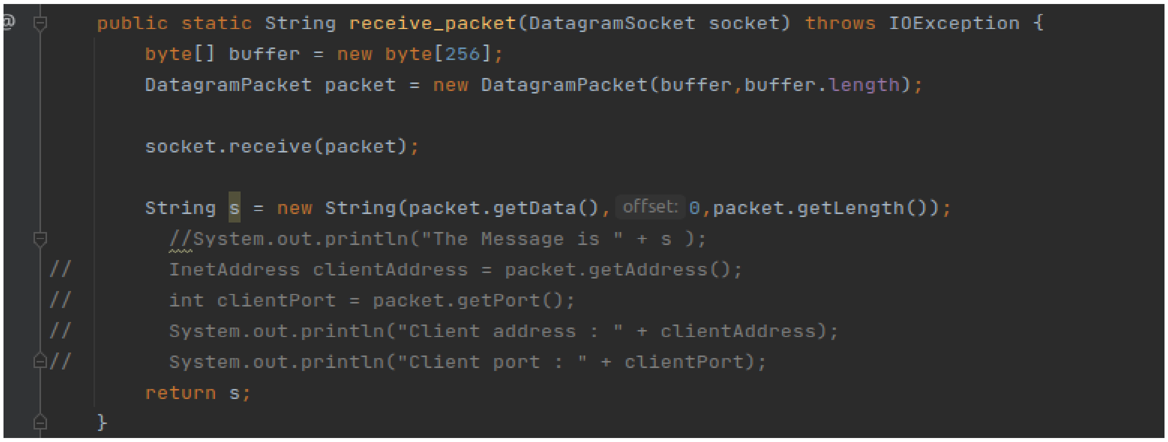
Send packet



```
public static void send_packet(String s) throws IOException {
    int server_port = 3500;
    DatagramSocket datagramSocket = new DatagramSocket();
    InetAddress server_address = InetAddress.getByName("hadoop-master");
    DatagramPacket datagramPacket = new DatagramPacket(s.getBytes(StandardCharsets.UTF_8) , s.length() , server_address , server_port );
    datagramSocket.send(datagramPacket);
    datagramSocket.close();
}
```

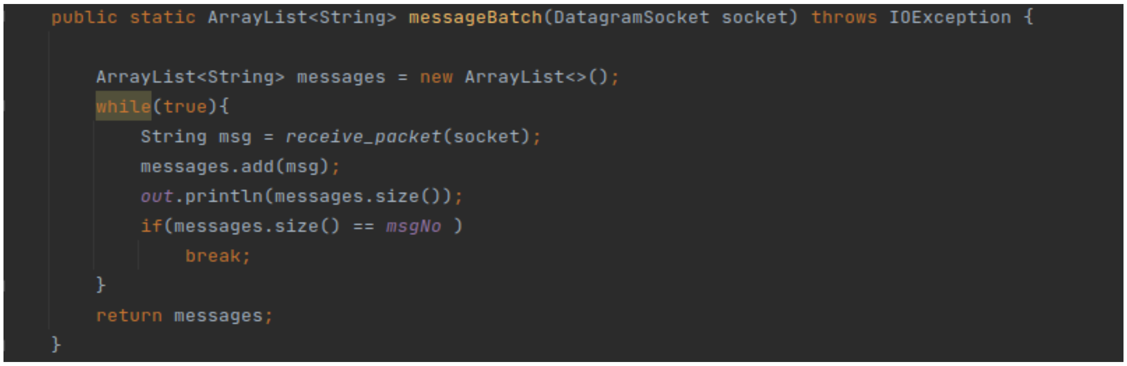
➢ Server:

