

# Report

## 1. Problem Description

Performance of sorting a large data set measured using Shared Memory Sorting, Hadoop and Spark. The performance of shared memory sorting, Hadoop sort and spark sort are measured on data of 10 GB with 1 node. The performance of Hadoop sort and spark sort is measured on dataset 100 GB on 17 nodes (1 master 16 slave).

## 2. Shared Memory Sorting

### a. Description:

Shared memory sorting is an implementation of the external merge sort written in java which uses a combination of in memory sort and merging the in memory sorted data.

### b. Environment settings:

The environment in this case was a Java Virtual Machine which was used to run the java code on the instance. The Heap Size had to be increased in order to match the high need of storing the in memory data.

OS distribution: Linux (preconfigured Hadoop AMI)

Java version: java-1.6.0-openjdk-devel (for jps and other developer tools)

### Problem Encountered

#### A) Java out of Memory

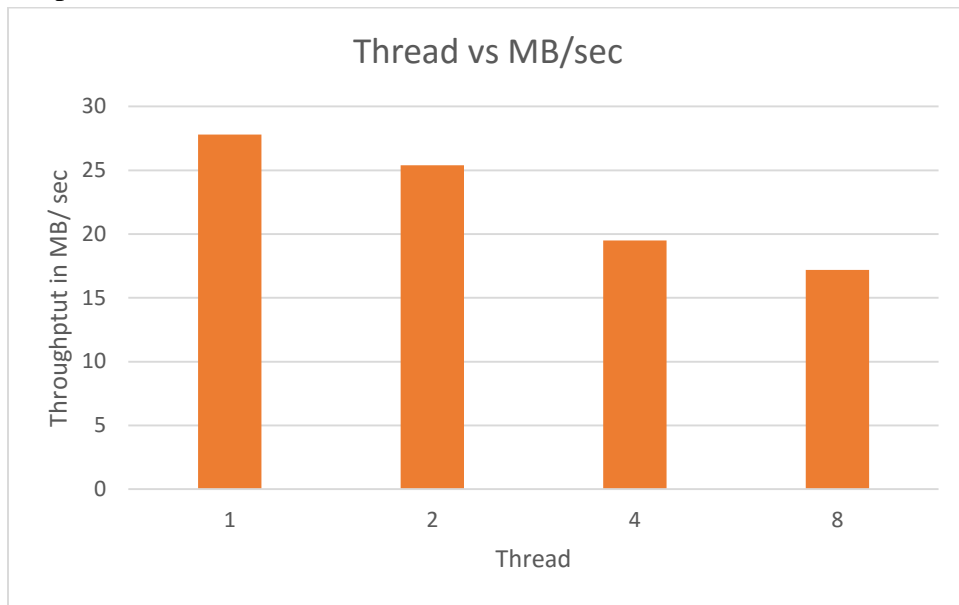
Solution: Increase Heap memory was increased –Xmx 3G

The program was run on a 10GB dataset with 1, 2, 4 and 8 threads.

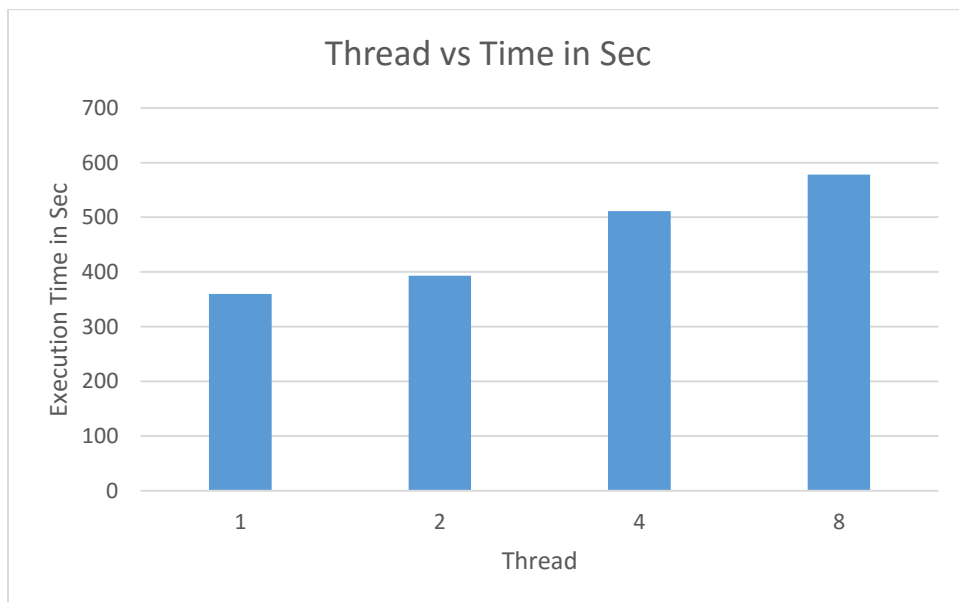
Results were:

Threads	Time(sec)	MB/sec
1	359.6	27.8
2	393.15	25.4
4	511.6	19.5
8	578.31	17.2

Graphs:



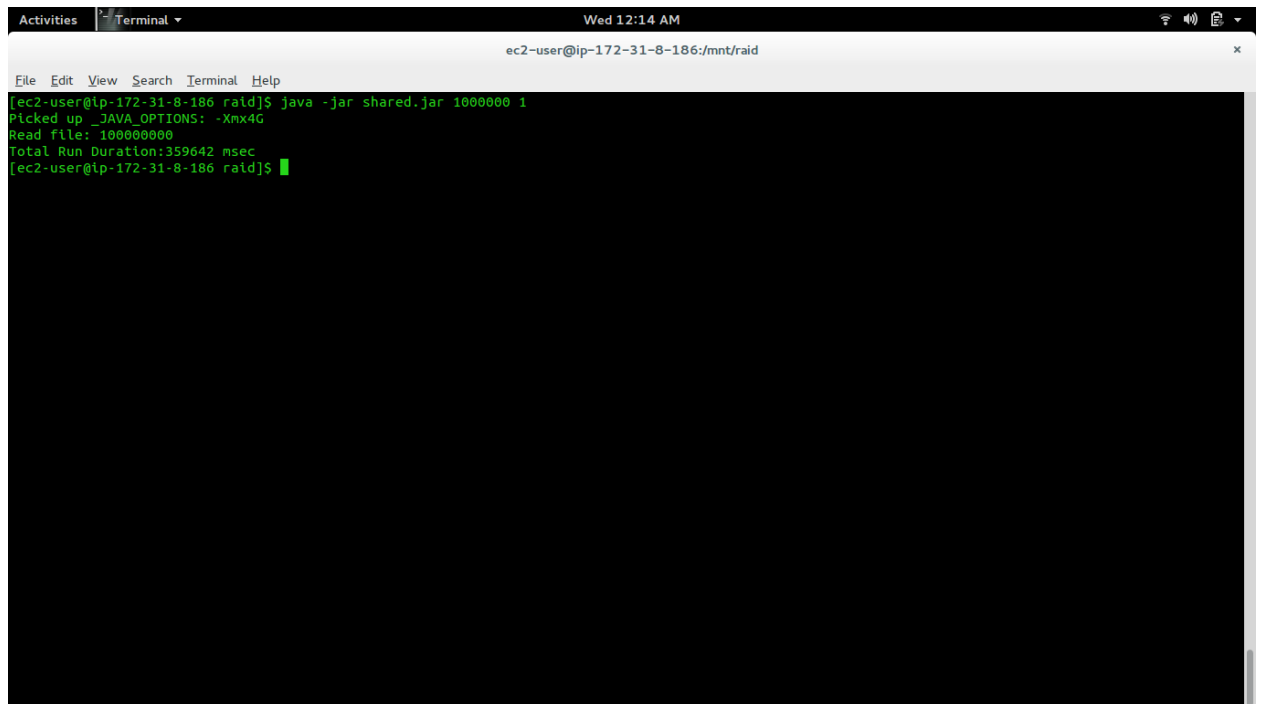
Graph: Threads vs MB/sec



Graph Execution Time vs Threads

Screenshots:

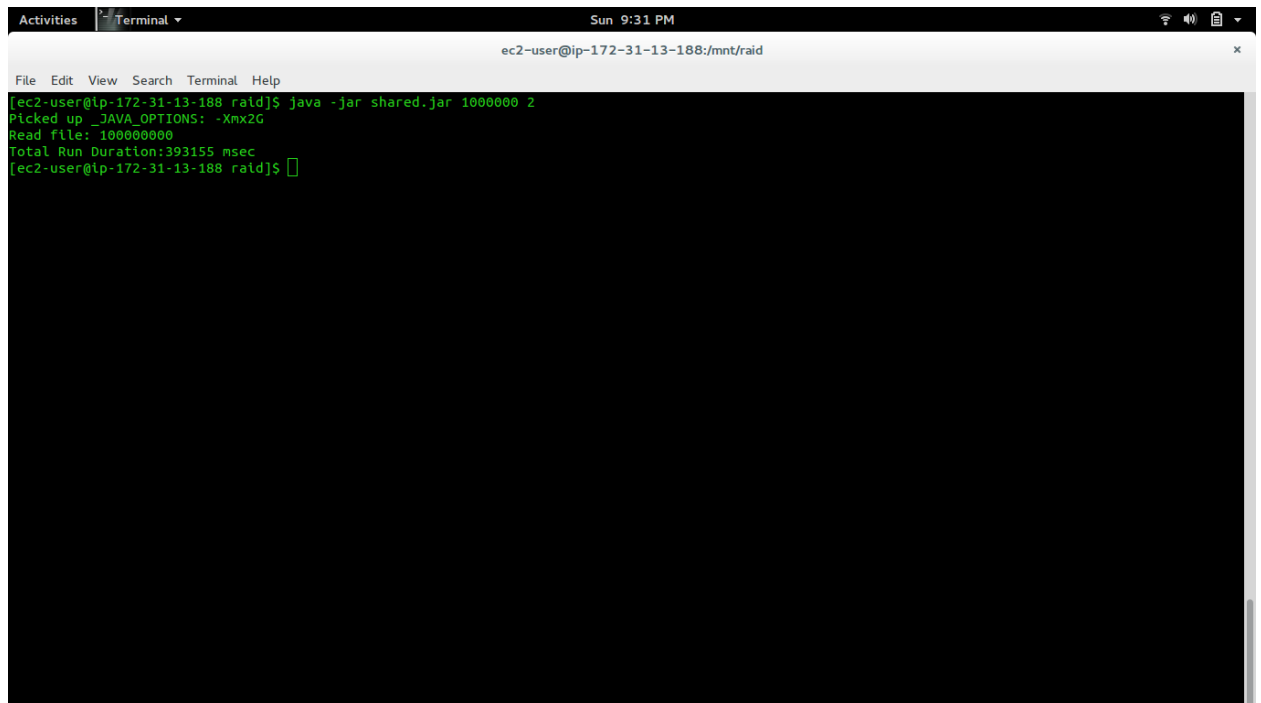
## 1 Thread



A terminal window titled "Terminal" with a subtitle "Wed 12:14 AM". The terminal shows the command `java -jar shared.jar 1000000 1` being executed. The output is as follows:

```
[ec2-user@ip-172-31-8-186 raid]$ java -jar shared.jar 1000000 1
Picked up _JAVA_OPTIONS: -Xmx4G
Read file: 1000000000
Total Run Duration:359642 msec
[ec2-user@ip-172-31-8-186 raid]$
```