Receive packets



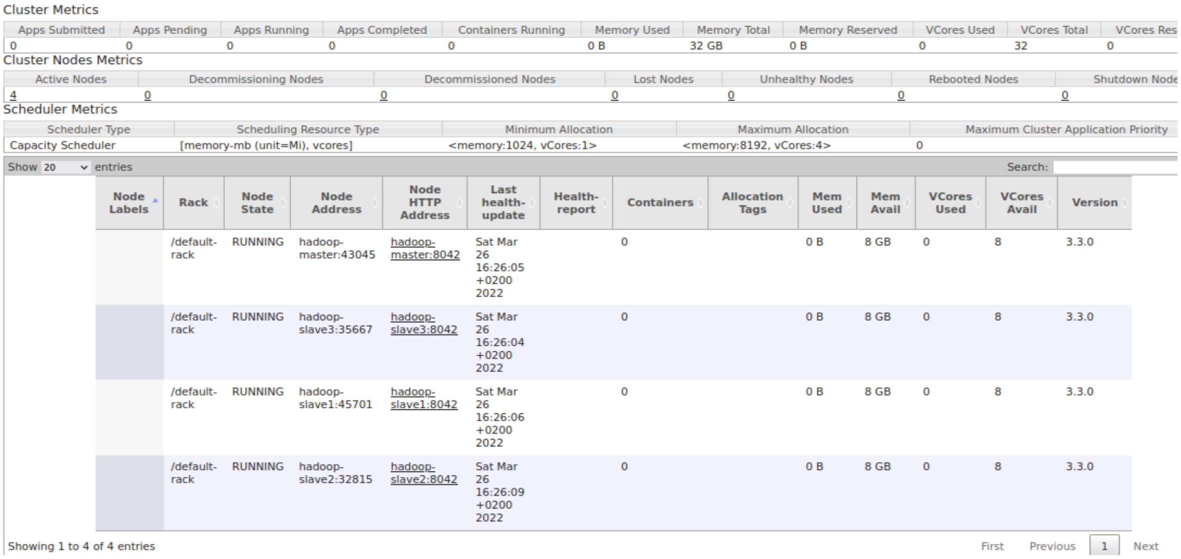
```
    public static String receive_packet(DatagramSocket socket) throws IOException {
        byte[] buffer = new byte[256];
        DatagramPacket packet = new DatagramPacket(buffer,buffer.length);

        socket.receive(packet);

        String s = new String(packet.getData(), offset: 0,packet.getLength());
          //System.out.println("The Message is " + s );
//        InetAddress clientAddress = packet.getAddress();
//        int clientPort = packet.getPort();
//        System.out.println("Client address : " + clientAddress);
//        System.out.println("Client port : " + clientPort);
        return s;
    }
```

Message batch



```java
public static ArrayList<String> messageBatch(DatagramSocket socket) throws IOException {

    ArrayList<String> messages = new ArrayList<>();
    while(true){
        String msg = receive_packet(socket);
        messages.add(msg);
        out.println(messages.size());
        if(messages.size() == msgNo )
            break;
    }
    return messages;
}
```

➢ HDFS:



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Res
0	0	0	0	0	0 B	32 GB	0 B	0	32	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Node
4	0	0	0	0	0	0

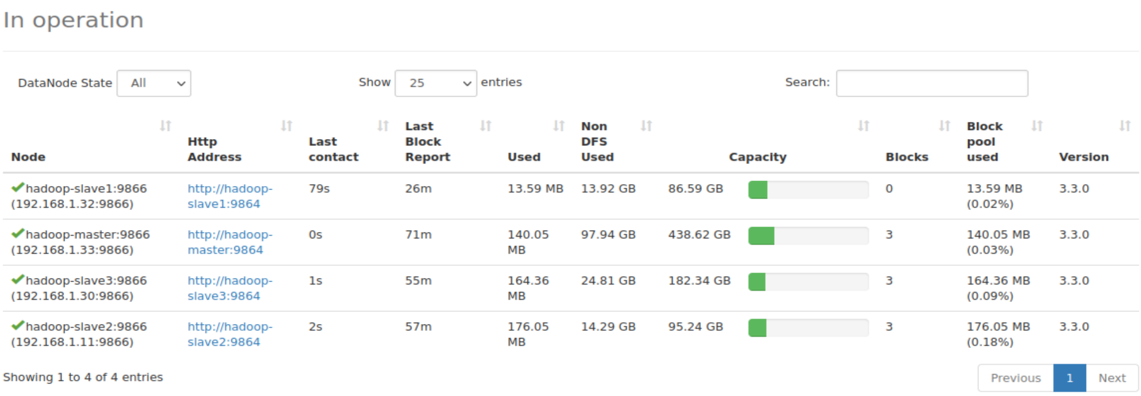
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	hadoop-master:43045	hadoop-master:8042	Sat Mar 26 16:26:05 +0200 2022		0		0 B	8 GB	0	8	3.3.0
	/default-rack	RUNNING	hadoop-slave3:35667	hadoop-slave3:8042	Sat Mar 26 16:26:04 +0200 2022		0		0 B	8 GB	0	8	3.3.0
	/default-rack	RUNNING	hadoop-slave1:45701	hadoop-slave1:8042	Sat Mar 26 16:26:06 +0200 2022		0		0 B	8 GB	0	8	3.3.0
	/default-rack	RUNNING	hadoop-slave2:32815	hadoop-slave2:8042	Sat Mar 26 16:26:09 +0200 2022		0		0 B	8 GB	0	8	3.3.0