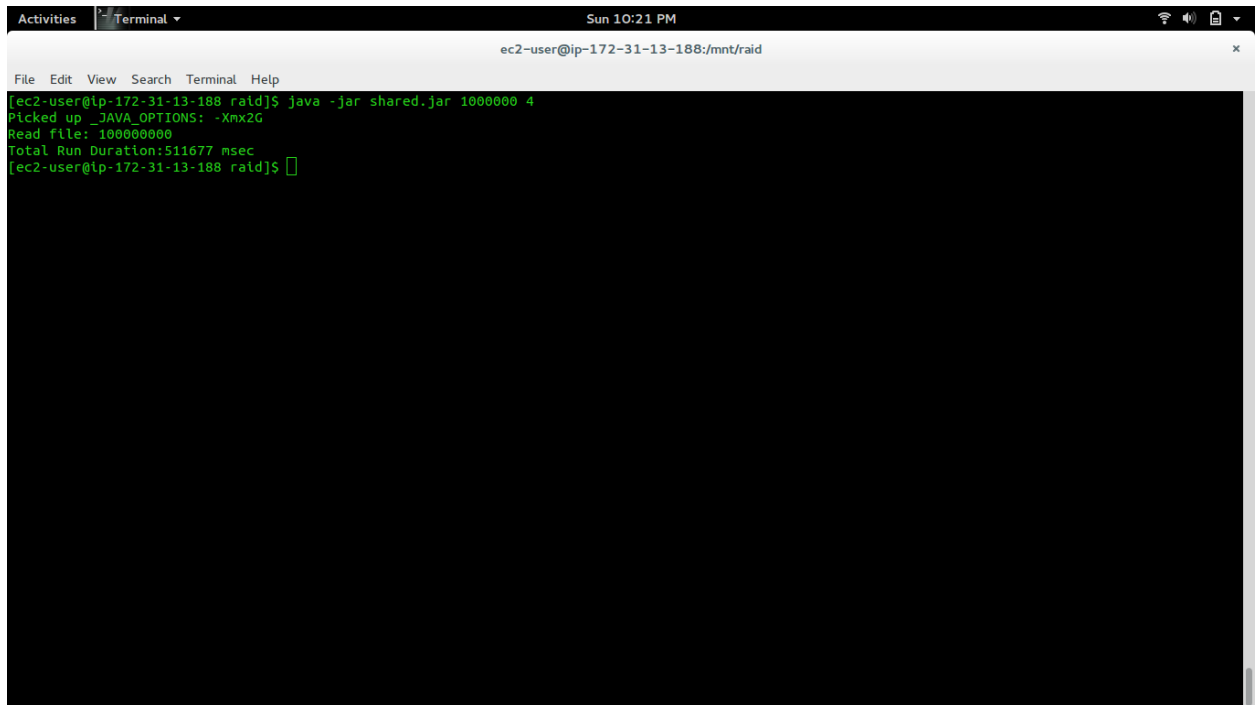
## 2. Threads



A terminal window titled "Terminal" with a subtitle "Sun 9:31 PM". The terminal shows the command `java -jar shared.jar 1000000 2` being executed. The output is as follows:

```
[ec2-user@ip-172-31-13-188 raid]$ java -jar shared.jar 1000000 2
Picked up _JAVA_OPTIONS: -Xmx2G
Read file: 1000000000
Total Run Duration:393155 msec
[ec2-user@ip-172-31-13-188 raid]$
```

## 4. Threads



```
Activities Terminal Sun 10:21 PM
ec2-user@ip-172-31-13-188:/mnt/raid
File Edit View Search Terminal Help
[ec2-user@ip-172-31-13-188 raid]$ java -jar shared.jar 1000000 4
Picked up _JAVA_OPTIONS: -Xmx2G
Read file: 100000000
Total Run Duration: 511677 msec
[ec2-user@ip-172-31-13-188 raid]$
```

Explaining Results:

The Best Output is found for 1 and 2 threads as maximum parallelism is achieved because the CPU has 2 vCPU.

### 3. Hadoop

a. Description:

Shared memory sorting is an implementation of the external merge sort written in java which uses a combination of in memory sort and merging the in memory sorted data.

b. Environment settings:

The Hadoop environment had to be configured and file had to be modified as described in the readme.txt. An AMI was built with Initial master preconfigured with IP addresses. Then the slaves were built from the AMI and raid storage was added to the instance. Pssh was used to configure the raid on the slaves.

Hadoop Version: 2.7.2

OS distribution: Linux

Java version: java-1.6.0-openjdk-devel (for jps and other developer tools)

#### Problem Encountered

B) Datanode not starting even though namenode and secondary namenode are running.

Solution: Delete Hadoop temp directory and format namenode and restart.

### C) Slave not detected

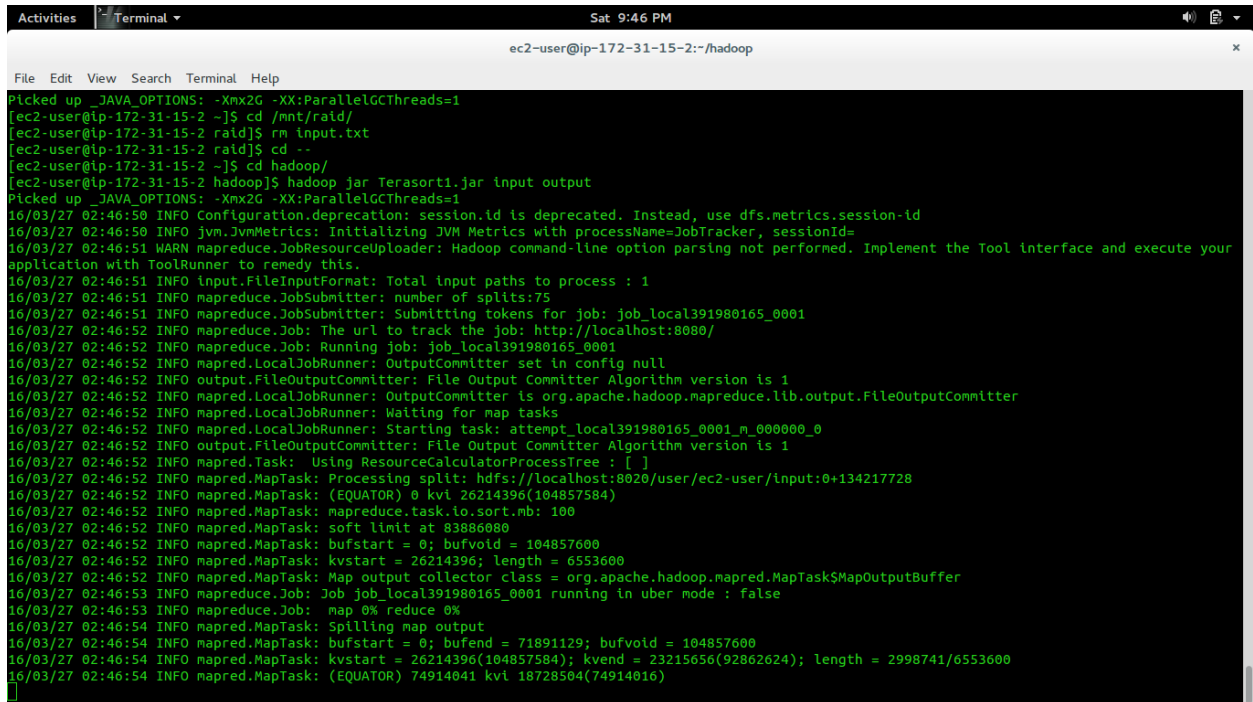
Solution: Change Slave file from localhost to IP address and add Master file at each slave.

### c. Results and Graphs

Nodes	Time	MB/sec
1	667.2	14.9
16	10220.2	9.78

## Screenshots

### I. 10 GB 1 Node Start



```
Activities Terminal Sat 9:46 PM
ec2-user@ip-172-31-15-2:~/hadoop

File Edit View Search Terminal Help
Picked up _JAVA_OPTIONS: -Xmx2G -XX:ParallelGCThreads=1
[ec2-user@ip-172-31-15-2 ~]$ cd /mnt/raid/
[ec2-user@ip-172-31-15-2 raid]$ rm input.txt
[ec2-user@ip-172-31-15-2 raid]$ cd --
[ec2-user@ip-172-31-15-2 ~]$ cd hadoop/
[ec2-user@ip-172-31-15-2 hadoop]$ hadoop jar Terasort1.jar input output
Picked up _JAVA_OPTIONS: -Xmx2G -XX:ParallelGCThreads=1
16/03/27 02:46:50 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
16/03/27 02:46:50 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
16/03/27 02:46:51 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/27 02:46:51 INFO Input.FileInputFormat: Total input paths to process : 1
16/03/27 02:46:51 INFO mapreduce.JobSubmitter: number of splits:75
16/03/27 02:46:51 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local391980165_0001
16/03/27 02:46:52 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
16/03/27 02:46:52 INFO mapreduce.Job: Running job: job_local391980165_0001
16/03/27 02:46:52 INFO mapred.LocalJobRunner: OutputCommitter set in config null
16/03/27 02:46:52 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
16/03/27 02:46:52 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
16/03/27 02:46:52 INFO mapred.LocalJobRunner: Waiting for map tasks
16/03/27 02:46:52 INFO mapred.LocalJobRunner: Starting task: attempt_local391980165_0001_m_000000_0
16/03/27 02:46:52 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
16/03/27 02:46:52 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
16/03/27 02:46:52 INFO mapred.MapTask: Processing split: hdfs://localhost:8020/user/ec2-user/input:0+134217728
16/03/27 02:46:52 INFO mapred.MapTask: (EQUATOR) 0 kvl 26214396(104857584)
16/03/27 02:46:52 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
16/03/27 02:46:52 INFO mapred.MapTask: soft limit at 83886080
16/03/27 02:46:52 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
16/03/27 02:46:52 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
16/03/27 02:46:52 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
16/03/27 02:46:53 INFO mapreduce.Job: Job job_local391980165_0001 running in uber mode : false
16/03/27 02:46:53 INFO mapreduce.Job: map 0% reduce 0%
16/03/27 02:46:54 INFO mapred.MapTask: Spilling map output
16/03/27 02:46:54 INFO mapred.MapTask: bufstart = 0; bufend = 71891129; bufvoid = 104857600
16/03/27 02:46:54 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 23215656(92862624); length = 2998741/6553600
16/03/27 02:46:54 INFO mapred.MapTask: (EQUATOR) 74914041 kvl 18728504(74914016)
```

### Ii. 10 GB 1 Node Finish

```
Activities Terminal Sat 10:06 PM
ec2-user@ip-172-31-15-2:/mnt/raid

File Edit View Search Terminal Help

Map output bytes=9589477482
Map output materialized bytes=9789477932
Input split bytes=7950
Combine input records=0
Combine output records=0
Reduce input groups=97470859
Reduce shuffle bytes=9789477932
Reduce input records=100000000
Reduce output records=100000000
Spilled Records=293952411
Shuffled Maps =75
Failed Shuffles=0
Merged Map outputs=75
GC time elapsed (ms)=11233
Total committed heap usage (bytes)=86941106176

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=10000303104
File Output Format Counters
Bytes Written=9589477482
Total Running time 667272milliseconds
[ec2-user@ip-172-31-15-2 ~]$ hadoop dfs -get /user/ec2-user/output /mnt/raid/output
Picked up _JAVA_OPTIONS: -Xmx2G -XX:ParallelGCThreads=1
[ec2-user@ip-172-31-15-2 ~]$ cd /mnt/raid/
[ec2-user@ip-172-31-15-2 raid]$ ls
gensort lost+found new output
[ec2-user@ip-172-31-15-2 raid]$ mkdir result1
[ec2-user@ip-172-31-15-2 raid]$ cat output/* | head -n 10 >result1/sample1.txt
[ec2-user@ip-172-31-15-2 raid]$ cat output/* | tail -n 10 >result1/sample2.txt
```