Showing 1 to 4 of 4 entries



In operation

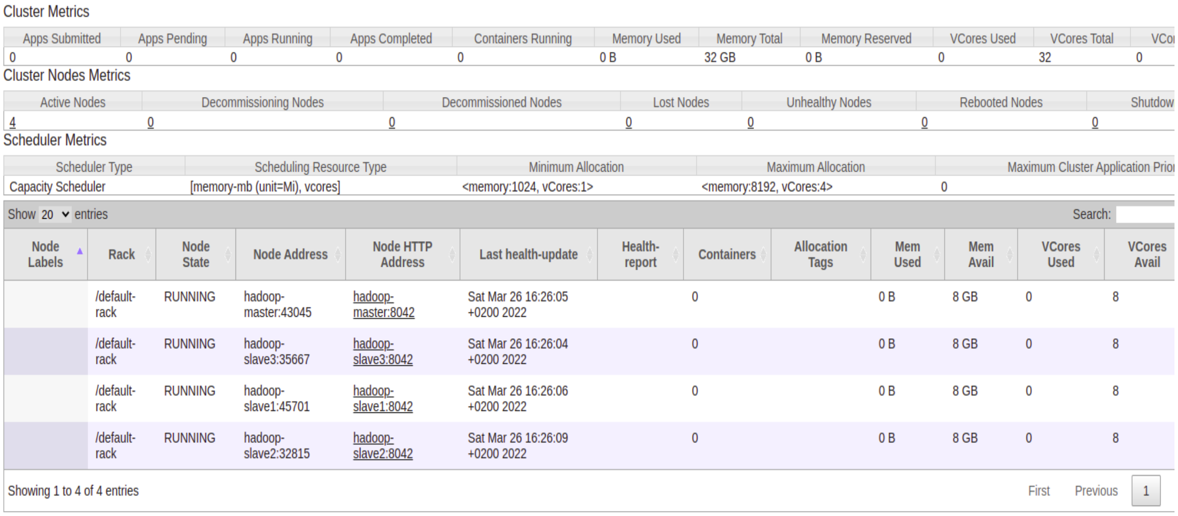
DataNode State All Show 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓hadoop-slave1:9866 (192.168.1.32:9866)	http://hadoop-slave1:9864	79s	26m	13.59 MB	13.92 GB	86.59 GB	0	13.59 MB (0.02%)	3.3.0
✓hadoop-master:9866 (192.168.1.33:9866)	http://hadoop-master:9864	0s	71m	140.05 MB	97.94 GB	438.62 GB	3	140.05 MB (0.03%)	3.3.0
✓hadoop-slave3:9866 (192.168.1.30:9866)	http://hadoop-slave3:9864	1s	55m	164.36 MB	24.81 GB	182.34 GB	3	164.36 MB (0.09%)	3.3.0
✓hadoop-slave2:9866 (192.168.1.11:9866)	http://hadoop-slave2:9864	2s	57m	176.05 MB	14.29 GB	95.24 GB	3	176.05 MB (0.18%)	3.3.0

Showing 1 to 4 of 4 entries

Configurations Steps

➢ Active Nodes



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCo
0	0	0	0	0	0 B	32 GB	0 B	0	32	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdow
4	0	0	0	0	0	0

Scheduler Metrics

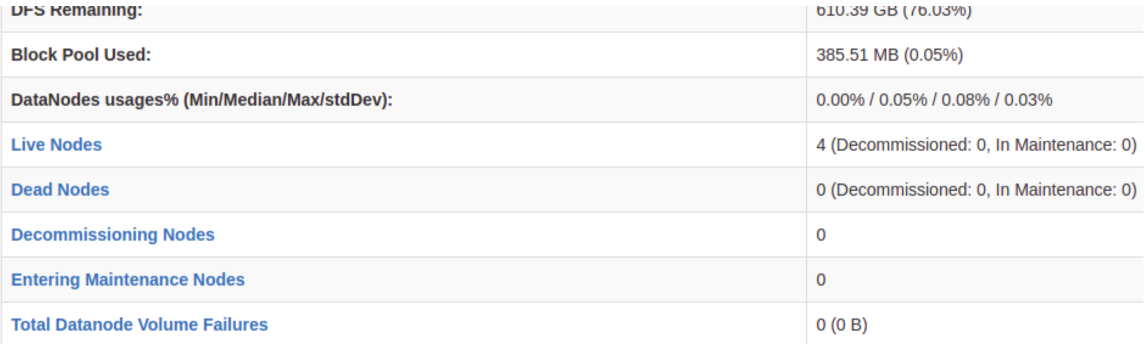
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Prio
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