## Iii. 16 node 100 GB Start

```
Activities Terminal Mon 11:44 PM
ec2-user@ip-172-31-11-5:~/hadoop

File Edit View Search Terminal Help

[ec2-user@ip-172-31-11-5 ~]$ hadoop jar Terasort1.jar input/ output2
Picked up _JAVA_OPTIONS: -Xmx2G -XX:ParallelGCThreads=1
```

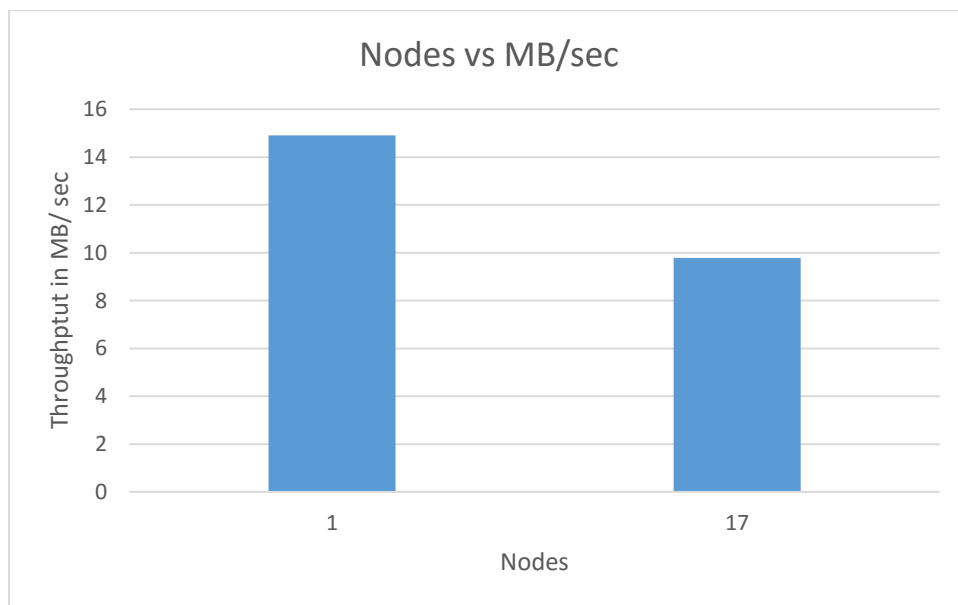
## Iv 16 node end

```
Activities Terminal Tue 2:47 AM
ec2-user@ip-172-31-11-5: ~/hadoop

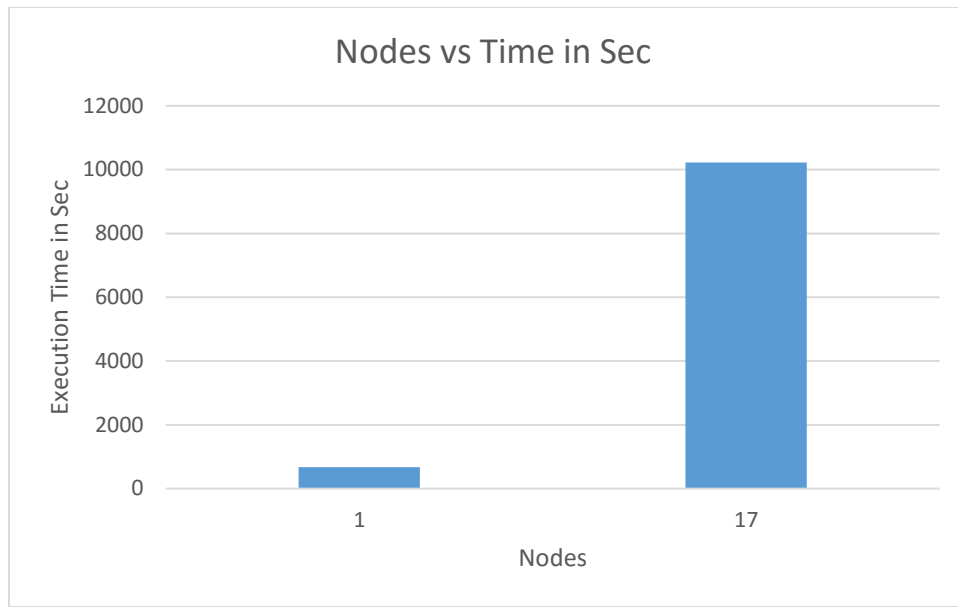
File Edit View Search Terminal Help

FILE: Number of write operations=0
HDFS: Number of bytes read=37404037074944
HDFS: Number of bytes written=95894739072
HDFS: Number of read operations=559501
HDFS: Number of large read operations=0
HDFS: Number of write operations=748
Map-Reduce Framework
  Map input records=1000000000
  Map output records=1000000000
  Map output bytes=95894739072
  Map output materialized bytes=97894743542
  Input split bytes=110260
  Combine input records=0
  Combine output records=0
  Reduce input groups=968948158
  Reduce shuffle bytes=97894743542
  Reduce input records=1000000000
  Reduce output records=1000000000
  Spilled Records=3998657822
  Shuffled Maps =745
  Failed Shuffles=0
  Merged Map outputs=745
  GC time elapsed (ms)=162112
  Total committed heap usage (bytes)=1504549273600
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=100003047424
File Output Format Counters
  Bytes Written=95894739072
Total Running time 10220344milliseconds
[ec2-user@ip-172-31-11-5: ~]$
```

Graphs:



Graph: Hadoop 1 node vs 17 nodes



Nodes vs Time in Seconds

#### D. questions

1) What is a Master node? What is a Slave node?

Ans. Master is the node that control the flow of data and execution over the cluster and manages all the slaves. The Namenode that manages storage and job tracker used to keep track of the running jobs.

Slaves usually run jobs assigned to them by the master and slaves can also be used as data store to maintain duplicates.

2) Why do we need to set unique available ports to those configuration files on a shared environment? What errors or side-effects will show if we use same port number for each user?

Ans. The unique ports help us identify, communicate and check if the node is active. Node can be uniquely identified in the network. If the same port is used for each user it will be difficult to communicate and identify.

3) How can we change the number of mappers and reducers from the configuration file?

Ans. The number of mappers and reducers can be changed by using `-D mapred.map.tasks` and `-D mapred.reducer.tasks` specified at runtime or the `mapred-site.xml` could be edited and `mapreduce.job.running.map.limit` could be set.

## 4. Spark



b. Description:

Shared memory sorting is an implementation of the external merge sort written in java which uses a combination of in memory sort and merging the in memory sorted data.

b. Environment settings:

The Spark Environment is built over the Hadoop version 2.6 and later. Hadoop is firstly configured and then spark is started over the Hadoop cluster. The Hadoop environment is adjusted according to the multimode cluster as setup above and then Spark for Hadoop 2.6 and later is configured to use the Hadoop HDFS and job tracker. The Spark environment is adjusted to use the raid for storage. Detailed description for files modified and changed is mentioned in read

Spark Version: 1.6.0

Hadoop Version: 2.7.2

OS distribution: Linux (preconfigured Hadoop AMI)

Java version: java-1.6.0-openjdk-devel (for jps and other developer tools)

Problem Encountered

A) Memory Full on device

Solution: Changed Temp directory for Spark to Raid 0

c. Results and Graphs

Nodes	Time	MB/sec
1	684.9	14.6
16	10024.9	9.97

Screenshot

I. 1 node start

```
Activities Terminal Tue 5:21 PM
ec2-user@ip-172-31-8-186:~/spark-1.6.0-bin-hadoop2.6

File Edit View Search Terminal Help
[ec2-user@ip-172-31-8-186 spark-1.6.0-bin-hadoop2.6]$ ./bin/spark-submit spark_sort.jar hdfs://ec2-54-172-183-63.compute-1.amazonaws.com:8020/input/in
put.txt /mnt/raid/output
Picked up _JAVA_OPTIONS: -Xmx2G
Picked up _JAVA_OPTIONS: -Xmx2G
█
```

## li 1 node end

```
Activities Terminal Tue 5:35 PM
ec2-user@ip-172-31-8-186:~/spark-1.6.0-bin-hadoop2.6

File Edit View Search Terminal Help
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/static,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/threadDump/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/threadDump,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/environment/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/environment,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/rdd/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/rdd,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/pool/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/pool,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/job/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/job,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/json,null}
16/03/29 22:32:57 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs,null}
16/03/29 22:32:57 INFO ui.SparkUI: Stopped Spark web UI at http://172.31.8.186:4040
16/03/29 22:32:57 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/03/29 22:32:57 INFO storage.MemoryStore: MemoryStore cleared
16/03/29 22:32:57 INFO storage.BlockManager: BlockManager stopped
16/03/29 22:32:57 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
16/03/29 22:32:57 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/03/29 22:32:57 INFO spark.SparkContext: Successfully stopped SparkContext
Total Running time 684995mlllseconds
16/03/29 22:32:57 INFO remote.RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/03/29 22:32:57 INFO util.ShutdownHookManager: Shutdown hook called
16/03/29 22:32:57 INFO remote.RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
16/03/29 22:32:57 INFO util.ShutdownHookManager: Deleting directory /mnt/raid/spark-ba394aa6-7b02-4966-9956-aa88f74b8d95/httpd-87c6feb0-905f-4665-8c44
-e321e75d30ad
16/03/29 22:32:57 INFO util.ShutdownHookManager: Deleting directory /mnt/raid/spark-ba394aa6-7b02-4966-9956-aa88f74b8d95
[ec2-user@ip-172-31-8-186 spark-1.6.0-bin-hadoop2.6]$ █
```