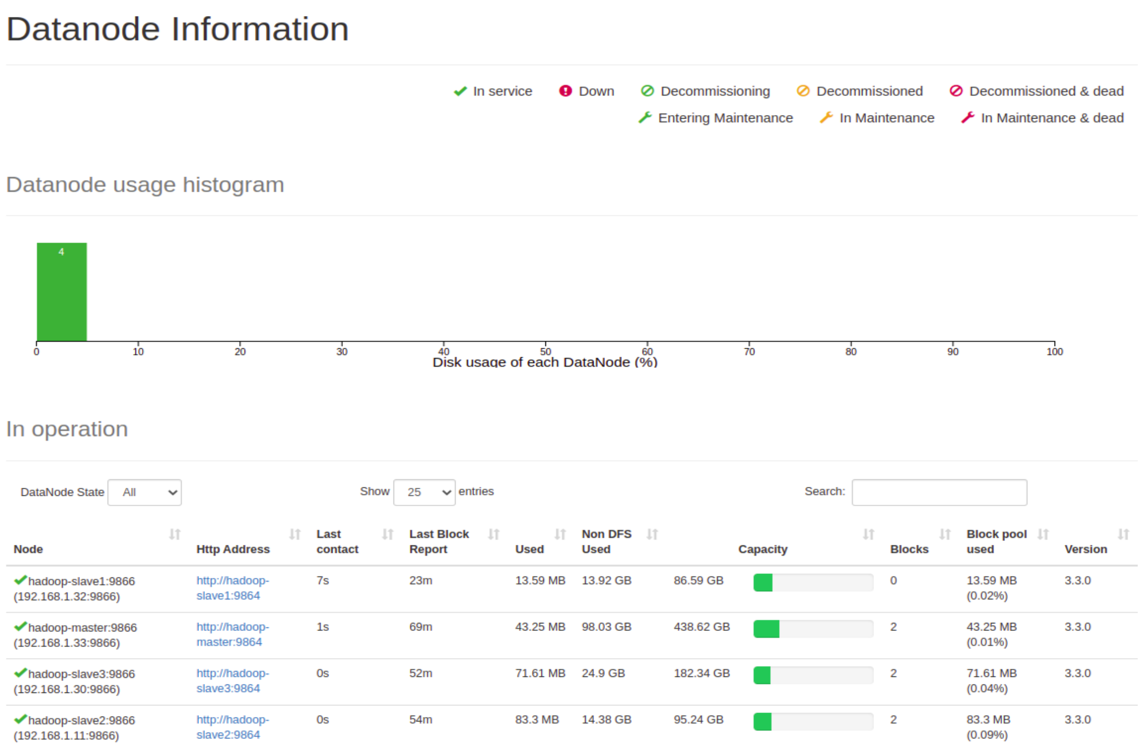
Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	VCores Used	VCores Avail
	/default-rack	RUNNING	hadoop-master:43045	hadoop-master:8042	Sat Mar 26 16:26:05 +0200 2022		0		0 B	8 GB	0	8
	/default-rack	RUNNING	hadoop-slave3:35667	hadoop-slave3:8042	Sat Mar 26 16:26:04 +0200 2022		0		0 B	8 GB	0	8
	/default-rack	RUNNING	hadoop-slave1:45701	hadoop-slave1:8042	Sat Mar 26 16:26:06 +0200 2022		0		0 B	8 GB	0	8
	/default-rack	RUNNING	hadoop-slave2:32815	hadoop-slave2:8042	Sat Mar 26 16:26:09 +0200 2022		0		0 B	8 GB	0	8

Showing 1 to 4 of 4 entries

➢ Live Nodes



DFS Remaining:	610.39 GB (76.03%)
Block Pool Used:	385.51 MB (0.05%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.05% / 0.08% / 0.03%
Live Nodes	4 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)



Datanode Information

✓ In service ● Down ⊘ Decommissioning ⊘ Decommissioned ⊘ Decommissioned & dead
Entering Maintenance In Maintenance In Maintenance & dead

Datanode usage histogram

In operation

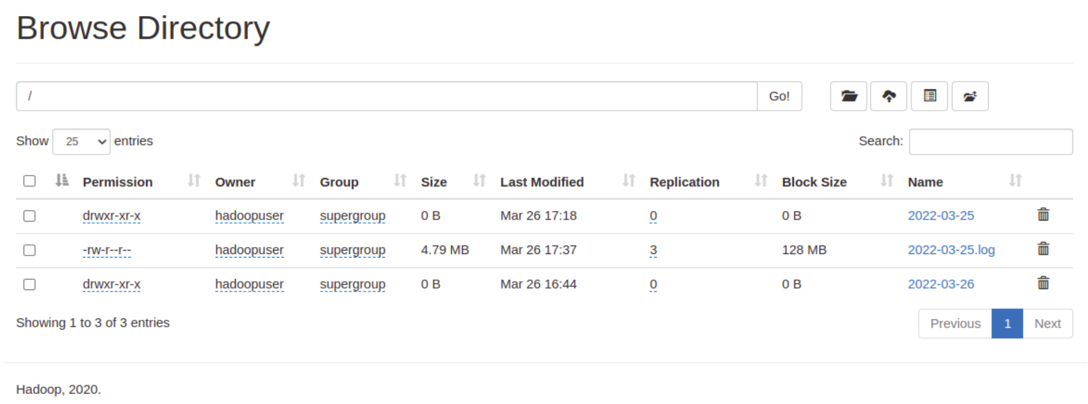
DataNode State All

Show 25 entries

Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓hadoop-slave1:9866 (192.168.1.32:9866)	http://hadoop-slave1:9864	7s	23m	13.59 MB	13.92 GB	86.59 GB	0	13.59 MB (0.02%)	3.3.0
✓hadoop-master:9866 (192.168.1.33:9866)	http://hadoop-master:9864	1s	69m	43.25 MB	98.03 GB	438.62 GB	2	43.25 MB (0.01%)	3.3.0
✓hadoop-slave3:9866 (192.168.1.30:9866)	http://hadoop-slave3:9864	0s	52m	71.61 MB	24.9 GB	182.34 GB	2	71.61 MB (0.04%)	3.3.0
✓hadoop-slave2:9866 (192.168.1.11:9866)	http://hadoop-slave2:9864	0s	54m	83.3 MB	14.38 GB	95.24 GB	2	83.3 MB (0.09%)	3.3.0

➢ Adding Files to HDFS



Browse Directory

/ Go!

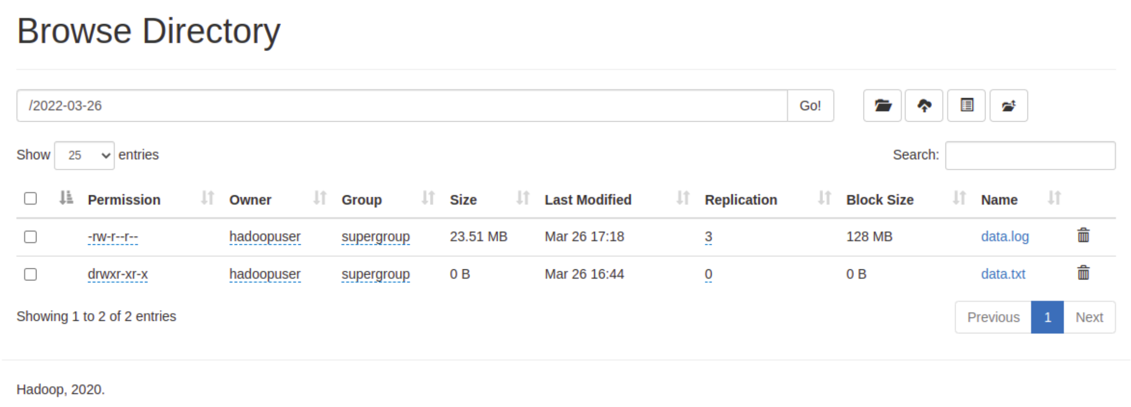
Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	hadoopuser	supergroup	0 B	Mar 26 17:18	0	0 B	2022-03-25
□	-rw-r--r--	hadoopuser	supergroup	4.79 MB	Mar 26 17:37	3	128 MB	2022-03-25.log
□	drwxr-xr-x	hadoopuser	supergroup	0 B	Mar 26 16:44	0	0 B	2022-03-26

Showing 1 to 3 of 3 entries

Hadoop, 2020.



Browse Directory

/2022-03-26 Go!

Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--	hadoopuser	supergroup	23.51 MB	Mar 26 17:18	3	128 MB	data.log	🗑
□	drwxr-xr-x	hadoopuser	supergroup	0 B	Mar 26 16:44	0	0 B	data.txt	🗑

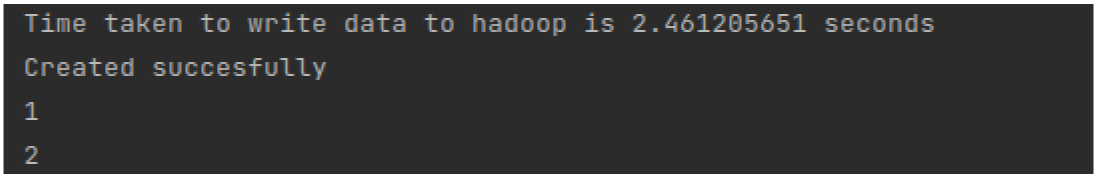
Showing 1 to 2 of 2 entries

Hadoop, 2020.

Performance Analysis

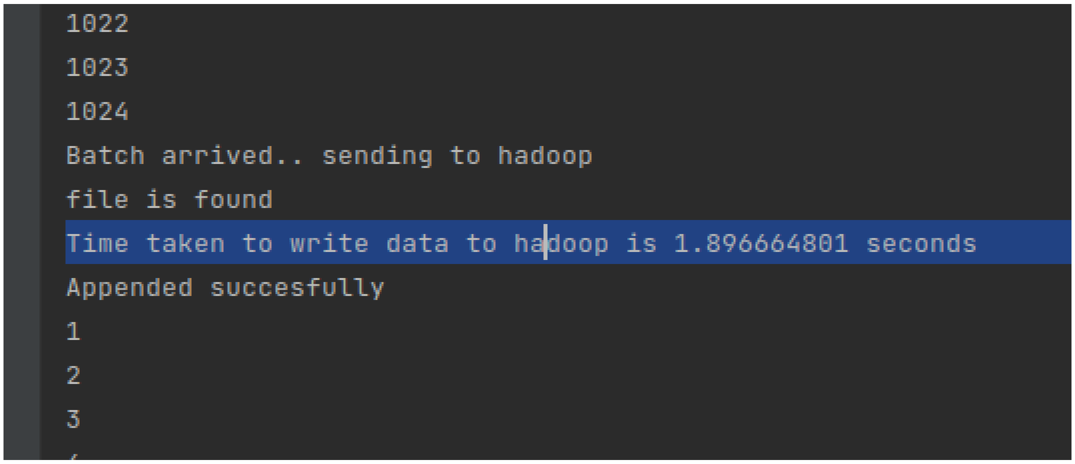
➢ What is the time taken to write data in the HDFS?