## Iii. 16 node start

```
Activities Terminal Tue 7:48 PM
ec2-user@ip-172-31-8-186:~/spark-1.6.0-bin-hadoop2.6

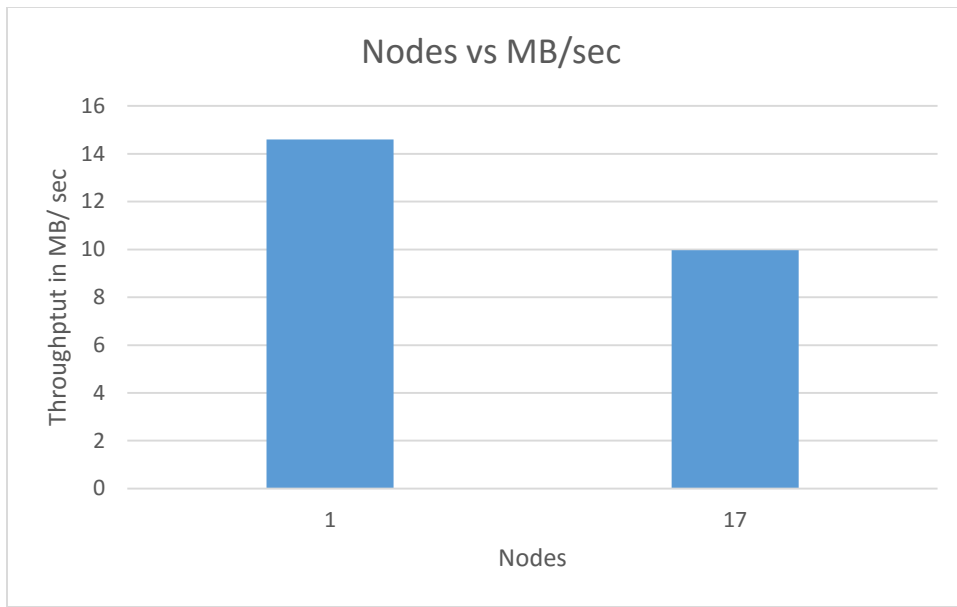
File Edit View Search Terminal Help
Picked up _JAVA_OPTIONS: -Xmx2G
16/03/30 00:48:50 INFO spark.SparkContext: Running Spark version 1.6.0
16/03/30 00:48:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/03/30 00:48:51 INFO spark.SecurityManager: Changing view acls to: ec2-user
16/03/30 00:48:51 INFO spark.SecurityManager: Changing modify acls to: ec2-user
16/03/30 00:48:51 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(ec2-user);
users with modify permissions: Set(ec2-user)
16/03/30 00:48:51 INFO util.Utils: Successfully started service 'sparkDriver' on port 57879.
16/03/30 00:48:52 INFO slf4j.Slf4jLogger: Slf4jLogger started
16/03/30 00:48:52 INFO Remoting: Starting remoting
16/03/30 00:48:52 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkDriverActorSystem@172.31.8.186:33258]
16/03/30 00:48:52 INFO util.Utils: Successfully started service 'sparkDriverActorSystem' on port 33258.
16/03/30 00:48:52 INFO spark.SparkEnv: Registering MapOutputTracker
16/03/30 00:48:52 INFO spark.SparkEnv: Registering BlockManagerMaster
16/03/30 00:48:52 INFO storage.DiskBlockManager: Created local directory at /mnt/raid/spark-c60fa56e-8367-49d9-a98a-46b764e17343/httpd-8b457dad-b2e5-46
16/03/30 00:48:52 INFO storage.MemoryStore: MemoryStore started with capacity 1140.4 MB
16/03/30 00:48:52 INFO spark.SparkEnv: Registering OutputCommitCoordinator
16/03/30 00:48:52 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/03/30 00:48:52 INFO server.AbstractConnector: Started SelectChannelConnector@0.0.0.0:4040
16/03/30 00:48:52 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
16/03/30 00:48:52 INFO ui.SparkUI: Started SparkUI at http://172.31.8.186:4040
16/03/30 00:48:52 INFO spark.HttpFileServer: HTTP File server directory is /mnt/raid/spark-c60fa56e-8367-49d9-a98a-46b764e17343/httpd-8b457dad-b2e5-46
10-adb4-d5fa7c421dce
16/03/30 00:48:52 INFO spark.HttpServer: Starting HTTP Server
16/03/30 00:48:52 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/03/30 00:48:52 INFO server.AbstractConnector: Started SocketConnector@0.0.0.0:42327
16/03/30 00:48:52 INFO util.Utils: Successfully started service 'HTTP file server' on port 42327.
16/03/30 00:48:53 INFO spark.SparkContext: Added JAR file:/home/ec2-user/spark-1.6.0-bin-hadoop2.6/spark_sort.jar at http://172.31.8.186:42327/jars/sp
ark_sort.jar with timestamp 1459298933070
16/03/30 00:48:53 INFO executor.Executor: Starting executor ID driver on host localhost
16/03/30 00:48:53 INFO util.Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 34976.
16/03/30 00:48:53 INFO netty.NettyBlockTransferService: Server created on 34976
16/03/30 00:48:53 INFO storage.BlockManagerMaster: Trying to register BlockManager
16/03/30 00:48:53 INFO storage.BlockManagerMasterEndpoint: Registering block manager localhost:34976 with 1140.4 MB RAM, BlockManagerId(driver, localh
ost, 34976)
16/03/30 00:48:53 INFO storage.BlockManagerMaster: Registered BlockManager
```

## Iv 16 node end

```
Activities Terminal Tue 10:50 PM
ec2-user@ip-172-31-8-186:~/spark-1.6.0-bin-hadoop2.6

File Edit View Search Terminal Help
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/ ,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/static,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/threadDump/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/threadDump,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/environment/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/environment,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/rdd/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/rdd,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/pool/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/pool,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/job/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/job,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/json,null}
16/03/30 03:44:29 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs,null}
16/03/30 03:44:29 INFO ui.SparkUI: Stopped Spark web UI at http://172.31.8.186:4040
16/03/30 03:44:29 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/03/30 03:44:30 INFO storage.MemoryStore: MemoryStore cleared
16/03/30 03:44:30 INFO storage.BlockManager: BlockManager stopped
16/03/30 03:44:30 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
16/03/30 03:44:30 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/03/30 03:44:30 INFO spark.SparkContext: Successfully stopped SparkContext
Total Running time 10024932milliseconds
16/03/30 03:44:30 INFO remote.RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/03/30 03:44:30 INFO remote.RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
16/03/30 03:44:30 INFO util.ShutdownHookManager: Shutdown hook called
16/03/30 03:44:30 INFO util.ShutdownHookManager: Deleting directory /mnt/raid/spark-bf79bab6-bdc3-476b-9b78-d1172dd8b9b5/httpd-f8eec080-3a67-4692-84f2
-d4ad0822eb26
16/03/30 03:44:30 INFO util.ShutdownHookManager: Deleting directory /mnt/raid/spark-bf79bab6-bdc3-476b-9b78-d1172dd8b9b5
[ec2-user@ip-172-31-8-186 spark-1.6.0-bin-hadoop2.6]$
```