➢ For Creating New File



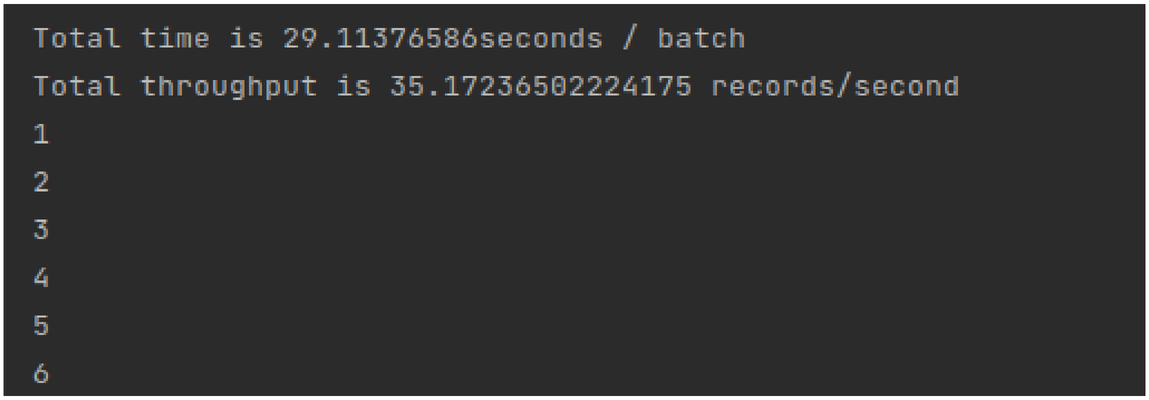
```
Time taken to write data to hadoop is 2.461205651 seconds
Created succesfully
1
2
```

➢ For Appending File



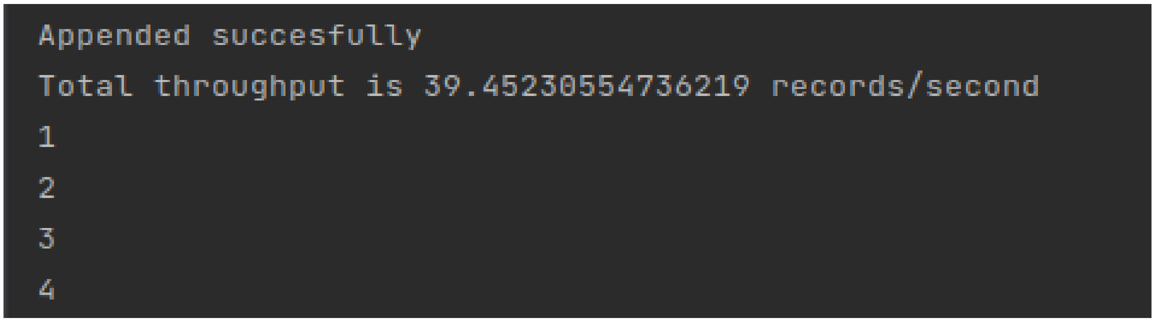
```
1022
1023
1024
Batch arrived.. sending to hadoop
file is found
Time taken to write data to hadoop is 1.896664801 seconds
Appended succesfully
1
2
3
4
```

➢ What is the average (and std) end-to-end time taken from the moment data is received in Health Monitor until it is written in the HDFS?



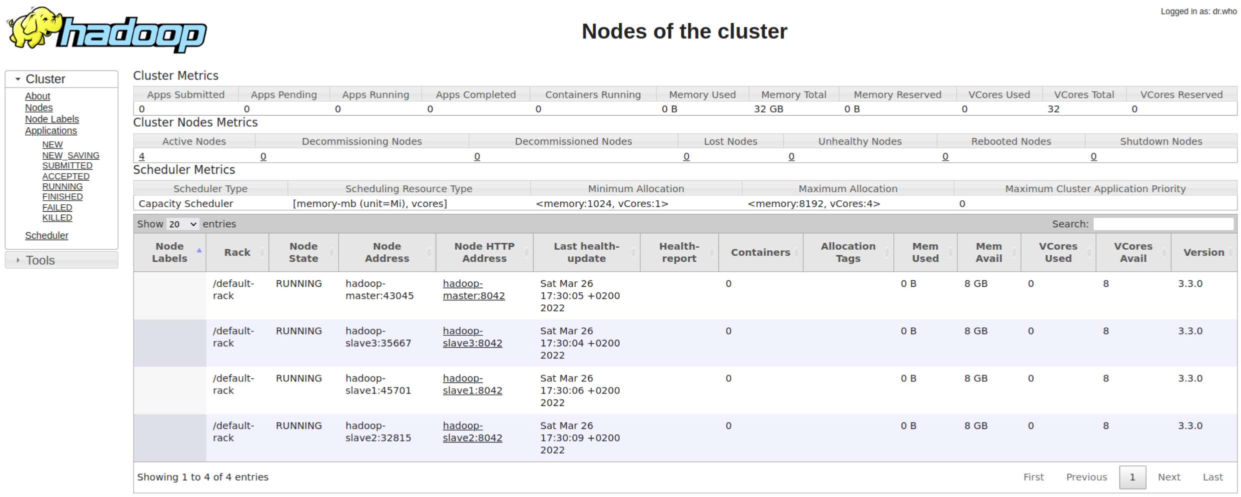
```
Total time is 29.11376586seconds / batch
Total throughput is 35.17236502224175 records/second
1
2
3
4
5
6
```