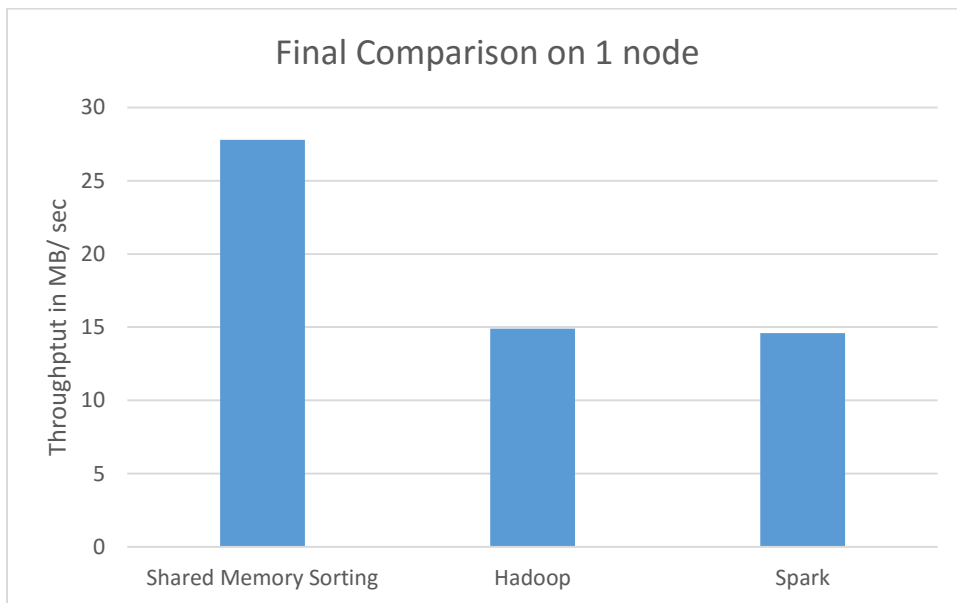
Graphs:



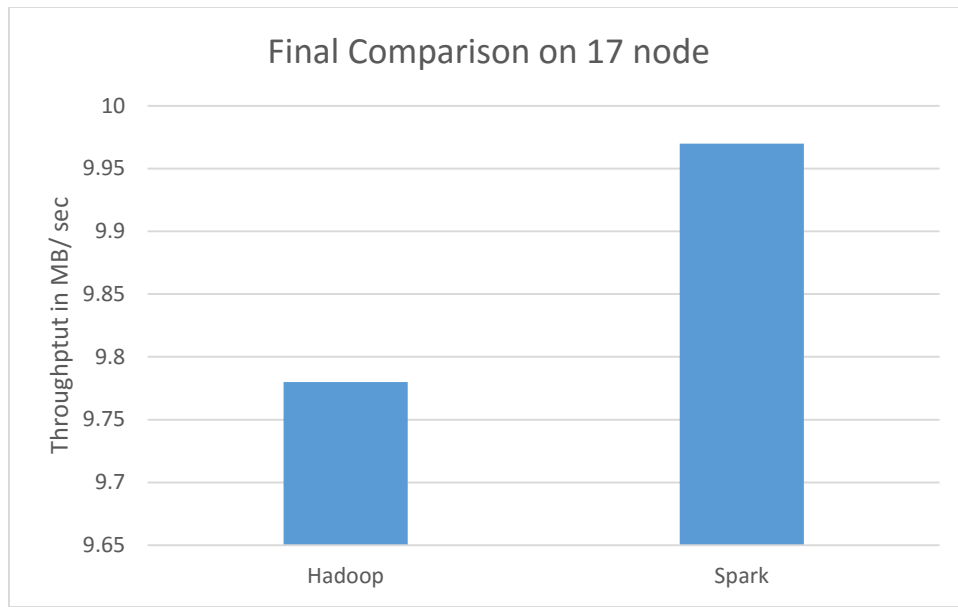
Graph: Hadoop 1 node vs 17 nodes

## 5. Performance

### a.Graphs



Graph: Comparing performance on 1 node for each framework.



Graph: Comparing performance on 17 node for Hadoop and Spark.

## b) Results and Conclusion

### 1) What conclusions can you draw?

Ans. For 1 node the best performance is given by Shared Memory sort as it doesn't involve overheads involved in Hadoop and Spark.

For 17 nodes the performance of Spark is slightly better than that of Hadoop but the factor is very less.

### 2) Can you predict which would be best at 100 node scale?

Ans. As seen from the comparison Spark performance much better in 17 nodes and close to Hadoop in case of 1 node. Hence Scaling up the node will boost the performance for Spark as compared to Hadoop.

### 3) Can you predict which would be best at 1000 node scale?

Ans. The above answer indicates the same as we scale the nodes the performance gets better for Spark, and hence I feel that Spark will perform well in case of 1000 nodes

### 4) Compare your results with those from the Sort Benchmark [9], specifically the winners in 2013 and 2014 who used Hadoop and Spark.

Ans. The winner of sort benchmark in 2014 with Apache Spark gave a throughput 4.27 TB/min 207 Instances and sorting a total of 100 TB. The winners in the 2013 experiment with Hadoop did sorting 104 TB with the throughput of 1.42 TB/sec implemented on 2100 nodes. The Throughput is very high as compared to our experiments which is approximately which is approximately 594 MB/min for Spark and 582 MB/min.

5) What can you learn from the CloudSort benchmark?

Ans. The CloudSort total cost ownership Benchmark that uses public cloud and majorly aims for IO intensive applications. It is developed for the teams that have limited budgets and require high end hardware to run the external sort algorithm. It prefers external sort as it is IO intensive and has high workloads.