➢ What is the overall system throughput (Health Monitor throughput)? in records/second



```
Appended succesfully
Total throughput is 39.45230554736219 records/second
1
2
3
4
```

Sample runs



Logged in as: dr.who

Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	32 GB	0 B	0	32	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
4	0	0	0	0	0	0

Scheduler Metrics

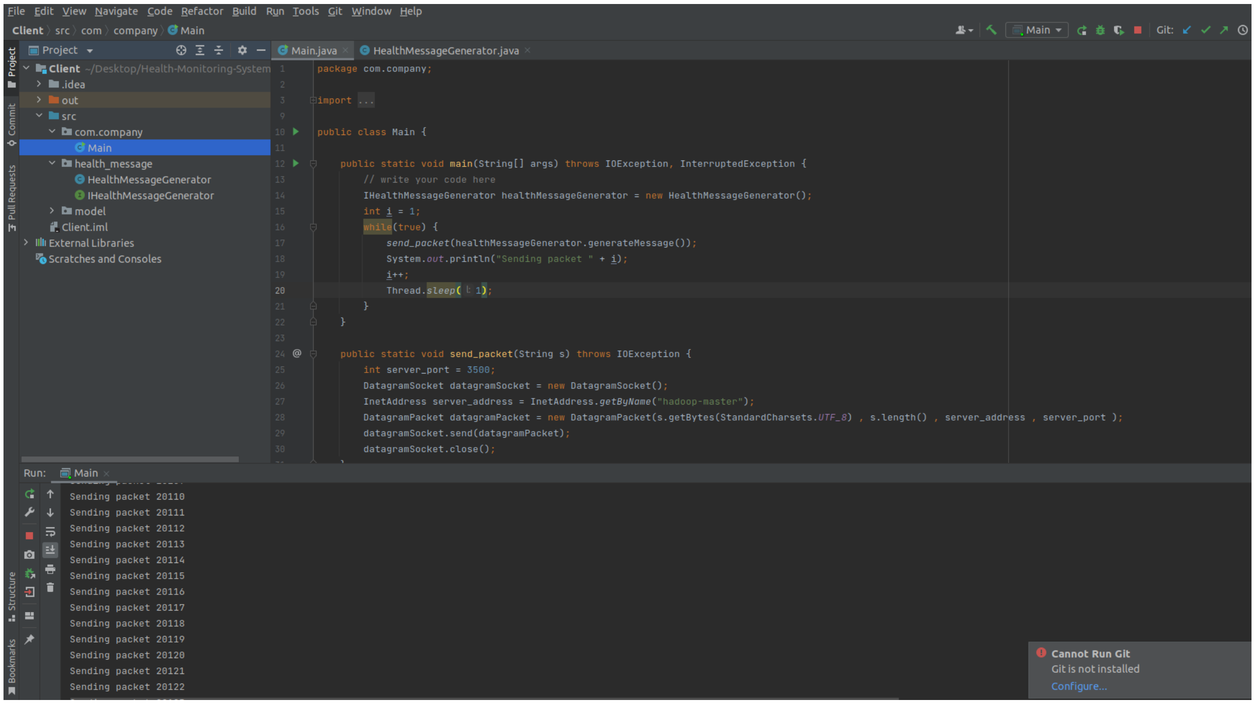
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries | Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	hadoop-master:43045	hadoop-master:8042	Sat Mar 26 17:30:05 +0200 2022		0		0 B	8 GB	0	8	3.3.0
	/default-rack	RUNNING	hadoop-slave3:35667	hadoop-slave3:8042	Sat Mar 26 17:30:04 +0200 2022		0		0 B	8 GB	0	8	3.3.0
	/default-rack	RUNNING	hadoop-slave1:45701	hadoop-slave1:8042	Sat Mar 26 17:30:06 +0200 2022		0		0 B	8 GB	0	8	3.3.0
	/default-rack	RUNNING	hadoop-slave2:32815	hadoop-slave2:8042	Sat Mar 26 17:30:09 +0200 2022		0		0 B	8 GB	0	8	3.3.0

Showing 1 to 4 of 4 entries

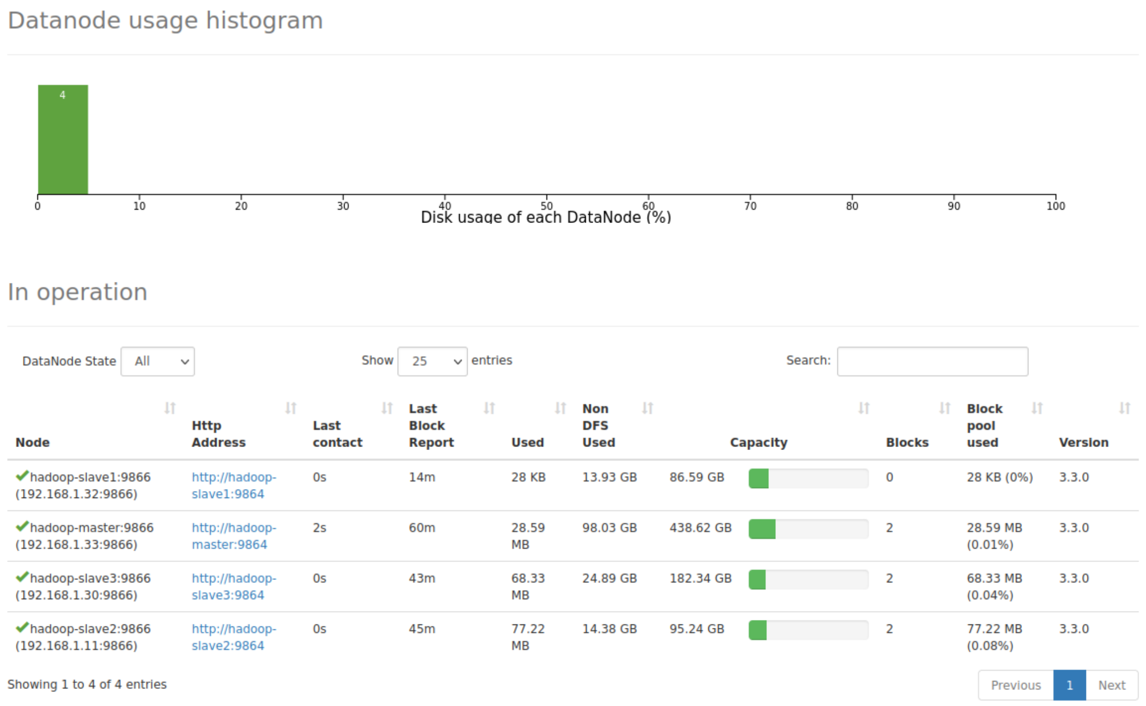
Slave 1 client sending…



```java
package com.company;

import ...

public class Main {

    public static void main(String[] args) throws IOException, InterruptedException {
        // write your code here
        IHealthMessageGenerator healthMessageGenerator = new HealthMessageGenerator();
        int i = 1;
        while(true) {
            send_packet(healthMessageGenerator.generateMessage());
            System.out.println("Sending packet " + i);
            i++;
            Thread.sleep( l: 1);
        }
    }

    public static void send_packet(String s) throws IOException {
        int server_port = 3500;
        DatagramSocket datagramSocket = new DatagramSocket();
        InetAddress server_address = InetAddress.getByName("hadoop-master");
        DatagramPacket datagramPacket = new DatagramPacket(s.getBytes(StandardCharsets.UTF_8) , s.length() , server_address , server_port );
        datagramSocket.send(datagramPacket);
        datagramSocket.close();
```

```
Sending packet 20110
Sending packet 20111
Sending packet 20112
Sending packet 20113
Sending packet 20114
Sending packet 20115
Sending packet 20116
Sending packet 20117
Sending packet 20118
Sending packet 20119
Sending packet 20120
Sending packet 20121
Sending packet 20122
```

Cannot Run Git
Git is not installed
Configure...



Datanode usage histogram

In operation

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓hadoop-slave1:9866 (192.168.1.32:9866)	http://hadoop-slave1:9864	0s	14m	28 KB	13.93 GB	86.59 GB	0	28 KB (0%)	3.3.0
✓hadoop-master:9866 (192.168.1.33:9866)	http://hadoop-master:9864	2s	60m	28.59 MB	98.03 GB	438.62 GB	2	28.59 MB (0.01%)	3.3.0
✓hadoop-slave3:9866 (192.168.1.30:9866)	http://hadoop-slave3:9864	0s	43m	68.33 MB	24.89 GB	182.34 GB	2	68.33 MB (0.04%)	3.3.0
✓hadoop-slave2:9866 (192.168.1.11:9866)	http://hadoop-slave2:9864	0s	45m	77.22 MB	14.38 GB	95.24 GB	2	77.22 MB (0.08%)	3.3.0

Showing 1 to 4 of 4 entries